SCM
Software for
Chemistry &
Materials

ADF Manual

Amsterdam Modeling Suite 2024.1

www.scm.com

Apr 05, 2024

CONTENTS

CHAPTER ONE

GENERAL

1.1 Introduction

In AMS2020 is ADF is only accessible through the AMS driver program. ADF (Amsterdam Density Functional) is an AMS engine for calculations on atoms and molecules (in gas phase or solution). It can be used for the study of such diverse fields as molecular spectroscopy, organic and inorganic chemistry, crystallography and pharmacochemistry. The AMS engine BAND is available for the study of periodic systems: crystals, surfaces, and polymers. The separate COSMO-RS program is used for calculating thermodynamic properties of (mixed) fluids.

The underlying theory is the Kohn-Sham approach to Density-Functional Theory (DFT). This implies a one-electron picture of the many-electron systems but yields in principle the exact electron density (and related properties) and the total energy.

If ADF is a new for you we recommend that you carefully read the section *Technical remarks, Terminology* (page 18), which presents a discussion of a few ADF-typical aspects and terminology. This will help you to understand and appreciate the output of an ADF calculation.

ADF has been developed since the early 1970s (at that time called HFS, later AMOL, see also refs.[1][2][3], mainly by the two theoretical chemistry groups of, respectively, the Vrije Universiteit in Amsterdam (http://www.chem.vu.nl/en/research/division-theoretical-chemistry/index.asp) and the University of Calgary, Canada. Other researchers have also contributed. As a major research tool of these academic development groups, ADF is in continuous development and retains a firm basis in the academic world.

Maintenance and distribution of the commercial (export) version of the program is done by Software for Chemistry & Materials B.V. (SCM) (http://www.scm.com), a company based in Amsterdam, formally split off from the theoretical chemistry group in Amsterdam but practically still very much a part of it. Documentation such as User manuals, Installation instructions, Examples, Theoretical documents can be found at the SCM web site.

Publications based on research with ADF should include appropriate references to the program. We recommend that references are made both to the program itself and to publications related to its development and structure. See the *Required Citations* (page 1089).

The installation of the Amsterdam Modeling Suite (AMS) program package is explained in the Installation manual. This User's Guide describes how to use the program, how input is structured, what files are produced, and so on. Some special applications of ADF are described in the *Examples* (page 401).

Where references are made to the operating system (OS) and to the file system on your computer the terminology of UNIX type OSs is used.

---

[1] E.J. Baerends, D.E. Ellis and P. Ros, *Self-consistent molecular Hartree-Fock-Slater calculations I. The computational procedure*, Chemical Physics 2, 41 (1973) (https://doi.org/10.1016/0301-0104(73)80059-X)

[2] E.J. Baerends and P. Ros, *Evaluation of the LCAO Hartree-Fock-Slater method: Applications to transition-metal complexes*, International Journal of Quantum Chemistry 14, S12, 169 (1978) (https://doi.org/10.1002/qua.560140814)

[3] G. te Velde, F.M. Bickelhaupt, E.J. Baerends, C. Fonseca Guerra, S.J.A. van Gisbergen, J.G. Snijders, T. Ziegler, *Chemistry with ADF*, Journal of Computational Chemistry 22, 931 (2001) (https://doi.org/10.1002/jcc.1056)

The AMS package is in continuous development to extend its functionality and applicability, to increase its efficiency and user-friendliness, and of course to correct errors. We appreciate comments and suggestions for improvement of the software and the documentation.

1.1.1 ADF is an AMS Engine

The most important change in AMS2020 is that ADF is only accessible via the AMS driver program. We recommend you to first read the General section of the AMS Manual More details this can be found in the section *AMSification of ADF* (page 9)

1.1.2 Functionality

- Computation of any electronic configuration
- Excitation energies, oscillator strengths, transition dipole moments, (hyper)polarizabilities, Van der Waals dispersion coefficients, CD spectra, ORD, MCD, VCD, magnetizabilities, Verdet constants, using Time-Dependent Density Functional Theory (TDDFT)
- ESR (EPR) g-tensors, A-tensors, NQCCs
- NMR chemical shifts and spin-spin coupling constants
- Mössbauer spectroscopy
- Transport properties: charge transfer integrals, NEGF
- Various other molecular properties
- Treatment of large systems and environment by the QM/MM (Quantum Mechanics / Molecular Mechanics) hybrid approach.

1.1.3 Applicability

All elements of the periodic table can be used (Z = 1-120). For each of the elements, basis sets of different sizes are available, ranging from minimal to high quality. Special basis sets are provided for relativistic calculations within the ZORA approach and for response calculations that require additional diffuse basis functions.

1.1.4 Model Hamiltonian

- A choice of Density Functionals, both for the Local Density Approximation (LDA), for the Generalized Gradient Approximation (GGA), for (range-separated) hybrid functionals (not for all properties available), and for meta-GGA functionals (not for all properties available) are available.
- Spin: restricted or unrestricted
- Relativistic effects: scalar approximation and spin-orbit (double-group symmetry), using the (now recommended) ZORA or the (previously used) Pauli formalism, X2C
- Environment: Solvent Effects, Homogeneous Electric Field, Point Charges (Madelung Fields), QM/MM method, FDE
- Constrained Density Functional Theory

1.1.5 Analysis

- Decomposition of the bond energy in ‘chemical’ components (steric interaction, Pauli repulsion, orbital interactions…)
- Natural orbitals for chemical valence (ETS-NOCV)
- Representation of data (Molecular Orbital coefficients, Mulliken Populations) in terms of the constituent chemical fragments in the molecule, along with the conventional representation in elementary basis functions
- Atomic charge determination by Hirshfeld analysis and by Voronoi analysis, multipole derived charges, along with the classical Mulliken populations, and Mayer bond orders
- QTAIM analysis based on local, atomic and non-local descriptors for bonding description.
- Conceptual DFT descriptors including global, atomic, non-local ones and a detailed analysis of the dual descriptor’s domains (predominantly electrophilic or nucleophilic regions).
- Bond energy decomposition based on the interacting quantum atoms (IQA) approach and using QTAIM real-space partition. Any atom-atom interaction can be evaluated and decomposed into electrostatic (ionic) and exchange (covalent) contributions.
- Localized molecular orbitals

1.1.6 Technical

- The implementation is based upon a highly optimized numerical integration scheme for the evaluation of matrix elements of the Fock operator, property integrals involving the charge density, etc. The code has been vectorized and parallelized.
- Basis functions are Slater-Type Orbitals (STOs). A database is available with several basis sets for each atom in the periodic table of elements.
- The Coulomb potential is evaluated via an accurate fitting of the charge density.
- A frozen core facility is provided for an efficient treatment of the inner atomic shells.
- Extensive use is made of point group symmetry. Most of the commonly encountered symmetry groups are available.
- Linear scaling techniques are used to speed up calculations on large molecules

1.1.7 Fragments

ADF has a fragment oriented approach: the poly-atomic system to be computed is conceptually built up from fragments, the molecular one-electron orbitals are calculated as linear combinations of fragment orbitals, and final analyzes of e.g. the bonding energy are in terms of fragment properties. The fragments may be single atoms or larger moieties.

When you compute a system in terms of its constituent fragments, these fragments must have been computed before and their properties must be passed on to the current calculation. This is done by attaching *fragment files*, which contain the necessary information. A fragment file is simply the standard result file of an ADF calculation on that fragment.

When using Basic Atoms as fragments, you do not need to create the fragment files yourself; ADF will create the required fragment files automatically. We therefore recommend this feature for starting ADF users.

Basic atoms

Obviously there must be a set of fundamental fragments that are not defined in terms of smaller fragments. Therefore ADF has two modes of execution: the normal mode, using fragments, and the create mode, in which a fundamental fragment is generated. Such a fundamental fragment *must* be a single atom, spherically symmetric and spin-restricted (i.e. spin-α and spin-β orbitals are spatially identical, they are equally occupied, and fractional occupations are applied, if necessary, to distribute the electrons equally over symmetry-degenerate states). Such a fundamental fragment is denoted a *basic atom*. The basic atoms are the smallest building blocks from which any 'real' calculations are started.

One should realize that the basic atoms are artificial objects that are convenient in the computational approach but that do not necessarily represent real atoms very well (in fact, usually not at all). The bonding energy of a molecule with respect to basic atoms, for instance, should be corrected for this discrepancy in order to get a decent comparison against experimental data. See ref.[4] for a discussion and for examples of applicable values.

A basic atom is computed in the conventional way. The one-electron orbitals are determined as linear combinations of basis functions; the frozen core approximation may be applied for the inner atomic states; a particular type of density functional can be chosen, et cetera. You may have, for instance, different basic Copper atoms by using different basis sets, by choosing different levels of frozen core approximations, or by applying different density functionals.

Automatic mode

If you are using 'Basic Atom' fragments only, you do not need to prepare the corresponding fragment files yourself; ADF will generate all the required fragment files for you. This makes your job scripts and ADF inputs simpler, it ensures that consistent options for the create runs and molecular runs are used.

1.1.8 Slater-type basis sets

ADF uses Slater-Type Orbitals (STO's) as basis functions. Slaters can display the correct nuclear cusp and asymptotic decay.

$$f(\mathbf{r}) = Y_{lm} r^n e^{-\zeta r}$$

The center of the function is at a nucleus, the Y_{lm} are spherical harmonics, and the exponential factor ζ (zeta) determines the long-range decay of the function.

ADF comes with a database of thoroughly tested basis set files, ranging in quality from single-zeta to quadruple-zeta basis sets with various diffuse and polarization functions. All-electron and frozen-core basis sets are available for all elements, including lanthanides and actinides. The frozen-core approximation can be used to considerably reduce the computation time for systems with heavy nuclei, in a controlled manner.

1.2 What's new in ADF 2024.1

1.2.1 New features

- Simplified procedure for calculating *pNMR chemical shieldings* (page 211) of open shell molecules
    - effects of ESR g-tensor and ESR A-tensor can be included, ESR D-tensor not
- Experimental combination of G_0W_0 (*GW* (page 227)) and embedding (*DRF* (page 112) or *COSMO* (page 102))
    - GW with embedding may yield inaccurate results

---

[4] E.J. Baerends, V. Branchadell and M. Sodupe, *Atomic reference-energies for density functional calculations*, Chemical Physics Letters 265,481 (1997) (https://doi.org/10.1016/S0009-2614(96)01449-2)

1.2.2 Removed features

- Solvation methods SCRF (Self-Consistent Reaction Field) and VSCRF no longer supported
  - see documentation in the ADF 2023 manual on SCRF (https://www.scm.com/doc.2023/ADF/Input/SCRF.html) and VSCRF (https://www.scm.com/doc.2023/ADF/Input/VSCRF.html)
- Old EPR/NMR program ($AMSBIN/epr) is no longer supported
  - see the documentation in the old ADF 2010 Properties document (http://downloads.scm.com/Doc/Doc2010/ADF/Properties/Properties.pdf).
- Old FCF program ($AMSBIN/oldfcf) is no longer supported
  - The new FCF program ($AMSBIN/fcf) has replaced the old FCF program in AMS 2022
  - see documentation in the AMS 2021 manual on the old FCF program (https://www.scm.com/doc.2021/AMS/Utilities/FCF_module.html)

1.3 What's new in ADF 2023.1

1.3.1 New features

- Model Hamiltonians
  - *r2SCAN-3c* (page 76): composite DFT method: r2SCAN, mTZ2P, gCP, D4
    - outperforms many hybrid/QZ approaches at a fraction of their cost
  - *sigma-functional* (page 86): New beyond-RPA method:
    - outperforms RPA at the same cost.
  - *ROKS for high spin open shell molecules* (page 57)
    - work-in-progress, only single points calculations
  - *TASKCC metaGGA* (page 76) improves TASKxc atomization energies
  - Calculate entropies of solvation with *3D-RISM* (page 137)
- Spectroscopy
  - *GW-BSE: Bethe Salpeter equation* (page 162)
    - G_0W_0-BSE, evGW-BSE, qsGW-BSE
    - spin-orbit coupled
  - *CD spectra with spin-orbit coupling* (page 179)
  - *excited state geometry optimizations for TD-DFT+TB* (page 185)
  - open shell excitation energies for *TD-DFT+TB, TDA-DFT+TB* (page 177), *sTDA, sTDDFT* (page 178)
  - open shell excitation energies with *Hybrid Diagonal Approximation (HDA)* (page 176)
  - *PolTDDFT with fitted HDA* (page 196)
  - Calculation of *vibrational polarizabilities* (page 154) under the harmonic approximation
- *Many-Body Perturbation Theory* (page 329)
  - *spin-orbit coupling with GW* (page 227): G_0W_0, evGW, qsGW, G3W2

- Analysis
  - MBPT: *qsGW quasi particle orbitals in terms of SFOs* (page 227)
  - *NOCV* (page 260)
    - T/V analysis of energy contributions
    - NOCVs in terms of SFOs
    - Mulliken population of the relaxation density in terms of primitive STOs
  - simplified calculation of site energies with *PRINT SFOSiteEnergies* (page 274)
  - *PolTDDFT* (page 196)
    - analysis in terms of single orbital transitions
    - fragment projection analysis using regions
  - *QTAIM* (page 267)
    - source function for all BCPs and RCPs
  - *IQA* (page 253)
    - extended to open-shell systems
- Visualization with AMS-GUI
  - MBPT: GW quasi particle orbitals and energies
  - SFO site energies
  - PolTDDFT: TCM, ICM-OS, ICM-RS
- Performance
  - reduced disk usage for some files containing matrices (TAPE15, adf.rkf a.k.a. TAPE21), which also improves performance for some file systems (NFS, GPFS)
- New AMS driver features can be used in combination with ADF
  - what's new in the AMS driver

1.3.2 Bug fixes

- *GW* (page 227): AMS2023.102: bug fix GW. Incorrect Fermi energy was used. Also affects BSE.
- *LFDFT analysis* (page 200): bug fix expectation values $< S^2 >, < L^2 >, < J^2 >$

1.4 What's new in ADF 2022.1

1.4.1 New features

- Model Hamiltonians
  - *QM/FQ Quantum Mechanics/Fluctuating Charges* (page 122)
    - extended to fluctuating dipoles (QM/FQFμ)
    - can be combined with Frozen Density Embedding (FDE)
  - *3D-RISM* (page 137) revised implementation and additional functionality

  - *IQA Interacting Quantum Atoms* (page 253)
    - intra-atomic terms
    - parallelization
  - *MP2* (page 85) and *double hybrid* (page 82) in combination with *spin-orbit coupling* (page 97)
    - in case of spin-orbit coupling approximate SS and OS contributions
- *Many-Body Perturbation Theory* (page 329)
  - *quasiparticle self-consistent GW: qsGW* (page 227)
  - *second-order self-energy GW: G3W2* (page 227)
  - improvements in stability of the MBPT results especially for larger basis sets
- New AMS driver features can be used in combination with ADF
  - what's new in the AMS driver
  - FCF module for the calculation of vibronic progressions and bandshapes improved

1.5 What's new in ADF 2021.1

1.5.1 New features

- Model Hamiltonians
  - *QM/FQ Quantum Mechanics/Fluctuating Charges* (page 122)
  - *r2SCAN-D4* (page 89) XC functional
- *Many-Body Perturbation Theory* (page 329)
  - *eigenvalue-only self-consistent GW: evGW* (page 227)
  - *Basis sets* (page 38) for use in *MBPT* (page 329)
    - Corr/TZ3P
    - Corr/QZ6P
- Spectroscopy
  - *LFDFT ESR (EPR) g-tensor doublets* (page 203)
  - *LFDFT MCD* (page 204)
  - *PolTDDFT: reduced fit sets for most elements for DZ, DZP, TZP basis sets* (page 196)
  - *unrelaxed dipole moment excited states* (page 160)
  - *transition dipole moment between excited states* (page 164)
- New AMS driver features can be used in combination with ADF. See the page What's new in the AMS driver for more details.

1.5.2 New Defaults

- *Perturbative inclusion of spin-orbit coupling for excitation energies* (page 171)
    - singlet ground state is included by default, keyword GSCORR now subkey of SOPERT
    - improved way to calculated transition dipole moment

1.6 What's new in ADF 2020.1

1.6.1 New features

- *Many-Body Perturbation Theory* (page 329)
    - *GW approximation* (page 227): G_0W_0
    - *RPA* (page 85)
- Spectroscopy
    - *VCD for open shell molecules* (page 154)
    - *Hybrid diagonal approximated (HDA) excitation energies* (page 176)
    - *Hybrid functionals with PolTDDFT* (page 196)
- STO basis sets
    - *STO basis sets for Uue (Z=119) and Ubn (Z=120) added* (page 38)
- PES point character (see AMS driver)
    - without calculating the full Hessian, characterize a PES point
- Symmetry (see AMS driver)
    - ADF does not symmetrize coordinates anymore
    - AMS can symmetrize coordinates, but that is not the default
    - more symmetries possible with irreducible representations whose characters are complex, like C(3), S(4)
    - labels for vibrational normal modes for these symmetries
    - remark: not more symmetries can be used for electronic states
- Speed
    - double parallelization (see AMS driver) possible for numerical calculation of IR frequencies

1.6.2 New Defaults

- *Scalar relativistic ZORA* (page 97) (instead of non-relativistic)
- AMS does not symmetrize coordinates by default, which ADF used to do. See also AMS driver system definition and *symmetry section of ADF* (page 380).
- $AMSHOME/atomicdata/ADF directory with ADF basis sets (instead of $ADFHOME/atomicdata)

1.7 ADF is an AMS Engine

Starting from AMS2020 ADF is only accessible via the AMS driver program. The standalone program 'adf' does not exist anymore. The job of the AMS driver is to handle all changes in the geometry, e.g. during a geometry optimization, using so-called engines like ADF for the calculation of energies and forces.

Important: In the AMS2020 release ADF is an AMS engine. We recommend you to first read the General section of the AMS Manual

---

If you use ADF exclusively via the Graphical User Interface (GUI), this change should not create any issues. If, on the other hand, you create input files by hand (or you use ADF via PLAMS), then you should be aware that shell scripts for ADF2019 and previous versions are not compatible with ADF2020 and have to be adjusted to the new setup (see also *Automatic tool for conversion of ADF2019 shell scripts* (page 13)).

Some of the changes:

- environment variables AMSHOME, AMSBIN (instead of ADFHOME, ADFBIN)
- ams (instead of adf)
- major restructuring of input and input keys
- output files in separate directory
- ams.rkf new binary output file, contains mainly geometry related data
- adf.rkf binary output file (instead of TAPE21), contains mainly single point related data
- AMS does not symmetrize coordinates by default, which ADF used to do. See also AMS driver system definition and *symmetry section of ADF* (page 380).
- QM/MM, QM/QM, Quild with the Hybrid engine
- $AMSHOME/atomicdata/ADF directory with ADF basis sets (instead of $ADFHOME/atomicdata)
- scalar relativistic ZORA is the default (instead of non-relativistic)

See also:

More details this can be found in the section *AMSification of ADF* (page 9)

1.8 AMSification of ADF

1.8.1 Restructuring of the input

The input for the following features belongs (mostly) to the AMS driver part of the input:

- Geometry, System definition
  - Geometry
  - Total charge
  - Masses of atoms (isotopes)
  - Electric field, point charges
  - Regions
- Structure and Reactivity, Molecular Dynamics

  - Geometry Optimizations
  - Linear Transit (LT), Transition States (TS), Nudged Elastic Band (NEB), TSRC
  - Intrinsic Reaction Coordinate (IRC)
  - Constraints, restraints
  - Excited state optimizations
  - Molecular Dynamics
- Gradients, Hessian, Thermodynamics
  - Nuclear gradients, Hessian, PES point character
  - Thermodynamics, gas phase Gibbs free energy
- Vibrational Spectroscopy
  - IR frequencies and intensities
  - Mobile Block Hessian (MBH), Mode Scanning, Mode Refinement, Mode Tracking
  - (Resonance) Raman, vibrational Raman optical activity (VROA)
  - Vibrational Circular Dichroism (VCD)
- Vibrational resolved electronic spectra
  - Vibrational resolved absorption, emission, phosphorescence, ionization
  - Adiabatic Hessian Franck-Condon
  - Vertical Gradient Franck-Condon

The input for QM/MM, QM/QM, Quild calculations uses the Hybrid engine

- Model Hamiltonians
  - QM/MM, QM/QM, Quild

The input for most other features belongs to the ADF engine part of the input.

1.8.2 Restructuring of input keys

ADF2019 key / feature	ADF2020 key / AMS2020 key / comments
$ADFBIN	$AMSBIN
$ADFHOME	$AMSHOME
$ADFRESOURCES	$AMSRESOURCES=$AMSHOME/atomicdata
$ADFHOME/atomicdata	$AMSHOME/atomicdata/ADF. Directory with ADF basis sets.
$ADFHOME/atomicdata/ZORA	$AMSHOME/atomicdata/ADF/ZORA. Directory with ADF ZORA basis sets.
‘ANALYTICALFREQ’	subkey ‘NormalModes’ of key ‘Properties’ in AMS driver.
‘AORESPONSE’	subkey ‘FreqRange’ and ‘Frequency’ removed. Use new subkey ‘Frequencies’.
‘ATOMPROPS’	key removed. See *alternative elements* (page 48).
‘ATOMS’	subkey ‘Atoms’ of key ‘System’ in AMS driver.
… f=fragtype/n	… adf.f=fragtype|n.
… R=CosmoRadius	… adf.R=CosmoRadius.
‘CHARGE’	subkey ‘Charge’ of key ‘System’ in AMS driver. key ‘SpinPolarization’ in ADF.
‘CONSTRAINTS’	key ‘Constraints’ in AMS driver.
‘COLLINEAR’	subkey ‘SpinOrbitMagnetization Collinear’ in key ‘Relativity’ in ADF.

continues on next page

Table 1.1 – continued from previous page

ADF2019 key / feature	ADF2020 key / AMS2020 key / comments
'CRDFILEXYZ'	key removed.
'CRDFILEMOL'	key removed.
EField	subkey 'ElectroStaticEmbedding%ElectricField' of key 'System' in AMS driver.
fcf	program 'fcf' documented in AMS driver
'GEOMETRY'	key 'Task' and key 'Properties' in AMS driver.
Geometry optimization	key 'Task GeometryOptimization' in AMS driver.
'GEOVAR'	key removed.
'GRADIENT'	subkey 'Gradients' of key 'Properties' in AMS driver.
'HESSDIAG'	key removed.
'HESSTEST'	key removed.
Initial Hessian	subkey 'InitialHessian' of key 'GeometryOptimization' in AMS driver.
Intrinsic Reaction Coordinate	key 'Task IRC' in AMS driver.
IR frequencies	subkey 'NormalModes' of key 'Properties' in AMS driver.
… Mobile Block Hessian	subkey 'Displacements Block' of key 'NormalModes' in AMS driver.
… symmetric displacements	subkey 'Displacements Symmetric' of key 'NormalModes' in AMS driver.
'IRCSTART'	key removed.
'ISOTOPICSHIFT'	key removed.
Linear Transit	key 'Task PESScan' in AMS driver.
'LINEARCONSTRAINTS'	key removed.
MBH	subkey 'Displacements Block' of key 'NormalModes' in AMS driver.
'MP2'	key 'MBPT'.
'NONCOLLINEAR'	subkey 'SpinOrbitMagnetization NonCollinear' in key 'Relativity' in ADF.
Nudged Elastic Band	key 'Task NEB' in AMS driver.
Partial Hessian	subkey 'SelectedRegionForHessian' and 'NormalModes' of key 'Properties' in AMS driver.
'POINTCHARGES'	subkey 'ElectroStaticEmbedding%MultipolePotential' of key 'System' in AMS driver.
'POLTDDFT'	subkey 'NFreq' and 'NGrid' added.
'QMMM'	Hybrid engine.
QM/MM; QM/QM; Quild	Hybrid engine.
Raman	subkey 'Raman' of key 'Properties' in AMS driver.
'RELATIVISTIC'	block key 'Relativity' in ADF.
'RESPONSE'	subkey 'FrqBeg', 'FrqEnd, and 'NFreq' removed. Use new subkey 'Frequencies'.
'RESTRAINT'	key 'Restraints' in AMS driver.
'Restart'	The file should be specified in 'EngineRestart' in AMS driver
'SYMMETRY'	both in ADF and AMS driver.
symmetrization	ADF does not symmetrize anymore. Subkey 'Symmetrize' of key 'System' in AMS driver.
'THERMO'	key 'Thermo' in AMS driver.
Transit State search	key 'Task TransitionStateSearch' in AMS driver.
'UNITS'	key removed. Possibility added to add units for many keys.
VCD	subkey 'VCD' of key 'Properties' in AMS driver.
vcdtools	program 'vcdtools' documented in AMS driver
VIBRON module	VIBRON module removed. For resonance Raman application see AMS driver.
VROA	subkey 'VROA' of key 'Properties' in AMS driver.

1.8.3 Example shell script changes

The example below shows how a shell script for ADF2019 is converted to ADF2020 (see also *Automatic tool for conversion of ADF2019 shell scripts* (page 13)).

ADF2019 shell script (obsolete):

```
#!/bin/sh

# This is a shell script for ADF2019 which will not work for ADF2020

$ADFBIN/adf <<eor
   Title WATER Geometry Optimization
   Atoms
      O             0.000000     0.000000     0.000000
      H             0.000000    -0.689440    -0.578509
      H             0.000000     0.689440    -0.578509
   End
   Geometry
      Converge grad=1e-4
   End
   Basis
      Type TZP
   End
   XC
      GGA PBE
   End
eor
```

ADF2020 shell script:

```
#!/bin/sh

# This is a shell script for ADF2020

# You should use '$ADFBIN/ams' instead of '$ADFBIN/adf'

$ADFBIN/ams <<eor
   # Input options for the AMS driver:

   System
      Atoms
         O             0.000000     0.000000     0.000000
         H             0.000000    -0.689440    -0.578509
         H             0.000000     0.689440    -0.578509
      End
   End
   Task GeometryOptimization
   GeometryOptimization
      Convergence gradients=1e-4
   End

   # The input options for ADF, which are described in this manual,
   # should be specified in the 'Engine ADF' block:

   Engine ADF
      Basis
         Type TZP
```

(continues on next page)

(continued from previous page)

```
      End
      XC
         GGA PBE
      End
   EndEngine
eor
```

1.8.4 Automatic tool for conversion of ADF2019 shell scripts

The executable `$AMSBIN/adf` no longer performs ADF calculations. Instead, it will call an **input-conversion tool**[1] which will automatically convert an ADF2019 input into an ADF2020 input.

For example, if you save and execute the following script (note that the `$AMSBIN/adf` executable is used):

```
#!/bin/sh

# This is a shell scripts of an ADF2019 calculation, which will not work for ADF2020
# If you execute this script, the ADF2019 input will be converted into an ADF2020␣
↪input.

$AMSBIN/adf <<eor
   Atoms
      O             0.000000     0.000000     0.000000
      H             0.000000    -0.689440    -0.578509
      H             0.000000     0.689440    -0.578509
   End

   Basis
      Type TZP
   End
eor
```

no ADF calculation will actually be performed. Instead, the input conversion tool will be called, and you will obtain the new shell script for ADF2020 (note the `$AMSBIN/ams` executable):

```
# =========
# IMPORTANT
# =========

# Starting from AMS2020, ADF can only be used through the AMS driver program.
# Many input options for ADF have changed since the 2019 version.
# Below you will find an automatically-converted input script for AMS2020.
# The automatic conversion is not 100% accurate, so double check your input!

#         ===================================================
#         Automatic conversion of ADF-2019 input to AMS-2020
#         ===================================================

# === NOTES ===
# - "TAPE21" is now called "adf.rkf" and is located in AMS results folder (by default
↪"ams.results")
# - Unlike ADF2019, AMS does not symmetrize the structure by default. See "System ->␣
↪Symmetrize" in the AMS driver manual.
```

(continues on next page)

---

[1] You can find the actual python script that performs the ADF2019 to ADF2020 conversion in `$AMSHOME/Utils/adf_to_ams_input_converter.py`.

(continued from previous page)

```
# - The AMS default symmetrization tolerance is larger than the ADF2019 one. See
↪"Symmetry -> SymmetrizeTolerance" in the AMS driver manual.
# - Scalar relativistic effects (ZORA) are included by default in the 2020 version of␣
↪ADF.


$AMSBIN/ams << eor
System
  Atoms
    O      0.0000000000      0.0000000000      0.0000000000
    H      0.0000000000     -0.6894400000     -0.5785090000
    H      0.0000000000      0.6894400000     -0.5785090000
  End
  Symmetrize True
End

Symmetry
  SymmetrizeTolerance 0.001
End

Engine ADF
  Relativity
    Level None
  End

  Basis
    Type TZP
  End
EndEngine

eor
```

The conversion tool might raise `WARNINGS`. Make sure to address them before using the converted the script!

1.9 Input, execution and output

See the AMS driver documentation:

- Input, execution and output section of the AMS driver manual

1.9.1 ADF specific files

The ADF engine may generate several output / result files, along that ADF writes part of the standard output file. The most important one is adf.rkf (previously ADF<=2019 known as TAPE21 or as a .t21 file), the general result file. The adf.rkf contains relevant information about the outcome of the calculation. It is a binary file that contains a lot of information about the calculation, such as the one-electron orbitals expressed in the basis functions. It can be used as a fragment file for subsequent calculations, or can be used as input to a property program, like NMR, CPL, or LFDFT.

Files attached to the job, such as fragment files, are by default also assumed to exist in the directory where you start AMS. You must take care to move or copy required files to that directory before starting the calculation, or to provide via input adequate information to the program where to find the files. In many cases you can specify a complete path to the file.

Most files that are generated by the program, in particular the standard result file that can be used as a fragment file in other calculations, are *binary* files. A binary file should usually not be moved from one machine to another, i.e. it may not be readable by another machine than the one that generated the file, unless the two machines are of the same type. The ADF package provides utilities to convert the ADF binary result files from binary to ASCII, and vice versa, so that you don't have to regenerate your fragment libraries when going to another machine. See the Scripting Section for further details regarding such aspects.

ADF is an engine that lends itself particularly well for chemical analysis. This is a direct result of the fragment-based approach, where properties of the molecule are related to the properties of the constituent fragments, which is precisely how the chemist thinks. Molecular Orbitals are (optionally) analyzed extensively as how they are composed from occupied and virtual fragment orbitals. This inherently implies a large amount of output. Even computations on small molecules may produce startlingly many pages of output. This is not necessarily so because you can regulate the production of output in detail. Obviously, some kind of *default* production of output had to be implemented. The field of ADF users is so wide and diverse that it is hard to satisfy everybody as regards this default level of output. Depending on your purposes the automatic settings, which determine how much output is generated without instructions to the contrary, may yield boringly many numbers that you just skip through in search for the one value you're interested in, or it may be widely insufficient. Therefore, take notice of the possibilities to regulate output.

Above all, however get familiar with the analysis tools that ADF provides to see in what ways these may help to interpret your results. In a later chapter a global description of output is given as it is normally produced. The chapter below gives an introduction in some of the essential features of ADF, which may be sufficiently different from what you are used to in other Quantum Chemistry codes to deserve your attention.

1.10 Feature List

1.10.1 Model Hamiltonians

- *XC energy functionals and potentials* (page 71)
  - *LDA* (page 72), *GGA* (page 73), *meta-GGA* (page 76), *model potentials* (page 87)
  - *(meta-)hybrid* (page 77), *range-separated* (page 78), *double-hybrids* (page 82)
  - *dispersion corrected* (page 90), *r2SCAN-3c* (page 76)
- *Many-Body Perturbation Theory* (page 329)
  - *MP2* (page 85), *RPA* (page 85)
  - *GW approximation* (page 227): G_0W_0, evGW, qsGW, G3W2
- *Relativistic effects* (page 97)
  - scalar relativity, spin-orbit coupling
  - ZORA, X2C
- Solvents and other environments
  - *COSMO* (page 102), *SM12* (page 108), *DIM/QM* (page 111), *QM/FQ* (page 122), *FDE* (page 127), *3D-RISM* (page 137)
  - Quild, QM/MM, QM/QM (see the Hybrid engine)
- Homogeneous electric field and point charges (see AMS driver)

1.10.2 Structure and Reactivity

- Geometry Optimizations (see AMS driver)
- Linear transit, PES scan, Transition state search, Nudged Elastic Band (NEB) (see AMS driver)
- Intrinsic Reaction Coordinate (IRC) (see AMS driver)
- *Excited state optimizations with TDDFT gradients* (page 185) (see also AMS driver)

Optimizations can be done in Cartesian and delocalized coordinates. Various restraints and constraints can be imposed. Hessians are available analytically for most GGAs, and numerically otherwise. Preoptimization is possible, for example, with DFTB.

1.10.3 Spectroscopic properties

- Vibrational Spectroscopy (see AMS driver)
  - IR frequencies and intensities (see AMS driver)
  - Mobile Block Hessian (MBH), Vibrational Circular Dichroism (VCD) (see AMS driver)
  - Raman intensities (see AMS driver)
  - Resonance Raman frequency-dependent polarizabilities or excited state gradients (see AMS driver)
  - vibrational Raman optical activity (VROA) (see AMS driver)
  - Franck-Condon Factors (see AMS driver)
- *Excitation energies: UV/Vis spectra, X-ray absorption, CD, MCD* (page 160)
  - *UV/Vis spectra, oscillator strengths* (page 160), *open shell excitations* (page 165), *spin-orbit coupled excitations* (page 171), *GW-BSE* (page 162)
  - *core excitations* (page 166), *Quadrupole oscillator strengths* (page 170), *XES* (page 170)
  - *TD-DFT+TB* (page 177), *sTDA, sTDDFT* (page 178)
  - vibrationally resolved electronic spectra (see AMS driver)
  - *excited state optimizations* (page 185) (see also AMS driver)
  - *CD spectra* (page 179), *MCD* (page 180)
  - *LFDFT* (page 200)
- *(Hyper-)Polarizabilities, dispersion coefficients, ORD, magnetizabilities, Verdet constants* (page 188)
  - *frequency-dependent (hyper)polarizabilities* (page 189), *lifetime effects* (page 191)
  - *PolTDDFT* (page 196)
  - *van der Waals dispersion coefficients* (page 198)
  - *optical rotatory dispersion (ORD)* (page 191)
  - *magnetizability* (page 191)
  - *Verdet constants, Faraday terms* (page 191)
- NMR
  - *chemical shifts* (page 205)
  - *spin-spin couplings* (page 213)

- ESR (EPR)
  - *g-tensors (g-factor)* (page 222)
  - *A-tensor (hyperfine interaction)* (page 222)
  - *zero-field splitting (ZFS, D-tensor)* (page 222)
- *Nuclear quadrupole interaction (EFG), ESR Q-tensor* (page 226)
- *Mössbauer* (page 226), *NRVS* (page 227)

1.10.4 Charge transport properties

- *charge transfer integrals* (page 239)
- *Non-self-consistent Green's function calculation* (page 244)

1.10.5 Analysis

- *Fragments* (page 3)
- *Bond energy decomposition* (page 22), *ETS-NOCV* (page 260)
- Advanced charge density and MO analysis
  - *Mulliken* (page 285), *Multipole-derived charges* (page 287)
  - *Hirshfeld charges, Voronoi deformation density* (page 285), *CM5 charges* (page 287), *FOD* (page 273)
  - *bond orders: Mayer, Nalewajski-Mrozek* (page 287)
  - *Bader (QT-AIM)* (page 267), *Conceptual DFT* (page 270)
  - *NBO 6.0* (page 262)
  - *(partial) DOS* (page 305)
- *Molecular symmetry* (page 20), *Schönfliess symbols and symmetry labels* (page 380)

1.10.6 Accuracy and Efficiency

- *Slater-type basis sets* (page 19)
  - *Z = 1 to 120, all electron, frozen-core, non-relativistic and relativistic* (page 37)
  - *SZ, DZ, DZP, TZP, TZ2P, QZ4P, even-tempered, diffuse* (page 37)
- *Integration scheme* (page 320)
- Parallelization (see AMS driver)
- *Linear scaling / distance cut-offs* (page 339)
- *Density fit* (page 325) and *frozen core approximation* (page 19)
- SCF convergence: *simple damping, DIIS, EDIIS, ADIIS, LISTi, ARH* (page 316)

1.11 Technical remarks, Terminology

A few words about ADF as regards its technical setup and the names and abbreviations used in this manual. References to these will be made in the discussion of output and print switches.

1.11.1 Density functional theory

The underlying theory of the ADF engine is the Kohn-Sham approach to the Density-Functional Theory (DFT). Kohn-Sham DFT is an important first-principles computational method to predict chemical properties accurately and to analyze and interpret these in convenient and simple chemical terms.

The reasons for its popularity and success are easy to understand. In the first place, the DFT approach is in principle exact. In particular, the Kohn-Sham method implies a one-electron picture of the many-electron systems but yields in principle the exact electron density (and related properties) and the total energy. The exact exchange-correlation (XC) functional is unknown, but the currently available XC functionals provide in most cases already a ‘chemical’ accuracy of a few kcal/mol for binding energies. Moreover, the quest for more accurate ones based on a more detailed understanding of their essential properties is continuing.

In the past two decades, computational chemistry has evolved from a curiosity of theoreticians into a mainstream tool used by all types of chemists, physicists and engineers who have an interest in research and development. In that time Density Functional Theory has become the dominant method for modeling chemistry at the molecular level.

In the second place, it preserves at all levels of approximation the appealing one-electron molecular orbital (MO) view on chemical reactions and properties. The computed orbitals are suitable for the typical MO-theoretical analyses and interpretations. The KS method effectively incorporates all correlation effects.

In the third place, it is a relatively efficient computational method, and its fundamental scaling properties do not deteriorate when methodological precision is increased, in particular, when a more accurate XC functional is applied. Recent research paves the way to implementations that scale only linearly with the system size. This brings within reach the treatment by fundamental quantum chemical methods of systems with hundreds, maybe even thousands of atoms.

DFT gives superior accuracy to Hartree-Fock theory and semi-empirical approaches, and it is well suited for molecules containing metal atoms. In contrast to conventional ab initio methods (MP2, CI, CC), it enables accurate treatment of systems with several hundreds of atoms (or several thousands with QM/MM).

Text is mostly taken from: *Chemistry with ADF*, G. te Velde, F.M. Bickelhaupt, E.J. Baerends, C. Fonseca Guerra, S.J.A. van Gisbergen, J.G. Snijders, T. Ziegler, , Journal of Computational Chemistry 22, 931 (2001) (https://doi.org/10.1002/jcc.1056)

1.11.2 The Kohn-Sham MO model

The basic postulate in Kohn-Sham DFT is that we can apply a one-electron formulation to the system of N interacting electrons by introducing a suitable local potential $V_{\mathrm{XC}}(r)$, in addition to any external potentials $V_{\mathrm{ext}}(r)$ and the Coulomb potential of the electron cloud $V_{\mathrm{C}}(r)$, and solving:

$$[T + V_{\mathrm{ext}}(r) + V_{\mathrm{C}}(r) + V_{\mathrm{XC}}(r)]\phi_i(r) = \epsilon_i \phi_i(r)$$

Here T is the kinetic energy operator. The potential $V_{\mathrm{XC}}(r)$ is the functional derivative with respect to the density ρ of $E_{\mathrm{XC}}\ [\rho]$, the exchange-correlation energy functional. The one-electron molecular orbitals (MOs) ϕ_{i} with corresponding orbital energies ϵ_i define the exact electronic charge density and give, in principle, access to all properties because these are expressible as functional of the density, in particular the energy. Moreover, they provide an intuitively appealing view of the system as being built from independent-electron orbitals with all ensuing interpretations. The exact form of the exact energy density $E_{\mathrm{XC}}(r)$, representing and incorporating all exchange and correlation (XC) effects is unknown. From general principles one can formulate conditions on what $E_{\mathrm{XC}}(r)$ should look like, and several, more and more advanced expressions have been advocated for it in the literature. Their application to real systems has been impressively successful, and it seems likely that a further increase of accuracy is a matter of time.

1.11.3 Basis functions and orbitals

Let us make a clear distinction between (basis) *functions* and *orbitals*, even where these phrases are sometimes mixed up in the traditional terminology. Orbitals are always specific combinations of the basis functions. Orbitals are related to the computed eigenfunctions of some Fock operator or Hamiltonian occurring in the run or in a related preceding calculation. Functions are merely the elementary mathematical entities in which the orbitals are expressed. A Slater Type Orbital (STO), for instance is a function, not an orbital.

The physical meaning of one-electron orbitals in DFT has often been questioned. We believe that they are useful quantities for interpretation, just like the HF orbitals. For a recent discussion see ref.[1].

See also

- ADF-GUI tutorial: basis set effects

Cartesian function sets, spurious components

ADF employs Slater-type exponential basis functions centered on the atoms. Such a function consists of an exponential part exp(-ar) and a polynomial pre-factor r^{kr} x^{kx} y^{ky} z^{kz} . A function *set* is characterized by its radial behavior (the exponential part and the power of r, kr) and by its angular momentum quantum number *l*. The functions in such a set consist of all possible combinations x^{kx} y^{ky} z^{kz} , such that kx+ky+kz=l. These are denoted the *Cartesian* spherical harmonics.

The Cartesian function sets are very suitable for computational manipulations, but they have a drawback. By inspection it is easily verified that a *d*-set consists of 6 Cartesian functions, while there can of course be only 5 true *d*-type functions among them: one (linear combination) of them is in fact an *s*-type function (x^2 +y^2 +z^2). Similarly, there are 10 *f*-type Cartesian functions, 3 of which are in fact *p*-functions. And so on. In ADF all such lower-*l* (combinations of) functions are projected out of the basis and not employed. As a consequence the basis set *size* in the sense of the number of degrees of freedom and hence the number of possible eigenfunctions of the Fock operator is smaller than the number of expansion coefficients that refer to the primitive (Cartesian) basis functions.

The abbreviation BAS is used for references to the elementary Cartesian basis functions.

Frozen core: Core Orbitals and Core Functions

To speed up the computation the innermost atomic shells are kept frozen. The frozen Core Orbitals (CO), which are solutions of a large-basis all-electron calculation on the isolated atom, are expressed in an auxiliary set of (Slater-type) basis functions cor-bas, distinct from the valence set. The core basis set and the COs expansion coefficients are stored in the basis set files.

Orthogonality of the valence Molecular Orbitals (MO) to the COs is achieved with the help of so-called Core Functions (CF). These functions are included in the valence set but they are not additional degrees of freedom. Each of the normal valence functions is combined with a linear combination of all CFs in the molecule in such a way that the transformed function (cbas) is orthogonal to all frozen COs in the molecule. There are exactly as many CFs as COs so the orthogonality condition for all valence basis functions amounts to the solution of a linear system where the number of conditions equals the number of parameters.

This aspect once more increases the discrepancy between the number of expansion coefficients of an MO and the number of MOs: the expansion coefficients in the most elementary bas representation run over all bas functions, including the CFs

---

[1] F.M. Bickelhaupt and E.J. Baerends, *Kohn-Sham DFT: Predicting and Understanding Chemistry*, in *Reviews in Computational Chemistry*, D.B. Boyd and K.B. Lipkowitz, Editors. 2000, Wiley-VCH: New York. p. 1-86.

among them. At some places there may, alternatively, be expansions in the core-orthogonalized BAS functions, CBAS, where the CFs do not count anymore: they are included implicitly in the cbas functions.

Symmetry

The Overlap and Fock matrices become block-diagonal by using symmetry-adapted combination of the (C)BAS functions, such that each such function transforms under the symmetry operators as one of the subspecies of the irreducible representations (irrep) of the symmetry group. Symmetry adapted functions are denoted (C)SBAS.

For a given irrep and subspecies not all elementary basis functions can participate in the symmetry adapted combinations. For instance, for an atom in a reflection plane a basis function that is anti-symmetric with respect to the reflection cannot be part of any symmetric combination of functions. In particular for higher symmetries the number of BAS functions that are relevant for a subspecies may be considerably smaller than the total number of BAS functions. This is used to cut down expansion lengths, both as used internally in the computation and construction of the Fock matrix, and in printed output. The printed expansion coefficients (in the bas representation) refer only to the participating BAS functions. A defining list of them is printed at an early stage of the run for each of the subspecies.

Orthonormal basis

It is often computationally convenient to use an orthonormal basis. This is constructed from the CSBAS basis by a Lowdin orthogonalization procedure. The resulting symmetry-adapted orthonormal basis is denoted low.

The MOs are computed by diagonalization of the Fock matrix in the LOW representation. The resulting eigenvectors are easily transformed back to any other representation whenever suitable, such as for instance to the primitive Cartesian bas representation (including the CFs).

Fragments

Except in Create mode, where a *basic atom* is constructed, the system is built up from fragments and the corresponding fragment files are attached to the run. The ADF engine reads from the files the fragment MOs and these are used as (compound) *basis functions* for the molecular calculation. The fragment MOs are called Fragment Orbitals (FO).

FOs belong of course to one of the symmetry representations of the fragment, but not necessarily to a symmetry representation of the new molecule. The FOs are therefore combined into symmetry-adapted combinations, SFOs, to serve as a symmetry-adapted basis in the molecule. These combinations may involve one or more FOs from the same fragment and/or from different fragments. In the latter case the fragments must be symmetry related by one of the operators of the molecule. Symmetry related fragments must of course be identical, apart from their spatial location: they must be of the same fragment *type*.

FOs are naturally orthogonal to the Core Orbitals of their own fragment, but not necessarily to COs of other fragments. By a suitable combination of the SFOs with all CFs in the molecule we obtain the core-orthogonalized symmetry-adapted CSFOs.

The CSFOs can be transformed to an orthonormal basis by a Lowdin transformation. The resulting basis is called low, as above.

Summary of functions and orbitals

In Create mode the (conceptual) approach is:

BAS → (core-orthogonalization) → CBAS → (symmetry) → CSBAS → (orthonormality) → LOW → (Fock diagonalization) → MO

In Fragment mode:

FO (=MO from fragment file) → (symmetry) → SFO → (core-orth.) → CSFO → (orthonormality) → LOW → (Fock diagonalization) → MO

Acronyms

BAS Elementary Cartesian basis functions, consisting of a radial part (exponential factor and power of r) and an angular part (Cartesian spherical harmonic). The complete BAS set contains spurious lower-*l* combinations; these combinations are projected out and not used in the calculation. The BAS set contains also Core Functions.

SBAS Symmetry-adapted combination of BAS functions.

CF Core Function, part of the bas set. The CFs do not represent degrees of freedom in the basis set but serve only to ensure orthogonalization of the valence space to all frozen Core Orbitals.

CBAS Core-orthogonalized elementary basis functions: the true valence (not-CF) BAS functions transformed by adding a suitable combination of the CFs. The total number of CBAS + the total number of of CFs equals the total number of BAS.

CSBAS Symmetry-adapted combination of cbas functions.

CO Frozen Core Orbitals, expressed as linear combinations of an auxiliary corbas basis set. The corbas set plays no role in the further discussion. The corbas functions are *not* the CFs.

The number of COs equals the number of CFs.

LOW Lowdin orthonormalized symmetry-adapted core-orthogonalized basis. In Create mode they are derived directly from the BAS functions, in Fragment mode from the Fragment Orbitals, which are themselves of course expressible in the BAS set.

FO Fragment Orbital: the MO of a fragment calculation, now used as a *basis function* in the molecule of which the fragment is part.

SFO Symmetry adapted combination of FOs.

CSFO Core-orthogonalized SFO.

1.11.4 Coulomb potential evaluation, density fitting

Using Slater-type basis functions yields awkward multi-center integrals in the evaluation of the Coulomb potential. We therefore first need to find an approximate density-representation for which the Coulomb integral can be evaluated efficiently. This procedure is commonly referred to as density fitting. The default density fitting procedure in ADF is described in Ref.[2].

One of the most important properties of a molecule is its energy, or its bonding energy with respect to the constituent fragments. The approximate nature of the fitting procedure introduces two types of errors. The first is that, since the Coulomb potential is only approximated, the SCF solution itself, i.e. the set of self-consistent Molecular Orbitals and their energy eigenvalues may be slightly wrong, yielding an error in the charge density and hence in the energy. Since the energy is to first order stable with respect to changes in the mo coefficients this error in the energy can be assumed very

---

[2] M. Franchini, P.H.T. Philipsen, E. van Lenthe, L. Visscher, *Accurate Coulomb Potentials for Periodic and Molecular Systems through Density Fitting*, Journal of Chemical Theory and Computation 10, 1994 (2014) (https://doi.org/10.1021/ct500172n)

small. The second type of error derives from the computation of the energy from the (self-consistent) charge density, via the Coulomb potential. Let

$$\rho \equiv \rho_{exact}(r) = \rho_{fit}(r) + \delta(r)$$

and

$$V_{fit}(r) = \int \rho_{fit}(r')/|r - r'|dr$$

For the Coulomb energy of the charge density we have

$$\begin{aligned} 2E_{Coul} &= \iint \rho(r)\rho(r')/|r - r'|drdr' = \int \rho(r)V_{fit}(r)dr + \iint \rho(r)\delta(r)/|r - r'|drdr' \\ &= \int V_{fit}(r)(\rho(r) + \delta(r))dr + \iint \delta(r)\delta(r')/|r - r'|drdr' \end{aligned}$$

from which we see that the fit error is corrected to first order (by adding the fit deficiency $\delta(r)$ to the exact charge density when integrating against the fit potential) and that only a second order term remains that cannot be evaluated, the last term in the right-hand-side of the last equation.

A fair impression of the fit quality and the importance of the second order error term is obtained by checking

- the size of the first order correction term $\int V_{fit}(r)\delta(r)dr$ and
- the norm of the deficiency function, $\int \delta^2(r)dr$.

Both are printed in standard output, at the end of the output of the SCF procedure computational report. They are usually very small, which gives some confidence that the second order fit error can be ignored.

1.11.5 Three-step build-up of the bonding

The approach of ADF is based on fragments. This applies not only in the analysis at the end of the computation but also in the set-up of the ADF engine. The computation of the molecule from its constituent fragments takes place in three steps, and these are reflected in the analysis of bond energy components.

First, the (free, unrelaxed) fragments are placed at their positions in the molecule. This implies an *electrostatic* interaction: for each fragment the Coulomb interaction of its undisturbed charge density with the fields of the other fragments.

Next, the Pauli exclusion principle is applied. Even without considering self-consistency the one-electron orbitals of the combined fragments cannot represent a correct one-determinant wave function because the orbitals of different fragments are not orthogonal to one another. The ADF engine performs an orthonormalization of the occupied Fragment Orbitals to obtain an antisymmetrized product. This implies a change in the total molecular charge density from the sum-of-fragments to what is called the sum-of-*orthogonalized*-fragments. The corresponding (repulsive) energy term is evaluated separately and is called *Exchange* repulsion, or alternatively *Pauli* repulsion. The phrase *orthogonal(ized) fragments*, if you find it elsewhere in this manual or in the source code of ADF, refers to this aspect. The sum of Pauli repulsion and electrostatic interaction is called the *steric interaction*.

The third phase is the relaxation to self-consistency, with of course the ensuing contributions to the bond energy.

1.11.6 Transition State procedure

This phrase stands for an analysis method described in ref.[4] and has no relation to transition states in chemical reactions. An extensive discussion of bond energy analysis by ADF is given in ref.[5][6].

The energy associated with a change in charge density, say the relaxation to self-consistency from the sum-of-orthogonal-fragments, can be computed by subtracting final and initial energies, each obtained from the corresponding charge density. For purposes of analysis the change in energy dE can be reformulated as

$$dE = \int dr \left((\rho_{\text{final}}(r) - \rho_{\text{initial}}(r)) \int_{\rho_{\text{initial}}}^{\rho_{\text{final}}} d\rho F[\rho(r)] \right) \quad (1.2.8)$$

F(ρ) is the Fock operator belonging to the charge density ρ

By writing the density difference ρ_{final} - ρ_{initial} a summation over contributions from the different irreducible representations Γ of the molecular symmetry group, an expression is obtained that lends itself for a decomposition of the bond energy into terms from the different symmetry representations:

$$dE = \sum_{\Gamma} \int dr \left(\Delta\rho^{\Gamma}(r) \int_{\rho_{\text{initial}}}^{\rho_{\text{final}}} d\rho F[\rho(r)] \right) \quad (1.2.9)$$

The integral of the Fock operator over the charge density is now approximated by a weighted summation (in fact, a Simpson integration):

$$\int_{\rho_{\text{initial}}}^{\rho_{\text{final}}} d\rho F[\rho(r)] \approx \frac{1}{6} F[\rho_{\text{initial}}] + \frac{2}{3} F[\rho_{\text{average}}] + \frac{1}{6} F[\rho_{\text{final}}] \quad (1.2.10)$$

where

$$\rho_{\text{average}} = \frac{1}{2}(\rho_{\text{initial}} + \rho_{\text{final}}) \quad (1.2.11)$$

The term with the Fock operator due to the average charge density has given rise to the phrase *transition state*. To avoid confusion we will often refer to it as to the transition *field*.

The approximate integral (1.2.10) involves two errors. The first, rather obvious, is the approximation of the exact integral in (1.2.9) by the weighted sum in (1.2.10). Except in pathological cases this approximation is highly accurate.

The second error comes from the fact that the Coulomb and XC potentials in the Fock operator are computed from the *fit* density. This is only an approximation to the true density, while in the original bond-energy expression (energy due to the final density minus energy due to the initial density) no potentials occur and the *exact* charge density can be used. As mentioned before, these fit-related errors are usually small. For the XC potential the true density can be used if one includes the keyword EXACTDENSITY.

All such errors in the total bonding energy are easily corrected by comparing the summation over the Γs with the correct value for the total bonding interaction term. The difference is simply added to the total bond energy, so no true error remains. We only have a (correction) term that can't be split in contributions from the distinct symmetry representations. In the printed bond energy analysis such small corrections are 'distributed' over the other terms by scaling the other terms such that their sum is the correct total value.

---

[4] T. Ziegler and A. Rauk, *On the calculation of Bonding Energies by the Hartree Fock Slater method. I. The Transition State Method*, Theoretica Chimica Acta 46, 1 (1977) (https://doi.org/10.1007/BF00551648)

[5] P.J. van den Hoek, A.W. Kleyn and E.J. Baerends, *What is the origin of the repulsive wall in atom-atom potentials*, Comments Atomic and Molecular Physics 23, 93 (1989)

[6] E.J. Baerends, *Pauli repulsion effects in scattering from and catalysis by surface*, in *Cluster models for surface and bulk phenomena*, ISBN13: 9780306441028, G. Puccchiori, P.S. Bagus and F. Parmigiani, Editors. 1992, Springer: New-York. p. 189-207.

CHAPTER

TWO

INPUT, EXECUTION, AND OUTPUT

2.1 Input

The input options for ADF are specified in a text file consisting of a series of key-value pairs, possibly nested in blocks. Much of the general remarks about input for ADF apply also to related property and analysis programs, which are also described in this document. The input for the description of the system, structure and reactivity, vibrational spectroscopy, belongs (mostly) to the AMS driver part of the input.

- Input, execution and output section of the AMS manual

The input is usually embedded in an executable shell script. This is the content of a typical shell script for running a ADF calculation:

```
#!/bin/sh

$AMSBIN/ams <<eor
   # This is the beginning of the input.
   # The input consists of key-value pairs and blocks.
   # Here we define the input option for the AMS driver:

   Task GeometryOptimization

   System
      Atoms
         H 0.0 0.0 0.0
         H 0.9 0.0 0.0
      End
   End

   # Next comes the ADF "Engine" block. The input options for ADF, which are
   # described in this manual, should be specified in this block:

   Engine ADF
      Basis
         Type DZP
      End

      XC
         GGA PBE
      End
   EndEngine
eor
```

To run the calculation above from command-line you should:

1. Create a text file called, for example, `test.run` and copy-paste the content of the script above
2. Make the script executable by typing in your shell `chmod u+x test.run`
3. Execute the script and redirect the output to a file: `./test.run > out`

The program will create a directory called `ams.results`. Inside it, you will find the *logfile* `ams.log` (which can be used to monitor the progress of the calculation) and the binary results files `ams.rkf` and `adf.rkf`. After the calculation is completed, you can examine the output file `out`. For more details, see

- Input, execution and output section of the AMS manual

See also:

The *Examples* (page 401) section contains a large number of input examples.

Important: Most options described in this manual should be specified in the ADF Engine block:

```
# All ADF keywords should be specified inside the 'Engine ADF' block
Engine ADF
   Basis
      Type DZP
   End

   XC
      GGA PBE
   End
EndEngine
```

2.1.1 General remarks on input structure and parsing

- Most keys are optionals. Defaults values will be used for keys that are not specified in the input
- Keys/blocks can either be *unique* (*i.e.* they can appear in the input only once) or *non-unique*. (i.e. they can appear multiple times in the input)
- The order in which keys or blocks are specified in the input does not matter. Possible exceptions to this rule are a) the content of non-standard blocks b) some non-unique keys/blocks)
- Comments in the input file start with one of the following characters: `#`, `!`, `::`

```
# this is a comment
! this is also a comment
:: yet another comment
```

- Empty lines are ignored
- The input parsing is **case insensitive** (except for string values):

```
# this:
UseSymmetry false
# is equivalent to this:
USESYMMETRY FALSE
```

- Indentation does not matter and multiple spaces are treaded as a single space (except for string values):

```
# this:
     UseSymmetry     false
# is equivalent to this:
UseSymmetry false
```

- Environment variables in the input will be replaced by their value. We only support simple substitution of variables (optionally surrounded by `{}`). All other parameter expansion features of the POSIX shell (https://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html#tag_18_06_02) (e.g. default values) are **not** supported. Expansion of a variable can be prevented by prefixing the leading dollar (`$`) with a backslash (`\`). Note that variables which are not set will expand to an empty string, just like in a shell:

```
File $SOME_DIRECTORY/file.txt
File ${SOME_DIRECTORY}_BACKUP/file.txt
Key  \$WILL_NOT_EXPAND
```

2.1.2 Keys

Key-value pairs have the following structure:

```
KeyName Value
```

Possible types of keys:

bool key The value is a single Boolean (logical) value. The value can be `True` (equivalently `Yes`) or `False` (equivalently `No`.). Not specifying any value is equivalent to specifying `True`. Example:

```
KeyName Yes
```

integer key The value is a single integer number. Example:

```
KeyName 3
```

float key The value is a single float number. For scientific notation, the E-notation is used (e.g. -2.5×10^{-3} can be expressed as `-2.5E-3`). The decimal separator should be a dot (`.`), and **not** a comma (`,`). Example:

```
KeyName -2.5E-3
```

Note that fractions (of integers) can also be used:

```
KeyName 1/3    (equivalent to: 0.33333333333...)
```

string key The value is a string, which can include white spaces. Only ASCII characters are allowed. Example:

```
KeyName Lorem ipsum dolor sit amet
```

multiple_choice key The value should be a single word among the list options for that key (the options are listed in the documentation of the key). Example:

```
KeyName SomeOption
```

integer_list key The value is list of integer numbers. Example:

```
KeyName 1 6 0 9 -10
```

Note that one can also specify ranges of integers by specifying the interval and (optionally) the step size separated by colons:

```
KeyName 1:5        (equivalent to: 1 2 3 4 5)
KeyName 2:10:2     (equivalent to: 2 4 6 8 10)
KeyName 20:10:-2   (equivalent to: 20 18 16 14 12 10)
```

Note also that ranges can be freely combined with individual numbers:

```
KeyName 1:5 10 20  (equivalent to: 1 2 3 4 5 10 20)
```

float_list key The value is list of float numbers. The convention for float numbers is the same as for Float keys. Example:

```
KeyName 0.1 1.0E-2 1.3
```

Float lists can also be specified as a range with equidistant points, by specifying the interval's boundaries (inclusive) as well as the number of desired subintervals separated by colons:

```
KeyName 1.0:1.5:5  (equivalent to: 1.0 1.1 1.2 1.3 1.4 1.5)
```

Range specifications can be freely combined with each other and single numbers:

```
KeyName 0.0 1.0:1.5:5 2.0:3.0:10
```

2.1.3 Blocks

Blocks give a hierarchical structure to the input, grouping together related keys (and possibly sub-blocks). In the input, blocks generally span multiple lines, and have the following structure:

```
BlockName
   KeyName1 value1
   KeyName2 value2
   ...
End
```

Headers

For some blocks it is possible (or necessary) to specify a *header* next to the block name:

```
BlockName someHeader
   KeyName1 value1
   KeyName2 value2
   ...
End
```

Compact notation

It is possible to specify multiple key-value pairs of a block on a single line using the following notation:

```
# This:
BlockName KeyName1=value1 KeyName2=value2

# is equivalent to this:
BlockName
   KeyName1 value1
   KeyName2 value2
End
```

Notes on compact notation:

- The compact notation cannot be used for blocks with headers.
- Spaces (blanks) between the key, the equal sign and the value are ignored. However, if a value itself needs to contain spaces (e.g. because it is a list, or a number followed by a unit), the entire value must be put in either single or double quotes:

```
# This is OK:
BlockName Key1=value Key2 = "5.6 [eV]" Key3='5 7 3 2'
# ... and equivalent to:
BlockName
   Key1  value
   Key2  5.6 [eV]
   Key3  5 7 3 2
End

# This is NOT OK:
BlockName Key1=value Key2 = 5.6 [eV] Key3=5 7 3 2
```

Non-standard Blocks

A special type of block is the *non-standard block*. These blocks are used for parts of the input that do not follow the usual key-value paradigm.

A notable example of a non-standard block is the `Atoms` block (in which the atomic coordinates and atom types are defined).

2.1.4 Including an external file

You can include an external ASCII file in the input with the `@include` directive:

```
@include FileName.in
@include "file name with spaces.in"
```

The file name should include the path, either absolute or relative to the run-directory. The content of the file is included in the input at the point where the `@include` directive occurs. The `@include` directive may occur any number of times in the input.

The `@include` feature makes it easy to pack your preferred settings in one file and use them in every run with minimum input-typing effort.

2.1.5 Units

Some keys have a default unit associated (not all keys have units). For such keys, the default unit is mentioned in the key documentation. One can specify a different unit within square brackets at the end of the line:

```
KeyName value [unit]
```

For example, assuming the key `EnergyThreshold` has as default unit `hartree`, then the following definitions are equivalent:

```
# Use defaults unit:
EnergyThreshold 1.0

# use eV as unit:
EnergyThreshold 27.211 [eV]

```

(continues on next page)

(continued from previous page)

```
# use kcal/mol as unit:
EnergyThreshold 627.5 [kcal/mol]

# hartree is the atomic unit of energy:
EnergyThreshold 1.0 [hartree]
```

Available units:

- **Energy**: `hartree`, `joule`, `eV`, `kJ/mol`, `kcal/mol`, `cm1`, `MHz`
- **Forces**: `hartree/bohr`, `eV/bohr`, `hartree/angstrom`, `eV/angstrom`, `kcal/mol/angstrom`, `kJ/mol/angstrom`
- **Length**: `bohr`, `angstrom`, `meter`
- **Angles**: `radian`, `degree`
- **Mass**: `el`, `proton`, `atomic`, `kg`
- **Pressure**: `atm`, `pascal`, `GPa`, `a.u.`, `bar`, `kbar`
- **Electric field**: `V/angstrom`, `V/meter`, `a.u.`

2.1.6 Title, comment

```
TITLE Title
```

Title may be any string. The program combines it (that is, the first approximately 50 characters) with date and time of the job to construct the* job identification.* The job identification is used to stamp an identification on result files, which will be read and printed if such a file is used again, for instance as a fragment file.

The job identification will also be echoed in the output header to identify the current run. By default the date and time are combined with a dummy string. In Create mode the title is first read from the data file that supplies the basis functions etc and can then be overwritten via input.

Note that, contrary to some other programs, ADF does *not* take the first input record as a title.

You can put more remarks in the input file to be echoed in the standard output file; these will not become part of the job identification:

```
COMMENT
   text
   ...
end
```

The text records are copied to the output header, directly after the job identification.

The key COMMENT may occur any number of times; all text blocks are printed in the output header with a blank line between any two text blocks.

2.1.7 Amsification of ADF in ADF2020

The standalone program 'adf' does not exist anymore in ADF2020. The job of AMS driver is to handle all changes in the geometry, e.g. during a geometry optimization, using so-called engines like ADF for the calculation of energies and forces. Details of this change can be found in the section on the

- *AMSification of ADF* (page 9)

2.1.8 Input parsing changes in ADF2018 and later

The input file parsing for ADF and its properties programs has changed from the 2017 to the 2018 version.

New syntax for a few keywords

In order to adapt ADF to the new (more strict) input format, the syntax of a few keywords had to be changed. The following table contains the list of keywords that have changed in ADF2018. Note that some keywords changed in AMS2020 again, see the *section on ADF amsification* (page 9). Note also that the block key DEFINE was removed.

key in ADF2017 and before	key in ADF2018 and later / comments
CorePotentials	*CorePotentials* (page 62) is now a block (and not an *general* key/block)
Define	removed
Dependency	*Dependency* (page 335) is now a block (and not a key)
EField	ADF2020: key 'System%ElectroStaticEmbedding' in AMS driver
ETSNOCV	*ETSNOCV* (page 261) is now a block (and not a key)
Excitations -> Davidson	*Excitations -> Davidson* (page 160) is now a block (and not a key)
Excitations -> Exact	*Excitations -> Exact* (page 160) is now a block (and not a key)
Geometry	ADF2020: key 'Task' and key 'Properties' in AMS driver
Integration Value	*Integration -> Accint Value* (page 322)
LinearScaling	*LinearScaling* (page 339) is now a block (and not an *general* key/block)
ModifyStartPotential	*ModifyStartPotential* (page 64) is now a block (and not an *general* key/block)
Occupations	Split into *Occupations* (page 59) and *IrrepOccupations* (page 61)
Relativistic	ADF2020: block key *Relativity* (page 97)
Restart -> Value	*Restart -> File -> Value* (page 344)
Response -> Units	Removed optional units for `FrqBeg` and `FrqEnd` in `Response` block
SCF -> ADIIS	*SCF -> ADIIS* (page 316) is now a block (and not a key)
SCF -> ARH	*SCF -> ARH* (page 319) is now a block (and not a key)
SCF -> DIIS	*SCF -> DIIS* (page 316) is now a block (and not a key)
SlaterDeterminants	*SlaterDeterminants* (page 370) is now a block (and not an *general* key/block)
SOPert	*SOPert* (page 171) is now a block (and not a key)
Thermo	ADF2020: key 'Thermo' in AMS driver

Strict parsing of input file

In ADF2018 and later **exact keyword matching** is used, meaning that **keywords abbreviations (or extensions) are not allowed**. In ADF2017 (and previous versions) the parsing of the input file was *tolerant* and it would allow for abbreviations and extension of keywords.

In the example below, only the first version is allowed in ADF2018 and later, while the second and third version will trigger an input syntax error:

```
# This is the proper input syntax:
SCF
   Converge 1.0E-7
End

# In ADF2017 you could (for some keywords) use abbreviations. e.g.:
SCF
   Conv 1.0E-7
End

# or extensions. e.g.:
SCF
   Convergence 1.0E-7
End

# Keywords abbreviations and extensions are NOT ALLOWED in ADF2018 and later.
```

2.2 Execution

See

- Input, execution and output section of the AMS manual

2.3 Structure of the Output

AMS produces two ASCII files: standard output and the ams.log file. ADF writes part of these files. See also

- Input, execution and output section of the AMS manual

The ams.log file is a very concise summary of the calculation's progress during the run. Furthermore, ADF produces and reads binary data files. Most of these files have the so-called KF format. KF stands for Keyed File: KF files are keyword oriented, which makes them easy to process by simple procedures. KF files are Direct Access binary files. Consult the Scripting Section for information about how to use some standard utilities for processing KF files.

The graphical user interface ADF-GUI provides graphical representations of calculated data fields:

See also:

GUI overview tutorials

ADF-GUI tutorials

2.3.1 Job Characteristics on standard Output

Input Echo, Output Header

- Header with the program name, the release number and a copyright statement.
- AMS part of the output file with the geometry
- Copy of the ADF engine part of the input file
- Title and any comments that may have been supplied via input (key COMMENT).

Main Job Characteristics

- The Model Parameters such as the Density Functional and relativistic options.
- A list of attached files: restart data files and fragment files.
- (Initial) geometric data: atomic positions, atom types, defined fragments, and the inter-atomic distance matrix.
- The point group symmetry, with a list of the irreducible representations and subspecies.
- The electronic configuration: occupation numbers (if specified), their distribution over spin-α and spin-β, and the net charge of the molecule.

Build Info: Fragments and Function Sets

- Correspondence between fragments in the molecule and the corresponding *master* fragments on the pertaining fragment file. (This output is by default off)
- SFOs: the Symmetry combinations of Fragment Orbitals. The SFOs are the basic conceptual entities for the analysis of MOs and other results.
- The elementary basis functions and the frozen-core levels of the atoms.

Technical Parameters

- Parallelization and vectorization characteristics.
- Update strategy parameters for the SCF procedure.
- General precision settings for numerical integration and neglect-of-small function values (in integral evaluations).

Computational Report

- Numerical integration parameter(s) and the number of generated (symmetry unique) integration points, and partitioning of the points in blocks.
- SCF procedure, at each cycle: for each irreducible representation: the one-electron orbital energies and the occupation numbers for a contiguous sequence of orbitals.
- Gross atomic charges, computed from a Mulliken population analysis.
- Property specific output, like spectroscopic properties.
- AMS part of the output

Exit Procedure

- Normal termination or an error message.

2.3.2 Log file, adf.rkf, TAPE13

The ams.log file (logfile) is generated during the calculation and flushed after (almost) each message that is sent to it by the program. Consequently, the user can inspect it and see what is going on without being delayed by potentially large system I/O buffers. Each message contains date and time of the message plus additional info.

A major part of the messages simply states the name of a procedure. Such messages are sent when the procedure is entered. During the SCF procedure, the SCF errors, which are a measure for non-self-consistency, are written at every cycle. In calculations where the geometry is changing (optimization, frequencies…) each set of new coordinates is sent to the log file (Cartesian, in angstrom and also Z-matrix, if a Z-matrix structure was provided in the input file). Other messages should be self-explanatory.

Be alert on error messages. Take them seriously: inspect the standard output carefully and try to understand what has gone wrong. Be also alert to warnings. They are not necessarily fatal but you should understand what they are about before being satisfied with the results of the calculation. Do not ignore them just because the program has not aborted: in some cases the program may not be able to determine whether or not you really want to do what appears to be wrong or suspicious. If you believe that the program displays erratic behavior, then the standard output file may contain more detailed information. Therefore, in such case save the complete standard output file, together with the logfile, in case we need these files for further analysis.

adf.rkf

adf.rkf is the general result file of an ADF calculation. It is a KF file: Direct-Access, binary, and keyword driven. It contains information about the calculation. You can use it as a fragment file in a subsequent calculation on a bigger molecule, where the current one may be a part, or in an analysis or property program. For more information on adf.rkf, see *Appendix on adf.rkf* (page 383).

TAPE13

TAPE13 is the checkpoint file for restarts after a crash. It is a concise version of adf.rkf, containing only the data the program uses for restarting the calculation. See the RESTART keyword. Like adf.rkf, TAPE13 is a binary, keyword driven KF file, see *Appendix on TAPE13* (page 385).

CHAPTER

THREE

COORDINATES, BASIS SETS, FRAGMENTS

See also:
ADF-GUI tutorial: building molecules , basis set effects , fragments

3.1 Atomic coordinates

See also

- Geometry, System definition section of the AMS manual

3.1.1 Cartesian

You can define the input atomic coordinates in the `Atoms` sub block key of the `System` key in AMS. Note that in AMS one can not add an integer by which the atoms are numbered.

```
System
   Atoms
      AtomType Coords {adf.F=Fragment}
      ...
   End
End
```

`AtomType` The name of an *atom type*. It must begin with the standard one- or two-character symbol for the chemical element (*e.g.* H, He, Li,...). Optionally it may be appended by .text, where text is any string (not containing delimiters). Examples: `H`, `Mn.3`, `Cu.dz-new`.

Note: *Atom type* is not identical to *chemical element*. An atom type is defined by all characteristics of the basic atom to which it in fact refers: the nuclear charge, the basis functions, the frozen core, the density functional and any other features that were applied in generating that basic atom.

`Coords` The xyz coordinates of the atom Example:

```
ATOMS
   O  0.000000  0.000000   0.000000
   H  0.758602  0.000000   0.504284
   H  0.758602  0.000000  -0.504284
END
```

`adf.F=Fragment` Specifies that the atom belongs to a particular fragment. The fragment name must be of the form `fragtype|n`, where fragtype is the name of one of the types of fragments in the molecule. The integer n, after the pipe `|`, counts the individual fragments of that type. The numbering suffix `|n` is not required if there is only one fragment of that type.

When `adf.f=fragment` is omitted altogether, the fragment type is taken to be the *atom type* that was specified earlier on the same line. (The numbering `|n` is then added automatically by the program, by counting the number of times that this single-atom fragment type occurs in the list of atoms).

note: Input items are generally case insensitive. Exceptions are the names of files and directories. Since (to be discussed below) the name of the fragment type as it is defined under atoms (explicitly with the `adf.f=option`, or implicitly as the name of the atom type) might also directly indicate the fragment file, the specification of fragment types is in principle case-sensitive. Errors may occur if you are sloppy in this respect. However, you must not give different fragment types names that differ only by case: at various places in the program fragment type names are compared in a case-insensitive way

3.1.2 Z-Matrix

It is possible to specify the input geometry as a Z-Matrix.

```
System
   Atoms Z-Matrix
      C
      H   1 1.089000
      H   1 1.089000  2  109.4710
      H   1 1.089000  2  109.4710  3  120.0000
      H   1 1.089000  2  109.4710  3 -120.0000
   End
End
```

Only the first atom may be a dummy atom in case of a Z-matrix, use as chemical symbol XX for the dummy atom.

3.1.3 Orientation of Local Atomic Coordinates

As discussed before the atomic positions are input with the key ATOMS. One option has thus far not been mentioned: the possibility to redefine the local coordinate frame of an atom.

```
ATOMS {type of coordinates}
   AtomType coordinates {adf.F=fragment} {Z="xx yy zz"} {X="xx yy zz"}
   ...
end
```

Except for the z= option all aspects have been examined already before.

`z="xx yy zz"` defines a reorientation of the local atomic z-axis; it is interpreted as a direction vector with components (xx,yy,zz) pointing away from the atom. In the local, reoriented frame the local atomic x-axis will be rotated to the plane defined by the directions of the molecular z-axis and the local atomic z-axis. This feature can be used only for single-atom fragments (otherwise it is ignored). Its purpose is to give more flexibility in the analysis of the final molecular orbitals in terms of the atomic orbitals. In such a case it may be very helpful to redefine the orientation of say the p-orbitals of an atom. For instance, you may orient all p-orbitals towards the origin by specifying for each atom z= -x -y -z (with x,y,z the coordinates of that atom). By default the local and molecular z-axes are identical.

`x="xx yy zz"` defines a reorientation of the local atomic x-axis; it is interpreted as a direction vector with components (xx,yy,zz) pointing away from the atom. Together with the z vector this defines the xz plane. The y axis is then given by the vector product $\vec{z} \times \vec{x}$. This is used for analysis (see orientation of the z-axis).

3.1.4 Symmetry Key

Note that this is a specification of the `Symmetry` key in the Engine ADF part of the input. You can use the System%Symmetrize key in the AMS part of the input, such that AMS will symmetrize the coordinates. ADF will not adjust coordinates. Thus input atomic coordinates that are off from their correct positions, even if they are only slightly off, are not adjusted by ADF. There exist a System%Symmetry key in the AMS part of the input, which can be used with the System%Symmetrize key to symmetrize coordinates. Note that the the symmetry used in AMS does not have to be the same as is used in ADF.

The point group symmetry and symmetry tolerance can be supplied.

```
Symmetry [AUTO | NOSYM | ATOM | C(LIN) | D(LIN) | C(I) | C(S) | C(2) | C(2V) | C(3V)␣
↪| C(4V) |
          C(5V) | C(6V) | C(7V) | C(8V) | C(2H) | D(2) | D(3) | D(4) | D(5) | D(6) |␣
↪D(7) |
          D(8) | D(2D) | D(3D) | D(4D) | D(5D) | D(6D) | D(7D) | D(8D) | D(2H) |␣
↪D(3H) |
          D(4H) | D(5H) | D(6H) | D(7H) | D(8H) | O(H) | T(D)]
```

`Symmetry`

Type Multiple Choice

Default value AUTO

Options [AUTO, NOSYM, ATOM, C(LIN), D(LIN), C(I), C(S), C(2), C(2V), C(3V), C(4V), C(5V), C(6V), C(7V), C(8V), C(2H), D(2), D(3), D(4), D(5), D(6), D(7), D(8), D(2D), D(3D), D(4D), D(5D), D(6D), D(7D), D(8D), D(2H), D(3H), D(4H), D(5H), D(6H), D(7H), D(8H), O(H), T(D)]

Description Use (sub)symmetry with this Schoenflies symbol. Can only be used for molecules. Orientation should be correct for the (sub)symmetry. Coordinates must be symmetric within SymmetryTolerance.

```
SymmetryTolerance float
```

`SymmetryTolerance`

Type Float

Default value 1e-07

Description Tolerance used to detect symmetry in the system. If symmetry Schoenflies symbol is specified, the coordinates must be symmetric within this tolerance.

The point group symmetry specified in input with a Schönfliess type symbol puts restrictions on the orientation of the atomic system. If symmetry is specified the user must take care of supplying the *Cartesian* coordinates in the appropriate orientation.

3.2 Basis sets and atomic fragments

See also:

- *What basis set should I use in ADF?* (page 349)
- *Basis set file format* (page 363)

3.2.1 STO basis sets

The basis functions used in ADF are commonly known as Slater Type Orbitals (STOs). The ADF package comes with a database of STO basis sets. The basis sets are simple ASCII files and they are located in the directory `$AMSHOME/atomicdata/ADF`. A description of the basis set file format can be found in the appendix *Basis set file format* (page 363).

A basis set can roughly be characterized by its size (single-, double-, triple-zeta; with or without polarization) and by the level of frozen core approximation. The standard basis sets available in ADF are:

- **SZ** Minimal basis sets: single-zeta without polarization. The exponents of the functions correspond to the standard STO-3g basis sets used in programs that employ Gaussian type basis functions. Type-SZ files are provided only for the lighter elements, up to Kr.
- **DZ** Double-zeta basis sets without polarization functions. A triple-zeta set is used for the 3d shells of the first row transition metals, the 4f shells of the Lanthanides, and the 5f shells of the Actinides. In all these cases a double-zeta set provides a rather poor expansion basis for the true (numerically computed) atomic orbital.
- **DZP** Double zeta polarized basis. The basis sets in DZP are derived from DZ, extended with a polarization function. This type of basis sets is thus far provided only for the elements up to Ar, and for the 4p series Ga through Kr.
- **TZP** Triple-zeta basis sets. A polarization function is added for H through Ar and for Ga through Kr (from DZP)
- **TZ2P** Triple-zeta with two polarization functions, for H through Ar and Ga through Kr (from DZP). Note that the TZ2P basis set files are provided only for the lighter elements, up to Kr. The ZORA/TZ2P basis set files are provided for all elements. Typically for all elements one polarization function is added compared to the corresponding TZP basis set. Note, however, that TZ2P will not always give you extra basis functions for most lanthanide and actinide frozen core basis sets.

In addition to the standard basis sets, the database contains directories with special basis sets:

- **TZ2P+** For transition metals Sc-Zn and lanthanides (ZORA) only: as TZ2P, but with extra d-STO (3d metals), and extra f-STO (lanthanides, ZORA)
- **ZORA** contains basis sets designed for *ZORA relativistic calculations* (page 97) (relativistic calculation have special basis set requirements, especially in the core region). ZORA basis sets with frozen core should be used **exclusively** in relativistic calculations with the ZORA approach, while all-electron ZORA basis sets can be used for both relativistic and non-relativistic calculations. The ZORA/QZ4P basis set can be loosely described as core triple zeta, valence quadruple zeta, with four sets of polarization functions.
- **ET** contains several even tempered basis sets which enables one to go to the basis set limit, such as ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE. The accuracy of the smallest basis set in this directory can loosely be described as quadruple zeta in the valence with three polarization functions added. This directory also contains basis sets with extra diffuse functions. In Response calculations one should use such large basis sets in case of small molecules. Very diffuse functions are absolutely necessary to get good results for excitation energies corresponding to high lying orbitals (Rydberg states).
- **AUG** contains several augmented standard basis sets which enables one to get reasonable results for excitation energies with relatively small basis sets, such as AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P.
- **Corr** contains several extended all electron ZORA basis sets, especially useful in MBPT calculations, Corr/TZ3P and Corr/QZ6P. For MBPT larger basis sets are needed to achieve the same accuracy as in a standard DFT calculation.

Furthermore, in `$AMSHOME/atomicdata/ADF` you will find:

- **Special/AE** contains non-relativistic basis sets for all-electron calculations. However, these files cannot be used as such, because they don't contain any fit sets. They serve as starting point for the development of (new) basis sets. For some of the all-electron sets appropriate fit sets have already been generated. The corresponding data base files can be found in the appropriate sub-directories SZ, DZ, DZP, et cetera.

- **Special/Vdiff** contains non-relativistic basis sets that include very diffuse functions. These were recommended to be used for Response calculations. Very diffuse functions are absolutely necessary to get good results for excitation energies corresponding to high lying orbitals. Recommendation: use the even tempered basis sets in the ET directory, since these basis sets are better.
- **Special/MDC** contains non-relativistic basis sets with optimized fit functions especially useful for accurate Multipole Derived Charges. These are available only for a limited number of basis sets.

The directory `$AMSHOME/atomicdata/Dirac` contains the input files for the DIRAC auxiliary program (see the RELATIVITY keyword).

Frozen core

Multiple occurrences of one chemical element in the same basis set sub-directory correspond to different levels of the *frozen core approximation* (page 19). Manganese for instance may have a basis set for an atom with a frozen 2p shell and another one with a frozen 3p shell. The file names are self-explanatory: Mn.2p stands for a data file for Manganese with frozen core shells up to the 2p level. An all-electron basis set would correspond to a file that has no frozen-core suffix in its name.

Electronic configurations specific basis sets

Another type of multiple occurrence of one element in one basis set directory may be found when basis sets have been developed for different electronic configurations: the Slater-type basis sets are fitted then to numerical orbitals from runs with different occupation numbers. Currently this applies only for Ni (in `$AMSHOME/atomicdata/ADF` `DZ`, `TZP` and `TZ2P`), where basis sets are supplied for the d8s2 and the d9s1 configurations respectively. Since in earlier releases only the d8s2 variety was available, the names of the basis set files are Ni.2p (for d8s2) and Ni_d9.2p, and likewise Ni.3p and Ni_d9.3p.

References on basis sets

- E. van Lenthe and E.J. Baerends, *Optimized Slater-type basis sets for the elements 1-118.* Journal of Computational Chemistry 24, 1142 (2003) (https://doi.org/10.1002/jcc.10255)
- D.P. Chong, E. van Lenthe, S.J.A. van Gisbergen and E.J. Baerends, *Even-tempered Slater-Type orbitals revisited: From Hydrogen to Krypton.* Journal of Computational Chemistry 25, 1030 (2004) (https://doi.org/10.1002/jcc.20030)
- D.P. Chong, *Augmenting basis set for time-dependent density functional theory calculation of excitation energies: Slater-type orbitals for hydrogen to krypton.* Molecular Physics 103, 749 (2005) (https://doi.org/10.1080/00268970412331333618)

Older references for STO basis sets are Refs.[1][2][3]. See also the paper by Raffennetti on design and optimization of even-tempered STO basis sets[4]. The paper by Del Chong describes completeness profiles as a visual tool in estimating the completeness of a basis set[5]. Finally, Zeiss and coworkers[6] describe field-induced polarization functions for STOs. These are useful for defining basis sets with diffuse functions for (hyper)polarizability and other property calculations.

The procedure for the usage and optimization of fit functions is described by Baerends et al.[7].

---

[1] E. Clementi, C. Roetti, *Roothaan-Hartree-Fock atomic wavefunctions: Basis functions and their coefficients for ground and certain excited states of neutral and ionized atoms,* $Z \leq 54$, Atomic Data and Nuclear Data Tables 14, 177 (1974) (https://doi.org/10.1016/S0092-640X(74)80016-1)

[2] A.D. McLean, R.S. McLean, *Roothaan-Hartree-Fock atomic wave functions Slater basis-set expansions for Z = 55-92*, Atomic Data and Nuclear Data Tables 26, 197 (1981) (https://doi.org/10.1016/0092-640X(81)90012-7)

[3] J.G. Snijders, P. Vernooijs, E.J. Baerends, *Roothaan-Hartree-Fock-Slater atomic wave functions: Single-zeta, double-zeta, and extended Slater-type basis sets for* $_{87}$ Fr-$_{103}$ Lr, Atomic Data and Nuclear Data Tables 26, 483 (1981) (https://doi.org/10.1016/0092-640X(81)90004-8)

[4] R.C. Raffenetti, *Eventempered atomic orbitals. II. Atomic SCF wavefunctions in terms of eventempered exponential bases*, Journal of Chemical Physics 59, 5936 (1973) (https://doi.org/10.1063/1.1679962)

[5] D.P. Chong, *Completeness profiles of one-electron basis sets*, Canadian Journal of Chemistry 73, 79 (1995) (https://doi.org/10.1139/v95-011)

[6] G.D. Zeiss, W.R. Scott, N. Suzuki, D.P. Chong, S.R. Langhoff, *Finite-field calculations of molecular polarizabilities using field-induced polarization functions: second- and third-order perturbation correlation corrections to the coupled Hartree-Fock polarizability of* H_2 O, Molecular Physics 37, 1543 (1979) (https://doi.org/10.1080/00268977900101121)

[7] E.J. Baerends, D.E. Ellis and P. Ros, *Self-consistent molecular Hartree-Fock-Slater calculations I. The computational procedure*, Chemical Physics 2, 41 (1973) (https://doi.org/10.1016/0301-0104(73)80059-X)

For documentation on how to make custom even-tempered basis/fit sets, see the old ADF 2014 documentation (https://downloads.scm.com/Doc/Doc2014/ADF/ADFUsersGuide/ADFUsersGuide.pdf).

3.2.2 Available basis sets

ADF has optimized STO basis sets for the whole periodic table (Z=1-120).

The next tables give an indication which all electron (ae) and frozen core (fc) standard basis sets are available for the different elements in ADF.

Table 3.1: Available standard basis sets for non-relativistic (non-rel) and ZORA calculations H-Kr (Z=1-36)

Element	ae or fc	SZ, DZ	DZP	TZP, TZ2P	TZ2P+	QZ4P, ET	AUG
H-He (Z=1-2)	ae	Yes	Yes	Yes		Yes	Yes
Li-Ne (Z=3-10)	ae	Yes	Yes	Yes		Yes	Yes
	.1s	Yes	Yes	Yes			non-rel
Na-Ar (Z=11-18)	ae	Yes	Yes	Yes		Yes	Yes
	.1s .2p	Yes	Yes	Yes			non-rel
K-Ca (Z=19-20)	ae	Yes	Yes	Yes		Yes	Yes
	.2p .3p	Yes	Yes	Yes			non-rel
Sc-Zn (Z=21-30)	ae	Yes		Yes	Yes	Yes	Yes
	.2p .3p	Yes		Yes	Yes		non-rel
Ga-Kr (Z=31-36)	ae	Yes	Yes	Yes		Yes	Yes
	.3p .3d	Yes	Yes	Yes			non-rel

Table 3.2: Available standard basis sets for non-relativistic calculations Rb-Cm (Z=37-96)

Element	fc	DZ, TZP
Rb-Sr (Z=37-38)	.3p .3d .4p	Yes
Y-Cd (Z=39-48)	.3d .4p	Yes
In-Ba (Z=49-56)	.4p .4d	Yes
La-Lu (Z=57-71)	.4d .5p	Yes
Hf-Hg (Z=72-80)	.4d .4f	Yes
Tl-Rn (Z=81-86)	.4d .4f .5p .5d	Yes
Fr-Ra (Z=87-88)	.5p .5d	Yes
Ac-Cm (Z=89-96)	.5d	Yes

Table 3.3: Available standard basis sets for ZORA calculations Rb-Og (Z=37-120)

Element	ae or fc	DZ, TZP, TZ2P	TZ2P+	QZ4P
Rb-Sr (Z=37-38)	ae	Yes		Yes
	.3p .3d .4p	Yes		
Y-Cd (Z=39-48)	ae	Yes		Yes
	.3d .4p	Yes		
In-Ba (Z=49-56)	ae	Yes		Yes
	.4p .4d	Yes		
La-Yb (Z=57-70)	ae	Yes	Yes	Yes
	.4d .5p	Yes	Yes	
Lu (Z=71)	ae	Yes		Yes
	.4d .5p	Yes		
Hf-Hg (Z=72-80)	ae	Yes		Yes
	.4d .4f	Yes		
Tl (Z=81)	ae	Yes		Yes
	.4d .4f .5p	Yes		
Pb-Rn (Z=82-86)	ae	Yes		Yes
	.4d .4f .5p .5d	Yes		
Fr-Ra (Z=87-88)	ae	Yes		Yes
	.5p .5d	Yes		
Ac-Lr (Z=89-103)	ae	Yes		Yes
	.4f .5d	Yes		
Rf-Cn (Z=104-112)	ae	Yes		Yes
	.4f .5d .5f	Yes		
Nh-Og (Z=113-118)	ae	Yes		Yes
	.5d .5f	Yes		
Uue-Ubn (Z=119-120)	ae	Yes		Yes
	.5f	Yes		

For heavier elements, from Rb on, the non-relativistic all electron basis sets are missing. In the ZORA basis sets directory you will find all-electron basis sets for all elements (Z = 1-120), which also could be used in non-relativistic calculations. Note, however, that these basis sets were optimized for ZORA calculations, which means that non-relativistic calculations will not always give you the expected accuracy. Non-relativistically optimized basis sets for the heavier elements are provided in a separate directory AE, which contains basis sets of single-, double- and triple-zeta quality indicated respectively by suffixes 'sz', 'dz', and 'tz'. The files in Special/AE/ are not complete basis set files, because they don't contain fit sets (the usage and relevance of fit functions is explained later).

Basis sets directories

Basis sets can be found in the directory `$AMSHOME/atomicdata/ADF`, for non-relativistic calculations in the sub-directories SZ, DZ, DZP, TZP, TZ2P, TZ2P+, for ZORA calculations in ZORA/SZ, ZORA/DZ, ZORA/DZP, ZORA/TZP, ZORA/TZ2P, ZORA/TZ2P+, ZORA/TZ2P-J, ZORA/QZ4P, ZORA/QZ4P-J, the augmented basis sets can be found in AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, the even tempered basis sets in ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, the basis sets for MBPT calculations in Corr/TZ3P, Corr/QZ6P. All electron basis sets can be used in non-relativistic and ZORA calculations.

Basis sets acronyms

- SZ: single zeta
- DZ: double zeta
- DZP: double zeta + 1 polarization function

- TZP: valence triple zeta + 1 polarization function
- TZ2P: valence triple zeta + 2 polarization function
- TZ2P+: = TZ2P + extra d (3d metals) or extra f (lanthanides)
- QZ4P: valence quadruple zeta + 4 polarization function, relativistically optimized
- ET: even tempered
  - pVQZ, QZ3P: valence quadruple zeta + 3 polarization function, even tempered
  - QZ3P-nD: = QZ3P + n diffuse sets of s, p, d, and f functions, even tempered
- AUG: augmented
  - ASZ, ADZ, ADZP, ATZP, ATZ2P: augmented for use in TDDFT
- Corr: for use in MBPT
  - TZ3P: = all electron (Z=1-54) ZORA/TZ2P + extra tight polarization function
  - QZ6P: = all electron (Z=1-118) ZORA/QZ4P + extra tight polarization functions
- TZ2P-J, QZ4P-J: for use in ESR hyperfine or NMR spin-spin couplings
  - TZ2P or QZ4P + extra tight (mainly 1s) functions

All electron or frozen core

- element name (without suffix): all electron
- .1s frozen: 1s
- .2p frozen: 1s 2s 2p
- .3p frozen: 1s 2s 2p 3s 3p
- .3d frozen: 1s 2s 2p 3s 3p 3d
- .4p frozen: 1s 2s 2p 3s 3p 3d 4s 4p
- .4d frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d
- .4f frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d 4f
- .5p frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d 5s 5p (La-Lu)
- .5p frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d 4f 5s 5p (other)
- .5d frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d 4f 5s 5p 5d
- .6p frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d 4f 5s 5p 5d 6s 6p (Ac-Lr)
- .5f frozen: 1s 2s 2p 3s 3p 3d 4s 4p 4d 4f 5s 5p 5d 5f 6s 6p

3.2.3 The Basis Key

The basis set can be specified in the input via the `Basis` key block. The most important subkeys Type and Core. Description of all subkeys:

```
Basis
   Type [...]
   Core [None | Small | Large]
End
```

`Basis`

Type Block

Description Definition of the basis set

`Type`

Type Multiple Choice

Default value DZ

Options [SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, mTZ2P, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, Corr/TZ3P, Corr/QZ6P, Corr/ATZ3P, Corr/AQZ6P, POLTDDFT/DZ, POLTDDFT/DZP, POLTDDFT/TZP]

GUI name Basis set

Description Select the basis set to use.

SZ: Single Z DZ: Double Z DZP: Double Z, 1 polarization function TZP: Triple Z, 1 polarization function TZ2P: Triple Z, 2 polarization functions QZ4P: Quad Z, 4 pol functions, all-electron AUG: Augmented (extra diffuse functions) ET: Even tempered all electron basis sets J: Extra tight functions

These descriptions are meant to give an indication of the quality, but remember that ADF uses Slater type functions.

For standard calculations (energies, geometries, etc.) the relative quality is:

SZ < DZ < DZP < TZP < TZ2P < ET-pVQZ < QZ4P

The basis set chosen will apply to all atom types in your molecule. If no matching basis set is found, ADF will try to use a basis set of better quality.

For TDDFT applications and small negatively charged atoms or molecules, use basis sets with extra diffuse functions.

J: TZ2P-J, QZ4P-J: for use in ESR hyperfine or NMR spin-spin couplings.

Use the Basis panel to select a basis set per atom type, and to see what basis set actually will be used.

`Core`

Type Multiple Choice

Default value Large

Options [None, Small, Large]

GUI name Frozen core

Description Select the size of the frozen core you want to use.

Small and Large will be interpreted within the basis sets available (of the selected quality), and might refer to the same core in some cases.

If you specify ‘None’ you are guaranteed to have an all-electron basis set.

Warning: Do not include the *Corepotentials* (page 62) keys when using the Basis key. Typically one should not include both the the Basis key and the *Fragments* (page 50) key.

Description of the other Basis subkeys:

```
Basis
   CreateOutput Yes/No
   Path string
   PerAtomType
      Core [None | Small | Large]
      File string
      Symbol string
      Type [...]
   End
   PerRegion
      Core [None | Small | Large]
      Region string
      Type [...]
   End
   FitType [...]
End
```

Basis

CreateOutput

Type Bool

Default value No

Description If true, the output of the atomic create runs will be printed to standard output. If false, it will be saved to the file CreateAtoms.out in the AMS results folder.

Path

Type String

Description The name of an alternative directory with basis sets to use. ADF looks for appropriate basis sets only within this directory. Default $AMSRESOURCES/ADF.

PerAtomType

Type Block

Recurring True

Description Defines the basis set for all atoms of a particular type.

Core

Type Multiple Choice

Options [None, Small, Large]

Description Size of the frozen core.

File

Type String

Description The path of the basis set file (the path can either absolute or relative to $AMSRESOURCES/ADF). Note that one should include ZORA in the path for relativistic calculations, for example 'ZORA/QZ4P/Au'. Specifying the path to the basis file explicitly overrides the automatic basis file selection via the Type and Core subkeys.

Symbol

Type String

Description The symbol for which to define the basis set.

`Type`

Type Multiple Choice

Options [SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, mTZ2P, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, Corr/TZ3P, Corr/QZ6P, Corr/ATZ3P, Corr/AQZ6P, POLTDDFT/DZ, POLTDDFT/DZP, POLTDDFT/TZP]

Description The basis sets to be used.

`PerRegion`

Type Block

Recurring True

Description Defines the basis set for all atoms in a region. If specified, this overwrites the values set with the Basis%Type and Basis%PerAtomType keywords for atoms in that region. Note that if this keyword is used multiple times, the chosen regions may not overlap.

`Core`

Type Multiple Choice

Default value Large

Options [None, Small, Large]

Description Size of the frozen core.

`Region`

Type String

Description The identifier of the region for which to define the basis set. Note that this may also be a region expression, e.g. 'myregion+myotherregion' (the union of two regions).

`Type`

Type Multiple Choice

Default value DZ

Options [SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, mTZ2P, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, Corr/TZ3P, Corr/QZ6P, Corr/ATZ3P, Corr/AQZ6P, POLTDDFT/DZ, POLTDDFT/DZP, POLTDDFT/TZP]

Description The basis sets to be used.

`FitType`

Type Multiple Choice

Default value Auto

Options [Auto, SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE]

GUI name STO fit set

Description Expert option.

Select the auxiliary fit to be used for STOfit or old Hartree-Fock RI scheme.

The fit set for a given atom is taken from the all-electron basis set file for the specified choice, for the same element as the atom.

By default (Auto) the fit set is taken from the original basis set file.

An example where you can use regions to define basis sets for parts of your system, see *Example: Multiresolution* (page 1079).

3.2.4 Automatic mode

The following input will run a geometry optimization on water, using a (almost) minimal input:

```
"$AMSBIN/ams" <<eor
   Task GeometryOptimization
   System
      Atoms
         O  0  0  0
         H  1  1  0
         H -1  1  0
      End
      Symmetrize Yes
   End

   Engine ADF
      Basis
         Type TZP
      End
   EndEngine
eor
```

- The `ATOMS` subblock key in the `System` block key specifies the geometry of the system;
- the `Task GeometryOptimization` key instructs AMS to perform a geometry optimization;
- the `Basis` block key instructs ADF to run the *create runs* automatically, using a TZP basis sets.

The *automatic mode* will be used when the `Basis` key is present in the input, or if no `Fragments` block key is present.

In *automatic mode* ADF will first create fragment files for all the basic atom fragments found in the `Atoms` block. Normally this means that for each atom type in your molecule a fragment file will be created.

You may have different fragments with the same atom: add a dot and a name (without spaces) after the name of the element. For example: `H.1` and `H.2`. In this example two fragment files will be created: one for the `H.1` fragment and one for the `H.2` fragment. Using the `PerAtomType` subkey you may assign different basis sets to these fragments. Another consequence is that the `H.1` and `H.2` atoms will never be symmetry equivalent to each other.

In case of a relativistic calculation, the DIRAC program will also be run automatically, and the create runs will include the correct relativistic key and corresponding basis sets. For **ZORA** calculations, ADF first tries to locate a special ZORA basis set. If this does not succeed it will use a normal basis set if the required basis set does not use a frozen core.

3.2.5 Create mode

Expert option. In Create mode the input file is relatively simple. First, the geometry is trivial: one atom at the origin.

Second, the problem is computationally so simple that default settings for precision aspects, such as convergence criteria and levels of numerical integration accuracy, are internally defined to be much more stringent than in normal calculations. These aspects don't have to be looked after. Also one should specify the wanted XC functional. If you use the Basis key all this will be handled automatically.

In Create mode you need an input file of the following form:

```
System
  Atoms
    Atomtype 0 0 0
  End
end
Task SinglePoint
Engine ADF
  CREATE Atomtype Datafile
EndEngine
```

`Create` is the keyword. The remainder of the record (atomtype datafile) is the argument.

`Atomtype` is a name for the basic atom that you want to create. The program reads and interprets this name. Therefore, the name must begin with the standard chemical symbol (H, He, Li, …) of the element to be created. Optionally the name may have an suffix of the form .text. The suffix begins with a period (.); the part after the period (text) is at your discretion as long as it does not contain a delimiter. A few examples:

Table 3.4: Examples of appropriate and inappropriate atom type names used with the keyword create.

appropriate names	inappropriate names
K	Si-with-core (no period after the chemical symbol)
Li.newbasis	$HOME/atomicdata/ADF/C.dzp (not beginning with the chemical symbol)
P.1992_Feb.30	Ga.nocore,smallbasis (contains a comma (a delimiter))
	Sodium.2s (Sodium is not the *symbol* for this element (Na))

`Datafile` specifies the data file that contains the basis set and related items. It may contain a full path if the file does not reside in the working directory of the job. The datafile part is optional. If you omit it, ADF assumes that the file name is identical to the atom type name, i.e. Create Atomtype is equivalent to and interpreted as Create Atomtype Atomtype In view of the restrictions that apply to the atom type name, the option to use the short form can only be used if the file name has the appropriate format. To make the input file easier to understand for a human reader you may, for Datafile, also type file=Datafile, where file= must be typed as such, and datafile is the name of the file.

So you could have a simple calculation as follows (the 'creation' of a Carbon atom);

```
$AMSBIN/ams << eor
  System
    Atoms
      C 0 0 0
    End
  end
  Task SinglePoint
  Engine ADF
    Create C $AMSHOME/atomicdata/ADF/DZ/C.1s
  EndEngine
eor
```

The presence of the keyword `create` sets the computational mode of ADF to: *create a basic atom*. Here a basis set file is located in `$AMSHOME/atomicdata/ADF`, where the file 'C.1s' in the sub-directory DZ/ (this contains basis sets of double-zeta quality). Examine logfile and out to check that everything has gone well.

A considerable number of basis set files are included in the ADF. You can also create basic atoms corresponding to so-called *Alternative Elements*, with for instance a non-integer nuclear charge or a different mass. See the next section.

3.2.6 Ghost Atoms, Non-standard Chemical Elements

The atom type names used under atoms (and in the create record) must begin with the standard chemical element symbol (H, He, Li…). The program uses this to deduce the nuclear charge and other elemental properties.

For the standard elements one can redefine the atomic mass (for instance to define a suitable isotope). Masses are specified by adding the desired mass (in Dalton) at the end of the atom's line (mass=).

It is also possible to define an artificial chemical element with user-specified properties. Such new elements are denoted *Alternative Elements*; and may for instance have a non-integer nuclear charge.

The chemical symbol of for a (ghost) atom that has zero nuclear charge is Gh. The `BASIS` key recognizes elements denoted with Gh.atom in the ATOMS key as being ghost atoms. If one does not specifically select a basis set for this ghost atom, the all electron basis set for the atom is selected in the creation of the ghost atom using the type of basis set chosen with the BASIS key. The atom name must begin with the standard one- or two-character symbol for the chemical element: Gh.H, Gh.He, Gh.Li, and so on. Optionally it may be appended by .text, where text is any string (not containing delimiters). Examples: Gh.H, Gh.Mn.3, Gh.Cu.dz-new. See also the Basis set superposition error (BSSE) tutorial.

For other alternative elements, for instance that have a non-integer nuclear charge, one should use one of the standard chemical symbols. The `BASIS` key will use this chemical symbol for selecting the basis set for this alternative element. Nuclear charges are specified by adding the nuclear charge at the end of the atom's line (nuclear_charge=).

You can create Gh-type basic atoms and other alternative elements and use them subsequently as fragments in a molecule.

Automatic mode

AMS allows to set user-defined masses for particular atoms. This can be used to simulate isotopes of different atoms. Masses are specified by adding the desired mass (in Dalton) at the end of the atom's line.

Example: three different hydrogen isotopes:

```
System
   Atoms
      N        0.000000    0.000000    0.010272
      H       -0.471582   -0.816803    0.407861
      H        0.943163    0.000000    0.407861 mass=2.014101778
      H       -0.471582    0.816803    0.407861 mass=3.01604927
   End
End
```

Use as fragment

Alternative basic atoms can be used like any other basic atoms to build up larger fragments and molecules. Gh can be considered just one more chemical symbol along with the other traditional ones. For other alternative elements one should define the nuclear charge, and possibly the nuclear mass.

You may have different alternative elements in a molecule, with different nuclear charges for instance.

Example: ghost atoms:

```
System
   atoms
      Gh.O  -0.525330     -0.050971     -0.314517
      Gh.H  -0.942007      0.747902      0.011253
      Gh.H   0.403697      0.059786     -0.073568
      O      2.316633      0.045501      0.071858
      H      2.684616     -0.526577      0.749387
      C      2.781638     -0.426129     -1.190301
      H      2.350821      0.224965     -1.943415
      H      3.867602     -0.375336     -1.264613
      H      2.453296     -1.445999     -1.389381
   end
end
```

Example: alternative elements:

```
System
   Atoms
      Mg 0.0  0.0  0.0
      F  1.0  1.0  1.0  nuclear_charge=9.5
      F  1.0 -1.0 -1.0  nuclear_charge=9.5
      F -1.0  1.0 -1.0  nuclear_charge=9.5
      F -1.0 -1.0  1.0  nuclear_charge=9.5
   End
End
```

3.2.7 Nuclear Model

By default in ADF a point charge model is used for the nuclear charge distribution. Alternatively, one can use a spherical Gaussian nuclear charge distribution model, see Ref.[8]. Nuclear finite size effects can have large effects on hyperfine interactions (ESR A-tensor, NMR spin-spin coupling) if heavy atoms like, for example, Mercury (Hg), are involved. In Ref.[?] it was asserted that the isotropic J-couplings (parameters in NMR spin-spin coupling) are typically reduced in magnitude by about 10 to 15 % for couplings between one of the heaviest NMR nuclei and a light atomic ligand, and even more so for couplings between two heavy atoms. This Ref.[?] gives more details on the parameters used in the Gaussian nuclear charge distribution model. Note that one needs basis sets with very tight functions to see any effect of using a finite size of the nucleus instead of a point nucleus. Such basis sets can be found for all elements in `$AMSRESOURCES/ZORA/TZ2P-J` and `$AMSRESOURCES/ZORA/QZ4P-J`, and for some elements in `$AMSRESOURCES/ZORA/jcpl`, which are basis sets especially designed for ESR hyperfine and NMR spin-spin coupling calculations.

```
NuclearModel [PointCharge | Gaussian]
```

`NuclearModel`

Type Multiple Choice

Default value PointCharge

Options [PointCharge, Gaussian]

Description Model for the nuclear charge distribution.

To see effects from your choice you will need to use a basis set with extra steep functions. For example you can find these in the ZORA/TZ2P-J basis directory.

---

[8] J. Autschbach, *Magnitude of Finite-Nucleus-Size Effects in Relativistic Density Functional Computations of Indirect NMR Nuclear Spin-Spin Coupling Constants*, ChemPhysChem 10, 2274 (2009) (https://doi.org/10.1002/cphc.200900271)

In the ADF output parameters will be shown for the Gaussian nuclear charge distribution if one includes in the input for ADF:

```
PRINT Nuclei
```

Starting from ADF2013 ADF also uses a finite distribution of the nuclear magnetic dipole moment for the calculation of the A-tensor.

3.3 Molecular fragments

3.3.1 Fragment mode

In Fragment mode you have to specify: (1) the atomic positions and (2) how the total system is built up from fragments. We recommended to specify also (3) the point group symmetry.

Example of an input file for the C2H4 molecule:

```
System
   Atoms
      C   0      0   0.668
      C   0      0  -0.668
      H   0.927  0  -1.203
      H  -0.927  0  -1.203
      H   0.927  0   1.203
      H  -0.927  0   1.203
   end
End

Engine ADF
   fragments
      C C_dzp.results/adf.rkf
      H H_dzp.results/adf.rkf
   end

   symmetry D(2h)
EndEngine
```

The *Atoms block key* (page 35) is in the AMS driver part of the input. Two keys are used in the engine ADF part: *fragments* (page 52) and *symmetry* (page 37).

atoms defines the atomic positions: each record in the data block contains the chemical symbol of an atom followed by its Cartesian coordinates in Angstroms (Z-matrix type input of atomic positions is also possible).

fragments lists the fragment files each record contains a fragment *type* followed by the corresponding fragment *file*. In the example the files are *local* files. Files in other directories are addressed by giving the complete file path. Note: if a *parallel* calculation is performed, be sure that each 'kid' finds the specified fragment files. This will usually require that the files are *not* local to the job, but first be moved to some shared volume, and that the references to the fragment files in the input contain the full path. An alternative is to ensure that the (local) files in the parent directory are copied first to the 'kid' directories before the parallel calculation starts.

symmetry specifies the point group symmetry by a Schönfliess type symbol. *Symmetry* (page 380) contains a complete list of all Schönfliess symbols that are recognized by ADF. If no symmetry is specified ADF will use the true symmetry of the nuclear frame. If *(electric) fields* (page 144) are used, see later, symmetry will be NOSYM. Note that the computed symmetry may not occur in the list of allowed symmetries (see *Symmetry* (page 380)), in which case you have to explicitly specify the (lower) point group symmetry you wish to apply.

The atomic coordinates must conform to the point group symmetry; the program will check this and abort if the atomic system does not have the specified symmetry. It is allowed, however, to specify a *lower* symmetry than what is actually present in the set of atomic positions. The *specified* symmetry determines how results are analyzed and how irreducible representations and subspecies are labeled. It also determines various algorithmic aspects: the program runs more efficiently with the highest possible symmetry.

The spatial orientation of the molecular coordinate system is not arbitrary. ADF requires for each point group symmetry a specific standard orientation. In axial groups for instance, the main rotation axis must be the z-axis. This implies a restriction on how you can define the atomic coordinates under atoms. The orientation requirements for all point groups are listed in *Symmetry* (page 380). If the specified symmetry equals the true symmetry of the nuclear frame ADF will adjust the input orientation of the molecule to the requirements (if necessary). If you have specified a subgroup of the true nuclear symmetry, no such orientation adjustment is carried out and the user has to make sure that his input data yield the correct orientation, lest an error will occur.

Restrictions apply to the symmetry (as specified) of the molecule, related to the symmetries of the fragments as they were stipulated in the preceding fragment calculations. All symmetry operators of the molecule that internally rotate or reflect a fragment but leave it at the same position in the molecule, must also be operators of the symmetry group in which the fragment has been computed. Furthermore, two fragments must not be symmetry-equivalent in the molecule only by an improper rotation. The implied internal reflection of the fragment must be one of the symmetry operators in the point group symmetry that is used in the fragment calculation *and* the molecular symmetry group must also contain a proper rotation that maps the two fragments onto each other.

The example of the C2H4 molecule implicitly assumes that all fragments are *single atom* fragments. When the fragments are larger the data records in the atoms key have to be extended: you must specify which atoms belong together in one fragment.

```
System
   Atoms
      Ni  0      0      0
      C  -1.06 -1.06  1.06 adf.f=CO|1
      C  -1.06  1.06 -1.06 adf.f=CO|2
      C  -1.06  1.06 -1.06 adf.f=CO|3
      C   1.06 -1.06 -1.06 adf.f=CO|4
      O   1.71  1.71  1.71 adf.f=CO|1
      O  -1.71 -1.71  1.71 adf.f=CO|2
      O  -1.71  1.71 -1.71 adf.f=CO|3
      O   1.71 -1.71 -1.71 adf.f=CO|4
   End
End

Engine ADF
   Fragments
      CO CO_yesterday.results/adf.rkf
      Ni Ni_dzp.results/adf.rkf
   End

   SYMMETRY T(D)
EndEngine
```

Another sample input file; using a single atom Ni fragment and four molecular CO fragments. The keys symmetry and fragments operate as before. Again we have two types of fragments (here: Ni and CO); for each of them, the fragment file is specified.

Under the key ATOMS the chemical symbols and the nuclear coordinates are listed. Added is the adf.f=…-part; f stands here for fragment and tells the program that the carbon and oxygen atoms belong to CO fragments. The last part in adf.f=CO|n enumerates the individual CO fragments: here you define which C and O belong together in one CO fragment.

The record for Ni contains no adf.f= part, implying the *default* for this atom: it is a fragment on its own. In the C2H4 example before the default applied to all atoms.

Note that one should use the adf.f= part for symmetry equivalent fragments. In the next example, ADF assumes the fragments CO1, CO2, CO3, and CO4, to be of different fragment types, even though they are coming from the same adf.rkf (TAPE21). Therefore ADF will assume symmetry NOSYM in the next calculation, and will not run in T(D) symmetry.

```
System
   Atoms
      Ni 0     0     0
      C -1.06 -1.06  1.06 adf.f=CO1
      C -1.06  1.06 -1.06 adf.f=CO2
      C -1.06  1.06 -1.06 adf.f=CO3
      C  1.06 -1.06 -1.06 adf.f=CO4
      O  1.71  1.71  1.71 adf.f=CO1
      O -1.71 -1.71  1.71 adf.f=CO2
      O -1.71  1.71 -1.71 adf.f=CO3
      O  1.71 -1.71 -1.71 adf.f=CO4
   End
End

Engine ADF
   Fragments
      CO1 CO_yesterday.results/adf.rkf
      CO2 CO_yesterday.results/adf.rkf
      CO3 CO_yesterday.results/adf.rkf
      CO4 CO_yesterday.results/adf.rkf
      Ni Ni_dzp.results/adf.rkf
   End
End
```

There are more possibilities with the keys atoms and fragments. This is worked out later. The purpose of this section was to provide a quick and easy start.

3.3.2 Fragment files

The adf.rkf (TAPE21) result files from the ADF computations on the fragments that constitute a molecule completely characterize these fragments. The fragment adf.rkf (TAPE21) files must be attached as *fragment files*. This is achieved with the key FRAGMENTS. See also the next section for the relation between Atom type, Fragment type and Fragment file names.

```
Fragments {Directory}
   FragType FragFile
   FragType FragFile
end
```

`FragType` One of the fragment *types* defined under atoms, either explicitly (adf.f=fragtypeln) or implicitly (fragment type=atom type, if the adf.f= option is not used).

`FragFile` The fragment file: the standard adf.rkf (TAPE21) result file from the computation of that fragment. The file name must contain the complete path relative to Directory (the argument of the key). By default, when no Directory is specified, this is the local directory where the job runs. You may therefore omit the directory and give simple (local) file names if all the files are present in the working directory of the job.

Obviously, FragFile is **case sensitive**. However, FragType is also treated as case sensitive; see also the *ATOMS key* (page 35) discussion (adf.f= option). The reason is that there are shortcuts possible to the effect that the FragType name (in the atoms block) is immediately interpreted as the name of the fragment file.

The key FRAGMENTS may be used any number of times in the input file. This is convenient if you employ a sizable number of fragment files, with subsets located in different directories. You can then use the key separately for each directory, to avoid typing long path names for all the files. Fragtypes that occur in the fragments block(s), but that are not referred by atoms are ignored. No fragment files must be specified for dummy atoms (xx).

It is allowed to use one and the same fragment file for different fragment types. Example:

```
ATOMS
   C.1 x1 y1 z1
   C.2 x2 y2 z2
end

Fragments
   C.1 C.results/adf.rkf
   C.2 C.results/adf.rkf
end
```

Two different atom types (and fragment types) C.1 and C.2 are defined. The properties of the two fragment types are now identical since they are characterized by the same fragment file, but from the program's point of view they are different and can therefore not be symmetry equivalent.

The reason you may want to specify different atom types will usually be related to analysis, in particular symmetry aspects. If you know in advance that the two atom types are not symmetry equivalent, or more generally, that they play a rather different role in the molecule, it can enhance clarity of printed output to assign different atom type names to them. However, see the notes below.

If you want your fragment file to be the result file of a spin-unrestricted calculation, you should use the *UNRESTRICTEDFRAGMENTS* (page 65) key. You can also analyze a molecule in terms of simulated unrestricted fragments, in which spin-restricted fragment files are used and the key *FRAGOCCUPATIONS* (page 66) is specified.

Suppose that you have done a calculation on a molecule *mol*, in which you have defined two different atom types for atoms of the same chemical element. Suppose furthermore, that you want to use that molecule now as a fragment in a new calculation.

You list under atoms all atoms of the molecule and you specify which atoms belong to the various fragments, among which the molecular fragment *mol*. The program will then have a problem deciding which atoms in your system are associated with the different atom types in the fragment. Normally, ADF analyzes this by comparing the chemical elements. That is not sufficient here because one chemical element corresponds with more than one type of atom in the *mol fragment* type. In such a case it is imperative to use *the same atom type names* in your new calculation as you used in the generation of the fragment. These names are stored in the fragment file, and they are printed in the output file of the calculation of *mol*.

The names of three items may be related to each other, depending on how you specify input: the *atom type*, the *fragment type*, and the *fragment file*.

The atom type is defined in the data block to atoms.

The fragment type is defined also in the data block to atoms: with the adf.f= option. For records in the data block that don't have the adf.f= option, the fragment type name is by definition identical to the atom type name.

The fragment file is defined in the data block to fragments, each record consisting of a fragment *type* name, followed by the fragment *file*. If a fragment type is not listed in the data block to fragments, so that no fragment file name is specified, the fragment *file* is by definition identical to the fragment *type* name.

CHAPTER

FOUR

MODEL HAMILTONIANS

See also:

spin-orbit coupling

Examples: *special XC functionals* (page 403), *relativistic effects* (page 432), *solvents, other environments* (page 445)

4.1 Electronic Configuration

The next few keys can be used to specify the electronic configuration. If you don't specify any such keys, certain defaults will apply. In principle, the program will (by default) attempt to find the lowest-energy spin-restricted (one-determinant) state. If SCF convergence is problematic the program may wind up at an excited state, by which (in this context) we mean a one-determinant state with a higher energy than some other one-determinant state with the same net spin polarization. In worse cases the program may fail to converge to any state at all. It is good practice to *always* verify which configuration you actually have computed.

When you specify a particular configuration and/or net charge and/or net spin-polarization of the system, the program will try to compute accordingly, even if the requested configuration has no physical or chemical meaning. The program has no knowledge about the existence of materials and will simply try to carry out what you tell it to do.

4.1.1 Charge and Spin

Spin: restricted vs. unrestricted

```
Unrestricted Yes/No
```

`Unrestricted`

Type Bool

Default value No

Description By default, a spin-restricted calculation is performed where the spin alpha and spin beta orbitals are spatially the same.

```
SpinPolarization float
```

`SpinPolarization`

Type Float

Description The spin polarization of the system, which is the number of spin-alpha electrons in excess of spin-beta electrons. Specification is only meaningful in a spin-unrestricted calculation. However, specification is not meaningful in an unrestricted Spin-Orbit coupled calculation using the (non-)collinear approximation.

In an `unrestricted` calculation spin-α and spin-β MOs may be spatially different and may have different occupation numbers. In the case of Spin-Orbit coupling, it means that Kramer's symmetry does not have to be satisfied, in which case the key `UNRESTRICTED` should be used in combination with the subkey `SpinOrbitMagnetization NONCOLLINEAR` or `SpinOrbitMagnetization COLLINEAR` of the key `Relativity`.

The unrestricted mode roughly doubles the computational effort. The actual numbers of spin-α and spin-β electrons respectively are controlled by the keys `SpinPolarization` and *OCCUPATIONS* (page 59). Note carefully, that using *only* the keyword `unrestricted`, without either `SpinPolarization` or `Occupations` (or both) will not result in any spin polarization. This implies that you would effectively perform a spin-restricted calculation, but with increased computational effort. Therefore, the program will check that in an unrestricted calculation at least one of the keys `SpinPolarization` or `Occupations` is applied.

The unrestricted feature is equivalent, in *ab-initio* terminology, to (Spin-)Unrestricted-Hartree-Fock (UHF); the N-particle wave function is a single determinant and not necessarily an eigenfunction of the spin operator S^2 .

A *restricted* calculation implies that the (spatial) orbitals *and* the occupation numbers are identical for spin-α and spin-β.

An unrestricted calculation with `SpinPolarization 0.0` (for instance by not specifying the spin polarization at all) is, in case one does not break the spin symmetry, in fact a restricted run: it should give exactly the same as the restricted calculation, but it will use more CPU time. If one does break the spin symmetry, for example with the key `MODIFYSTARTPOTENTIAL` or the `SPINFLIP` option in the key *RESTART* (page 344), the solution may also be a broken spin symmetry solution. For example one may want to start a calculation in broken symmetry with spin-α density on one fragment and spin-β density on another, e.g. in a spin-unrestricted calculation of H_2 at large separation.

The Fock operator, both in an unrestricted and in a restricted run, commutes with the spin operator Sz, but not (unless accidentally) with S^2 . The obtained one-determinant wave function may for instance be a mixture of a singlet and a triplet state.

In an unrestricted calculation the expectation value of S^2 is computed in ADF (note 29 in ref.[1]). The implementation of an evaluation of S^2 is not quite trivial. DFT is essentially a one-particle formalism, so the S-operator for the n-particle system has to be written out in single-particle operators[2]. The equations used in ADF to calculate the expectation value of S^2 can be found in Szabo and Ostlund[3]. Note that the so called exact value $(S_{exact})^2$, which is printed in the ADF output, is defined as $(S_{exact})^2 = (|N_a - N_b|/2)(|N_a - N_b|/2+1)$, where N_a and N_b are the number of spin-α and spin-β electrons, respectively. The expectation value of S^2 is not calculated in a Spin-Orbit coupled calculation.

If one includes the *UNRESTRICTEDFRAGMENTS* (page 65) key, molecules that have been calculated using the spin-unrestricted formalism can be employed as fragments. In this case, the calculation on the full complex also needs to use the spin-unrestricted formalism. See also the examples *Example: unrestricted fragments: CH3I* (page 1019) and *Example: unrestricted EDA: Cu(C2H4)2* (page 1017).

A fair approximation to a computation with spin-unrestricted fragments can be achieved with the key *FRAGOCCUPATIONS* (page 66), which uses spin restricted fragments, but allows for spin-unrestricted fragment occupations. See also the examples *Example: Spin-unrestricted Fragments: H2* (page 991), *Example: Bond Energy analysis meta-GGA, (meta-)hybrids: Zn2, Cr2, CrH* (page 1012) and *Example: Bond Energy analysis open-shell fragments: PCCP* (page 997).

---

[1] R.E. Bulo, A.W. Ehlers, S. Grimme and K. Lammertsma, *Vinylphosphirane.Phospholene Rearrangements: Pericyclic [1,3]-Sigmatropic Shifts or Not?* Journal of the American Chemical Society 124, 13903 (2002) (https://doi.org/10.1021/ja027925u)

[2] R. Pauncz, *Spin Eigenfunctions*, ISBN13: 9780306401411, 1979, New York: Plenum Press

[3] A. Szabo and N.S. Ostlund, *Modern Quantum Chemistry*, ISBN13: 9780070627390, 1st ed. revised ed. 1989: McGraw-Hill

Spin: unrestricted vs restricted open shell

In unrestricted calculations the spin-α and spin-β orbitals, energies and occupations may differ. In restricted open shell calculations the spin-α and spin-β orbitals and energies should be identical, only the occupations may differ. In ADF2023 a restricted open shell method is implemented valid for high spin open shell molecules, similar as in Ref.[11], in which case the 1-determinantal wave function is an eigenfunction of S_z and S^2. The implementation of the method is work-in-progress. At the moment it is only valid in case of single point calculations. This method can not be used in case of spin-orbit coupling, TD-DFT, or for many other properties.

The method implemented in ADF needs integer occupation numbers, and a positive spin polarization. It requires the subkey ROSCF of the key SCF, the keyword UNRESTRICTED, and the key SPINPOLARIZATION. Part of the input could like like:

```
Occupations IntegerAufbau
Unrestricted Yes
SpinPolarization 2
SCF
    ROSCF
    End
End
```

The ROSCF options are:

```
SCF
   ROSCF
      Alpha float_list
   End
End
```

SCF

ROSCF

Type Block

Description Settings for the ROSCF method.

Alpha

Type Float List

Default value [0.5, 0.5, 0.5]

Description Coefficients to build the alpha-spin orbital contribution to the diagonal closed-, open-, and virtual-shell blocks of the Fock matrix. The beta-spin orbital contributions are 1.0 minus the alpha ones.

---

[11] P. Ramos, M. Pavanello, *Constrained subsystem density functional theory*, Physical Chemistry Chemical Physics 18, 21172 (2016) (https://doi.org/10.1039/C6CP00528D)

Unrestricted and Spin-Orbit Coupling

In the case of Spin-Orbit coupling there are two ways to do spin-polarized calculations, either using the collinear approximation or the noncollinear approximation[4][5]. Using the unrestricted feature in order to assign different numbers of electrons to a and b spin, respectively, cannot be applied as such, if one includes Spin-Orbit coupling, since the electrons are not directly associated with spin-α and spin-β. For the collinear and noncollinear approximation one should use `Symmetry NOSYM` (see *Symmetry key* (page 37)), and each level can allocate 1 electron. One should not specify the spin-polarization.

Collinear approximation

```
Unrestricted Yes
Symmetry NOSYM
Relativity
   Level Spin-Orbit
   SpinOrbitMagnetization COLLINEAR
End
```

The `SpinOrbitMagnetization` key is only relevant in the case of Spin-Orbit coupling.

In the collinear approximation in each point in space the spin-polarization has the same direction (default is in the direction of the z-axis). Kramer's symmetry does not have to be satisfied. Symmetry used in the calculation should be NOSYM. The default direction of the spin-polarization can be overruled using as argument `COLLINEARX` for spin-polarization only in the direction of the x-axis, or as argument `COLLINEARY` for spin-polarization only in the direction of the y-axis.

Noncollinear approximation

```
Unrestricted Yes
Symmetry NOSYM
Relativity
   Level Spin-Orbit
   SpinOrbitMagnetization NONCOLLINEAR
End
```

In the noncollinear approximation in each point in space the spin-polarization can have a different direction. Kramer's symmetry does not have to be satisfied. Symmetry used in the calculation should be NOSYM.

Net Charge

The net charge of the molecule can be controlled with the key CHARGE in the System block in the AMS part of the input:

```
System header
   Charge float
End
```

`System`

`Charge`

Type Float

Default value 0.0

GUI name Total charge

---

[4] H. Eschrig and V.D.P. Servedio, *Relativistic density functional approach to open shells*, Journal of Computational Chemistry 20, 23 (1999) (https://doi.org/10.1002/(SICI)1096-987X(19990115)20:1%3C23::AID-JCC5%3E3.0.CO;2-N)

[5] C. van Wüllen, *Spin densities in two-component relativistic density functional calculations: Noncollinear versus collinear approach*, Journal of Computational Chemistry 23, 779 (2002) (https://doi.org/10.1002/jcc.10043)

Description The system's total charge in atomic units.

4.1.2 Orbital occupations: electronic configuration, excited states

With the keys `OCCUPATIONS` and `IRREPOCCUPATIONS` you can specify in detail the assignment of electrons to MOs

Aufbau, smearing, freezing

```
OCCUPATIONS Options
```

`Options` May contain IntegerAufbau, Keeporbitals, ElectronicTemperature, Smear, Freeze, or Steep:

`IntegerAufbau` Electrons are assigned to MOs according to the Aufbau principle, trying to use integer occupations, also in case of degeneracy at the Fermi level. ADF normally may use fractional occupation numbers in that case. Note that for multi-dimensional irreps, using the subkey IntegerAufbau may still not prevent fractional occupation numers to be used.

`Keeporbitals=NKeep` Until SCF cycle Nkeep electrons are assigned to MOs according to the Aufbau principle, using at each cycle the then current orbital energies of the MOs. Thereafter the KeepOrbitals feature is applied. As soon as this is activated the program will on successive SCF cycles assign electrons to the MOs that maximally resemble - in spatial form - those that were occupied in a 'reference cycle number'. The default for Nkeep is 20, except:

a) When orbital occupations for MOs are specified explicitly in the data block of the occupations key, these apply throughout.

b) In a Create run fixed occupations are derived from a database in the program.

c) When electron smearing is explicitly turned on by the user (see the Smear option below) Nkeep is by default 1,000,000 so the program will 'never' compare the spatial forms of MOs to determine the occupation numbers. The 'reference cycle number' is by default the previous cycle, which will suppress jumps in the spatial occupations during the SCF development while at the other hand allowing the system to let the more-or-less-frozen configuration relax to self-consistency.

`Freeze` Occurrence of this word in the option list specifies that the 'reference cycle number' will be the cycle number on which the KeepOrbitals feature is activated: during all subsequent SCF cycles the program will assign electrons to MOs that resemble the MOs of that specific SCF cycle. This may be used when the MOs of that cycle are already reasonably close to the final ones, and it will suppress unwanted step-by-step charge-transfers from occupied to empty orbitals that are very close in energy. By default this option is not active.

`ElectronicTemperature=T` Invokes orbital population using the Fermi-Dirac distribution with temperature `T` (in Kelvin). This may be used to achieve SCF convergence in an otherwise problematically converging system. Note that the default SCF acceleration method A-DIIS is not compatible with non-aufbau orbital occupations so it is disabled automatically and the classical Pulay DIIS scheme is used. For this reason it is strongly recommended to use either MESA or one of the methods from the LIST family in combination with `ElectronicTemperature` or `Smear`.

`Smear=Smear1[,Smear2,Smear3,...,Smear10]` Another form of specifying the electronic temperature (in Hartree). The smearing parameter Smear*N* is related to the `ElectronicTemperature` in such a way that the occupation number for any orbital outside the smearing interval will deviate from an integer value (0 and 1 for spin-unrestricted, 0 and 2 for spin-restricted) by no more than 1% (0.01 electron for spin-unrestricted, 0.02 for spin-restricted). More precisely, a smearing value of 1 Hartree is equivalent to an electronic temperature of 34285 Kelvin. When a comma-delimited list of values is specified, the next value from the list is picked after SCF has converged, and the SCF is continued. This way one can specify a list of gradually decreasing values to get sort of annealing effect. NOTE: No spaces are allowed when specifying the list of values for Smear.

`Steep=Lambda[,Nmax]` The occupation number for each orbitals are updated according to steepest-descent method (Ref: F. W. Averill and G. S. Painter, Phys. Rev. B **46**, 2498 (1992)). During an SCF cycle, the occupation number for each new orbital is initially determined by decomposing the old charge density with new orbitals. Then, the occupation numbers are modified so that the total energy of the system will decrease. The Lambda parameter gives the coefficient for the charge transfer in 1/au unit. The second parameter, Nmax, is an additional limit for the amount of the charge transfer. Nmax would be useful for early steps of cycle when the Lambda parameter gives too large charge transfer. Too small Nmax results in irregular behavior in SCF convergence. In the case of difficult SCF convergence, you should make mixing and Lambda smaller. From our experience, Nmax=0.1 or 0.2 is usually OK. This method should be used with turning off DIIS method (DIIS N=0), and the choice of the mixing parameter in SCF cycle is also important. This option is especially useful for systems with many quasi-degenerate orbitals around Fermi level. For instance, cluster models of surface systems usually suffer from dangling bonds and should be converged with this method. Note though that slow convergence is an intrinsic feature of this method so one should specify a large limit for the number of SCF cycles, say 500 or even 1000, depending on the cluster size.

`OptimizeSpin=Delta`, `OptimizeSpinRound=Delta` The spin polarization can be optimized by minimizing the difference between Fermi level values for spin-alpha and spin-beta MOs. This method works reliably only when a reasonably large finite electron temperature is used (300K is found to be sufficiently large for small transition metal clusters). The polarization value is optimized up to a specified tolerance Delta (fractional values are allowed) by step-wise reduction of the root-bracketing interval for the objective function f=FermiLevel(alpha)-FermiLevel(beta). The SCF procedure is converged for each spin polarization. When the interval becomes smaller than Delta the optimization is considered converged. After that the SCF may optionally be repeated with the spin polarization set to the nearest even or odd number (depending on the total number of electrons in the system) or, in other words, with the total number of electrons per spin rounded to the nearest integer. The method is invoked using either the OptimizeSpin or the OptimizeSpinRound Occupation option. As mentioned above, it must be used together with the ElectronicTemperature option. The OptimizeSpin and OptimizeSpinRound argument specifies the tolerance value. The OptimizeSpinRound option specifies that after polarization has converged the SCF must be completed with the total number of electrons per spin rounded to the nearest integer. It should be noted that, as usual, the Unrestricted and SpinPolarization keys must be specified as for any other spin-unrestricted calculation. The argument of the SpinPolarization key sets the initial spin polarization value from which the optimization is started. The result of the optimization may depend on the initial value because the objective function may have more than one root.

Notes about the occupations options:

- When occupation numbers are explicitly defined via the block `IRREPOCCUPATIONS` (see next section), the Smear option cannot be used.
- The aufbau principle does not determine or adjust the distribution of electrons over spin-α versus spin-β in an unrestricted calculation. This aspect is controlled by the key `SpinPolarization` and by any explicit occupations in the data block of occupations.
- Smearing cannot be used in combination with the keeporbitals option.

Explicit occupation numbers

```
IrrepOccupations
  irrep orbitalnumbers
  irrep orbitalnumbers
  ...
End
```

irrep The name of one of the irreducible representations (not a subspecies) of the point group of the system. See the *Symmetry* (page 380) for the irrep names as they are used in ADF.

orbitalnumbers A series of one or more numbers: the occupation numbers for one-electron *valence* orbitals in that irrep. The orbitals are ordered according to their energy eigenvalue; higher states than those listed get an occupation number zero.

For degenerate representations such as the 2-dimensional E-representations or the 3-dimensional T-representations, you must give the *total* occupation, i.e. the sum over the partner representations; ADF assigns each partner an occupation equal to the appropriate fraction of what appears here.

In an unrestricted calculation, two sequences of numbers must be specified for each irrep; the sequences are separated by a double slash (//). The first set of numbers is assigned to the spin-α orbitals, the second set to the spin-β orbitals. Example unrestricted calculation in symmetry NOSYM with two unpaired electrons:

```
IrrepOccupations
   A 28 // 26
End
SpinPolarization 2
Symmetry NOSYM
```

Note that this is not meaningful in an unrestricted Spin-Orbit coupled calculation using the (non-)collinear approximation, where one should use one sequence of occupation numbers for each irrep.

Notes about the occupations data block:

- When specifying electron configurations, all valence electrons in the calculation must be explicitly assigned to MOs and the IRREPOCCUPATIONS keyword must be used. In this context the concept *valence electrons* and hence *valence orbitals* is not necessarily identical to what you may normally assume to be the valence space of an atom or molecule. The meaning of *valence* is here strictly defined as whatever electrons are outside the frozen core. It depends therefore on the level of frozen core approximation applied in the calculation. This traces back to the Create runs in which the basic atoms were generated that are now used to build the molecule.
- When for some irrep there is a rather long list of occupation numbers, corresponding to * consecutive fully occupied* states, you can combine these numbers and enter their sum instead: ADF knows the maximum occupation for an irrep, and when you put a larger number the program will split it up. For instance, if you give for the *p*-representation (in a single atom calculation):

  ```
  P 17 3
  ```

  ADF will interpret this as:

  ```
  P 6 6 5 3
  ```

  i.e. the occupation number 17 is interpreted as denoting two fully occupied p-shells and the remaining five electrons in the next higher shell. This example also illustrates how to specify an excited state: here we have defined a hole in the third p-shell.
- Fractional occupation numbers in input are allowed. For a discussion of the interpretation of fractional occupation

numbers see ref.[6]. The program even allows you (technically) to use a non-integer total number of electrons, whatever the physical meaning of such a calculation is.

- The data block of occupations is not parsed (see the section *Input* (page 25)). The program does not replace expressions by their value and it does not recognize constants or functions defined with the define key.
- In a numerical frequencies run (without symmetric displacements) the symmetry used internally in the program is NOSYM, irrespective of any Schönfliess symbol in the input file. As a consequence the program will recognize only the A representation (the only irrep in nosym), but not the representations belonging to the input point group symmetry. (The symmetry in the equilibrium geometry, defined by the input Schönfliess symbol, is used to enhance efficiency and stability in the construction of the matrix of Force constants).

CHARGE and SPINPOLARIZATION vs. IRREPOCCUPATIONS

The contents of the data block of IRREPOCCUPATIONS, if used, defines the total number of valence electrons and hence the net total charge. In an unrestricted run it also defines the net spin polarization. If the keys `CHARGE` and `SPINPOLARIZATION` are also used, the program will check that the specifications are consistent.

We strongly recommend to employ this and always specify the net total charge and spin polarization with charge whenever explicit occupation numbers are supplied with IRREPOCCUPATIONS, to that the program will check that your occupation numbers result in the total charge and spin polarization that you have in mind.

Create mode

In Create mode occupation numbers are predefined (see Appendix *Elements of the Periodic Table* (page 367)), and these are applied unless you specify occupations in input yourself. Conceivably this may result in a non-aufbau configuration. In Create mode the program always operates as if the occupations were set in input.

4.1.3 Frozen core approximation

Frozen core vs. pseudopotentials

Pseudopotentials are not supported. The frozen core approximation is automatic in a normal (Fragment mode) calculation and is defined by the basic atomic fragments. The data file used in the Create run specifies the frozen core for the atom, which is then used in all molecules that incorporate that atomic fragment.

Core Potentials

In the standard approach the Coulomb potential and the charge density due to the atomic frozen core are computed from the frozen one-electron orbitals. ADF stores the computed core density and core potential for each atom type in the molecule on a file TAPE12. Alternatively, you may attach a file with (core) potentials and densities. The file must have the same structure as the standard TAPE12. It should contain one or more sections, each with the core information for one type of atom. With the block COREPOTENTIALS you specify the core file and (optionally) which sections pertain to the distinct atom types in the molecule.

```
COREPOTENTIALS corefile
  {atomtype index}
  {atomtype index}
  ...
end
```

`corefile` The file with core potentials and charge densities. The name may contain a path.

---

[6] S.G. Wang and W.H.E. Schwarz, *Simulation of nondynamical correlation in density functional calculations by the optimized fractional orbital occupation approach: Application to the potential energy surfaces of* O_3 and SO_2, Journal of Chemical Physics 105, 4641 (1996) (https://doi.org/10.1063/1.472307)

atomtype One of the atom type names as defined by atoms.

index Points to the core section on the attached file that applies to the atom type. Different atom types may use the same section. A non-positive index tells the program that the atoms of that type don't have a frozen core. If the information on the corresponding fragment file (or data file in Create mode) indicates the contrary the program will abort with an error message.

If the key is used as a simple key (specifying only the core file) the sections on the file are associated with the atom types in order: the first section is used for the first atom type, et cetera. This is overruled by applying the block form. However, since the key *must* have the core file as argument, the block form requires that you apply the continuation symbol: an ampersand (), separated from the core file name by at least one blank.

If you omit an atom type from the data block it gets a zero index (no core).

The attached file may contain more sections than used in the calculation, and the indices specified in the data block don't have to be in ascending order, consecutive, or cover a specific interval.

When a file with non-standard (e.g. relativistic) cores is attached and used in the calculation of an atom or molecule, and the result is used as fragment in a subsequent calculation, you should attach and use the same core potentials again. Otherwise, the program will internally compute the standard core potentials and hence implicitly employ another fragment than you may think, i.e. a fragment with other properties. ADF will not check anything in this respect and corepotentials should therefore be handled with great care.

The primary application of the corepotentials option is to include (scalar) relativistic corrections in the (frozen core part of the) Fock operator. The relativistic core potentials can be computed with the auxiliary program dirac (see the RELATIVITY keyword).

4.1.4 Spin-polarized start-up potential

The Fock matrix for the first SCF cycle is built by combining the Fock matrices from the fragment files. When performing a *Restart* (page 344), the initial Fock Matrix may be read from the specified `adf.rkf` file.

Spin-flip method for broken symmetries

It is possible to exchange alpha and beta electrons for selected atoms when performing a restart from a previous spin-unrestricted calculation.

In many cases, one wishes to perform a calculation of a low-spin complex where spin-density is positive on some atoms and negative on the others. It is usually very difficult to achieve SCF convergence if one starts from scratch. Sometimes, the `ModifyStartPotential` feature (see next section) helps with this problem but sometimes it does not. A more robust way is to first perform a high-spin calculation and then modify the resulting adf.rkf file by “flipping” the spin on some atoms. This file then can be used to restart a subsequent low-spin calculation.

Such a “flipping” can be performed during restart by specifying the `SpinFlip` keyword in the *Restart* (page 344) input block as shown below:

```
Restart
   File HighSpin.results/adf.rkf
   ! SpinFlip keyword is followed by the indices of the
   ! atoms for which the flipping will be performed
   SpinFlip 1
End
```

See also:

The example *Broken spin-symmetry: Fe4S4* (page 731) and the AMS-GUI tutorial Spin Coupling in Fe4S4 Cluster demonstrate the use of the spin-flip option.

Modify the starting potential

In some applications you may want to modify the initial potential (from the restart file or the fragment files), see also the previous section. This is achieved with the block `ModifyStartPotential`. It allows you to scale the potential as to represent user-chosen amounts of spin-α and spin-β density on some or all of the fragments. This will adjust the spin-α and spin-β initial potentials.

This option applies only to *unrestricted* calculations of course. It may be used to help the program find a particular state. This might, for instance, be hard to find otherwise due to the a-b symmetry in the start-up situation. It may also be useful to speed up the SCF convergence in case you know what the final distribution of spin-α and spin-β density over the molecule will approximately be.

```
ModifyStartPotential {specification}
{ frag alfa // beta
 frag alfa // beta
 ...}
end
```

`specification` Must be *two numbers*, ASPIN and BSPIN, if provided at all. They specify the (relative) amounts of spin-α and spin-β fit density to define the spin-dependent potential at the first SCF cycle. The coefficients retrieved from the fragment files (or from the restart file in case of a SCF restart) are scaled accordingly. This will not affect the *total* amount of fit density: the absolute values of ASPIN and BSPIN play no role, only their ratio. In case of a restart run the restart file must have been generated in a *restricted* calculation, while the continuation run must be an *unrestricted* one.

If no argument is given a data block must be supplied with records frag alfa // beta. This is very much similar to the main option with ASPIN and BSPIN: you specify ASPIN and BSPIN now separately for each fragment. This involves somewhat more input but increases the possibilities to tune the initial potential. Again this can be applied only in an unrestricted calculation. It cannot be used in a restart: the affected fit coefficients are those from the fragment files, while in an SCF restart run these are ignored and replaced by the coefficients on the adf.rkf (TAPE21) restart file.

Each line specifies a frag with its corresponding ASPIN and BSPIN fit partitioning. If frag is the name of a fragment *type*, the specified ASPIN-BSPIN is applied to all individual fragments of that type. Alternatively an *individual* fragment can be specified, using the format fragtype/n, where *n* is an index between one and the total number of fragments of that type. In such a case the ASPIN-BSPIN data applies only to that particular fragment while different values may be supplied for the other fragments of the same type.

It is allowed to specify for certain fragment types individual fragments and for other fragment types only the type. Duplicate specifications are not allowed; an individual fragment must not be specified if its fragment type is also specified as a whole. If the data block form is used, only the fit coefficients of the referenced fragments are affected. For the not-referenced fragments the fit densities are used as they are defined on the corresponding fragment files.

The SCF convergence of a spin-unrestricted calculation usually improves when you start with potentials that correspond to the correct ratio of spin-α and spin-β electrons. By default ASPIN=BSPIN=0.5, as implied by the spin-restricted start density of the fragments or restricted molecule.

The total amount of fit density used on the first iteration is defined by the sum-of-fragment densities (or the density on the restart file). This may be different from the total nr. of electrons in the actual calculation. On the second SCF cycle the fit density will internally be normalized so as to represent the correct number of electrons.

The block-form of the key makes the start up of broken symmetry calculations easy. For example one may want to start a calculation in broken symmetry with spin-α density on one fragment and spin-β density on another, e.g. in a spin-unrestricted calculation of H_2 at large separation. It is particularly useful for larger systems, e.g. for magnetic coupling between spin-polarized magnetic centers, as in Fe-S complexes[7]: start with oppositely polarized Fe centers, but with, for instance, the remaining bridge and terminal ligands unpolarized.

---

[7] L. Noodleman, and E.J. Baerends, *Electronic Structure, Magnetic Properties, ESR, and Optical Spectra for 2-Fe Ferredoxin Models by LCAO-Xa Valence Bond Theory*, Journal of the American Chemical Society 106, 2316 (1984) (https://doi.org/10.1021/ja00320a017)

See also:

The example *Localized Hole: N2+* (page 729) demonstrates the use of the `ModifyStartPotential` option.

4.1.5 Unrestricted fragments

In ADF you can use spin-unrestricted fragments if the key UNRESTRICTEDFRAGMENTS is used.

```
UNRESTRICTEDFRAGMENTS
UNRESTRICTED
```

If spin-unrestricted fragments are used the full complex must also be calculated spin-unrestricted, which means that the key UNRESTRICTED is required, and in addition the SPINPOLARIZATION key and/or the IRREPOCCUPATIONS block key must be included. The key FRAGOCCUPATIONS should not be included. The spin-α and spin-β occupations are read from the spin-unrestricted fragment files. For spin-restricted fragments the spin-α and spin-β occupations will be set equal and sum up to the spin-restricted occupation numbers.

The unrestricted fragments can be self-consistent: different numbers of spin-α and spin-β electrons usually result in different spatial orbitals and different energy eigenvalues for spin-α and spin-β when you go to self-consistency.

Prepared for bonding

Typically an unrestricted electron configuration for the fragments is used, such that the Pauli repulsion between the fragments is minimal, which means that the fragments are so called 'prepared for bonding'. For example if one has two fragments which both have one unpaired electron, one would put the unpaired electron of the first fragment in the spin-α orbital and the unpaired electron of the second fragment in the spin-β orbital. If one wants to calculate separately the electron pair bonding see key *REMOVEALLFRAGVIRTUALS* (page 67).

4.1.6 Simulated unrestricted fragments with key FRAGOCCUPATIONS

In the method that simulates unrestricted fragments with the key FRAGOCCUPATIONS, the fragments from which the molecule is built must be spin-restricted, that is: the fragment files must be result files of spin-restricted calculations. For purposes of analysis, however, it may be desirable in some applications to build your molecule from fragments that have an *un*restricted electronic configuration. This can be simulated as follows.

You tell ADF that you want to *treat* the fragments as if they were unrestricted; this causes the program to duplicate the one-electron orbitals of the fragment: one set for spin-α and one set for spin-β. You can then specify occupation numbers for these spin-unrestricted fragments, and occupy spin-α orbitals differently from spin-β orbitals.

Of course, the unrestricted fragments that you use in this way, are not self-consistent: different numbers of spin-α and spin-β electrons usually result in different spatial orbitals and different energy eigenvalues for spin-α and spin-β when you go to self-consistency, while here you have spatially identical fragment orbitals. Nevertheless it is often a fair approximation which gives you a considerable extension of analysis possibilities.

Prepared for bonding

Typically an unrestricted electron configuration for the fragments is used, such that the Pauli repulsion between the fragments is minimal, which means that the fragments are so called ‘prepared for bonding’. For example if one has two fragments which both have one unpaired electron, one would put the unpaired electron of the first fragment in the spin-α orbital and the unpaired electron of the second fragment in the spin-β orbital. If one wants to calculate separately the electron pair bonding see key *REMOVEALLFRAGVIRTUALS* (page 67).

FRAGOCCUPATIONS

```
FRAGOCCUPATIONS
 fragtype
   irrep spin-a // spin-b
   irrep spin-a // spin-b
   ...
 subend
 fragtype
   irrep spin-a // spin-b
   ...
 subend
end
```

`fragtype` One of the fragment types and functions as a (block type) subkey. The data block for the subkey ends with the standard end code for block type subkeys (subend).

`irrep` One of the irreducible representations (irreps) for the point group symmetry that was used in the computation of that fragment.

`spin-a // spin-b` Two sequences of occupation numbers, which will be applied to the spin-α and spin-β versions of the Fragment Orbitals. The sequences must be separated by a double slash (//).

The sum of spin-α and spin-β occupations must, for each fragment orbital in each irrep separately, be equal to the total (restricted) occupation of that orbital as it is stored on the fragment file. In other words: you can only change the distribution over spin-α and spin-β electrons within one orbital.

(Without this restriction the spatial distribution of the total (sum over spins) fragment charge density would be changed, leading to an incorrect bonding energy analysis after the calculation).

Be aware that in more-dimensional irreps (E, T) the number of electrons in a fully occupied orbital is input as the dimension of the irrep times the one-electron orbital occupation. Compare the key IRREPOCCUPATIONS.

For irreps that are not mentioned in this input block, and hence for all irreps of fragment(type)s that are not mentioned at all, the spin-α and spin-β occupations will be set equal, which is of course what they in fact are on the (restricted) fragment file.

For an example of applying this option see ref.[8].

---

[8] F.M. Bickelhaupt, N.M. Nibbering, E.M. van Wezenbeek and E.J. Baerends, *The Central Bond in the Three CN* Dimers NC_CN, CN-CN, and CN-NC: Electron Pair Bonding and Pauli Repulsion Effects*, Journal of Physical Chemistry 96, 4864 (1992) (https://doi.org/10.1021/j100191a027)

4.1.7 Remove Fragment Orbitals

By default all fragment orbitals (the MOs of the fragment computation), which are stored on the fragment file, are used as basis functions for the overall molecule. You can remove one or more of these fragment orbitals from the basis set of the molecule. This may be useful for special analyzes, like calculating the electron pair bonding in case one has open shell fragments, or for instance to study the effect of deleting all virtual MOs of a particular fragment (CSOV analysis, Constrained Space Orbital Variation). It may also enhance the efficiency since you effectively reduce the size of the basis set, but you should be aware of the potential effects on the results.

The pure orbital interaction effect of forming electron pair bonding between open shell molecules can approximately be calculated with a bond energy analysis in which all virtual orbitals are removed from the fragments, see Ref.[9]. For calculating the effect of electron pair bonding best is to specify an unrestricted electron configuration for the fragments, such that the Pauli repulsion is minimal. Removing of all virtuals from an open shell fragment, that is calculated spin-restricted, means that all fragment orbitals with zero occupation are removed. Thus, for example, a singly occupied fragment orbital will not be removed. This singly occupied fragment orbital will result in a spin-α and a spin-β fragment orbital. In combination with other singly occupied fragment orbitals they may form an electron pair bonding combination, but also an anti-bonding combination. In practice this means that the orbital interaction calculated with a bond energy analysis in which all virtual orbitals are removed from open shell fragments might be due to more than electron pair bonding. The situation is not much different if one uses unrestricted fragments, because technically always the same number of alpha and beta orbitals must be removed, and the removed orbitals must all have zero occupation.

If one wants to remove all virtual fragment orbitals use the key REMOVEALLFRAGVIRTUALS.

```
REMOVEALLFRAGVIRTUALS
```

If one wants to remove all virtual fragment orbitals except on the first fragment use the key REMOVEOTHERFRAGVIRTUALS.

```
REMOVEOTHERFRAGVIRTUALS
```

If one does not want to remove all virtual fragment orbitals then one should use the block key REMOVEFRAGORBITALS.

```
REMOVEFRAGORBITALS
 fragtype
   subspecies nremove
   subspecies nremove
   ...
 subend
 fragtype
   subspecies nremove
   ...
 subend
 ...
 (etc.)
 ...
end
```

fragtype One of the fragment types in the system. Any subset of the available fragment types can be used here as subkey. The subkeys are block type keys; their data blocks end subend.

subspecies One of the subspecies of the irreducible representations of the point group symmetry that was used in the calculation of the fragment itself. This requires of course that one knows the symmetry that has been used for the fragment calculation.

---

[9] F.M. Bickelhaupt, M. Solà, C. Fonseca Guerra, *Highly polar bonds and the meaning of covalency and ionicity – structure and bonding of alkali metal hydride oligomers*, Faraday Discussions 135, 451 (2007) (https://doi.org/10.1039/B606093E)

nremove The number of fragment orbitals of the pertaining representation that will not be used as basis functions for the overall system. The *highest* (in energy eigenvalue) nremove orbitals are discarded. You must not remove *occupied* fragment orbitals.

By default (omission of the key) all fragment orbitals are used in the basis set for the system.

Important Note

It is imperative that any removal of fragment orbitals will not break the symmetry of the molecule. This consideration is relevant when for instance two different subspecies of a fragment irrep contribute to different partner subspecies in one of the irreps of the molecule. In such a case, when one removes an orbital in such a fragment subspecies, its partner orbital should also be removed. If this is violated an error may occur or the results will simply be wrong. Quite likely, the program will detect the error, but this may occur only in the final (analysis) stage of the calculation so that a lot of CPU time may have been wasted.

Example: consider a single-atom fragment, computed in atom symmetry, used as fragment in a c(lin) molecule and assume that the p:x and p:y fragment orbitals contribute to respectively the pi:x and pi:y subspecies of the molecule. Then, when you remove one or more p:x fragment orbitals, you should also remove the same number of p:y fragment orbitals. Practical cases may be more complicated and whenever you use this key, make sure that you've fully analyzed and understood how the fragment irreps combine into the molecular symmetry representations. Hint: run the molecule, without removing any fragment orbitals, and stop at an early stage after the program has computed and printed the build-up of the molecular SFOs from the fragment orbitals. To control early aborts via input, use the key STOPAFTER.

4.1.8 CDFT: Constrained Density Functional Theory

CDFT is a tool for carrying out DFT calculations in the presence of a constraint. The nature of the constraint is general in theory, however, in the current implementation the user can constrain the CHARGEs or the SPINs of a set of moieties (as identified by sets of atoms) to be a specific real number given in input. Note that the use of CDFT as implemented in ADF is an **expert option**, and it is a **work in progress**. Implemented in ADF by M. Pavanello and P. Ramos[?], based on the method described in Ref.[10]. At the moment SYMMETRY NOSYM and an all electron basis set are required.

Warning: CDFT is an expert option that may not work for your system or the specific constraint you had in mind.

The simplest way to run CDFT is using the following keyword combination

```
CDFT
  NCONSTRAINTS 1
  NATOMSPERSET N
  THEATOMS atom1 atom2 ... atomN
  CONSTRAINTS charge
END
```

All the CDFT block key options are:

```
CDFT
   AllAtoms Yes/No
   AnalyticalHessian integer
   ChargeAndSpin Yes/No
   Constraints float_list
   DoNotOptimize Yes/No
   ExcitedCDFT Yes/No
   InitialMultipliers float_list
```

(continues on next page)

---

[10] Q. Wu, T. Van Voorhis, *Direct optimization method to study constrained systems within density-functional theory*, Physical Review A 72, 024502 (2005) (https://doi.org/10.1103/PhysRevA.72.024502)

(continued from previous page)

```
   MaxIter integer
   Metric Yes/No
   NAtomsPerSet integer_list
   NConstraints integer
   OnlyCharge Yes/No
   OnlySpin Yes/No
   PopType [yukawalike | fuzzyvoronoibecke | fuzzyvoronoifermi]
   Print [low | medium | high]
   SelfConsistent Yes/No
   StepSize float
   TheAtoms integer_list
   Threshold float
End
```

`CDFT`

Type Block

Description CDFT is a tool for carrying out DFT calculations in the presence of a constraint.

`AllAtoms`

Type Bool

Default value No

Description If AllAtoms is true, then TheAtoms is overridden and all the atoms in the active fragment are included in the set.

`AnalyticalHessian`

Type Integer

Default value 0

Description This will calculate the analytical derivative of the energy w.r.t. the Lagrange multiplier up to the specified SCF iteration. This key is not recommended due to the high computational cost that comes with it. The calculation is equivalent to a ground state Hessian, and it is carried out with the full sum-over-states formula.

`ChargeAndSpin`

Type Bool

Default value No

Description will constrain both the charge and the spin

`Constraints`

Type Float List

Description The values of the constraints. If CHARGEANDSPIN, constraints to the alpha and beta electrons need to be specified sequentially. One more electron => CONSTRAINTS -1.0. One less electron => CONSTRAINTS 1.0. If the CDFT type is EXCITEDCDFT, CONSTRAINTS=1.0 is recommended. Other values are technically possible but have not been tested yet.

`DoNotOptimize`

Type Bool

Default value No

Description If true, the multipliers chosen in INITIALMULTIPLIERS will not be optimized and will be constant throughout the entire SCF procedure.

`ExcitedCDFT`

Type Bool

Default value No

Description will generate an excited state with CONSTRAINTS number of ALPHA electrons constrained to occupy the virtual space of a ground state reference calculation. This is the essence of the eXcited Constrained DFT (XCDFT) method(P. Ramos, M. Pavanello, Low-lying excited states by constrained DFT, Journal of Chemical Physics 148, 144103 (2018) https://doi.org/10.1063/1.5018615) for the calculation of low-lying single excitations. XCDFT is found to correctly reproduce the energy surface topology at conical intersections between the ground state and the first singly excited state and can also accounts for the condensed phase effects in solvated chromophores where typical Delta SCF methods variationally collapse.

`InitialMultipliers`

Type Float List

Description If available, a guess for the Lagrange multipliers can be entered.

`MaxIter`

Type Integer

Default value 200

Description Maximum number of CDFT iterations. CDFT carries out a loop nested inside the SCF cycle.

`Metric`

Type Bool

Default value No

Description Relevant for XCDFT. In the XCDFT method orthogonality is not imposed between the KS-orbitals of ground and excited states. If METRIC is specified, the degree of mixing of the single excited state with the ground state or high-order excitations is calculated. Three parameters are calculated: p, m and d. The parameters p and m will give information about the amount of mixing with the ground state, while parameter d will determine the mixing with high order excitations. Additional information about the origin of these parameters can be found in the literature (P. Ramos, M. Pavanello, Low-lying excited states by constrained DFT, Journal of Chemical Physics 148, 144103 (2018) https://doi.org/10.1063/1.5018615)

`NAtomsPerSet`

Type Integer List

Description The number of atoms in each moiety (set).

`NConstraints`

Type Integer

Default value 1

Description This specifies the number of sets of atoms to be considered. For example, if the user wishes to constrain a positive charge on one part of the system, and a negative charge on another part, NCONSTRAINTS should be set to two. There is no limit on the number of constraints. However, SCF convergence becomes an issue with more than 2 constraints. Note: NCONSTRAINTS>1 is untested.

`OnlyCharge`

Type Bool

Default value Yes

Description Will constrain only the charge, letting spin relax (and potentially delocalize)

`OnlySpin`

Type Bool

Default value No

Description Will constrain only the spin

`PopType`

Type Multiple Choice

Default value yukawalike

Options [yukawalike, fuzzyvoronoibecke, fuzzyvoronoifermi]

Description The population analysis chosen for determining the constraint.

`Print`

Type Multiple Choice

Default value low

Options [low, medium, high]

Description Print level and debugging.

`SelfConsistent`

Type Bool

Default value No

Description Self-Consistent CDFT

`StepSize`

Type Float

Default value 0.5

Description The amount of the Lagrange multipliers step taken in each CDFT iteration

`TheAtoms`

Type Integer List

Description The atom numbers of the moieties in the input geometry order. If NCONSTRAINTS is larger than 1, the sets of atoms are entered as a single list.

`Threshold`

Type Float

Default value 1e-10

Description The threshold for convergence of the CDFT constraints. The tighter the SCF convergence criteria, the tighter the THRESHOLD should be.

4.2 Density Functionals (XC)

The Density Functional, also called the exchange-and-correlation (XC) functional, consists of an LDA, a GGA part, a Hartree-Fock exchange part (hybrids), and a meta-GGA part (meta-GGA or meta-hybrid). Possibly, it also depends on virtual Kohn-Sham orbitals through inclusion of an orbital-dependent correlation (double-hybrids). LDA stands for the Local Density Approximation, which implies that the XC functional in each point in space depends only on the (spin) density in that same point. GGA stands for Generalized Gradient Approximation and is an addition to the LDA part, by including terms that depend on derivatives of the density. A hybrid GGA (for example B3LYP) stands for some combination of a standard GGA with a part of Hartree-Fock exchange. A meta-GGA (for example TPSS) has a GGA part, but also depends on the kinetic energy density. A meta-hybrid (for example TPSSh) has GGA part, a part of Hartree-Fock exchange and a part that depends on the kinetic energy density. For these terms ADF supports a large number of the formulas advocated in the literature. For post-SCF energies only, ADF supports also various other meta-GGA functionals and more hybrid functionals. A double-hybrid has a hybrid or a meta-hybrid part, but also contains a contribution from second-order Møller-Plesset perturbation theory (MP2). Here, only the hybrid (meta-hybrid) part is evaluated self-consistently, whereas the MP2 part is evaluated post-SCF and added to the hybrid (meta-hybrid) energy.

The key that controls the Density Functional is `XC`. All subkeys are optional.

```
XC
  {LDA LDA {Stoll}}
  {GGA GGA}
  {MetaGGA metagga}
  {Model MODELPOT [IP]}
  {HartreeFock}
  {HYBRID hybrid {HF=HFpart}}
  {MetaHYBRID metahybrid}
  {DOUBLEHYBRID doublehybrid}
  {RPA {option}}
  {XCFUN}
  {RANGESEP {GAMMA=X} {ALPHA=a} {BETA=b}}
  (LibXC functional}
  {DISPERSION [s6scaling] [RSCALE=r0scaling] [Grimme3] [BJDAMP] [PAR1=par1] ␣
↪[PAR2=par2]  [PAR3=par3]  [PAR4=par4] }
  {Dispersion Grimme4 {s6=...} {s8=...} {a1=...} {a2=...}}
  {DISPERSION dDsC}
  {DISPERSION UFF}
end
```

If the `XC` key is omitted, the program will apply only the Local Density Approximation (no GGA terms). The chosen LDA form is then VWN.

4.2.1 LDA

```
XC
  LDA {functional} {Stoll}
End
```

`LDA` Defines the LDA part of the XC functional. If `functional` is omitted, VWN will be used (also if LYP is specified in the GGA part).

Available LDA functionals:

- **Xonly**: The pure-exchange electron gas formula. Technically this is identical to the Xalpha form (see next) with a value 2/3 for the X-alpha parameter.

- **Xalpha**: The scaled (parametrized) exchange-only formula. When this option is used you may (optionally) specify the X-alpha *parameter* by typing a numerical value after the string Xalpha (separated by a blank). If omitted this parameter takes the default value 0.7
- **VWN**: The parametrization of electron gas data given by Vosko, Wilk and Nusair (ref[1], formula version V). Among the available LDA options this is the more advanced one, including correlation effects to a fair extent.
    - `Stoll`: For the VWN variety of the LDA form you may include Stoll's correction[2] by typing Stoll on the same line, after the main LDA specification. You must not use Stoll's correction in combination with the Xonly or the Xalpha form for the Local Density functional. The Stoll formula is considered to be a *correlation* correction to the *Local* Density Approximation. It is conceptually not correct to use the Stoll correction *and* apply gradient (GGA) corrections to the correlation. It is the user's responsibility, in general and also here, to avoid using options that are not solidly justified theoretically.
- **PW92**: the parametrization of electron gas data given by Perdew and Wang (ref[3]).

4.2.2 GGA

```
XC
   GGA functional
End
```

GGA Specifies the GGA part of the XC Functional (in earlier times often called the 'non-local' correction to the LDA part of the density functional). It uses derivatives (gradients) of the charge density.

Available GGA functionals:

- **BP86**: Exchange: Becke, Correlation: Perdew
- **PW91**: Exchange: pw91x, Correlation: pw91c
- **mPW**: Exchange: mPWx, Correlation: pw91c
- **PBE**: Exchange: PBEx, Correlation: PBEc
- **RPBE**: Exchange: RPBEx, Correlation: PBEc
- **revPBE**: Exchange: revPBEx, Correlation: PBEc
- **mPBE**: Exchange: mPBEx, Correlation: PBEc
- **PBEsol**: Exchange: PBEsolx, Correlation: PBEsolc
- **HTBS**: Exchange: HTBSx, Correlation: PBEc
- **BLYP**: Exchange: Becke, Correlation: LYP
- **OLYP**: Exchange: OPTX, Correlation: LYP
- **OPBE**: Exchange: OPTX, Correlation: PBEc[4]
- **BEE**: Exchange: BEEx, Correlation: PBEc

---

[1] S.H. Vosko, L. Wilk and M. Nusair, *Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis*, Canadian Journal of Physics 58 (8), 1200 (1980) (https://doi.org/10.1139/p80-159)

[2] H. Stoll, C.M.E. Pavlidou, and H. Preuss, *On the calculation of correlation energies in the spin-density functional formalism*, Theoretica Chimica Acta 49, 143 (1978) (https://doi.org/10.1007/BF00553794)

[3] J.P. Perdew and Y. Wang, *Accurate and simple analytic representation of the electron-gas correlation energy*, Physical Review B 45, 13244 (1992) (https://doi.org/10.1103/PhysRevB.45.13244)

[4] M. Swart, A.W. Ehlers and K. Lammertsma, *Performance of the OPBE exchange-correlation functional*, Molecular Physics 2004 102, 2467 (2004) (https://doi.org/10.1080/0026897042000275017)

- **XLYP**: Exchange: XLYPx[5] (exchange, not available separately from LYP) + LYP
- **SSB-D**: Dispersion corrected functional by Swart-Solà-Bickelhaupt[62][63]. The SSB-D functional by definition already includes a dispersion correction by Grimme (factor 0.847455). There are some numerical issues with the GGA implementation in ADF of SSB-D (Ref.[??]) for some systems. Because of this, the GGA SSB-D option is only available for single-points (and NMR). Geometry optimizations (etc.) are still possible by using instead:

  ```
  XC
    METAGGA SSB-D
  END
  ```

  This METAGGA implementation is only possible with all-electron basis sets. Use GGA SSB-D for NMR calculations.
- **S12g**: Dispersion corrected (Grimme-D3) functional by Swart, successor of SSB-D[6].
- **LB94**: By Van Leeuwen and Baerends[7].
- **KT1**: By Keal and Tozer[8].
- **KT2**: By Keal and Tozer[?].

If only a GGA part is specified (omitting the `LDA` sub key) the LDA part defaults to VWN, except when the LYP correlation correction is used: in that case the LDA default is Xonly: pure exchange. The reason for this is that the LYP formulas assume the pure-exchange LDA form, while for instance the Perdew-86 correlation correction is a correction to a *correlated* LDA form. The precise form of this correlated LDA form assumed in the Perdew-86 correlation correction is not available as an option in ADF but the VWN formulas are fairly close to it.

Separate choices can be made for the GGA exchange correction and the GGA correlation correction respectively. Both specifications must be typed (if at all) on the same line, after the `GGA` subkey.

For the **exchange** part the options are:

- **Becke**: Becke (1988)[9].
- **PW86x**: Perdew-Wang (1986)[10].
- **PW91x**: Perdew-Wang (1991)[11]
- **mPWx**: Modified PW91 by Adamo-Barone (1998)[12]
- **PBEx**: Perdew-Burke-Ernzerhof (1996)[13]

---

[5] X. Xu and W.A. Goddard III, *The X3LYP extended density functional for accurate descriptions of nonbond interactions, spin states, and thermochemical properties*, Proceedings of the National Academy of Sciences 101, 2673 (2004) (https://doi.org/10.1073/pnas.0308730100)

[62] M. Swart, M. Solà and F.M. Bickelhaupt, *A new all-round DFT functional based on spin states and SN2 barriers*, Journal of Chemical Physics 131, 094103 (2009) (https://doi.org/10.1063/1.3213193)

[63] M. Swart, M. Solà and F.M. Bickelhaupt, *Switching between OPTX and PBE exchange functionals*, Journal of Computational Methods in Science and Engineering 9, 69 (2009) (https://doi.org/10.3233/JCM-2009-0230)

[6] M. Swart, *A new family of hybrid density functionals*, Chemical Physics Letters 580, 166 (2013) (https://doi.org/10.1016/j.cplett.2013.06.045)

[7] R. van Leeuwen and E.J. Baerends, *Exchange-correlation potential with correct asymptotic behavior*, Physical Review A 49, 2421 (1994) (https://doi.org/10.1103/PhysRevA.49.2421)

[8] T.W. Keal and D.J. Tozer, *The exchange-correlation potential in Kohn.Sham nuclear magnetic resonance shielding calculations*, Journal of Chemical Physics 119, 3015 (2003) (https://doi.org/10.1063/1.1590634)

[9] A.D. Becke, *Density-functional exchange-energy approximation with correct asymptotic behavior*, Physical Review A 38, 3098 (1988) (https://doi.org/10.1103/PhysRevA.38.3098)

[10] J.P. Perdew and Y. Wang, *Accurate and simple density functional for the electronic exchange energy: generalized gradient approximation*, Physical Review B 33, 8822 (1986) (https://doi.org/10.1103/PhysRevB.33.8800)

[11] J.P. Perdew, J.A. Chevary, S.H. Vosko, K.A. Jackson, M.R. Pederson, D.J. Sing and C. Fiolhais, *Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation*, Physical Review B 46, 6671 (1992) (https://doi.org/10.1103/PhysRevB.46.6671)

[12] C. Adamo and V. Barone, *Exchange functionals with improved long-range behavior and adiabatic connection methods without adjustable parameters: The mPW and mPW1PW models*, Journal of Chemical Physics 108, 664 (1998) (https://doi.org/10.1063/1.475428)

[13] J.P. Perdew, K. Burke and M. Ernzerhof, *Generalized Gradient Approximation Made Simple*, Physical Review Letters 77, 3865 (1996) (https://doi.org/10.1103/PhysRevLett.77.3865)

- **RPBEx**: revised PBE by Hammer-Hansen-Norskov (1999)[14]
- **revPBEx**: revised PBE by Zhang-Wang (1998)[15]
- **mPBEx**: Modified PBE by Adamo-Barone (2002)[16]
- **PBEsolx**: Perdew-Ruzsinszky-Csonka-Vydrov-Scuseria (2008)[17]
- **HTBSx**:[18]
- **OPTX**: Handy-Cohen (2001)[19]
- **BEEx**: Mortensen-Kaasbjerg-Frederiksen-Nørskov-Sethna-Jacobsen (2005)[20]

For the **correlation** part the options are:

- **Perdew**: Perdew (1986)[21].
- **PBEc**: Perdew-Burke-Ernzerhof (1996)[?] .
- **PBEsolc**: The PBEsol correlation correction by Perdew-Ruzsinszky-Csonka-Vydrov-Scuseria (2008)[?]
- **PW91c**: Perdew-Wang (1991), see[?].
- **LYP**: Lee-Yang-Parr (1988) correlation correction[64][65][66].

The string GGA must contain not more than one of the exchange options and not more than one of the correlation options. If options are applied for both they must be separated by a blank or a comma. Example:

```
XC
  GGA Becke Perdew
End
```

is equivalent to

```
XC
  GGA BP86
End
```

It is questionable to apply gradient corrections to the *correlation*, while not doing so at the same time for the exchange. Therefore, the program will check this and stop with an error message. This check can be overruled with the key ALLOW.

---

[14] B. Hammer, L.B. Hansen, and J.K. Norskøv, *Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals*, Physical Review B 59, 7413 (1999) (https://doi.org/10.1103/PhysRevB.59.7413)

[15] Y. Zhang and W. Yang, *Comment on "Generalized Gradient Approximation Made Simple"*, Physical Review Letters 80, 890 (1998) (https://doi.org/10.1103/PhysRevLett.80.890)

[16] C. Adamo and V. Barone, *Physically motivated density functionals with improved performances: The modified Perdew.Burke.Ernzerhof model*, Journal of Chemical Physics 1996 116, 5933 (1996) (https://doi.org/10.1063/1.1458927)

[17] J.P. Perdew, A. Ruzsinszky, G.I. Csonka, O.A. Vydrov, G.E. Scuseria, *Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces*, Physical Review Letters 100, 136406 (2008) (https://doi.org/10.1103/PhysRevLett.100.136406)

[18] Ph. Haas, F. Tran, P. Blaha, and K.H. Schwartz, *Construction of an optimal GGA functional for molecules and solids*, Physical Review B83, 205117 (2011) (https://doi.org/10.1103/PhysRevB.83.205117).

[19] N.C. Handy and A.J. Cohen, *Left-right correlation energy*, Molecular Physics 99, 403 (2001) (https://doi.org/10.1080/00268970010018431)

[20] J.J. Mortensen, K. Kaasbjerg, S.L. Frederiksen, J.K. Nørskov, J.P. Sethna, and K.W. Jacobsen, *Bayesian Error Estimation in Density-Functional Theory*, Physical Review Letters 95, 216401 (2005) (https://doi.org/10.1103/PhysRevLett.95.216401)

[21] J.P. Perdew, *Density-functional approximation for the correlation energy of the inhomogeneous electron gas*, Physical Revied B 33, 8822 (1986) (https://doi.org/10.1103/PhysRevB.33.8822) Erratum: J.P. Perdew, Physical Review B 34, 7406 (1986) (https://doi.org/10.1103/PhysRevB.34.7406)

[64] C. Lee, W. Yang and R.G. Parr, *Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density*, Physical Review B 37, 785 (1988) (https://doi.org/10.1103/PhysRevB.37.785)

[65] B.G. Johnson, P.M.W. Gill and J.A. Pople, *The performance of a family of density functional methods*, Journal of Chemical Physics 98, 5612 (1993) (https://doi.org/10.1063/1.464906)

[66] T.V. Russo, R.L. Martin and P.J. Hay, *Density Functional calculations on first-row transition metals*, Journal of Chemical Physics 101, 7729 (1994) (https://doi.org/10.1063/1.468265)

4.2.3 MetaGGA

```
XC
  MetaGGA functional
End
```

`MetaGGA` Specifies that a meta-GGA should be used during the SCF. **All electron basis sets should be used** (see *Basis key* (page 42)).

Available meta-GGA functionals:

- **M06-L**: Functional by Yan-Truhlar[58][59]
- **TPSS**: Functional by Tao-Perdew-Staroverov-Scuseria[60][61]
- **revTPSS**: Revised TPSS functional[22]
- **SSB-D**: Dispersion corrected GGA functional by Swart-Solà-Bickelhaupt[?][?]. Use GGA SSB-D for NMR calculations.
- **MVS**: Functional by Sun-Perdew-Ruzsinszky[23]
- **MS0**: Functional by Sun et al.[24]
- **MS1**: Functional by Sun et al.[25]
- **MS2**: Functional by Sun et al.[?]
- **SCAN**: Functional by Sun et al.[26]
- **TASKxc**: Functional by Aschebrock et al.[32], intended for charge transfer systems.
- **TASKCC**: Functional by Lebeda et al.[34], improves TASKxc atomization energies
- **r2SCAN-3c**: Composite DFT method by Gasevic et al.[33]

The r2SCAN-3c composite method uses the r^2 SCAN (r2SCAN) exchange-correlation functional, in combination with a tailor-made all electron polarized basis set (mTZ2P), the semiclassical London dispersion correction (D4), and a geometrical counterpoise (gCP) correction. NumericalQuality should be Good, and ZORA should be used. Note that internally LibXC will be used for the r2SCAN functional, and automatically the D4 and gCP corrections will be included. The

---

[58] Y. Zhao and D.G. Truhlar, *A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions*, Journal of Chemical Physics 125, 194101 (2006) (https://doi.org/10.1063/1.2370993)

[59] Y. Zhao and D.G. Truhlar, *The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals*, Theoretical Chemical Accounts 120, 215 (2008) (https://doi.org/10.1007/s00214-007-0310-x)

[60] J. Tao, J.P. Perdew, V.N. Staroverov and G.E. Scuseria, *Climbing the Density Functional Ladder: Nonempirical MetaGeneralized Gradient Approximation Designed for Molecules and Solids* Physical Review Letters 91, 146401 (2003) (https://doi.org/10.1103/PhysRevLett.91.146401)

[61] V.N. Staroverov, G.E. Scuseria, J. Tao and J.P. Perdew, *Comparative assessment of a new non empirical density functional: Molecules and hydrogen-bonded complexes* Journal of Chemical Physics 119, 12129 (2003) (https://doi.org/10.1063/1.1626543)

[22] J.P. Perdew, A. Ruzsinszky, G.I. Csonka, L.A. Constantin, and J. Sun, *Workhorse Semilocal Density Functional for Condensed Matter Physics and Quantum Chemistry*, Physical Review Letters 103, 026403 (2009) (https://doi.org/10.1103/PhysRevLett.103.026403).

[23] J. Sun, J.P. Perdew, and A. Ruzsinszky, *Semilocal density functional obeying a strongly tightened bound for exchange*, Proceedings of the National Academy of Sciences 112, 685 (2015) (https://doi.org/10.1073/pnas.1423145112)

[24] J. Sun, B. Xiao, A. Ruzsinszky, *Communication: Effect of the orbital-overlap dependence in the meta generalized gradient approximation*, Journal of Chemical Physics 137, 051101 (2012) (https://doi.org/10.1063/1.4742312).

[25] J. Sun, R. Haunschild, B. Xiao, I.W. Bulik, G.E. Scuseria, J.P. Perdew, *Semilocal and hybrid meta-generalized gradient approximations based on the understanding of the kinetic-energy-density dependence*, Journal of Chemical Physics 138, 044113 (2013) (https://doi.org/10.1063/1.4789414).

[26] J. Sun, A. Ruzsinszky, J.P. Perdew, *Strongly Constrained and Appropriately Normed Semilocal Density Functional*, Physical Review Letters 115, 036402 (2015) (https://doi.org/10.1103/PhysRevLett.115.036402).

[32] T. Aschebrock, and S. Kümmel, *Ultranonlocality and accurate band gaps from a meta-generalized gradient approximation*, Phys. Rev. Research 1, 033082 (2019) (https://doi.org/10.1103/PhysRevResearch.1.033082)

[34] T. Lebeda, T. Aschebrock, and S. Kümmel, *First steps towards achieving both ultranonlocality and a reliable description of electronic binding in a meta-generalized gradient approximation*, Phys. Rev. Research 4, 023061 (2022) (https://doi.org/10.1103/PhysRevResearch.4.023061)

[33] T. Gasevic, J.B. Stückrath, S. Grimme, and M. Bursch, *Optimization of the r2SCAN-3c Composite Electronic-Structure Method for Use with Slater-Type Orbital Basis Sets*, Journal of Physical Chemistry A 126, 3826 (2022) (https://doi.org/10.1021/acs.jpca.2c02951)

STO-optimized r2SCAN-3c outperforms many conventional hybrid/QZ approaches in most common applications at a fraction of their cost.

The M06-L functional needs high integration accuracy (at least *BeckeGrid quality good* (page 321)) for reasonable gradients. For TPSS moderate integration accuracy for reasonable gradients is sufficient. For heavier elements (Z>36) and if one uses the M06-L functional it is also necessary to include the following keyword

```
FragMetaGGAToten
```

Using this key FRAGMETAGGATOTEN the difference in the meta-hybrid or meta-GGA exchange-correlation energies between the molecule and its fragments will be calculated using the molecular integration grid, which is more accurate than the default, but is much more time consuming. Default is to calculate the meta-GGA exchange-correlation energies for the fragments in the numerical integration grid of the fragments.

4.2.4 Hartree-Fock

```
XC
  HartreeFock
End
```

`HartreeFock` Specifies that the Hartree-Fock exchange should be used during the SCF.

4.2.5 Hybrid

```
XC
  HYBRID functional {HF=HFpart}
End
```

`HYBRID` Specifies that a hybrid functional should be used during the SCF.

Available Hybrid functionals:

- **B3LYP**: ADF uses VWN5 in B3LYP. functional (20% HF exchange) by Stephens-Devlin-Chablowski-Frisch[27].
- **B3LYP***: Modified B3LYP functional (15% HF exchange) by Reiher-Salomon-Hess[28].
- **B1LYP**: Functional (25% HF exchange) by Adamo-Barone[29].
- **KMLYP**: Functional (55.7% HF exchange) by Kang-Musgrave[30].
- **O3LYP**: Functional (12% HF exchange) by Cohen-Handy[31].
- **X3LYP**: Functional (21.8% HF exchange) by Xu-Goddard[?].
- **BHandH**: 50% HF exchange, 50% LDA exchange, and 100% LYP correlation.
- **BHandHLYP**: 50% HF exchange, 50% LDA exchange, 50% Becke88 exchange, and 100% LYP correlation.
- **B1PW91**: Functional by (25% HF exchange) Adamo-Barone[?].

---

[27] P.J. Stephens, F.J. Devlin, C.F. Chabalowski and M.J. Frisch, *Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields*, Journal of Physical Chemistry 98, 11623 (1994) (https://doi.org/10.1021/j100096a001)

[28] M. Reiher, O. Salomon and B.A. Hess, *Reparameterization of hybrid functionals based on energy differences of states of different multiplicity*, Theoretical Chemistry Accounts 107, 48 (2001) (https://doi.org/10.1007/s00214-001-0300-3)

[29] C. Adamo and V. Barone, *Toward reliable adiabatic connection models free from adjustable parameters*, Chemical Physics Letters 274, 242 (1997) (https://doi.org/10.1016/S0009-2614(97)00651-9)

[30] J.K. Kang and C.B. Musgrave, *Prediction of transition state barriers and enthalpies of reaction by a new hybrid density-functional approximation*, Journal of Chemical Physics 115, 11040 (2001) (https://doi.org/10.1063/1.1415079)

[31] A.J. Cohen and N.C. Handy, *Dynamic correlation*, Molecular Physics 99, 607 (2001) (https://doi.org/10.1080/00268970010023435)

- **mPW1PW**: Functional (25% HF exchange) by Adamo-Barone?.
- **mPW1K**: Functional (42.8% HF exchange) by Lynch-Fast-Harris-Truhlar[35].
- **PBE0**: Functional (25% HF exchange) by Ernzerhof-Scuseria[36] and by Adamo-Barone[37], hybrid form of PBE.
- **OPBE0**: Functional (25% HF exchange) by Swart-Ehlers-Lammertsma?, hybrid form of OPBE.
- **S12H**: Dispersion corrected (Grimme-D3) functional (25% HF exchange) by Swart?.

`HFpart` Specifies the amount of HF exchange that should be used in the functional, instead of the default HF exchange percentage for the given hybrid. Example HF=0.25 means 25% Hartree-Fock exchange.

4.2.6 Meta-Hybrid

```
XC
  MetaHYBRID functional
End
```

`MetaHYBRID` Specifies that a meta-hybrid functional should be used during the SCF.

Available meta-hybrid functionals:

- **M06**: Functional (27% HF exchange) by Yan-Truhlar??
- **M06-2X**: Functional (54% HF exchange) by Yan-Truhlar??
- **M06-HF**: Functional (100% HF exchange) by Yan-Truhlar??
- **TPSSH**: Functional (10% HF exchange) by Tao-Perdew-Staroverov-Scuseria??

4.2.7 Range separated hybrids

In ADF there are two (mutually exclusive) ways of specifying range separated hybrids functionals:

- Through the `RANGESEP` and `XCFUN` keys. This will use the Yukawa potential as switching function, see Ref.[38];
- By specifying a range separated functional via the `LibXC` key.

See also the advanced tutorial: Tuning the range separation in LC-wPBE for organic electronics (https://www.scm.com/news/tuning-range-separation-lc-wpbe-organic-electronics/)

---

[35] B.J. Lynch, P.L. Fast, M. Harris and D.G. Truhlar, *Adiabatic Connection for Kinetics*, Journal of Physical Chemistry A 104, 4811 (2000) (https://doi.org/10.1021/jp000497z)

[36] S. Grimme, *Accurate description of van der Waals complexes by density functional theory including empirical corrections*, Journal of Computational Chemistry 25, 1463 (2004) (https://doi.org/10.1002/jcc.20078)

[37] M. Ernzerhof and G. Scuseria, *Assessment of the Perdew.Burke.Ernzerhof exchange-correlation functional*, Journal of Chemical Physics 110, 5029 (1999) (https://doi.org/10.1063/1.478401)

[38] M. Seth and T. Ziegler, *Range-Separated Exchange Functionals with Slater-Type Functions*, Journal of Chemical Theory and Computation 8, 901 (2012) (https://doi.org/10.1021/ct300006h)

RangeSep + XCFun: Yukawa-range separated hybrids

`RANGESEP {GAMMA=X} {ALPHA=a} {BETA=b}` If RANGESEP is included, by default a long-range corrected (LC) functional is created with range separation parameter GAMMA of 0.75. As switching function in ADF the Yukawa potential is utilized, see Ref.?. Range separated functionals require XCFUN and are limited to GGA, meta-GGA, and CAMY-B3LYP. The CAMY-B3LYP functional is not the same as the CAM-B3LYP functional, since a different switching function is used. No other hybrids or meta-hybrids are supported. The special CAMYB3LYP functional is defined by three parameters, ALPHA, BETA and the attenuation parameter GAMMA. For CAMYB3LYP by default ALPHA is 0.19, BETA is 0.46, and GAMMA is 0.34.

Range-separated functionals make use of a modified form of the Coulomb operator that is split into pieces for exact exchange and DFT. As switching function in ADF the Yukawa potential is utilized, see Ref.?. Global hybrids can be thought of as a special case of a range-separated functional where the split is independent of the inter-electronic distance and is a simple X exact and 1-X DFT in all space.

In a general RS-functional the split depends on the inter-electronic distance. How the split is achieved depends on the functional in question but it is achieved using functions that smoothly go from 1 to 0. In ADF an exponential function is used (the error function is common in Gaussian based codes). In a range-separated function the potential goes from a Coulomb interaction to a sum of Coulomb functions attenuated by an exponential function.

In practical terms, this means that a range-separated functional looks very much like a hybrid (or meta-hybrid) functional but with additional integrals over the attenuated interaction with fit functions on the exact exchange side and a modified functional on the DFT side.

DFT part of RS-functionals

Using Hirao's approach for creating RS-functionals, the RS form of a given exchange functional is created by multiplying the standard energy density by a factor that depends on the energy density. The factor is the same for all functionals and the only difference is introduced by the individual energy densities.

The range-separation comes in at the level of the integrals over the operator with fit functions. They are very similar to the standard Coulomb integrals.

RS-functionals

An RS-functional is described by a series of regions describing each of the pieces of the Coulomb operator. The total function is built up by looping over the regions and adding up all the pieces. Currently, simple LC functionals can be defined where the exact exchange goes from 0 to 1 as the inter-electronic distance increases and the DFT part does the reverse. In addition, CAMY-B3LYP type functionals can be defined. More general functionals are not possible yet.

Functionality/Limitations

RS functionals with XCFUN are limited to the GGA and meta-GGA functionals and one hybrid CAMY-B3LYP. The following functionals can be evaluated with range-separation at the present time:

- LDA: VWN5, X-ALPHA PW92
- GGA exchange: Becke88, PBEX, OPTX, PW91X, mPW, revPBEX
- GGA correlation: LYP, Perdew86, PBEC
- MetaGGA: TPSS, M06L, B95
- Hybrids: CAMY-B3LYP

The following functionality has been tested: XC potential, energy, ground state geometry, TDDFT. Starting from ADF2018 singlet-triplet excitation calculations and excited state geometry optimizations are possible. See for possible limitations in case of excitation calculations or excited state geometry optimizations the corresponding part of the ADF manual.

Numerical stability

The range-separated implementation requires that the range-separation parameter is not too close to the exponent of a fit function. In practice this means that values of the separation parameter between 1.0 and 50 can cause numerical problems. Typical useful values are in the range 0.2 to 0.9 so this should not be too serious a limitation.

```
XC
  XCFUN
  RANGESEP {GAMMA=X} {ALPHA=a} {BETA=b}
END
```

Range separation is activated by putting RANGESEP in the XC block. Inclusion of XCFUN is required, see the *XCFUN description* (page 89). By default a long-range corrected (LC) functional is created with range separation parameter of 0.75. The parameter can be changed by modifying X in GAMMA=X in the RANGESEP card. Range separation typically will be used in combination with a GGA or METAGGA functional.

Range separation can not be included with a hybrid or meta-hybrid, with one exception, the special RS functional: CAMY-B3LYP. This is entered as HYBRID CAMY-B3LYP and must be used in combination with XCFUN (see *XCFUN description* (page 89)) and RANGESEP. The CAMY-B3LYP functional is defined by three parameters, alpha, beta and the attenuation parameter gamma. The gamma parameter can be modified as for the LC functionals. For CAMY-B3LYP it defaults to 0.34. The alpha and beta parameters can be modified through ALPHA=a and BETA=b in the RANGESEP card. They default to 0.19 and 0.46 respectively.

```
XC
  HYBRID CAMY-B3LYP
  XCFUN
  RANGESEP GAMMA=0.34 ALPHA=0.19 BETA=0.46
END
```

List of the most important functionals, for which one can use range separation:

```
LDA VWN
GGA BLYP
GGA BP86
GGA PBE
HYBRID CAMY-B3LYP
```

Range-separated hybrids with LibXC

One can simply specify a range separated hybrid functional in the *LibXC* (page 89) key, *e.g.*:

```
XC
  LibXC CAM-B3LYP
End
```

See the *LibXC* (page 89) section for a list of available range separated hybrid functionals.

For the HSE03 and HSE06 short range-separated hybrids you can (optionally) specify the switching parameter omega, *e.g.*:

```
XC
  LibXC HSE06 omega=0.1
End
```

4.2.8 Notes on Hartree-Fock and (meta-)hybrid functionals

If a functional contains a part of Hartree-Fock exchange then the LDA, GGA, metaGGA, or MODEL key should not be used in combination with this key, and one should only specify one of HartreeFock, HYBRID or MetaHYBRID. Dispersion can be added. Note that it is not recommended to use (part of the) Hartree-Fock exchange in combination with frozen cores, since at the moment the frozen core orbitals are not included in the Hartree Fock exchange operator. In ADF one can do unrestricted Hartree-Fock (or hybrid or meta-hybrid) calculations, as long as one has integer occupation numbers. The default implementation in ADF for unrestricted Hartree-Fock calculations is UHF. In case of a high spin electron configuration one can do ROHF, see *ROKS for high spin open shell molecules* (page 57). You need to use the same XC-potential in the create run of the atoms, which is done automatically if you use the BASIS key.

Starting from ADF2009.01 the meta-hybrids M06, M06-2X, M06-HF, and TPSSH can be used during the SCF. Also starting from ADF2009.01 Hartree-Fock and the (meta-)hybrid potentials can be used in combination with geometry optimization, TS, IRC, LT, and numerical frequencies; hybrids can be used in calculating *NMR chemical shift* (page 205); PBE0 can be used in calculating *NMR spin-spin coupling* (page 213); Hartree-Fock and (meta-)hybrid can be used in calculating excitation energies, in which the kernel consists of the Hartree-Fock percentage times the Hartree-Fock kernel plus one minus the Hartree-Fock percentage times the ALDA kernel (thus no (meta-)GGA kernel). Hartree-Fock and the (meta-)hybrid potentials still can not or should not be used in combination with analytical frequencies, the (AO)RESPONSE key, EPR/ESR g-tensor, and frozen cores. Starting from ADF2010 it is possible to use Hartree-Fock and hybrids to calculate CD spectra.

In ADF one can do unrestricted Hartree-Fock (or hybrid or meta-hybrid) calculations (UHF, UKS), as long as one has integer occupation numbers, or, in case of a high spin electron configuration, one can do ROHF or ROKS, see *ROKS for high spin open shell molecules* (page 57).

It is possible to change the amount of HF exchange in the input for hybrids (not for meta-hybrids and Hartree-Fock). For many hybrid functionals the sum of the amount of Hartree-Fock exchange and the amount of LDA exchange (or GGA exchange) is one. If that is the case, then if one changes the amount of Hartree-Fock exchange in the input the amount of LDA exchange (or GGA exchange) will also be changed, such that the sum remains one. Example:

```
XC
  Hybrid B3LYP HF=0.25
END
```

In this case the amount of Hartree-Fock for the B3LYP functional will be changed to 25% (instead of 20%), and the amount of LDA exchange to 75% (instead of 80%). The LDA correlation and GGA exchange and correlation part will be left unaltered.

An accuracy issue is relevant for some of the meta-GGA functionals, in particular the M06 functionals. These need high integration accuracy (at least BeckeGrid quality good) for reasonable gradients. For TPSSH moderate integration accuracy for reasonable gradients is sufficient. For heavier elements (Z>36) and if one uses one of the M06 functionals it is also necessary to include the following keyword

```
FragMetaGGAToten
```

Using this key `FRAGMETAGGATOTEN` the difference in the metahybrid or metagga exchange-correlation energies between the molecule and its fragments will be calculated using the molecular integration grid, which is more accurate than the default, but is much more time consuming. Default is to calculate the meta-hybrid or meta-GGA exchange-correlation energies for the fragments in the numerical integration grid of the fragments.

For benchmark calculations one would like to use a large basis set, like the QZ4P basis set. In such cases it is recommended to use a good numerical quality. Thus for accurate hybrid calculations of **small** molecules one could use:

```
basis
  type QZ4P
end
AddDiffuseFit
```

(continues on next page)

(continued from previous page)

```
Dependency bas=1e-4
NumericalQuality good
```

4.2.9 MP2, Double Hybrids, RPA

To calculate treat correlation energies beyond DFT, ADF offers MP2 and random-phase approximation (RPA) based methods. In addition, ADF offers a large number of modern Double hybrid functionals which combine MP2 correlation with a hybrid functional.

ADF implements canonical MP2 using density fitting. Additionally, ADF implements RPA and direct MP2 using an efficient atomic orbital based algorithm. The algorithm is described in this paper.[89] The algorithm is continuously improved over the last years and currently allows to perform single-point calculations for systems with up to 1000 atoms one a single modern compute node. Many of the most accurate Double Hybrid functionals only use direct MP2.

Double Hybrid Functionals

```
XC
  DOUBLEHYBRID functional
End
```

DOUBLEHYBRID Specifies that a double-hybrid functional[68] should be used.

See also:

The paper Double hybrid DFT calculations with Slater type orbitals (https://doi.org/10.1002/jcc.26209) and the page Double hybrids: recommendations for accurate thermochemistry and kinetics (https://www.scm.com/highlights/double-hybrids-recommendations-for-accurate-thermochemistry-kinetics/) contain useful recommendations for Double Hybrid calculations in ADF.

Double hybrids usually yield considerably better energies than (meta-)GGA and (meta-)hybrid functionals for (main group) thermochemistry and kinetics, transition metal chemistry and non-covalent interactions. For an overview of the capabilities of double-hybrids implemented in ADF we refer to a recent review.[69]

The MP2 correlation energy consists of two terms,

$$
\begin{aligned}
E_{MP2} &= \frac{1}{2}(E_{\text{direct}} - E_{\text{ex}}) \\
E_{\text{direct}} &= \sum_{ij}\sum_{ab} \frac{(ia|jb)(ia|jb)}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \\
E_{\text{ex}} &= \sum_{ij}\sum_{ab} \frac{(ia|jb)(ib|ja)}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \\
(ia|jb) &= \iint \phi_i^\dagger(1)\phi_a(1)\frac{1}{r_{12}}\phi_j^\dagger(2)\phi_b(2)d1d2
\end{aligned}
$$

---

[89] A. Förster, M. Franchini, E. van Lenthe, L. Visscher, *A Quadratic Pair Atomic Resolution of the Identity Based SOS-AO-MP2 Algorithm Using Slater Type Orbitals*, Journal of Chemical Theory and Computation 16, 875 (2020) (https://doi.org/10.1021/acs.jctc.9b00854)

[68] S. Grimme, *Semiempirical hybrid density functional with perturbative second-order* J. Chem. Phys. 2006, 124, 034108 (https://doi.org/10.1063/1.2148954)

[69] A. Förster, L. Visscher, *Double hybrid DFT calculations with Slater type orbitals*, Journal of Computational Chemistry 41 1660–1684 (2020) (https://doi.org/10.1002/jcc.26209)

For a closed-shell system, the MP2 correlation energy can also be partitioned as

$$\begin{aligned} E_{MP2} &= E_{\text{OS}} + E_{\text{SS}} \\ E_{\text{OS}} &= E_{\text{direct}} \\ E_{\text{SS}} &= E_{\text{direct}} - E_{\text{ex}} \end{aligned}$$

Here, OS (opposite spin) denotes the contribution to the correlation energy from electrons with unpaired spins and SS (same spin) denotes the contribution to the correlation energy from electron with paired spins. In case of spin-orbit coupling *approximate SS and OS contributions* (page 87) are calculated.

There are three classes of Double hybrid functionals:

- Opposite-spin only functionals (Only the OS term is used and scaled by an empirical factor)
- Spin-component scaled functionals (Both, the OS and SS components are individually scaled by empirical factors)
- Standard Double Hybrid functionals (Both, the OS and SS components are scaled by the same empirical factor)

We recommend to use opposite-spin only functionals for large systems (50-100 atoms and larger) since they are computationally more efficient than the other functionals with also include the same-spin contribution. **An opposite-spin only functional calculation is always feasible when a hybrid calculation is feasible too!**

For additional technical details of the algorithm and how to tweak the technical parameters, see *the MBPT section* (page 331).

Opposite-spin-only Double Hybrids

Currently, ADF supports the following opposite-spin-only Double Hybrid functionals:

- **SOS1-PBE-QIDH**: 1-parameter functional with PBE exchange, PBE correlation (69 % HF, 44 % OS-MP2[70]
- **rev-DOD-BLYP**: B88 exchange, LYP correlation, Grimme3 dispersion (71 % HF, 62.2 % OS-MP2)[71]
- **rev-DOD-BLYP-D4**: B88 exchange, LYP correlation, Grimme4 dispersion (71 % HF, 63.5 % OS-MP2)[?]
- **rev-DOD-PBE**: PBE exchange, PBE correlation, Grimme3 dispersion (68 % HF, 61.3 % OS-MP2)[?]
- **rev-DOD-PBE-D4**: PBE exchange, PBE correlation, Grimme4 dispersion (68 % HF, 61.8 % OS-MP2)[?]
- **rev-DOD-PBEP86**: PBE exchange, P86 correlation, Grimme3 dispersion (69 % HF, 60.6 % OS-MP2)[?]
- **rev-DOD-PBEP86-D4**: PBE exchange, P86 correlation, Grimme4 dispersion (69 % HF, 61.2 % OS-MP2)[?]
- **DOD-SCAN**: SCAN exchange and correlation, Grimme3 dispersion (66 % HF, 63,0 % OS-MP2)[?]
- **rev-DOD-SCAN-D4**: SCAN exchange and correlation, Grimme4 dispersion (66 % HF, 63.4 % OS-MP2)[?]

Except for **SOS1-PBE-QIDH**, all functionals include dispersion correction by default which cannot be switched off.

Standard Double Hybrids

Currently, ADF supports the following standard double hybrid functionals

- **B2PLYP**: B88 exchange and LYP correlation (53 % HF, 27 % MP2)[?]
- **B2PIPLYP**: B88 exchange and LYP correlation (60 % HF, 27 % MP2), parametrized for pi-pi interactions[72]
- **ROB2PLYP**: B88 exchange and LYP correlation (59 % HF, 28 % MP2), restricted-open-shell version of B2PLYP[73]

---

[70] Js. Garcia, E. Bremond, M. Savarese, AJ. Perez-Gimenez, C. Adamo, *Partnering dispersion corrections with modern parameter-free double-hybrid density functionals* Phys. Chem. Chem. Phys., 19, 13481 (2017) (https://doi.org/10.1021/acs.jctc.8b01203)

[71] G. Santra, N. Sylvetsky, and J.M.L. Martin, *Minimally Empirical Double-Hybrid Functionals Trained against the GMTKN55 Database: revDSD-PBEP86-D4 revDOD-PBE-D4, and DOD-SCAN-D4* J. Chem. Phys. 2019, 123, 5129 (https://doi.org/10.1021/acs.jpca.9b03157)

[72] J.C. Sancho-García, A.J. Pérez-Jiménez, *Assessment of double-hybrid energy functionals for PI-conjugated systems* J. Chem. Phys. 2009, 131, 084108 (https://doi.org/10.1063/1.3212881)

[73] D.C. Graham, A.S. Menon, L. Goerigk, S. Grimme, L. Radom *Optimization and basis-set dependence of a restricted-open-shell form of B2-PLYP double-hybrid density functional theory* J. Phys. Chem. A 2009, 113, 9861 (https://doi.org/10.1021/jp9042864)

- **B2TPLYP**: B88 exchange and LYP correlation (60 % HF, 31 % MP2), parametrized for thermodynamics[74]
- **B2GPPLYP**: B88 exchange and LYP correlation (65 % HF, 36 % MP2), 'General Purpose' parametrization[75]
- **B2KPLYP**: B88 exchange and LYP correlation (72 % HF, 42 % MP2), parametrized for kinetics[?]
- **B2NCPLYP**: B88 exchange and LYP correlation (70 % HF, 49 % MP2), parametrized for non-covalent interactions[76]
- **mPW2PLYP**: mPW exchange and LYP correlation (55 % HF, 25 % MP2)[77]
- **mPW2KPLYP**: mPW exchange and LYP correlation (72 % HF, 42 % MP2)[?]
- **mPW2NCPLYP**: mPW exchange and LYP correlation (42 % HF, 49 % MP2)[?]
- **DH-BLYP**: 1-Parameter functional with B88 exchange and LYP correlation (65 % HF, 42 % MP2)[78]
- **PBE0-DH**: 1-Parameter functional with PBE exchange and PBE correlation (50 % HF, 13 % MP2)[79]
- **PBE-QIDH**: 1-parameter functional with PBE exchange and PBE correlation (69 % HF, 33 % MP2)[?]
- **LS1-DH**: 1-Parameter functional with PBE exchange and PBE correlation (75 % HF, 42 % MP2)[80]
- **PBE0-2**: 1-Parameter functional with PBE exchange and PBE correlation (79 % HF, 50 % MP2)[81]
- **LS1-TPSS**: 1-Parameter functional with TPSS meta-GGA (85 % HF, 61 % MP2)[82]
- **DS1-TPSS**: 1-Parameter functional with TPSS meta-GGA (73 % HF, 53 % MP2)[?]

Empirical dispersion corrections can be requested in the XC block in the usual way. Some functionals can be combined with Grimme's D4 empirical dispersion correction with optimized parameters: B2PLYP, B2GPPLYP, mPW2PLYP, PBE0-DH, PBE0-2. All functionals in this category can be combined with Grimme's D3(BJ) empirical dispersion correction with optimized parameters. Note that for Grimme's D3(BJ) the parameters for B2PIPLYP, ROB2PLYP, B2TPLYP, B2KPLYP, mPW2PLYP, mPW2KPLYP, DH-BLYP, LS1-DH, PBE0-2, and DS1-TPSS are modified B2PLYP parameters, see Ref.[?]. If no optimized empirical dispersion parameters exist for a certain functional, default parameters are used, which may not give the expected results.

Spin-component-scaled functionals

Currently, ADF supports the following spin-component scaled Double Hybrid functionals:

- **DSD-BLYP**: B88 exchange and LYP correlation, Grimme3 dispersion (69 % HF, 46 % OS-MP2, % 37 % SS-MP2)[83]
- **rev-DSD-BLYP**: revised version of DSD-BLYP, Grimme3 dispersion (71 % HF, 54.7 % os, 19.8 % SS-MP2)[?]

---

[74] A. Tarnopolsky, A. Karton, R. Sertchook, D. Vuzman, J.M.L Martin, *Double-hybrid functionals for thermochemical kinetics* J. Phys. Chem. A 2008, 112, 3 (https://doi.org/10.1021/jp710179r)

[75] A. Karton, A. Tarnopolsky, J.F. Lamere, G.C. Schatz, J.M.L, Martin, *Highly accurate first-principles bench- mark data sets for the parametrization and validation of density functional and other approximate methods. Derivation of a robust, generally applicable, double-hybrid functional or thermochemistry and thermochemical kinetics.* J. Phys. Chem. A 2008, 112, 12868 (https://doi.org/10.1021/jp801805p)

[76] Y. Feng, *Double-Hybrid Density Functionals Free of Dispersion and Counterpoise Corrections for Non-Covalent Interactions* J. Phys. Chem. A 2014, 118, 3175 (https://doi.org/10.1021/jp5005506)

[77] T. Schwabe, S. Grimme, *Towards chemical accuracy for the thermodynamics of large molecules: new hybrid density functionals including non-local correlation effects.* Phys. Chem. Chem. Phys. 2006, 8, 4398 (https://doi.org/10.1039/B608478H)

[78] K. Sharkas, J. Toulouse, A. Savin, *Double-hybrid density-functional theory made rigorous* J. Chem. Phys. 2011, 134, 064113 (https://doi.org/10.1063/1.3544215)

[79] E. Bremond, C. Adamo, *Seeking for parameter-free double-hybrid functionals: The PBE0-DH model* J. Chem. Phys. 2011, 135, 024106 (https://doi.org/10.1063/1.3604569)

[80] J. Toulouse, K. Sharkas, E. Bremond, C. Adamo, *Rationale for a new class of double-hybrid approximations* J. Chem. Phys. 2011, 135, 101102 (https://doi.org/10.1063/1.3640019)

[81] J.D. Chai, S.P. Mao, *Seeking for reliable double-hybrid density functionals without fitting parameters: the PBE0-2 functional* Chem. Phys. Lett. 2012, 538, 121 (https://doi.org/10.1016/j.cplett.2012.04.045)

[82] S.O. Souvi, K. Sharkas and J. Toulouse, *Double-hybrid density functional theory with meta-generalized gradient approximations* J. Chem. Phys. 2014, 140, 084107 (https://doi.org/10.1016/j.cplett.2012.04.045)

[83] S. Kozuch, D. Gruzman, J.M.L. Martin, *DSD-BLYP: a general purpose double hybrid density functional including spin component scaling and dispersion correction* J. Phys. Chem. C 2010, 114, 20801 (https://doi.org/10.1021/jp1070852)

- **rev-DSD-BLYP-D4**: revised version of DSD-BLYP, Grimme4 dispersion (71 % HF, 55.9 % OS-, 19.7 % SS-MP2)[?]
- **DSD-PBEP86**: PBE exchange and P86 correlation (69 % HF, 52 % OS-, % 22 % SS-MP2)[84]
- **rev-DSD-PBEP86**: revised version of DSD-PBEP86 (69 % HF, 57.9 % OS-, 8 % SS-MP2)[?]
- **rev-DSD-PBEP86-D4**: PBE exchange and P86 correlation, Grimme4 dispersion (69 % HF, 59.2 % OS-, 6.4 % SS-MP2)[?]
- **DSD-PBE**: PBE exchange and PBE correlation, Grimme3 dispersion (68 % HF, 55 % OS-, % 13 % SS-MP2)[?]
- **rev-DSD-PBE**: revised version of DSD-PBE, Grimme3 dispersion (68 % HF, 58.5 % os, 7 % SS-MP2)[?]
- **rev-DSD-PBE-D4**: revised version of DSD-PBE, Grimme4 dispersion (68 % HF, 60 % os, 4.2 % SS-MP2)[?]
- **rev-DSD-SCAN-D4**: based on SCAN meta-GGA (66 % HF, 63.2 % OS-, 1.3 % SS-MP2)[?]
- **SD-SCAN69**: Based on SCAN meta-GGA, no dispersion correction (69 % HF, 62 % OS-, 26 % SS-MP2)[?]

Except for **SD-SCAN69**, all functionals include dispersion correction by default which cannot be switched off.

MP2

```
XC
  MP2
  EmpiricalScaling {NONE|SOS|SCS|SCSMI}
END
```

In addition to double-hybrids, ADF also implements MP2 including some popular spin-scaled variants. Technically, they are not distinct from double-hybrids, however, the all rely on a HF instead of a DFT calculation. The following variants are supported.

- **SOS-MP2**: pure HF reference (100 % HF, 130 % OS-MP2)[85]
- **MP2**: pure HF reference (100 % HF, 100 % MP2 correlation)
- **SCS-MP2**: pure HF reference (100 % HF, 120 % OS-MP2, 33 % SS-MP2)[86]
- **SOS-MI-MP2**: pure HF reference (100 % HF, 40 % OS-MP2, 129 % SS-MP2)[87]

In case of spin-orbit coupling *approximate SS and OS contributions* (page 87) are calculated.

The spin-scaling variant can be requested in the XC block together with the `MP2` keyword:

```
XC
  MP2
  EmpiricalScaling SOS
END
```

requests an SOS-MP2 calculation.

For additional technical details of the algorithm and how to tweak parameters, see *the MBPT section* (page 331).

---

[84] S. Kozuch, J.M.L. Martin, *DSD-PBEP86: In search of the best double-hybrid DFT with spin-component scaled MP2 and dispersion corrections* Phys. Chem. Chem. Phys. 2011, 13, 20104 (https://doi.org/10.1039/C1CP22592H)

[85] Y. Jung, R.C. Lochan, A. D. Dutoi, M. Head-Gordon, *Scaled opposite-spin second order Møller-plesset correlation energy: An economical electronic structure method.* Journal of Chemical Physics 2004, 121, 9793 (https://doi.org/10.1063/1.1809602)

[86] S. Grimme, *Improved second-order Møller-Plesset perturbation theory by separate scaling of parallel- and antiparallel-spin pair correlation energies.* Journal of Chemical Physics 2003, 118, 9095 (https://doi.org/10.1063/1.1569242)

[87] R.C. Lochan, Y. Jung, M. Head-Gordon, *Scaled opposite spin second order Møller-Plesset theory with improved physical description of long-range dispersion interactions* Journal of Physical Chemistry A2005,109, 7598 (https://doi.org/10.1021/jp0514426)

RPA

Note: In AMS2022, the keyword for RPA+SOX was RPASOX.

The RPA goes beyond MP2 by accounting explicitly for the polarizability of the system which screens the electron-electron interaction. It can therefore be applied to large system for which MP2 typically diverges.[90] The following RPA based methods are available.

- **RPA** : Standard (direct)RPA without exchange
- **RPA + SOX** : Standard RPA plus statically screened second-order exchange[91]
- **RPA + SOSEX**: Standard RPA plus dynamically screened second-order exchange[92]

A detailed overview of the RPA algorithm in ADF and a detailed assessment of the performance of second-order exchange corrections can be found in[94].

An RPA calculation is requested in the XC block:

```
XC
  RPA {NONE|DIRECT|SOSEX|SOSSX|SIGMA}
End
```

An RPA calculation needs to be combined with an XC functional. For instance,

```
XC
  hybrid pbe0
  RPA DIRECT
End
```

will perform a PBE0 calculation followed by a direct RPA calculation. RPA and all of its variants can be used in conjunction with LDA, GGAs, hybrid, and RSH functionals.

For additional technical details of the algorithm and how to tweak parameters, see *the RPA section* (page 331).

sigma-functional

Starting from AMS2023, the sigma-functional by Görling and coworkers is implemented.[93] In this method, the correlation kernel is calculated form the adiabatic fluctuation-dissipation theorem. In addition to the direct RPA (Hartree) kernel, higher-order contributions to the kernel are included by the so-called sigma-kernel which is fitted to relative energies. Sigma-functionals are as fast as RPA.

As an RPA calculation, a sigma-functional calculation needs to be combined with an XC functional. For instance,

```
XC
  hybrid pbe0
  RPA sigma
End
```

---

[90] B. D. Nguyen, G. P. Chen, M. M. Agee, A. M. Burow, M. P. Tang, F. Furche, *Divergence of Many-Body Perturbation Theory for Noncovalent Interactions of Large Molecules*, Journal of Chemical Theory and Computation 16(4), 2258-2273 (2020) (https://doi.org/10.1021/acs.jctc.9b01176)

[91] A. Förster, L. Visscher, *Exploring the statically screened G3W2 correction to the GW self-energy : Charged excitations and total energies of finite systems*, Physical Review B 105, 125121 (2022) (https://journals.aps.org/prb/abstract/10.1103/PhysRevB.105.125121)

[92] X.Ren, P. Rinke, G.E. Scuseria, M. Scheffler, *Renormalized second-order perturbation theory for the electron correlation energy: Concept, implementation, and benchmarks*, Physical Review B 88(3), 035120 (2013) (https://journals.aps.org/prb/abstract/10.1103/PhysRevB.88.035120)

[94] A. Förster, *Assessment of the second-order statically screened exchange correction to the random phase approximation for correlation energies*, Journal of Chemical Theory and Computation 18(10), 5948-5965 (2022) (https://doi.org/10.1021/acs.jctc.2c00366)

[93] E. Trushin, A. Thierbach, A. Görling *Toward chemical accuracy at low computational cost: Density-functional theory with sigma-functionals for the correlation energy*, The Journal of Chemical Physics 154, 014104 (2021) (https://doi.org/10.1063/5.0026849)

requests to use the sigma-functional with the W1 parametrization for PBE0.[95] sigma-functionals can only be used with a limited number of exchange-correlation functionals only, since they need to be explicitly parametrized for each functional. Currently, the sigma-functional can be used in conjunction with the GGA PBE, and the hybrids PBE0 and B3LYP. The available parametrizations for each functional are listed in the following table:

functional	available parametrizations[?][96]
PBE	W1 S1 S2
PBE0	W1 S1 S2 W2
B3LYP	W1

The parametrization can be changed in the MBPT block, see *the MBPT block* (page 331). For instance:

```
MBPT
  SigmaFunctionalParametrization S1
End
```

Spin-orbit coupling

In case of spin-orbit coupling approximate SS and OS contributions are calculated, which is relevant for open shell molecules with double hybrids or MP2 variants that use different scaling factors for these contributions:

$$\begin{aligned}
(ia|jb) &= \iint \phi_i^\dagger(1)\phi_a(1)\frac{1}{r_{12}}\phi_j^\dagger(2)\phi_b(2)d1d2 \\
m_{ij} &= \int \phi_i^\dagger(1)\vec{\sigma}\phi_i(1)d1 \cdot \int \phi_j^\dagger(2)\vec{\sigma}\phi_j(2)d2 \\
E_2^{SS} &= -\sum_{ijab}\frac{(1+m_{ij})(ia|jb)(ai|bj)}{\epsilon_i+\epsilon_j-\epsilon_a-\epsilon_b} + \sum_{ijab}\frac{2(ia|jb)(aj|bi)}{\epsilon_i+\epsilon_j-\epsilon_a-\epsilon_b} \\
E_2^{OS} &= -\sum_{ijab}\frac{(1-m_{ij})(ia|jb)(ai|bj)}{\epsilon_i+\epsilon_j-\epsilon_a-\epsilon_b}
\end{aligned}$$

with, i, j occupied spinors, a, b virtual spinors, ϵ spinor energies, $\vec{\sigma}$ Pauli spin matrices. Note with pure α and β orbitals, $m_{i^\alpha j^\alpha} = m_{i^\beta j^\beta} = 1, m_{i^\alpha j^\beta} = m_{i^\beta j^\alpha} = -1$, one has the familiar SS and OS energy expressions.

4.2.10 Model Potentials

Several asymptotically correct XC potentials have been implemented in ADF, such as the (now somewhat outdated) LB94 potential[?], the gradient-regulated asymptotic correction (GRAC)[39], and the statistical average of orbital potentials (SAOP)[42][40]. These can currently be used only for response property calculations, not for geometry optimizations. For spectroscopic properties, they usually give results superior to those obtained with LDA or GGA potentials, (see Ref.[41]

---

[95] S.Fauser, E. Trushin, C. Neiss, A. Görling *Chemical accuracy with sigma-functionals for the Kohn–Sham correlation energy optimized for different input orbitals and eigenvalues*, The Journal of Chemical Physics 155, 034111 (2021) (https://doi.org/10.1063/5.0059641)

[96] J.Erhard, S. Fauser, E. Trushin, A. Görling *Scaled sigma-functionals for the Kohn–Sham correlation energy with scaling functions from the homogeneous electron gas*, The Journal of Chemical Physics 157, 114105 (2022) (https://doi.org/10.1063/5.0101641)

[39] M. Grüning, O.V. Gritsenko, S.J.A. van Gisbergen and E.J. Baerends, *Shape corrections to exchange-correlation Kohn-Sham potentials by gradient-regulated seamless connection of model potentials for inner and outer region*, Journal of Chemical Physics 114, 652 (2001) (https://doi.org/10.1063/1.1327260)

[42] O.V. Gritsenko, P.R.T. Schipper and E.J. Baerends, *Approximation of the exchange-correlation Kohn-Sham potential with a statistical average of different orbital model potentials*, Chemical Physics Letters 302, 199 (1999) (https://doi.org/10.1016/S0009-2614(99)00128-1)

[40] P.R.T. Schipper, O.V. Gritsenko, S.J.A. van Gisbergen and E.J. Baerends, *Molecular calculations of excitation energies and (hyper)polarizabilities with a statistical average of orbital model exchange-correlation potentials*, Journal of Chemical Physics 112, 1344 (2000) (https://doi.org/10.1063/1.480688)

[41] M. Grüning, O.V. Gritsenko, S.J.A. van Gisbergen and E.J. Baerends, *On the required shape correction to the LDA and GGA Kohn Sham potentials for molecular response calculations of (hyper)polarizabilities and excitation energies*, Journal of Chemical Physics 116, 9591 (2002) (https://doi.org/10.1063/1.1476007)

for applications to (hyper)polarizabilities Cauchy coefficients, etc. of small molecules). This is particularly true if the molecule is small and the (high-lying) virtual orbitals are important for the property under study.

It was also shown that, simply using the orbital energies of the occupied Kohn-Sham orbitals of a SAOP calculation, quite good agreement with experiment vertical ionization potentials is obtained. This is true not only for the HOMO orbital energy, which should be identical to (minus) the experimental ionization potential with the exact XC potential, but also for lower-lying occupied orbital energies. The agreement becomes worse for deep-lying core orbital energies. A theoretical explanation and practical results are given in Ref.[43].

```
XC
  Model ModelPotential [IP]
End
```

MODEL Specifies that one of the less common XC potentials should be used during the SCF. These potentials specify both the exchange and the correlation part. No LDA, GGA, MetaGGA, HartreeFock, HYBRID or MetaHYBRID key should be used in combination with these keys. It is also not advised to use any energy analysis in combination with these potentials. For energy analysis we recommend to use one of the GGA potentials. It is currently not possible to do a Create run with these potentials. It is possible to do a one atom regular ADF calculation with these potentials though, using a regular adf.rkf (TAPE21) file from an LDA or GGA potential as input. Available model potentials:

- **LB94**: This refers to the XC functional of Van Leeuwen and Baerends[?]. There are no separate entries for the Exchange and Correlation parts respectively of LB94. Usually the GRAC or SAOP potentials give results superior to LB94.
- **GRAC**: The gradient-regulated asymptotic correction, which in the outer region closely resembles the LB94 potential[?]. It requires a further argument: the ionization potential [IP] of the molecule, in hartree units. This should be estimated or obtained externally, or calculated in advance from two GGA total energy calculations.
- `IP`:Should be supplied only if GRAC is specified.
- **SAOP**: The statistical average of orbital potentials[??]. It can be used for all electron calculations only. It will be expensive for large molecules, but requires no further parameter input.

The LB94, GRAC, and SAOP functionals have only a SCF (=Potential) implementation, but **no Energy** counterpart.

The LB94, GRAC, and SAOP forms are density functionals specifically designed to get the correct asymptotic behavior. This yields much better energies for the highest occupied molecular orbital (HOMO) and better excitation energies in a calculation of response properties (Time Dependent DFT). Energies for lower lying orbitals (sub-valence) should improve as well (in case of GRAC and SAOP, but not LB94). The energy expression underlying the LB94 functional is very inaccurate. This does not affect the response properties but it does imply that the energy and its derivatives (gradients) should not be used because LB94-optimized geometries will be wrong, see for instance[44]. The application of the LB94 functional in a runtype that involves the computation of energy gradients is disabled in ADF. You can override this internal check with the key *ALLOW* (page 337).

In case of a GRAC calculation, the user should be aware that the potential in the outer region is shifted up with respect to the usual level. In other words, the XC potential does not tend to zero in the outer region in this case. The size of the shift is the difference between the HOMO orbital energy and the IP given as input. In order to compare to regular GGA orbital energies, it is advisable to subtract this amount from all orbital energies. Of course, orbital energy differences, which enter excitation energies, are not affected by this shift in the potential.

---

[43] D.P. Chong, O.V. Gritsenko and E.J. Baerends, *Interpretation of the Kohn-Sham orbital energies as approximate vertical ionization potentials*, Journal of Chemical Physics 116, 1760 (2002) (https://doi.org/10.1063/1.1430255)

[44] R. Neumann, R.H. Nobes and N.C. Handy, *Exchange functionals and potentials*, Molecular Physics 87, 1 (1996) (https://doi.org/10.1080/00268979600100011)

4.2.11 XCFun

XCFUN XCFun is a library of approximate exchange-correlation functionals, see Ref.[45], for which functional derivatives can be calculated automatically. For example, with XCFUN the full (non-ALDA) kernel can be evaluated and this has been implemented in the calculation of TDDFT excitations. The Full kernel can not be used in combination with symmetry or excited state geometry optimizations. The following functionals can be evaluated with XCFUN at the present time:

- LDA: VWN5, X-ALPHA, PW92
- GGA exchange: Becke88, PBEX, OPTX, PW91X, mPW, revPBEX
- GGA correlation: LYP, Perdew86, PBEC
- MetaGGA: TPSS, M06L, B95
- MetaHybrids: M06, M05, M062X, M06HF
- Hybrids: PBE0, B3LYP, BHandH, B1LYP, B3LYP*, PBEFALFX
- Yukawa range separated Hybrids: CAMY-B3LYP and more, see *Yukawa RS hybrids with XCFUN* (page 79)

Here MetaGGA B95 means Becke88 exchange + B95c correlation. The Metahybrids PW6B95 and PWB6K have been removed from this list, since they do not agree with the LibXC implementation.

4.2.12 LibXC

LibXC functional LibXC is a library of approximate exchange-correlation functionals, see Ref.[46][47]. **All electron basis sets should be used** (see *Basis key* (page 42)). Version 5.1.2 of LibXC is used. The following functionals can be evaluated with LibXC (incomplete list):

- LDA: LDA, PW92, TETER93
- GGA: AM05, BCGP, B97-GGA1, B97-K, BLYP, BP86, EDF1, GAM, HCTH-93, HCTH-120, HCTH-147, HCTH-407, HCTH-407P, HCTH-P14, PBEINT, HTBS, KT2, MOHLYP, MOHLYP2, MPBE, MPW, N12, OLYP, PBE, PBEINT, PBESOL, PW91, Q2D, SOGGA, SOGGA11, TH-FL, TH-FC, TH-FCFO, TH-FCO, TH1, TH2, TH3, TH4, XLYP, XPBE, HLE16
- MetaGGA: M06-L, M11-L, MN12-L, MS0, MS1, MS2, MVS, PKZB, RSCAN, R2SCAN, REVSCAN, SCAN, TPSS, HLE17
- Hybrids: B1LYP, B1PW91, B1WC, B3LYP, B3LYP*, B3LYP5, B3LYP5, B3P86, B3PW91, B97, B97-1 B97-2, B97-3, BHANDH, BHANDHLYP, EDF2, MB3LYP-RC04, MPW1K, MPW1PW, MPW3LYP, MPW3PW, MPWLYP1M, O3LYP, OPBE, PBE0, PBE0-13, REVB3LYP, REVPBE, RPBE, SB98-1A, SB98-1B, SB98-1C, SB98-2A, SB98-2B, SB98-2C, SOGGA11-X, SSB, SSB-D, X3LYP
- MetaHybrids: B86B95, B88B95, BB1K, M05, M05-2X, M06, M06-2X, M06-HF, M08-HX, M08-SO, MPW1B95, MPWB1K, MS2H, MVSH, PW6B95, PW86B95, PWB6K, REVSCAN0, SCAN0, REVTPSSH, TPSSH, X1B95, XB1K
- Range-separated: CAM-B3LYP, CAMY-B3LYP, HJS-PBE, HJS-PBESOL, HJS-B97X, HSE03, HSE06, LRC_WPBE, LRC_WPBEH, LCY-BLYP, LCY-PBE, M06-SX, M11, MN12-SX, N12-SX, TUNED-CAM-B3LYP, WB97, WB97X

---

[45] U. Ekström, L. Visscher, R. Bast, A.J. Thorvaldsen, and K. Ruud, *Arbitrary-Order Density Functional Response Theory from Automatic Differentiation*, Journal of Chemical Theory and Computation 6, 1971 (2010) (https://doi.org/10.1021/ct100117s)

[46] M.A.L. Marques, M.J.T. Oliveira, and T. Burnus, *Libxc: a library of exchange and correlation functionals for density functional theory*, Computer Physics Communications 183, 2272 (2012) (https://doi.org/10.1016/j.cpc.2012.05.007)

[47] S. Lehtola, C. Steigemann, M.J.T. Oliveira, M.A.L. Marques, *Recent developments in LibXC – A comprehensive library of functionals for density functional theory*, SoftwareX 7, 1 (2018) (https://doi.org/10.1016/j.softx.2017.11.002)

One of the acronyms in the list above can be used, or one can also use the functionals described at the LibXC website http://www.tddft.org/programs/Libxc. Note that ADF can not calculate VV10 dependent LibXC functionals, like VV10, LC-VV10, B97M-V, WB97X-V. Example usage for the BP86 functional:

```
XC
  LibXC BP86
End
```

Alternative

```
XC
  LibXC XC_GGA_X_B88 XC_GGA_C_P86
End
```

In case of LibXC the output of the ADF calculation will give the reference for the used functional, see also the LibXC website http://www.tddft.org/programs/Libxc.

Do not use any of the subkeys LDA, GGA, METAGGA, MODEL, HARTREEFOCK, HYBRID, METAHYBRID, XCFUN, RANGESEP in combination with the subkey LIBXC. One can use the DISPERSION key with LIBXC. For a selected number of functionals the optimized dispersion parameters will then be used automatically, please check the output in that case. Note that in many cases you have to include the DISPERSION key and include the correct dispersion parameters yourself.

The LibXC functionals can not be used with frozen cores, NMR calculations, the (AO)RESPONSE key, EPR/ESR g-tensor. Most LibXC functionals can be used in combination with geometry optimization, TS, IRC, LT, numerical frequencies, and excitation energies (ALDA kernel used). For a few GGA LibXC functionals analytical frequencies can be calculated, and one can use the full kernel in the calculation of excitation energies (if FULLKERNEL is included as subkey of the key EXCITATIONS). In case of LibXC (meta-)hybrids and calculating excitation energies, the kernel consists of the Hartree-Fock percentage times the Hartree-Fock kernel plus one minus the Hartree-Fock percentage times the ALDA kernel (thus no (meta-)GGA kernel). For the LibXC range separated functionals, like CAM-B3LYP, starting from ADF2016.102 the kernel consists of range separated ALDA plus the kernel of the range separated exact exchange part. In ADF2016.101 the kernel for LibXC range separated functionals, like CAM-B3LYP, was using a 100% ALDA plus range separated exact exchange kernel (the ALDA part was not range-separated corrected). For the range separated functionals WB97 and WB97X one can use the full kernel in the calculation of excitation energies.

4.2.13 Dispersion corrections

`Dispersion Grimme4 {s6=...} {s8=...} {a1=...} {a2=...}` If `Dispersion Grimme4` is present in the `XC` block the D4(EEQ) dispersion correction (with the electronegativity equilibrium model) by the Grimme group[48] will be added to the total bonding energy, gradient and second derivatives, where applicable.

The D4(EEQ) model has four parameters: s_6, s_8, a_1 and a_2 and their value should depend on the XC functional used. For the following functionals the D4(EEQ) parameters are **predefined**: B1B95, B3LYP, B3PW91, BLYP, BP86, CAM-B3LYP, HartreeFock, OLYP, OPBE, PBE, PBE0, PW6B95, REVPBE, RPBE, TPSS, TPSSH. For these functionals it is enough to specify `Dispersion Grimme4` in the input block. E.g.:

```
XC
  GGA BLYP
  Dispersion Grimme4
END
```

For all other functionals you should explicitly specify the D4(EEQ) parameters in the `Dispersion` key (otherwise the PBE parameters will be used). For example, for the PW91 functional you should use the following input:

---

[48] E. Caldeweyher, S. Ehlert, A. Hansen, H. Neugebauer, S. Spicher, C. Bannwarth, S. Grimme, *A Generally Applicable Atomic-Charge Dependent London Dispersion Correction Scheme*, ChemRxiv 7430216 v2 (https://doi.org/10.26434/chemrxiv.7430216)

```
XC
  GGA PW91
  Dispersion Grimme4 s6=1.0 s8=0.7728 a1=0.3958 a2=4.9341
END
```

The D4(EEQ) parameters for many functionals can be found in the supporting information of the following paper:[?].

For Double-Hybrids, see the *Double Hybrid Functionals* (page 82) section of the user manual.

DISPERSION Grimme3 BJDAMP If DISPERSION Grimme3 BJDAMP is present a dispersion correction (DFT-D3(BJ)) by Grimme[49] will be added to the total bonding energy, gradient and second derivatives, where applicable. Parametrizations are implemented e.g. for B3LYP, TPSS, BP86, BLYP, PBE, PBEsol, and RPBE. For SCAN parameters from Ref.[50] are used.

For example, this is the input block for specifying the PBE functional with Grimme3 BJDAMP dispersion correction (PBE-D3(BJ)):

```
XC
  GGA PBE
  DISPERSION Grimme3 BJDAMP
End
```

The D3(BJ) dispersion correction has four parameters. One can override the default parametrization by using *PAR1=.. PAR2=..*, etc. In the table the relation is shown between the parameters and the real parameters in the Dispersion correction.

variable	variable on Bonn website (https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3/functionalsbj)
PAR1	s6
PAR2	a1
PAR3	s8
PAR4	a2

For example, this is the input block for specifying the PBE-D3(BJ)-GP parametrization by Proppe et.al.[88] (i.e. $a_1 = 0, s_8 = 0, a_2 = 5.6841$):

```
XC
  GGA PBE
  DISPERSION Grimme3 BJDAMP PAR2=0 PAR3=0 PAR4=5.6841
End
```

DISPERSION Grimme3 If DISPERSION Grimme3 is present a dispersion correction (DFT-D3) by Grimme[51] will be added to the total bonding energy, gradient and second derivatives, where applicable. Parametrizations are available e.g. for B3LYP, TPSS, BP86, BLYP, revPBE, PBE, PBEsol, and RPBE, and will be automatically set if one of these functionals is used. There are also parameters directly recognized for S12g and S12h. For SCAN parameters from Ref.[?] are used. For all other functionals, PBE-D3 parameters are used as default. You can explicitly specify the three parameters.

---

[49] S. Grimme, S. Ehrlich, and L. Goerigk, *Effect of the Damping Function in Dispersion Corrected Density Functional Theory*, Journal of Computational Chemistry 32, 1457 (2011) (https://doi.org/10.1002/jcc.21759).

[50] J.G. Brandenburg, J.E. Bates, J. Sun, and J.P. Perdew, *Benchmark tests of a strongly constrained semilocal functional with a long-range dispersion correction*, Physical Review B 94, 115144 (2016) (https://doi.org/10.1103/PhysRevB.94.115144).

[88] J. Proppe, S. Guglerb and M. Reiher, *Gaussian Process-Based Refinement of Dispersion Corrections* https://arxiv.org/pdf/1906.09342.pdf

[51] S. Grimme, J. Anthony, S. Ehrlich, and H. Krieg, *A consistent and accurate* ab initio *parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu*, Journal of Chemical Physics 132, 154104 (2010) (https://doi.org/10.1063/1.3382344).

variable	variable on Bonn website (https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3/functionals)
PAR1	s6
PAR2	sr,6
PAR3	s8

`DISPERSION {s6scaling]} {RSCALE=r0scaling}` If the DISPERSION keyword is present (without the argument Grimme3) a dispersion correction (DFT-D) by Grimme[?] will be added to the total bonding energy, gradient and second derivatives, where applicable. The global scaling factor with which the correction is added depends on the exchange-correlation functional used at SCF but it can be modified using the *s6scaling* parameter. The following scaling factors are used (with the XC functional in parentheses): 1.20 (BLYP), 1.05 (BP), 0.75 (PBE), 1.05 (B3LYP). In all other cases a factor 1.0 is used unless modified via the s6scaling parameter. The SSB-D functional includes the dispersion correction (factor 0.847455) by default.

The van der Waals radii used in this implementation are hard coded in ADF. However, it is possible to modify the global scaling parameter for them using the *RSCALE=r0scaling* argument. The default value is 1.1 as proposed by Grimme[?]. Please also see *additional documentation* (page 90) for more information about this topic.

`DISPERSION dDsC` The DISPERSION dDsC key invokes the density dependent dispersion correction[54], which has been parametrized for the functionals BLYP, PBE, BP, revPBE, B3LYP, PBE0 and BHANDHLYP.

`DISPERSION UFF` The DISPERSION UFF key invokes the universal correction of density functional theory to include London dispersion (DFT-ulg)[52], which has been parametrized for all elements up to Lr (Z=103), and for the functional PBE, PW91, and B3LYP. For other functionals the PBE parameters will be used.

`DISPERSION MBD` The DISPERSION MBD key invokes the MBD@rsSCS method[53], which is designed to accurately describe long-range correlation (and thus dispersion) in finite-gap systems, including at the same time a description of the short-range interactions from the underlying DFT computation of the electronic structure.

DFT-D4 functionals

Grimme's latest dispersion correction, D4(EEQ)[?], has been added in the 2019.3 release of the Amsterdam Modeling Suite. This is the latest dispersion correction in the DFT-D family. In contrast to the earlier D3 dispersion correction, in D4(EEQ) the atomic coordination-dependent dipole polarizabilities are scaled based on atomic partial charges obtained from an electronegativity equilibrium model (EEQ). Compared to D3 the introduced charge dependence improves thermochemical properties, especially for systems containing metals. The authors recommend D4(EEQ) as a physically improved and more sophisticated dispersion model in place of D3.

DFT-D3 functionals

The D3 dispersion correction by Stefan Grimme is available in ADF. Grimme and his coworkers at the Universität Münster outlined the parametrization of this new correction, dubbed DFT-D3, in Ref.[?]. A slightly improved version with a more moderate BJ damping function appeared later, and was called DFTB-D3-BJ.[?] Here they list the advantages of the new method as the following:

- It is less empirical, i.e., the most important parameters are computed from first principles by standard Kohn-Sham (KS)-(TD)DFT.

---

[54] S.N. Steinmann, and C. Corminboeuf, *Comprehensive Benchmarking of a Density-Dependent Dispersion Correction*, Journal of Chemical Theory and Computation 7, 3567 (2011) (https://doi.org/10.1021/ct200602x).

[52] H. Kim, J.-M. Choi, W.A. Goddard, *Universal Correction of Density Functional Theory to Include London Dispersion (up to Lr, Element 103)*, Journal of Physical Chemistry Letters 3, 360 (2012) (https://doi.org/10.1021/jz2016395)

[53] A. Ambrosetti, A.M. Reilly, Robert A. DiStasio Jr., A. Tkatchenko, *Long-range correlation energy calculated from coupled atomic response functions*, Journal of Chemical Physics 140, 18A508 (2014) (https://doi.org/10.1063/1.4865104)

- The approach is asymptotically correct with all DFs for finite systems (molecules) or nonmetallic infinite systems. It gives the almost exact dispersion energy for a gas of weakly interacting neutral atoms and smoothly interpolates to molecular (bulk) regions.
- It provides a consistent description of all chemically relevant elements of the periodic system (nuclear charge Z = 1-94).
- Atom pair-specific dispersion coefficients and cutoff radii are explicitly computed.
- Coordination number (geometry) dependent dispersion coefficients are used that do not rely on atom connectivity information (differentiable energy expression).
- It provides similar or better accuracy for "light" molecules and a strongly improved description of metallic and "heavier" systems.

DFT-D3-BJ is invoked with the XC block, for example

```
XC
  GGA BLYP
  Dispersion Grimme3 BJDAMP
END
```

Parametrizations are available for: B3LYP, TPSS, BP86, BLYP, revPBE, PBE, PBEsol, RPBE, and some more functionals, and will be automatically set if one of these functionals is used. Otherwise PBE parameters will be used. The parameters can be set manually, see the XC key block. In ADF2016 parameters for Grimme3 and Grimme3 BJDAMP were updated according to version 3.1.1 of the coefficients, available at the Bonn website (https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3/dft-d3)

DFT-D functionals

An implementation for dispersion corrections based, called DFT-D is available in ADF. Like DFT-D3 this implementation is easy to use and is also supported by the GUI.

This DFT-D implementation is based on the paper by Grimme[?] and is extremely easy to use. The correction is switched on by specifying *DISPERSION*, possibly with parameters, in the XC input block. See *description of the XC input block* (page 72) for details about the DISPERSION keyword.

Energies calculated Post-SCF using different DFT-D or GGA-D functionals are also present in table printed when METAGGA keyword is specified. These include: BLYP-D, PBE-D, BP86-D, TPSS-D, B3LYP-D, and B97-D. NOTE: this option does not require specifying a DISPERSION keyword in the XC block and thus there is **no correction added to the energy gradient** in this case. Please also note that although the original B97 functional includes HF exchange (and is thus a hybrid functional), the B97-D is a pure GGA. B3LYP-D is, however, a hybrid functional. The following functional-dependent global scaling factors s_6 are used: 1.2 (BLYP-D), 0.75 (PBE-D), 1.05 (BP86-D), 1.0 (TPSS-D), 1.05 (B3LYP-D), and 1.25 (B97-D). These are fixed and cannot be changed.

Regarding performance of different functionals, testing has shown that BLYP-D gives good results for both energies and gradients involving VdW interactions. Post-SCF energy-only calculations at fixed geometries showed that also B97-D gives good binding energies compared to high-level reference data. Thorough comparison of different DFT-D functionals can be found in ref.[67]

Note: The original paper by Grimme included parameters for elements H throughout Xe. In ADF2009.01 values for dispersion parameters for DFT-D functionals for heavier elements (Cs-Rn) have been added. These new values have not been tested extensively. Thus, in this implementation, no dispersion correction is added for interactions involving atoms heavier than Radon.

DFT-D is invoked with the XC block, for example

---

[67] S. Grimme, , J. Antony, T. Schwabe and C. Mück-Lichtenfeld, *Density Functional Theory with Dispersion Corrections for Supramolecular Structures, Aggregates, and Complexes of (Bio)Organic Molecules*, Organic & Biomolecular Chemistry 5, 741 (2007) (https://doi.org/10.1039/B615319B)

```
XC
  GGA BLYP
  Dispersion
END
```

dDsC: density dependent dispersion correction

The DISPERSION dDsC key invokes the density dependent dispersion correction[?], which has been parametrized for the functionals BLYP, PBE, BP, revPBE, B3LYP, PBE0 and BHANDHLYP.

```
XC
  GGA BLYP
  Dispersion dDsC
END
```

For other functionals one can set the dDsC parameters ATT0 and BTT0 with

```
XC
  ...
  DISPERSION dDsC ATT0=att0 BTT0=btt0
END
```

The dispersion dDsC in ADF can not be used with fragments larger than 1 atom. The reason is that ADF uses the Hirshfeld partitioning on fragments for dDsC, which is only correct if the fragments are atoms.

DFT-ulg

The DISPERSION UFF key invokes the universal correction of density functional theory to include London dispersion (DFT-ulg)[?], which has been parametrized for all elements up to Lr (Z=103), and for the functional PBE, PW91, and B3LYP. For other functionals the PBE parameters will be used. Example:

```
XC
  GGA PBE
  Dispersion UFF
END
```

DFT-MBD functionals

The DISPERSION MBD key invokes the MBD@rsSCS method[?], which is designed to accurately describe long-range correlation (and thus dispersion) in finite-gap systems, including at the same time a description of the short-range interactions from the underlying DFT computation of the electronic structure. The MBD (many-body dispersion) method[55] obtains an accurate description of van der Waals (vdW) interactions that includes both screening effects and treatment of the many-body vdW energy to infinite order. The revised MBD@rsSCS method[?] employs a range-separation (rs) of the self-consistent screening (SCS) of polarizabilities and the calculation of the long-range correlation energy. It has been parametrized for the elements H-Ba, Hf-Rn, and for the functional PBE and PBE0. Note that the MBD@rsSCS method depends on Hirshfeld charges. In calculating forces the dependence of the Hirshfeld charges on the actual geometry is neglected. The MBD method is implemented in case the BeckeGrid is used for the numerical integration. Example for PBE MBD@rsSCS:

---

[55] A. Tkatchenko, R.A. DiStasio Jr., R. Car, M. Scheffler *Accurate and Efficient Method for Many-Body van der Waals Interactions*, Physical Review Letters 108, 236402 (2012) (https://doi.org/10.1103/PhysRevLett.108.236402)

```
XC
  GGA PBE
  Dispersion MBD
END
```

One can use user defined values with:

```
XC
  Dispersion MBD {RSSCS|TS} {BETA=beta}
END
```

MBD {RSSCS|TS} {BETA=beta} The default method for MBD is MBD@rsSCS. Optionally one can use MBD@TS or change the used parameter β with setting beta.

4.2.14 Post-SCF energy functionals

GGA energy functionals

In principle you may specify different functionals to be used for the *potential,* which determines the self-consistent charge density, and for the *energy* expression that is used to evaluate the (XC part of the) energy of the charge density. To be consistent, one should generally apply the same functional to evaluate the potential and energy respectively. Two reasons, however, may lead one to do otherwise:

- The evaluation of the GGA part in the *potential* is more time-consuming than LDA. The effect of the GGA term in the potential on the self-consistent charge density is often not very large. From the point of view of computational efficiency it may, therefore, be attractive to solve the SCF equations at the LDA level (i.e. not including GGA terms in the potential), and to apply the full expression, including GGA terms, to the energy evaluation *a posteriori*: post-SCF.
- A particular XC functional may have only an implementation for the potential, but not for the energy (or vice versa). This is a rather special case, intended primarily for fundamental research of Density Functional Theory, rather than for run-of-the-mill production runs.

One possibility is to calculate a whole list of post-SCF energy functionals using the METAGGA keyword, see next section. For some functionals the next possibility is enough. One has to specify different functionals for potential and energy evaluations respectively, using:

```
XC
 {LDA {Apply} LDA {Stoll}}
 {GGA {Apply} GGA}
end
```

Apply States whether the functional defined on the pertaining line will be used self-consistently (in the SCF-potential), or only post-SCF, i.e. to evaluate the XC energy corresponding to the charge density. The value of apply must be SCF or Energy. A value postSCF will also be accepted and is equivalent to Energy. A value Potential will also be accepted and is equivalent to SCF. For each record separately the default (if no Apply value is given in that record) is SCF. For each of the two terms (LDA, GGA) in the functional: if no record with Energy specification is found in the data block, the evaluation of the XC energy will use the same functional as is applied for the potential.

LDA, GGA See the XC potential section for all possible values.

Meta-GGA and hybrid energy functionals

The post SCF energy calculation is an easy and cheap way to get a reasonable guess for the bond energies for different XC functionals at the same time. Note that post-SCF energy calculations for a certain XC functional will not be so accurate if the functional form of the XC functional used in the SCF is very different from the XC functional used post SCF. The relative accuracy of post-SCF energies may not be so high if one looks at small energy differences. For accurate energy calculations it is recommended to use the same XC functional during the SCF as for the energy.

The calculation of a large, pre-specified list of LDA, GGA, and meta-GGA energy functionals is invoked by specifying

```
METAGGA
```

as a separate keyword. The following (incomplete) list gives an idea of the (meta-)GGA density functionals that will then be calculated (the t-MGGA functional is the θ-MGGA functional of Ref.[56]):

```
BP, PW91, mPW, BLYP, PBE, RPBE, revPBE, mPBE, OLYP, OPBE, KCIS, PKZB, VS98, FT97,␣
↪BLAP3,
HCTH, tau-HCTH, BmTau1, BOP, OLAP3, TPSS, KT1, KT2, B97, M06-L, t-MGGA.
```

The hybrid GGA and hybrid meta-GGA energy functionals are calculated if in addition to the METAGGA key, the key

```
HARTREEFOCK
```

is included. The following (incomplete) list gives an idea of the extra hybrid (meta-)GGA density functionals that will then be calculated:

```
B3LYP, B3LYP*, B1LYP, KMLYP, O3LYP, X3LYP, BHandH, BHandHLYP, B1PW91, MPW1PW, MPW1K,
PBE0, OPBE0, TPSSh, tau-HCTH-hybrid, B97, M05, M05-2X, M06, M06-2X.
```

The keys METAGGA and HARTREEFOCK can be used in combination with any XC potential. Note that at the moment hybrid functionals can not be used in combination with frozen cores. Also most METAGGA functionals will give wrong results if used in combination with frozen cores. Thus it is best to use an all electron basis set if one of the keywords METAGGA or HARTREEFOCK is used. One should include the `HARTREEFOCK` keyword also in the create runs of the atoms. In ADF the hybrid energies only make sense if the calculation is performed with completely filled orbitals. In case of a high spin electron configuration one can do ROKS, see *ROKS for high spin open shell molecules* (page 57).

The Examples document describes an application to the OH molecule for the METAGGA option. More output, on the total XC energy of the system, can be obtained by specifying

```
PRINT METAGGA
```

This latter option is intended for debugging purposes mainly and is not recommended for general use.

The implementation calculates the total XC energy for a system and writes it to a file. This is always done in Create runs. If the basic fragments are atoms, the keyword

```
ENERGYFRAG
   ATOM [filename]
   ATOM [filename]
   ... ...
END
```

specifies that different atomic fragment files are to be used in the meta-GGA energy analysis than the regular atomic fragment files from the create runs. This keyword cannot be used for molecular fragment files. In order to compare meta-GGA energy differences between molecular fragments and the total molecule, results from the various calculations need to be combined by hand.

---

[56] P. de Silva and C. Corminboeuf, *Communication: A new class of non-empirical explicit density functionals on the third rung of Jacob's ladder*, Journal of Chemical Physics 143, 111105 (2015) (https://doi.org/10.1063/1.4931628)

In such situations, it is advisable to use a somewhat higher integration accuracy than one would normally do, at least for the smaller fragments, as there is no error cancellation as in a regular ADF bond energy analysis.

A general comment is that some functionals show a more stable behavior than others (at least in our current implementation). In general, the functionals which are dependent on the Laplacian of the density may display a large variation with respect to basis set changes or different numerical integration accuracy. For this reason we currently recommend FT97 in favor of FT98. Similarly, the results with the BmTau1 functional should still be carefully checked. In our test calculations on the G2 set of molecules, the VS98 showed best performance, both for the average error and for the maximum error. The G2 set consists only of small molecules with elements up to Cl. The relative performance for transition metals and heavy elements is unknown and may well be very different from the ordering for the G2 set.

Post Hartree-Fock energy functionals

This is mostly taken from text by the authors of Ref.[57]: In the early days of DFT, non-self-consistent Kohn-Sham energy was often evaluated upon Hartree-Fock (HF) densities as a way to test new approximations. This method was called HF-DFT. It has been discovered that in some cases, HF-DFT actually gave more accurate answers when compared to self-consistent DFT calculations. In Ref.[57], it was found that DFT calculations can be categorized into two different types of calculations. The error of an approximate functional can be decomposed into two parts: error from the functional (functional error), and error from the density (density-driven error). For most calculations, functional error is dominant, and here self-consistent DFT is usually better than non-self consistent DFT on more accurate densities (called density corrected DFT (DC-DFT)). Unlike these ‘normal’ calculations, there is a class of calculations where the density-driven error is much larger, so DC-DFT give better a result than self-consistent DFT. These calculations can be classified as ‘abnormal’. HF-DFT is a simple implementation of DC-DFT and a small HOMO-LUMO gap is an indicator of an ‘abnormal’ calculation, thus, HF-DFT would perform better in such cases.

In ADF one can do HF-DFT with:

```
XC
  HartreeFock
END
MetaGGA
```

This will produce a large, pre-specified list of LDA, GGA, meta-GGA, hybrid, and metahybrid energy functionals.

4.3 Relativistic effects

Scalar relativistic effects using the ZORA formalism are included by default in ADF calculations. Different relativistic options, such as spin-orbit coupling, can be included using the `Relativity` key

Note: Starting from the 2020 version, ZORA scalar relativistic effects are included by default. In 2019.3 and previous versions of ADF, relativistic effects were **not** included by default.

```
Relativity
   Formalism [Pauli | ZORA | X2C | RA-X2C]
   Level [None | Scalar | Spin-Orbit]
   Potential [MAPA | SAPA]
   SpinOrbitMagnetization [NonCollinear | Collinear | CollinearX | CollinearY |␣
↪CollinearZ]
End
```

---

[57] M.-C. Kim, E. Sim, and K. Burke, *Understanding and Reducing Errors in Density Functional Calculations*, Physical Review Letters 111, 2073003 (2013) (https://doi.org/10.1103/PhysRevLett.111.073003)

`Relativity`

Type Block

Description Options for relativistic effects.

`Formalism`

Type Multiple Choice

Default value ZORA

Options [Pauli, ZORA, X2C, RA-X2C]

Description Note that if Level is None, no relativistic effects are taken into account, irrespective of the chosen formalism.

Pauli stands for the Pauli Hamiltonian.

ZORA means the Zero Order Regular Approximated Hamiltonian, recommended.

X2C and RA-X2C both stand for an exact transformation of the 4-component Dirac equation to 2-components.

X2C is the modified Dirac equation by Dyall.

RA-X2C is the regular approach to the modified Dirac equation.

`Level`

Type Multiple Choice

Default value Scalar

Options [None, Scalar, Spin-Orbit]

GUI name Relativity

Description None: No relativistic effects.

Scalar: Scalar relativistic. This option comes at very little cost.

Spin-Orbit: Spin-orbit coupled. This is the best level of theory, but it is (4-8 times) more expensive than a normal calculation. Spin-orbit effects are generally quite small, unless there are very heavy atoms in your system, especially with p valence electrons (like Pb).

See also the SpinOrbitMagnetization subkey.

`Potential`

Type Multiple Choice

Default value MAPA

Options [MAPA, SAPA]

Description Starting from ADF2017 instead of SAPA (the Sum of neutral Atomical potential Approximation) MAPA is used by default for ZORA. The MAPA (the Minumium of neutral Atomical potential Approximation) at a point is the minimum of the neutral Atomical potentials at that point. Advantage of MAPA over SAPA is that the gauge dependence of ZORA is reduced. The ZORA gauge dependency is small for almost all properties, except for the electron density very close to a heavy nucleus. The electron density very close to a heavy nucleus can be used for the interpretation of isomer shifts in Mossbauer spectroscopy.

`SpinOrbitMagnetization`

Type Multiple Choice

Default value CollinearZ

Options [NonCollinear, Collinear, CollinearX, CollinearY, CollinearZ]

Description Relevant only for spin-orbit coupling and if unrestricted key has been activated.

Most XC functionals have as one ingredient the spin polarization in case of unrestricted calculations. Normally the direction of the spin quantization axis is arbitrary and conveniently chosen to be the z-axis.

However, in a spin-orbit calculation the direction matters, and it is arbitrary to put the z-component of the magnetization vector into the XC functional. There is also the exotic option to choose the quantization axis along the x or y axis.

It is also possible to plug the size of the magnetization vector into the XC functional. This is called the non-collinear approach.

- NonCollinear: the non-collinear method. - CollinearXYZ: use the x, y, or z component as spin polarization for the XC functional. - Collinear: the same as CollinearZ.

MAPA and SAPA

The SAPA method is described in Ref.[1] for the BAND program. The same potential was used in the ADF program. However, starting from ADF 2017 instead of SAPA (the Sum of neutral Atomical potential Approximation) MAPA is used by default for ZORA. The MAPA (the Minimum of neutral Atomical potential Approximation) at a point is the minimum of the neutral Atomical potentials at that point. Advantage of MAPA over SAPA is that the gauge dependence of ZORA is reduced. The ZORA gauge dependency is small for almost all properties, except for the electron density very close to a heavy nucleus. The electron density very close to a heavy nucleus can be used for the interpretation of isomer shifts in Mössbauer spectroscopy

Recommendations:

```
Relativity
   Level Scalar
   Formalism ZORA
   Potential MAPA
End
```

or

```
Relativity
   Level Spin-Orbit
   Formalism ZORA
   Potential MAPA
End
```

---

[1] P.H.T. Philipsen, E. van Lenthe, J.G. Snijders and E.J. Baerends, *Relativistic calculations on the adsorption of CO on the (111) surfaces of Ni, Pd, and Pt within the zeroth-order regular approximation*, Physical Review B 56, 13556 (1997) (https://doi.org/10.1103/PhysRevB.56.13556)

4.3.1 Pauli

Specification of the Pauli formalism means that the first order relativistic corrections (the Pauli Hamiltonian) will be used[2,3,4,5,6,7,8,9,10,11]. In a *scalar* relativistic run ADF employs the single point group symmetry and only the so-called *scalar* relativistic corrections, Darwin and Mass-Velocity. The treatment is not strictly first-order, but is *quasi*-relativistic, in the sense that the first-order scalar relativistic Pauli Hamiltonian is diagonalized in the space of the non-relativistic solutions, i.e. in the non-relativistic basis set.

The quasi-relativistic approach improves results considerably over a first-order treatment. There are, however, theoretical deficiencies due to the singular behavior of the Pauli Hamiltonian at the nucleus. This would become manifest in a complete basis set but results are reasonable with the normally employed basis sets. However, this aspect implies that it is not recommended to apply this approach with an all-electron basis set for the heavy atoms, and for very heavy elements even a frozen core basis set often fails to give acceptable results. The problems with the quasi relativistic approach of the Pauli Hamiltonian are discussed for example in Ref.[12].

4.3.2 ZORA

The ZORA approach gives generally better results than the Pauli formalism. For all-electron calculations, and in fact also for calculations on very heavy elements (Actinides), the Pauli method is absolutely unreliable. Therefore the ZORA method is the recommended approach for relativistic calculations with ADF.

ZORA refers to the Zero Order Regular Approximation[?][17,18,19,20]. This formalism requires special basis sets, primarily to include much steeper core-like functions; applying the ZORA method with other, not-adapted basis sets, gives unreliable results. The ZORA basis sets can be found in subdirectories under the $AMSHOME/atomicdata/ADF/ZORA directory.

The ZORA formalism can also be used in Geometry Optimizations. However, there is a slight mismatch between the energy expression and the potential in the ZORA approach, which has the effect that the geometry where the gradients are zero does not exactly coincide with the point of lowest energy. The differences are very small, but not completely negligible, in the order of 0.0001 Angstrom.

---

[2] J.G. Snijders and E.J. Baerends, *A perturbation theory approach to relativistic calculations. I. Atoms*, Molecular Physics 36, 1789 (1978) (https://doi.org/10.1080/00268977800102771)

[3] J.G. Snijders, E.J. Baerends and P. Ros, *A perturbation theory approach to relativistic calculations. II. Molecules*, Molecular Physics 38, 1909 (1979) (https://doi.org/10.1080/00268977900102941)

[4] T. Ziegler, J.G. Snijders and E.J. Baerends, *Relativistic effects on bonding*, Journal of Chemical Physics 74, 1271 (1981) (https://doi.org/10.1063/1.441187)

[5] R.L. DeKock, E.J. Baerends, P.M. Boerrigter and J.G. Snijders, *On the nature of the first excited states of the uranyl ion*, Chemical Physics Letters 105, 308 (1984) (https://doi.org/10.1016/0009-2614(84)85036-8)

[6] R.L. DeKock, E.J. Baerends, P.M. Boerrigter and R. Hengelmolen, *Electronic structure and bonding of* $Hg(CH_3)_2$, $Hg(CN)_2$, $Hg(CH_3)(CN)$, $Hg(CCCH_3)_2$, *and* $Au(PMe)_3(CH_3)$, Journal of the American Chemical Society 106, 3387 (1984) (https://doi.org/10.1021/ja00324a001)

[7] P.M. Boerrigter, *Spectroscopy and bonding of heavy element compounds*, 1987, Vrije Universiteit.

[8] P.M. Boerrigter, M.A. Buijse and J.G. Snijders, *Spin-Orbit interaction in the excited states of the dihalogen ions* $F_2{}^+$, $Cl_2{}^+$ and $Br_2{}^+$, Chemical Physics 111, 47 (1987) (https://doi.org/10.1016/0301-0104(87)87007-6)

[9] P.M. Boerrigter, E.J. Baerends and J.G. Snijders, *A relativistic LCAO Hartree-Fock-Slater investigation of the electronic structure of the actinocenes* $M(COT)_2$, *M=Th, Pa, U, Np and Pu*, Chemical Physics 122, 357 (1988) (https://doi.org/10.1016/0301-0104(88)80018-1)

[10] T. Ziegler, V. Tschinke, E.J. Baerends, J.G. Snijders and W. Ravenek, *Calculation of bond energies in compounds of heavy elements by a quasi-relativistic approach*, Journal of Physical Chemistry 93, 3050 (1989) (https://doi.org/10.1021/j100345a036)

[11] J. Li, G. Schreckenbach and T. Ziegler, *A Reassessment of the First Metal-Carbonyl Dissociation Energy in* $M(CO)_4$ (M = Ni, Pd, Pt), $M(CO)_5$ (M = Fe, Ru, Os), and $M(CO)_6$ (M = Cr, Mo, W) *by a Quasirelativistic Density Functional Method*, Journal of the American Chemical Society 117, 486 (1995) (https://doi.org/10.1021/ja00106a056)

[12] E. van Lenthe, A.E. Ehlers and E.J. Baerends, *Geometry optimization in the Zero Order Regular Approximation for relativistic effects*, Journal of Chemical Physics 110, 8943 (1999) (https://doi.org/10.1063/1.478813)

[17] E. van Lenthe, E.J. Baerends and J.G. Snijders, *Relativistic regular two-component Hamiltonians*, Journal of Chemical Physics 99, 4597 (1993) (https://doi.org/10.1063/1.466059)

[18] E. van Lenthe, E.J. Baerends and J.G. Snijders, *Relativistic total energy using regular approximations*, Journal of Chemical Physics 101, 9783 (1994) (https://doi.org/10.1063/1.467943)

[19] E. van Lenthe, J.G. Snijders and E.J. Baerends, *The zero-order regular approximation for relativistic effects: The effect of spin-orbit coupling in closed shell molecules*, Journal of Chemical Physics 105, 6505 (1996) (https://doi.org/10.1063/1.472460)

[20] E. van Lenthe, R. van Leeuwen, E.J. Baerends and J.G. Snijders, *Relativistic regular two-component Hamiltonians*, International Journal of Quantum Chemistry 57, 281 (1996) (https://doi.org/10.1002/(SICI)1097-461X(1996)57:3%3C281::AID-QUA2%3E3.0.CO;2-U)

4.3.3 X2C and RA-X2C

X2C stands for an exact transformation of the 4-component Dirac equation to 2-components[21][22]. RA-X2C also stands for an exact transformation of the 4-component Dirac equation to 2-components, in this case using a regular approach to calculate the transformation matrix. In practice, however, often approximations are made, and this is also true for the implementation in ADF. In the X2C and RA-X2C method implemented in ADF, first the 4-component Dirac equation for a model potential (MAPA) of the molecule is calculated for the given ADF basis set, using the modified Dirac equation by Dyall[13] for X2C, or using the regular approach[14] to the modified Dirac equation for RA-X2C.

With the method used in ADF in the basis set limit X2C and RA-X2C should lead to the same results for the model potential (MAPA). However, since in practice always a finite basis set is used, even for the model potential the results for X2C and RA-X2C will differ. Next the (electronic) 4-component Dirac solutions are transformed exactly to 2-components[15]. The transformation used is also used to calculate an effective one-electron 2-component kinetic energy operator in a basis set representation. This kinetic energy matrix is then assumed to be constant in further ADF SCF calculation on the molecule. The full Hamiltonian is approximated using the electron density in the 2-component picture, the so called Foldy-Wouthuysen picture.

In a similar way a spin-free (scalar relativistic) form of X2C and RA-X2C are calculated. The spin-free form of X2C and RA-X2C will be different even in the basis set limit, see Ref.[16].

X2C and RA-X2C in ADF can be used in single point calculations only. An all electron basis set is needed. For bond energies, polarizabilities and TD-DFT excitation energies the Foldy-Wouthuysen picture will be used. For the calculation of the EFG, ESR g-value, and ESR A-tensor, the Dirac picure will be used. X2C and RA-X2C is not implemented for frozen cores, optimization of nuclear coordinates, frequencies, NMR properties.

4.3.4 Spin-Orbit coupling

The Spin-Orbit option uses double-group symmetry. The symmetry-adapted orbitals are labeled by the quantum number J rather than L and any references in input to subspecies, such as a specification of occupation numbers, must refer to the double group labels.

Create runs must *not* use the Spin-Orbit formalism. The SFO analysis of Molecular Orbitals for a Spin-Orbit calculation is only implemented in the case of a scalar relativistic fragment file, which is the whole molecule.

In a Spin-Orbit run each level can allocate 2 electrons (times the dimension of the irreducible representation) as in a normal restricted calculation. However, contrary to the normal case these two electrons are not directly associated with spin-α and spin-β, but rather with the more general Kramer's symmetry. Using the unrestricted feature in order to assign different numbers of electrons to a and b spin respectively cannot be applied as such. However, one can use the unrestricted option in combination with the collinear or non-collinear approximation. In that case one should use symmetry NOSYM, and each level can allocate 1 electron.

---

[21] K.G. Dyall, *Interfacing relativistic and nonrelativistic methods. I. Normalized elimination of the small component in the modified Dirac equation*, Journal of Chemical Physics 106, 9618 (1997) (https://doi.org/10.1063/1.473860)

[22] W. Kutzelnigg and W. Liu, *Quasirelativistic theory equivalent to fully relativistic theory*, Journal of Chemical Physics 123, 241102 (2005) (https://doi.org/10.1063/1.2137315)

[13] K.G. Dyall, *An exact separation of the spin-free and spin-dependent terms of the Dirac-Coulomb-Breit Hamiltonian*, Journal of Chemical Physics 100, 2118 (1994) (https://doi.org/10.1063/1.466508)

[14] A.J. Sadlej and J.G. Snijders, *Spin separation in the regular Hamiltonian approach to solutions of the Dirac equation*, Chemical Physics Letters 229, 435 (1994) (https://doi.org/10.1016/0009-2614(94)01067-6)

[15] E. van Lenthe, E.J. Baerends, and J.G. Snijders, *Construction of the Foldy-Wouthuysen transformation and solution of the Dirac equation using large components only*, Journal of Chemical Physics 105, 2373 (1996) (https://doi.org/10.1063/1.472104)

[16] L. Visscher and E. van Lenthe, *On the distinction between scalar and spin-orbit relativistic effects*, Chemical Physics Letters 306, 357 (1999) (https://doi.org/10.1016/S0009-2614(99)00458-3)

4.4 Solvents and other environments

The solvation methods SCRF (Self-Consistent Reaction Field) and VSCRF (Vertical Excitation SCRF) are no longer supported in ADF2024. In older versions it is available, see also the ADF 2023 manual on SCRF (https://www.scm.com/doc.2023/ADF/Input/SCRF.html) and VSCRF (https://www.scm.com/doc.2023/ADF/Input/VSCRF.html)

4.4.1 COSMO: Conductor like Screening Model

You can study chemistry in solution, as contrasted to the gas phase, with the implementation in ADF[1] of the Conductor like Screening Model (COSMO) of solvation[2][3][4]. The energy derivatives can also be calculated, so geometry optimization, harmonic frequencies, et cetera are available within this model.

The COSMO model is a dielectric model in which the solute molecule is embedded in a molecule-shaped cavity surrounded by a dielectric medium with given dielectric constant ϵ. Energy-related terms are computed for a conductor first, then scaled by the function

$$f(\epsilon) = \frac{\epsilon - 1}{\epsilon + x} \qquad (2.1.1)$$

The empirical scaling factor x is specified in the input data block for the SOLVATION key. The block key SOLVATION turns the solvation calculation on. In most cases default values are available for the involved parameters. In order to construct the cavity atomic cavity radii are used. The default COSMO atomic cavity radii used in ADF are the Van der Waals radii from the MM3 method by Allinger (Ref.[5]) divided by 1.2.

It is also possible to include a linear parametrization of non-electrostatic terms as a function of surface area. To include such term can be specified in the input data block for the SOLVATION key. The default is to include only the part of this term that is proportional to the surface area (default CAV0=0.0, CAV1=0.0067639):

$$E_{\text{non-elst}} = f(\epsilon)(CAV0 + CAV1 \times \text{area}) \qquad (2.1.2)$$

If a calculations was done on a fragment, then the wave function obtained would be optimal for the fragment in solution, but not optimal for gas phase. The energies with the gas phase Hamiltonian would be higher, and the apparent solvation contribution to bonding would also be higher. The net point is that normally the COSMO procedure reports the energy of Esolv(AB), but to get the solvation energy, you need to subtract the E(AB, solv) from E(AB,gas) because the wave function changes (unless you are doing it post-SCF.) For this you need to have the same reference fragments in each case A(g) and B(g).

Because of precision issues in ADF2016 the default surface has been changed to Delley, instead of Esurf.

```
SOLVATION
 {SURF Surface {NOKEEP}}
 {SOLV {Name=solvent} {Eps=78.4} {Del=1.4} {Rad=1.4}
       {Neql=1.9}{Emp=0.0}{Cav0=0.0}{Cav1=0.0067639} }
 {DIV {Ndiv=3} {NFdiv=1} {Min=0.5} {OFAC=0.8}
      {leb1=23} {leb2=29}{rleb=1.5} }
```

(continues on next page)

---

[1] C.C. Pye and T. Ziegler, *An implementation of the conductor-like screening model of solvation within the Amsterdam density functional package*, Theoretical Chemistry Accounts 101, 396 (1999) (https://doi.org/10.1007/s002140050457)

[2] A. Klamt and G. Schüürmann, *COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient*, Journal of the Chemical Society: Perkin Transactions 2, 799 (1993) (https://doi.org/10.1039/P29930000799)

[3] A. Klamt, *Conductor-like Screening Model for real solvents: A new approach to the quantitative calculation of solvation phenomena*, Journal of Physical Chemistry 99, 2224 (1995) (https://doi.org/10.1021/j100007a062)

[4] A. Klamt and V. Jones, *Treatment of the outlying charge in continuum solvation models*, Journal of Chemical Physics 105, 9972 (1996) (https://doi.org/10.1063/1.472829)

[5] N.L. Allinger, X. Zhou, J. Bergsma, *Molecular mechanics parameters*, Journal of Molecular Structure: THEOCHEM 312, 69 (1994) (https://doi.org/10.1016/S0166-1280(09)80008-0)

(continued from previous page)

```
 {NOASS}
 {RADII
   name1=value1
   name2=value2
   ...
 subend }
 {CHARGED {Method=meth} {Conv=1e-8} {Omega=1.0} {Iter=1000}
          {Corr} {LoCosmo LoCosmoDist}}
 {C-MAT How {SCF} tol=1e-10 }
 {DISC {SC=0.01} {LEG=4} {TOL=0.1} }
 {SCF {When} {How}
 {NOCSMRSP}
 {LPRT}
End
```

Presence of the SOLVATION key block triggers the solvent calculation and does not require additional data. With subkeys you can customize various aspects of the model, for instance to specify the type of solute. None of the subkeys is obligatory. Follows a description of the subkeys

SURF Surface {NOKEEP} Defines the type of cavity to be used. `Surface` can be one of the following: Wsurf, Asurf, Esurf, Klamt, or Delley. **Default: Delley**. The Wsurf, Asurf, and Esurf surfaces are constructed with the GEPOL93 algorithm[6]

Wsurf Wsurf triggers the Van der Waals surface (VdW), which consists of the union of all atomic spheres. Not recommended to be used.

Asurf Asurf gives the Solvent-Accessible-Surface (SAS). This is similar to VdW but consists of the path traced by the center of a spherical solvent molecule rolling about the VdW surface or, equivalently, a VdW surface created by atomic spheres to which the solvent radius has been added. These two surface types contain cusps at the intersection of spheres. Not recommended to be used.

Esurf Esurf gives the Solvent-Excluding-Surface (SES), which consists of the path traced by the *surface* of a spherical solvent molecule rolling about the VdW surface. Primarily, this consists of the VdW surface but in the regions where the spheres would intersect, the concave part of the solvent sphere replaces the cusp.

Klamt The fourth surface option is Klamt as described in[7]. It excludes the cusp regions also. Note that this surface might give an incomplete COSMO surface in case of more complicated molecules. Not recommended to be used.

Delley The fifth surface (default surface) is the so called Delley surface, see also Ref.[7]. This Delley type of cavity construction is recommended to be used in COSMO calculations, which results are used as input for COSMO-RS calculation, see the corresponding manual for COSMO-RS. This Delley surface is the default COSMO surface in ADF.

NOKEEP The optional parameter NOKEEP controls surface creation during calculation of frequencies by numerical differentiation. By default, the surface is constructed only once at the central geometry and is used for the rest of the calculation. If the NOKEEP is specified then ADF will construct a new surface at each displaced geometry.

DIV The actual construction of the surface involves a few technical parameters controlled with the subkey DIV

Ndiv, NFdiv Ndiv controls how fine the spheres that in fact describe the surface are partitioned in small surface triangles, each containing one point charge to represent the polarization of the cavity surface. Default division

---

[6] J.L. Pascual-ahuir, E. Silla and I. Tuñon, *GEPOL: An improved description of molecular surfaces. III. A new algorithm for the computation of a solvent-excluding surface*, Journal of Computational Chemistry 15, 1127 (1994) (https://doi.org/10.1002/jcc.540151009)

[7] B. Delley, *The conductor-like screening model for polymers and surfaces*, Molecular Simulation 32, 117 (2006) (https://doi.org/10.1080/08927020600589684)

level for triangles Ndiv=3. Default final division level for triangles NFdiv=1 (NFdiv ≤ Ndiv). Not used in the Delley surface.

`Min` Min specifies the size, in angstrom, of the smallest sphere that may be constructed by the SES surface. For VdW and SAS surfaces it has no meaning. Default Min=0.5. Not used in the Delley surface.

`Ofac` Ofac is a maximum allowed overlap of new created spheres, in the construction procedure. Default Ofac=0.8. Not used in the Delley surface.

`leb1, leb2, rleb` Only used in the Delley surface. For the Delley type of construction one needs to set the variables leb1 (default value 23), leb2 (default value 29), and rleb (default value 1.5 Angstrom) to set the number of surface points. If the cavity radius of an atom is lower than rleb use leb1, otherwise use leb2. These values can be changed: using a higher value for leb1 and leb2 gives more surface points (maximal value leb1, leb2 is 29). A value of 23 means 194 surface points in case of a single atom, and 29 means 302 surface points in case of a single atom Typically one could use leb1 for the surface point of H, and leb2 for the surface points of other elements.

`NOASS` By default all new spheres that are created in the surface-construction are assigned to atoms, for the purpose of gradient computations (geometry optimization). Specifying the noass subkey turns this off. It has no argument.

`SOLV` Solvent details.

`Eps, Rad` Eps specifies the dielectric constant (the default relates to water). In ADF an infinite value for Eps is chosen if Eps is specified to be lower than 1.0. Rad specifies the radius of the (rigid sphere) solvent molecules, in angstrom. Instead of specifying Eps and Rad one can specify a solvent name or formula after ‘name=’. The following table lists names and formulas that are recognized with the corresponding values for Eps and Rad. The Rad in this table is calculated from the density, the molar mass, and a spherical approximation for the solvent. The names and formulas are case-insensitive.

Name	Formula	Eps	Rad
AceticAcid	CH3COOH	6.19	2.83
Acetone	CH3COCH3	20.7	3.08
Acetonitrile	CH3CN	37.5	2.76
Ammonia	NH3	16.9	2.24
Aniline	C6H5NH2	6.8	3.31
Benzene	C6H6	2.3	3.28
BenzylAlcohol	C6H5CH2OH	13.1	3.45
Bromoform	CHBr3	4.3	3.26
Butanol	C4H9OH	17.5	3.31
isoButanol	(CH3)2CHCH2OH	17.9	3.33
tertButanol	(CH3)3COH	12.4	3.35
CarbonDisulfide	CS2	2.6	2.88
CarbonTetrachloride	CCl4	2.2	3.37
Chloroform	CHCl3	4.8	3.17
Cyclohexane	C6H12	2	3.5
Cyclohexanone	C6H10O	15	3.46
Dichlorobenzene	C6H4Cl2	9.8	3.54
DiethylEther	(CH3CH2)2O	4.34	3.46
Dioxane	C4H8O2	2.2	3.24
DMFA	(CH3)2NCHO	37	3.13
DMSO	(CH3)2SO	46.7	3.04
Ethanol	CH3CH2OH	24.55	2.85
EthylAcetate	CH3COOCH2CH3	6.02	3.39
Dichloroethane	ClCH2CH2Cl	10.66	3.15
EthyleneGlycol	HOCH2CH2OH	37.7	2.81

continues on next page

Table 4.1 – continued from previous page

Formamide	HCONH2	109.5	2.51
FormicAcid	HCOOH	58.5	2.47
Glycerol	C3H8O3	42.5	3.07
HexamethylPhosphoramide	C6H18N3OP	43.3	4.1
Hexane	C6H14	1.88	3.74
Hydrazine	N2H4	51.7	2.33
Methanol	CH3OH	32.6	2.53
MethylEthylKetone	CH3CH2COCH3	18.5	3.3
Dichloromethane	CH2Cl2	8.9	2.94
Methylformamide	HCONHCH3	182.4	2.86
Methypyrrolidinone	C5H9NO	33	3.36
Nitrobenzene	C6H5NO2	34.8	3.44
Nitrogen	N2	1.45	2.36
Nitromethane	CH3NO2	35.87	2.77
PhosphorylChloride	POCl3	13.9	3.33
IsoPropanol	(CH3)2CHOH	19.9	3.12
Pyridine	C5H5N	12.4	3.18
Sulfolane	C4H8SO2	43.3	3.35
Tetrahydrofuran	C4H8O	7.58	3.18
Toluene	C6H5CH3	2.38	3.48
Triethylamine	(CH3CH2)3N	2.44	3.81
TrifluoroaceticAcid	CF3COOH	8.55	3.12
Water	H2O	78.39	1.93

Del Del is the value of Klamt's delta_sol parameter, only relevant in case of Klamt surface.

Neql If Neql = ϵ_{NEQL} is included a non-equilibrium solvation is used, i.e. that the dielectric constant ϵ_{NEQL} used in RESPONSE is different from the ground state dielectric constant ϵ. Only relevant in case of TDDFT calculations. Default $\epsilon_{\mathrm{NEQL}} = \epsilon$. The reason for using two different dielectric constants is that the electronic transition can so fast that only the electronic component of the solvent dielectric can respond, i.e., one should use the optical part of the dielectric constant. This is typically referred to as non-equilibrium solvation. The optical dielectric constant can be obtaining from the (frequency dependent) refractive index n of the solvent as: $\epsilon_{\mathrm{NEQL}} = \mathrm{n}^2$.

Emp Emp addresses the empirical scaling factor *x* in the formula 2.1.1 above.

Cav0, Cav1 Other options specify a linear parametrization of non-electrostatic terms as a function of surface area, see the formula 2.1.2 above. Possible values for CAV0 and CAV1 are CAV0 = 1.321 and CAV1 = 0.0067639, see Ref.[8]). In ADF2010 the default values for CAV0 and CAV1 are CAV0 = 0.0 and CAV1 = 0.0. However, starting from ADF2012 the default values for CAV0 and CAV1 are CAV0 = 0.0 and CAV1 = 0.0067639, If CAV0 is not zero, Esolv(AB) is not the same as Esolv(A) + Esolv(B) if A and B are far apart. This is the reason why CAV0 is set to zero, by default. By default CAV1 is not set to zero, thus by default there is a solvation energy term that does depend on the size of the cavity (surface area).

COSMO Radii In order to construct the surface you have to specify the atomic ('Van der Waals') radii. There are three ways of doing this. In the first method you append 'adf.R=value' to the atomic coordinates record, in the ATOMS key block. This would look like, for instance:

```
C 1 2 3 CC CCO CCOH adf.f=C.dz adf.R=2.0
```

It assigns a radius of 2.0 to the Carbon atom. In the second method you apply the same format, but specify a symbol (identifier) rather than a value:

---

[8] T.N. Truong and E.V. Stefanovich, *A new method for incorporating solvent effect into the classical, ab initio molecular orbital and density functional theory frameworks for arbitrary shape cavity*, Chemical Physics Letters 240, 253 (1995) (https://doi.org/10.1016/0009-2614(95)00541-B)

```
C 1 2 3 CC CCO CCOH adf.f=C.dz adf.R=C-sp3
```

The identifiers must be defined in the (optional) RADII subkey block in the Solvation data block (see next). In the third method, you don't modify the Atoms block at all. In this case, the RADII subkey must be used and the 'identifiers' in it must be exactly the atom type names in the Atoms block.

RADII This subkey is block type. Its data block (if the subkey is used) must terminate with a record subend. In the Radii data block you give a list of identifiers and values:

```
SOLVATION
   ...
   Radii
      name1=value1
      name2=value2
      ...
   Subend
   ...
End
```

The values are the radii of the atomic spheres, in the same units of length as used in the Atoms block (angstrom or bohr). The names specify to which atoms these values apply. As discussed for the Solv subkey this depends on the Atoms block. If in the specification of atomic coordinates you have used the 'adf.R=' construct to assign radii, with identifiers rather than values for the R-value, these identifiers must be defined in the Radii sub block. If no 'adf.R=' construct was applied in the Atoms block, you must use the atom type names as they occurred in the Atoms data block. Referring to the example given in the Solv subkey discussion, you might have:

```
Radii
   C-sp3=2.0
   ...
Subend
```

A simple atom type reference might look like:

```
Radii
   C=2.0
   ...
Subend
```

When no radius specified a default value is used. The default value for an atom is the corresponding Van der Waals radius from the MM3 method by Allinger (Ref.[?]) divided by 1.2, see Ref.[9].

Another popular choice are the so called Klamt atomic cavity radii (Ref.[10]), which are optimized for a few elements:

```
Radii
   H  1.30
   C  2.00
   N  1.83
   O  1.72
   F  1.72
   Si 2.48
   P  2.13
   S  2.16
   Cl 2.05
```

(continues on next page)

---

[9] M. Swart, E. Rösler, F.M. Bickelhaupt, *Proton Affinities in Water of Maingroup-Element Hydrides - Effects of Hydration and Methyl Substitution*, European Journal of Inorganic Chemistry 2007 3646 (2007) (https://doi.org/10.1002/ejic.200700228)

[10] A. Klamt, V. Jonas, T. Bürger and J.C. Lohrenz, *Refinement and Parametrization of COSMO-RS*. J. Phys. Chem. A 102, 5074 (1998) (https://doi.org/10.1021/jp980017s)

(continued from previous page)

```
   Br 2.16
   I  2.32
Subend
```

CHARGED {Method=meth} {Conv=1e-8} {Omega=1.0} {Iter=1000} {Corr} {LoCosmo LoCosmoDist}
This addresses the determination of the (point) charges that model the cavity surface polarization. In COSMO calculations you compute the surface point charges *q* by solving the equation A*q*=-*f*, where *f* is the molecular potential at the location of the surface charges *q* and A is the self-interaction matrix of the charges. The number of charges can be substantial and the matrix A hence very large. A direct method, i.e. inversion of A, may be very cumbersome or even impossible due to memory limitations, in which case you have to resort to an iterative method. Meth specifies the equation-solving algorithm. Meth=INVER requests direct inversion. Meth=GAUS calls for the Gauss-Seidel iterative method. Meth=Jacobi activates another standard iterative procedure. The latter two methods require a positive-definite matrix (which may fail to be the case in an actual calculation) and can be used with a relaxation technique, controlled by the relaxation parameter OMEGA (1.0=no relaxation). Meth=CONJ (default) uses the preconditioned biconjugate gradient method. This is guaranteed to converge and does not require huge amounts of memory. CONV and ITER are the convergence criterion and the maximum number of iterations for the iterative methods. Some of the molecular electronic charge distribution may be located outside the cavity. This affects the assumptions underlying the COSMO equations. LOCOSMO can only be used in combination with FDE.

Specifying the CORR option to the CHARGED subkey constrains the computed solvent surface charges to add up to the negative of the molecular charge. In ADF2016 this outlying charge correction is also done during the SCF. Note that before ADF2016 it was only a post-SCF correction on the COSMO charges. In ADF2018 a bug that was introduced in ADF2016 in case of this outlying charge correction was fixed for the energy (typically a small effect).

C-MATRIX How: For the potential *f* we need the Coulomb interaction between the charges *q* and the molecular electronic density (and nuclei). Three methods are available, specified by the first option to the C-Matrix subkey.

a) EXACT: compute the straightforward Coulomb potential due to the charge *q* in each point of the molecular numerical integration grid and integrate against the electronic charge density. This is, in principle, exact but may have inaccuracies when the numerical integration points are very close to the positions of a charge *q*. To remedy this, the point charges *q* can be 'smeared out' and represented by a disc, see the next subkey DISC.

b) FIT: same as EXACT, but the *q*-potentials are now integrated not against the exact electronic charge density, but against the (much cheaper-to-compute) fitted density. The same DISC considerations apply.

c) POT: evaluate the molecular potential at the position of the charge *q* and multiply against the *q*-strength. Since the molecular Coulomb potential is computed from the fit density, any difference in results between the FIT and the POT approach should be attributed to the DISC issue. POT is the default, because it is faster, and is only inadequate if the fit density is very inaccurate, which would be a problem anyway.

SCF: If you specify this option, the computation of the Coulomb interaction matrix (between electrons and surface charges) is carried out during the SCF procedure, but this turns out to hamper the SCF convergence behavior. Therefore: not recommended. *IF* you use it, the program will switch to one of the other 3 methods, as given by the 'How' option, as soon as the SCF convergence error drops below TOL: (applies only to the SCF option, which is not recommended).

DISC Applies only when the C-matrix method is EXACT or FIT. Note, however, that the default for the C-matrix method is POT, in which case the DISC subkey has no meaning. The DISC key lets the program replace the point charges *q* by a solid uniformly charged spherical surface disc whenever the numerical integration accuracy requires so, i.e. for those charges that are close to numerical integration points. Options: SC defines a shrinking factor, by which the actual disc radius used is reduced from its 'normal' value: an inscribed disc in the triangular surface partitions that define the distribution of surface charges, see the subkey DIV. LEG gives the polynomial expansion order of the disc potentials. The Legendre expansion converges rapidly and the default should be adequate.

TOL is a tolerance parameter to control the accuracy of the disc potential evaluations.

SCF In COSMO calculations you can include the surface charges in the Fock operator self-consistently, i.e. by recomputing the charges q at every SCF cycle and include them in the equations, or in a perturbational manner, i.e. post-SCF. This is controlled with the first option. The When option must be either VAR or PERT, for variational and perturbational, respectively. Default is VAR. The second (HOW) option applies only to the WHEN=VAR case and may affect the speed of SCF convergence. The COSMO calculation implies a considerable increase in CPU time! Values for HOW:

- ALL: This includes it in all SCF cycles (except for the first SCF cycle, which is gas-phase)
- LAST: This lets the program first converge the SCF completely without any solvent effects. Thereafter, the COSMO is turned on, hopefully converging in fewer cycles now, to compensate for the 'double' SCF effort.
- TOL=0.1 (or another value) is an in-between approach: converge the gas-phase SCF until the SCF error is below TOL, then turn on COSMO.

NOCSMRSP Relevant only in combination with the time-dependent DFT (TDDFT) applications: the EXCITATION, the RESPONSE, or the AORESPONSE key. If this subkey NOCSMRSP is included the induced electronic charges which are present in the TDDFT calculations, will not influence the COSMO surface charges. No dielectric constant in the response might be closer to the optical dielectric constant than using the full dielectric constant, see also subargument NEQL of the subkey SOLV of the key SOLVATION. By default, in absence of this subkey NOCSMRSP, the induced electronic will influence the COSMO surface charges. If one does geometry optimization of the excited state this makes sense, since then the solvent dielectric has time to fully respond.

LPRT This is a debug switch and triggers a lot more output related to the cavity construction etc.

Warning

Numerical frequencies calculated with COSMO should be checked for stability with respect to the NumericalDifferentiation%NuclearStepSize in the AMS part of the input, the numerical differentiation step size. The problem is that the COSMO surface changes slightly when a nucleus is moved from its equilibrium position. The change is usually small but in some cases it may result is creation or annihilation of surface points, which will lead to discontinuities in the potential energy surface and may result in inaccurate frequencies.

Thus, when calculating vibrational frequencies numerically with COSMO, one should try decreasing the NumericalDifferentiation%NuclearStepSize value in the AMS part of the input, until no changes in frequencies are observed. However, the value should not be too small because then the total numerical noise may become too large compared to the generated forces. A general recommendation would be to try to decrease NuclearStepSize by a factor of 2 at a time. Of course, this procedure may be very expensive for a large molecule. If this the case, one could use the NormalModes%ReScanModes or NormalModes%ReScanFreqRange in the AMS part of the input, in order to recalculate frequencies along a normal mode. It should be noted that generally frequencies that have a small force constant are more sensitive to the numerical noise.

4.4.2 SM12: Solvation Model 12

Continuum solvation can be done with the Minnesota's Solvation Model 12 (SM12)[1]. Details on the implementation of SM12 in ADF can be found in Ref.[2]. The energetics of solvation is calculated using:

$$\Delta G_S^{\otimes} = \Delta E_E + G_P + G_{CDS} + \Delta G_N + \Delta G_{conc}^{\otimes}$$

where the symbol $\otimes$ denotes an arbitrary choice of standard states, ΔE_E is the change in the solute's internal electronic energy in transferring from the gas phase to the liquid phase at the same geometry, G_P is the polarization free energy of the solute–solvent system when the solute is inserted, G_{CDS} is the component of the free energy that is nominally associated with cavitation, dispersion, and solvent structure, ΔG_N is the change in ΔG_S^0 due to a change in nuclear

---

[1] A.V. Marenich, C.J. Cramer, and D.G. Truhlar, *Generalized Born Solvation Model SM12*, Journal of Chemical Theory and Computation 9, 609 (2013) (https://doi.org/10.1021/ct300900e)

[2] C.A. Peeples and G. Schreckenbach, *Implementation of the SM12 Solvation Model into ADF and Comparison with COSMO*, Journal of Chemical Theory and Computation 12, 4033 (2016) (https://doi.org/10.1021/acs.jctc.6b00410)

coordinates, and $\Delta G^{\otimes}_{conc}$ accounts for the difference in concentrations, if any, in the gas-phase standard state and the solution-phase one. In case of 1 M concentration in both solution and gas, then $\Delta G^{\otimes}_{S}$ = 0 kcal/mol, which yields ΔG^{*}_{S}. If the same geometry is used in solution and gas phase calculation, then ΔG_N is zero.

SM12 makes use of the Generalized Born approximation to calculate the bulk electrostatic contribution. This is comprised of several terms that are together known as the ENP (Electronic, Nuclear, and Polarization) term G_P. The SM12 model in ADF uses *CM5 charges* (page 287)[3]. CM5 is a class 4 charge model, making use of both empirical and density related terms. It is comprised of Hirshfeld charges[4], a simple bond order calculation, atomic distances, and atom specific parameters[?]. The covalent radii utilized are based on the atomic covalent radius from the Handbook of Chemistry and Physics[5]. The Coulomb integral is described with the use of an approximation from Still et al.[6]. Several parameters go into describing this, which include: inter atomic distance, an empirical Born constant, and the Born area, which is calculated with the Analytical Surface Area (ASA) algorithm[7]. The Born area is calculated using Legendre-Gauss quadrature from the atomic radii to a sphere that encapsulates the entire molecule.

The ASA algorithm is also used to calculate the solvent accessible surface area (SASA), which is computed within the CDS (Cavitation, Dispersion, Solvation) term of SM12. The CDS term depends on three terms:

- SASA (ASA Algorithm)
- Atomic surface tension
- Macroscopic surface tension

Atomic surface tension is based on atom-atom distances and the solvent. Macroscopic surface tension is solvent specific. The SM12 implementation in ADF reports energies in an atom specific way. You can attribute exact CDS and polarization energies to each atom in your solute. The parameters for SM12 are derived to explicitly incorporate organic elements (N, C, O, F, Si, P, S, Cl, Br, I), with less emphasis on non-organics. Also, while most solvents have a generic atomic surface tension reliance for atoms, water has its own explicit set of parameters to better describe it.

```
SOLVATION SM12
  {SOLV NAME=solvent}
  {CUST {NAME=solvent}}
  {REF 1.33}
  {ACID 0.82}
  {BASE 0.35}
  {TENS 103.62}
  {EPS 78.36}
  {ARO 0.0}
  {HALO 0.0}
  {RADSOLV 0.4}
  {CHGAL 2.474}
  {BORNC 3.70}
  {PRINTSM12}
End
```

Presence of the SOLVATION SM12 key block key triggers the solvent calculation and does not require additional data. Curly brackets are not part of the key. If no other keywords are present the default is water. With subkeys you can customize various aspects of the model, for instance to specify the type of solute. None of the subkeys is obligatory. Follows a description of the subkeys

---

[3] A.V. Marenich, S.V. Jerome, C.J. Cramer, D.G. Truhlar, *Charge Model 5: An Extension of Hirshfeld Population Analysis for the Accurate Description of Molecular Interactions in Gaseous and Condensed Phases*, Journal of Chemical Theory and Computation 8, 527 (2012) (https://doi.org/10.1021/ct200866d)

[4] F.L. Hirshfeld, *Bonded-atom fragments for describing molecular charge densities*, Theoretica Chimica Acta 44, 129 (1977) (https://doi.org/10.1007/BF00549096)

[5] M. Mantina, R. Valero, C.J. Cramer, D.G. Truhlar, in *CRC Handbook of Chemistry and Physics*, 91st ed. (2010-2011), ISBN13 9781439820773, W.M. Haynes, Ed., CRC Press: Boca Raton, FL, 2010.

[6] W.C. Still, A. Tempczyk, R.C. Hawley, and T. Hendrickson, *Semianalytical treatment of solvation for molecular mechanics and dynamics*, Journal of the American Chemical Society 112, 6127 (1990) (https://doi.org/10.1021/ja00172a038)

[7] D.A. Liotard, G.D. Hawkins, G.C. Lynch, C.J. Cramer, and D.G. Truhlar, *Improved Methods for Semiempirical Solvation Models*, Journal of Computational Chemistry 16, 422 (1995) (https://doi.org/10.1002/jcc.540160405)

`SOLV NAME = solvent` Defines which solvent to be used. Each solvent has parameters set for each of the variables defined below. All of which can be found in the SM12 Ref.[?]. Any variable can be changed with the inclusion of its key. `solvent` can be one of the following:

- ACETICACID
- ACETONITRILE
- ACETOPHENONE
- ANILINE
- ANISOLE
- BENZENE
- BENZONITRILE
- BENZYLALCOHOL
- BROMOBENZENE
- BROMOETHANE
- BROMOFORM
- BROMOOCTANE
- N-BUTANOL
- SEC-BUTANOL
- BUTANONE
- BUTYLACETATE
- N-BUTYLBENZENE
- SEC-BUTYLBENZENE
- T-BUTYLBENZENE
- CARBONDISULFIDE
- CARBONTETRACHLORIDE
- CHLOROBENZENE
- CHLOROFORM
- CHLOROHEXANE
- M-CRESOL
- CYCLOHEXANE
- CYCLOHEXANONE
- DECALIN
- DECANE
- DECANOL
- 1-2-DIBROMOETHANE
- DIBUTYLETHER
- O-DICHLOROBENZENE
- 1-2-DICHLOROETHANE
- DIETHYLETHER
- DIISOPROPYLETHER
- N-N-DIMETHYLACETAMIDE
- N-N-DIMETHYLFORMAMIDE
- 2-6-DIMETHYLPYRIDINE
- DIMETHYLSULFOXIDE
- DODECANE
- ETHANOL
- ETHOXYBENZENE
- ETHYLACETATE
- ETHYLBENZENE
- FLUOROBENZENE
- 1-FLUORO-N-OCTANE
- HEPTANE
- HEPTANOL
- HEXADECANE
- HEXADECYLIODIDE
- HEXANE
- HEXANOL
- IODOBENZENE
- ISOBUTANOL
- ISOOCTANE
- ISOPROPANOL
- ISOPROPYLBENZENE
- P-ISOPROPYLTOLUENE
- MESITYLENE
- METHANOL
- METHOXYETHANOL
- METHYLENECHLORIDE
- N-METHYLFORMAMIDE
- 2-METHYLPYRIDINE
- 4-METHYL-2-PENTANONE
- NITROBENZENE
- NITROETHANE
- NITROMETHANE
- O-NITROTOLUENE
- NONANE
- NONANOL
- OCTANE
- OCTANOL
- PENTADECANE
- PENTANE
- PENTANOL
- PERFLUOROBENZENE
- PHENYLETHER
- PROPANOL
- PYRIDINE
- TETRACHLOROETHENE
- TETRAHYDROFURAN
- TETRAHYDROTHIOPHENEDIOXIDE
- TETRALIN
- TOLUENE
- TRIBUTYLPHOSPHATE
- TRIETHYLAMINE
- 1-2-4-TRIMETHYLBENZENE
- UNDECANE
- WATER
- XYLENE
- 1-2-DIBROMOETHANE_WATER
- 1-2-DICHLOROETHANE_WATER
- BENZENE_WATER
- CARBONTETRACHLORIDE_WATER
- CHLOROBENZENE_WATER
- CHLOROFORM_WATER
- CYCLOHEXANE_WATER
- DIBUTYLETHER_WATER

- DIETHYLETHER_WATER
- ETHYLACETATE_WATER
- HEPTANE_WATER
- HEXANE_WATER
- NITROBENZENE_WATER
- OCTANOL_WATER

`CUST {NAME=solvent}` A solvent with user specified parameters can be used. A user specified name of the solvent `solvent` can be provided, but should not match one of the solvent names listed above. Any un-specified parameters will be set to zero, with the exception of the dielectric constant, which will be set to the dielectric constant of water.

`REF` Index of refraction at 293K (n).

`ACID` Abrahams hydrogen bond acidity (α).

`BASE` Abrahams hydrogen bond basicity (β).

`TENS` Macroscopic surface tension of the solvent at the air interface cal mol^{-1} Å^{-2})(γ).

`EPS` Dielectric constant. If not specified it will be set to the default of water. (ϵ).

`ARO` Square root of the fraction of non-hydrogen atoms in the solvent molecule that are aromatic carbon atoms (carbon aromaticity) (ϕ).

`HALO` Square of the fraction of non-hydrogen atoms in the solvent molecule that are halogens (Electronegative halogenicity)(ψ).

`RADSOLV` Solvent radius, this is added to the van der Waals radius used for the cavity calculation.

`CHGAL` CM5 alpha parameter for Pauling bond order.

`BORNC` Empirical Born constant, see Ref.[?]

`PRINTSM12` Print more information on the SM12 calculation: CDS atom terms, ENP charges.

4.4.3 Quild

The QUILD (Quantum-regions Interconnected by Local Description) program has been developed for enabling calculations through multi-level approaches, in which different computational treatments are used for different regions of the system under study, see the separate Quild manual.

4.4.4 Hybrid Engine (QM/MM, QM/QM')

The Hybrid engine is meant for mechanical or electrostatic embedding and QM/MM (QM/QM') calculations that combine multiple engines. Some of the functionality of QUILD can be achieved using the Hybrid engine. See the separate Hybrid Engine manual.

4.4.5 DIM/QM: Discrete Interaction Model/Quantum Mechanics

The Discrete Interaction Model/Quantum Mechanics (DIM/QM) method facilitates calculating the optical properties of molecules under the influence of a discrete solvent or a metal nanoparticle, see for example Ref.[1]. DIM/QM relies on one of three descriptions of the system: Discrete Reaction Field (DRF), where the atoms interact via induced dipoles and static charge, Capacitance Polarizability Interaction Model (CPIM), where the atoms interact via induced dipoles and induced charges, and Polarizability Interaction Model (PIM), where the atoms interact via induced dipoles only. DRF is best for solvents, CPIM is best for small metal nanoparticles, and PIM is best for large metal nanoparticles.

---

[1] J.L. Payton, S.M. Morton, Justin E. Moore, and Lasse Jensen, *A discrete interaction model/quantum mechanical method for simulating surface-enhanced Raman spectroscopy*, Journal of Chemical Physics 136, 214103 (2012) (https://doi.org/10.1063/1.4722755)

The DRF module is now a part of DIM/QM, so DRF is now a submodule of DIM/QM.

To perform a DIM/QM calculation, two block keys are required. The first is the DIMQM block which activates the DIM/QM module. The parameters for each DIM atom must also be given, and they can be given with either the DIMPAR block which is most convenient for metal nanoparticles, or with the EXTERNALS block which is designed for DRF.

In ADF2019 the bond energies calculated with DIM/QM in ADF have been corrected.

DRF

The Discrete Solvent Reaction Field (DRF) model is a hybrid Quantum mechanical and Molecular Mechanics (QM/MM) model for studying solvation effects on (time-dependent) molecular properties such as dipole moments, excitation energies and (hyper)polarizabilities[7][8][9][10][11]. The classical solvent molecules are represented using distributed atomic charges and polarizabilities.

With the ADF-GUI it is simple to setup a DRF calculation. Another possibility is to use and/or modify a python script that is explained in the example *DIMQM_DRF_plams* (page 521). Both the ADF-GUI and the DRF python script have default settings which can be easily modified if this is required. In the simplest case the user just needs to provide the QM and DRF regions. By default, atomic charges that are used, in case of the ADF-GUI or the DRF python script, for the DRF are computed with LDA functional, DZP basis set and normal numerical quality. Also by default, atomic polarizabilities (computed with the Thole's model) are taken from a inner database including H, C, N, O, F, S, Cl, Br, I atoms[2]. If one does not use the ADF-GUI or the DRF python script, the user has to specify this by hand.

Within the Discrete Solvent Reaction Field model the QM/MM operator is

$$H_{\text{QM/MM}} = \sum_i v_{\text{DRF}}(r_i, \omega) = \sum_i V^{\text{el}}(r_i) + V^{\text{pol}}(r_i, \omega)$$

where the first term, v^{el} , is the electrostatic operator and describes the Coulombic interaction between the QM system and the permanent charge distribution of the solvent molecules. The second term, v^{pol} , is the polarization operator and describes the many-body polarization of the solvent molecules, i.e. the change in the charge distribution of the solvent molecules due to interaction with the QM part and other solvent molecules. The charge distribution of the solvent is represented by atomic point charges and the many-body polarization by induced atomic dipoles at the solvent molecules. The induced atomic dipole at site *s* is found by solving a set of linear equations

$$\mu^{\text{ind}}_{s,\alpha}(\omega) = \alpha_{s,\alpha\beta}\left(F^{\text{init}}_{s,\beta}(\omega) + \sum_{t\neq s} T^{(2)}_{st,\beta\gamma}\mu^{\text{ind}}_{t,\gamma}(\omega)\right)$$

where $\alpha_{s,\alpha\beta}$ is a component of the atomic polarizability tensor at site *s*. The screened dipole interaction tensor is given by

$$T^{(2)}_{st,\beta\gamma} = 3f^{T}_{st}R_{st,\alpha}R_{st,\beta}/R^5_{st} - f^{E}_{st}\delta_{\alpha\beta}/R^3_{st}$$

---

[7] L. Jensen, P.T. van Duijnen and J.G. Snijders, *A discrete solvent reaction field model within density functional theory*, Journal of Chemical Physics 118, 514 (2003) (https://doi.org/10.1063/1.1527010)

[8] L. Jensen, P.T. van Duijnen and J.G. Snijders, *A discrete solvent reaction field model for calculating molecular linear response properties in solution*, Journal of Chemical Physics 119, 3800 (2003) (https://doi.org/10.1063/1.1590643)

[9] L. Jensen, P.T. van Duijnen and J.G. Snijders, *A discrete solvent reaction field model for calculating frequency-dependent hyperpolarizabilities of molecules in solution*, Journal of Chemical Physics 119, 12998 (2003) (https://doi.org/10.1063/1.1627760)

[10] L. Jensen, M. Swart and P.T. van Duijnen, *Microscopic and macroscopic polarization within a combined quantum mechanics and molecular mechanics model*, Journal of Chemical Physics 122, 34103 (2005) (https://doi.org/10.1063/1.1831271)

[11] L. Jensen, Modelling of optical response properties: Application to nanostructures, (http://downloads.scm.com/Doc/jensen.pdf) PhD thesis, Rijksuniversiteit Groningen, 2004.

[2] P.T. van Duijnen and M. Swart, *Molecular and Atomic Polarizabilities: Thole's Model Revisited*, Journal of Physical Chemistry A 102, 2399 (1998) (https://doi.org/10.1021/jp980221f)

where the damping functions $f^T{}_{st}$ and $f^E{}_{st}$ have been introduced, see also?. A smeared-out point charge model[3] is used for short-range damping of the QM/MM operator

$$1/R_{st} \to 1/S_{st} = erf(R_{st})/R_{st}$$

The scaled distance, S_{st} , then replaces the normal distance, R_{st} , in the QM/MM operator.

In order to perform a DRF calculation two types of parameters (model atomic charges and atomic polarizabilities) for each type of atom in the MM part are required. The point charges should represent at least the permanent molecular dipole moment, and the distributed atomic polarizabilities the full molecular polarizability tensor. The atomic charges can straightforward be obtained using e.g. Multipole Derived Charges (MDC) [See *section on MDC* (page 287) and the distributed polarizabilities by adopting standard parameters or refitting them to match the calculated polarizability tensor??. This allows for a simple procedure to obtain the solvent model parameters which subsequently can be used in the DRF calculation.

DRF gradients can be calculated by using the DRF2 keyword in the DIMQM block. This is an expert option and should be used with great caution. The DRF2 keyword uses DRF but with Gaussian screening like that of the PIM method. It also allows for input of parameters by atom type like the PIM method. The user must specify a radius and charge for the atom type like seen below.

```
DIMPAR
  O
    rad=0.9028
    char=-0.6690
    STATIC
  SUBEND
  H
    rad=0.2479
    char=0.3345
    STATIC
  SUBEND
  Ag
    rad=1.445
    char=0.0000
  SUBEND
END
```

Here, the STATIC keyword is used to tell DIM/QM to ignore the frequency dependent response for those atom types.

Surface-enhanced response properties

In addition to the calculation of solvation effects on response properties, DIM/QM can also be used for simulating nonlinear optical properties of molecules near metal surfaces of a nanoparticle. A TDDFT description of the molecule is coupled to an electrodynamical treatment of the nanoparticle, including electrostatics and polarization of the atoms. In this way DIM/QM can be used to study surface-enhance Raman scattering (SERS)?, surface-enhanced Raman optical activity (SEROA)[4], and plasmon-enhanced photochemistry.

Starting from ADF2018 the Polarizability Interaction Model (PIM) can be used to calculate damped first hyperpolarizabilities β and damped second hyperpolarizabilities γ. Different types of β-tensors can be calculated, such as STATIC, OPTICALR, EOPE, and SHG. This allows for simulation of surface-enhanced hyper-Raman scattering (SEHRS)[5]. All the (sub)keywords for regular damped β, such as BETA and QUADRATIC, can also be used. Also different types of

---

[3] L. Jensen, P.-O. Astrand, A. Osted, J. Kongsted and K.V. Mikkelsen, *Polarizability of molecular clusters as calculated by a dipole interaction model*, Journal of Chemical Physics 116, 4001 (2002) (https://doi.org/10.1063/1.1433747)

[4] D.V. Chulhai and L. Jensen, *Simulating Surface-Enhanced Raman Optical Activity Using Atomistic Electrodynamics-Quantum Mechanical Models*, Journal of Physical Chemistry A 118, 9069 (2014) (https://doi.org/10.1021/jp502107f)

[5] Z. Hu, D.V. Chulhai, and L. Jensen, *Simulating Surface-Enhanced Hyper-Raman Scattering Using Atomistic Electrodynamics-Quantum Mechanical Models*, Journal of Chemical Theory and Computation 12, 5968 (2016) (https://doi.org/10.1021/acs.jctc.6b00940)

γ-tensors can be calculated, such as STATIC, EFIOR, OKE, IDRI, EFISHG, THG and TPA. This allows for simulation of plasmon-enhanced two-photon absorption (PETPA). Except CUBIC, all the other (sub)keywords for regular damped γ can also be used.

Input options

```
DIMQM
  <DRF|PIM|CPIM>
  NOPOL
  NOCHAR
  NOCROSS
  DDA
  FULLGRID
  LOCALFIELD <Resp|All|Static>
  EFIELD x y z
  SCREEN <ERF|EXP|ESP|GAU|NONE> {length}
  :: FREQUENCY-DEPENDENT PARAMETERS
  FREQUENCY
  OM_H value
  OM_C value
  OM_O value
  OM_N value
  :: CONTROL OVER SOLVER
  ALGORITHM <BEST|DIRECT|BRUTE|SINGLE|MULTI>
  TOLERANCE tol
  NITER iterations
  VOLUME vol_in_nm^3
  MULTIPLIER aterm bterm cterm
  GRID <COARSE|MEDUIM|FINE>
  :: GRADIENT OPTIONS
  FORCEFIELD
  CHEMBOND qmindex dimindex
  COORDDEPEND
  CHEMISORPTION
  COORDPAR atomtype e0 e1 r0 r1 CNmax Rmax Rmin
  CHEMPAR atomtype e0 e1 r0 r1 cutoff
  PROJECTIONMATRIXPOINTS <ALL|CUTOFF radius|OFF>
  :: OUTPUT CONTROL
  PRINTATOMICDIPOLES
  PRINTLJPAR
  DEBUG
  LOCALDIM
  LOCALDIRECT
  DIMBUFFER
  QMBUFFER
  CUTOFFDIST
END
```

`<DRF|CPIM|PIM>` DIM/QM relies on one of three descriptions of the system: Discrete Reaction Field (DRF), where the atoms interact via induced dipoles and static charge, Capacitance Polarizability Interaction Model (CPIM), where the atoms interact via induced dipoles and induced charges, and Polarizability Interaction Model (PIM), where the atoms interact via induced dipoles only. DRF is best for solvents, CPIM is best for small metal nanoparticles, and PIM is best for large metal nanoparticles. One and only one of these three keys must be included in every DIM/QM calculation. However, in ADF2019 there is also the expert DRF2 possibility, which allows for DRF gradients but should be used cautiously.

`NOPOL` The NOPOL key turns off the polarization terms, and thus all induced dipoles are zero. This key is only valid for DRF or CPIM calculations.

NOCHAR The NOCHAR key turns off the charge terms, and thus all induced or static charges are zero. This key is only valid for DRF or CPIM calculations.

NOCHOSS The NOCROSS key turns off the charge-dipole interactions. This key is only valid for CPIM calculations.

DDA By default, the dipole-dipole, charge-dipole and charge-charge interactions are screened to take into account that atoms are not point charges. The DDA key will turn off this screening so that the results can be compared directly to the discrete dipole approximation (DDA).

FULLGRID This is used in conjunction with the frozen density approximation.

LOCALFIELD <Resp|All|Static> Default is All, when no option is specified. When the molecule interacts with a (for example) metal nanoparticle, there are two types of interactions: the image field and the local field. The image field is caused by the dipoles induced into the nanoparticle by the molecule's electron density. This is always taken into account in a DIM/QM calculation. The local field arises by direct interactions of the nanoparticle with an external field. Addition of the LOCALFIELD key causes the DIM/QM calculation to include this effect, but by default this is not included in a DIM/QM calculation. In ADF2016 the correct behavior with local fields for the electric dipole-dipole, electric dipole-quadrupole, and electric dipole-magnetic dipole response tensors in the velocity gauge is implemented. This allows for correct calculation of surface-enhanced Raman optical activity (SEROA) (Ref.[?]) and plasmonic circular dichroism (Ref.[6]).

Default option should be used in most situations as other options are expert options. In effort to speed up calculation times, one can request to just calculate the Local Fields during the response section of the calculation by using the Resp keyword after local fields. All will calculate the local field for both ground state calculations and the response calculations, while static will calculate the local field during the ground state only.

EFIELD x y z The EFIELD key is used to include an external static electric field in the vector x y z. Internally, the static charges used in a DRF calculation use AMS's System%ElectrostaticEmbedding, and therefore use of the AMS System%ElectrostaticEmbedding is not allowed in combination with the DIMQM block. This key is included so the user may include an external static electric field.

SCREEN <ERF|EXP|ESP|GAU|NONE> {length} The SCREEN key indicates what functional form is used to screen the interactions between each DIM atom and the QM density. The choices are ERF (error function), EXP (exponential), ESP (error function for potential operator only), GAU (Gaussian), or NONE. For CPIM and PIM the default is GAU; for DRF, the default is ESP. In all cases, the default screening length is 1.0, but this may be changed with the optional length parameter.

FREQUENCY The FREQUENCY key turns on frequency-dependent parameters.

OM_[HCON] value The OM_H, OM_C, OM_O, and OM_N keys provide the resonance frequency (in atomic units) for the elements H, C, O and N, respectively. These keys are only for use with DRF and only when the older EXTERNALS block is used.

ALGORITHM <BEST|DIRECT|BRUTE|SINGLE|MULTI> DIM/QM can choose between several solver algorithms. The DIRECT method solves the linear system of equations directly with a LAPACK routine; this should be considered the most robust method, but scales poorly with the number of atoms ($O(N^3)$ where N is the number of atoms in the system). The other three methods use an iterative technique, The BRUTE method (brute force) takes into account all atoms in the matrix-vector multiply step, and scales as $O(N^2)$. The SINGLE method uses the single-level cell-multipole-method (CMM), wherein dipoles that are spatially similar are collected into a multipole moment which effectively reduces the system size. This also scales as $O(N^2)$ but with a lower prefactor than BRUTE. The MULTI method uses the multi-level cell-multipole-method, which uses larger and large multipole the farther apart the dipoles are. This is the fastest method and scales as O(N log N).

Due to technical limitations, CPIM can only use DIRECT. Further, depending on the system size DIRECT or SINGLE may be more efficient than MULTI. To simplify choosing the solving algorithm, there is a BEST option that chooses the best algorithm for the particular system. BEST is the default option for algorithm.

---

[6] D.V. Chulhai and L. Jensen, *Plasmonic Circular Dichroism of 310- and α-Helix Using a Discrete Interaction Model/Quantum Mechanics Method*, Journal of Physical Chemistry A 119, 5218 (2015) (https://doi.org/10.1021/jp5099188)

`TOLERANCE tol` The TOLERANCE key allows the user to specify a tolerance for the iterative solver. By default the tolerance is based on ADF's INTEGRATION key. This has no effect with the DIRECT solver.

`NITER iterations` The NITER key allows the user to specify the maximum number of iterations for the iterative solver. By default this is MAX(N/100, 200) where N is the number of DIM atoms.

`VOLUME vol_in_nm3` The VOLUME key is used to specify the DIM system volume. The volume is used to determine how to partition the system for the cell multipole method (ALGORITHM options SINGLE or MULTI), and is also use to determine the scattering efficiencies for frequency-dependent polarizability calculations. The volume does not need to be supplied; if it is missing, it will be calculated based on each atom's radius and the MULTIPLIER key.

`MULTIPLIER aterm bterm cterm` An efficient way to get a approximation for the volume of the system is to sum the volume of each atom in the system, modified to account for the space between the atoms. This is done by modifying the atomic radius by a formula that takes into account the number of DIM atoms so that the effective radius changes with surface-to-bulk ratio. This formula is given by

$r_{\mathrm{eff}} = -ar/\mathrm{N}^{\mathrm{b}} + c$

where r is the atomic radius, r_{eff} is the effective radius, N is the number of atoms, and the a, b, and c terms are the three parameters defined by the MULTIPLIER key. If the MULTIPLIER key is missing, the default values are 0.7, 0.5, and 1.13, respectively.

`GRID <COARSE|MEDIUM|FINE>` In the cell multipole method (ALGORITHM options SINGLE or MULTI), a certain number of the closest atom interactions must be calculated explicitly. The GRID key controls how many atoms must be calculated this way, with COARSE being the least and FINE being the most. COARSE will be the fastest to calculate but may be numerically unstable. FINE is slowest to calculate but is the most stable. If the GRID key is missing, the default is MEDIUM.

`FORCEFIELD` The FORCEFIELD key indicates that the DIM/QM calculation will include the DIM/QM force field. Currently the only maintained potential is the Lennard-Jones 12-6 potential (see Ref.[?]). This key is required to perform a DIM/QM geometry optimization and vibrational frequencies. By default, the DIM/QM force field is not included into the calculation. DRF gradients are not implemented.

Currently, DIM/QM geometry optimizations must be done in Cartesian coordinates which is specified in the GEOMETRY block. The user should be aware that ADF's default convergence criterion for a geometry optimization are relatively low, thus it is strongly suggested for a DIM/QM calculation to set the numerical integration quality (BeckeGrid) to good and change the convergence criterion of the max gradient to 1E-4. It is not possible to do geometry optimization with DRF.

`COORDDEPEND` The COORDDEPEND key indicates that the DIM/QM force field will be coordination dependent. This only effects the Lennard-Jones parameters for DIM atoms (see Ref.[?]). By default, the DIM/QM force field is not coordination dependent.

`CHEMISORPTION` The CHEMISORPTION key will include chemisorption corrections for all atoms that have chemisorption parameters within a given cutoff radius. By default, the DIM/QM force field does not included chemisorption corrections.

`COORDPAR atomtype e0 e1 r0 r1 CNmax Rmax Rmin` The COORDPAR key allows the user to add additional coordination dependent parameter for a selected element type. atomtype specifies the element type (i.e., Ag for silver) for the given parameter set. e0, e1, r0, and r1 are the coordination dependent Lennard-Jones parameters; see Ref.[?]) for more details. The coordination numbers of the DIM atoms are computed as an effective coordination number. This scheme requires a maximum and minimum cutoff distances, Rmax and Rmin respectively, and a maximum coordination number, CNmax. All parameters need to be in atomic units.

`CHEMPAR atomtype e0 e1 r0 r1 CUTOFF` The CHEMPAR key allows the user to add additional chemisorption dependent parameter for a selected element type. atomtype specifies the element type (i.e., N for nitrogen) for the given parameter set. e0, e1, r0, and r1 are the coordination dependent Lennard-Jones parameters; see Ref.[?]) for more details. The code determines if there is a chemical bond by a cutoff distance parameter, CUTOFF. If the QM-DIM bond is within the cutoff, the code uses the chemisorption parameter; otherwise, the code uses the standard parameter set. All parameters need to be in atomic units.

`CHEMBOND qmindex dimindex` The CHEMBOND key indicates that there will a chemisorption correction used for the bond between the specified QM and DIM atoms. The user may repeat the CHEMBOD key up to 50 times to specify up to 50 different chemical bonds for the force field. The qmindex is an integer based on the order of atoms in the ATOMS block; i.e. the fifth QM atom in the ATOMS block would have qmindex = 5. The dimindex is the same but corresponds to the DIM atom involved in the bond. This key should only be used when the CHEMISORPTION key is also specified. When using CHEMBOND, the cutoff distance parameter for chemisorption correction parameter key will be ignored. It is suggested to use CHEMBOND if the user is generating a potential energy surface with a chemisorbed QM system.

`PROJECTIONMATRIXPOINTS <ALL|CUTOFF radius|OFF>` The PROJECTIONMATRIXPOINTS key specifies what DIM atoms to include for the projection matrix when removing rigid motions out of the gradient. The methods available are ALL, CUTOFF, or OFF. The ALL option causes PROJECTIONMATRIXPOINTS to include all DIM atoms. OFF will turn off the removal of rigid motions. CUTOFF includes any DIM atom points within a cutoff radius from the center of mass of the QM system to the DIM atom points and requires a cutoff radius to be given in Angstrom. This key only applies to a geometry optimization. If the PROJECTIONMATRIXPOINTS key is not given, the option CUTOFF with a cutoff radius of 25.4 Angstrom is assumed.

`PRINTATOMICDIPOLES` The PRINTATOMICDIPOLES key causes all the induced dipole moments of each DIM atom to be printed at the conclusion of each SCF cycle and each RESPONSE or AORESPONSE polarizability calculation. Because DIM/QM is typically used with many thousands of atoms, this can result in a large output file, but they may be useful for debugging purposes or to calculate electric fields. By default these are not printed.

`PRINTLJPAR` The PRINTLJPAR key specifies that all Lennard-Jones parameters used for the calculation will be printed in the output file. The QM atoms' Lennard-Jones parameters are also printed with the DEBUG key.

`DEBUG` The DEBUG key will print out extra information in the process of the calculation.

`LocalDIM` LocalDIM performs the calculation fully for a specified CutOffDist (in Bohr) but ignores the image dipoles from atoms outside of this distance. This option should converge to the full simulation and the distance should be chosen wisely. Must have LocalDirect keyword as well.

`LocalDirect` Tells LocalDIM calculation to use the direct solver for the DIM dipoles. This solver is generally much faster than the other solvers available when the system is small. Currently, LocalDIM only works with the intention of using this faster solver, so this option must be used with LocalDIM

`DIMBUFFER` Default value = 0.85. This keyword tells DIM how to decide for DRF2 calculations which atoms are in the same molecular grouping. This grouping decides if the charges polarize other atomic dipoles. For example, the charge for atoms in the same molecule will not polarize each other but they will polarize atomic dipoles in other molecules. This value should be adjusted based on the XYZ you supply and checked in the output file to ensure proper grouping behavior.

`QMBUFFER` Default value = 0.85. This keyword tells DIM how to decide if the QM atoms are "bonded" to one another to determine the force field parameters needed calculations requiring DIM/QM gradient terms. As with the DIMBUFFER keyword, this value should be adjusted for each system and checked in the output file to ensure proper behavior.

`CutOffDist` Distance in Bohr that determines if the image dipoles of a DIM atom is included in a LocalDIM calculation. Any DIM atom that is inside the supplied distance from any QM atom will be fully included in the calculation.

DRF atomic parameters

The EXTERNALS block key controls the input data for the MM atoms. The EXTERNALS block is designed for DRF calculations. For each MM atom the following data are required:

```
EXTERNALS
 atm num grp-nam grp-num, char, x, y, x, pol
 ...
 GROUP
 {...}
end
```

atm Type of atom, i.e., H, O, …

num number of atoms (optional)

grp-nam Name of the group to which the atom belongs

grp-num Number of the group to which the atom belong

char atomic charge (in atomic units)

x x-coordinate

y y-coordinate

z z-coordinate

pol atomic polarizability (in atomic units)

GROUP Indicates the end of group

The separation of molecules into GROUP's are important. Since in the many-body polarization operator only intermolecular interactions, i.e. only interaction between sites which do not belong to the some group, are included. Therefore, it is important that the combined string (grp-nam + grp-num) is unique for each GROUP.

An example of a EXTERNALS block for two water molecules:

```
EXTERNALS
 O 4 water 2, -0.6690, -11.380487, -11.810553, -4.515226, 9.3005
 H 5 water 2,  0.3345, -13.104751, -11.837669, -3.969549, 0.0690
 H 6 water 2,  0.3345, -10.510898, -12.853311, -3.320199, 0.0690
 GROUP
 O 7 water 3, -0.6690,  -1.116350,   9.119186, -3.230948, 9.3005
 H 8 water 3,  0.3345,  -2.822714,   9.717033, -3.180632, 0.0690
 H 9 water 3,  0.3345,  -0.123788,  10.538199, -2.708607, 0.0690
 GROUP
 {...}
end
```

DIM atomic parameters

In this block, the parameters for the DIM atoms are defined.

```
DIMPAR
  Element
    RAD val
    POL val
    CAP val
    CHAR val
```

(continues on next page)

(continued from previous page)

```
    OM val
    OM1 val
    OM2 val
    GM1 val
    GM2 val
    SIZE val
    BOUND val
    STATIC
    EXP /path/to/experimental/dielectric/file
    DRUDE plasma damping {EV}
    FERMI val
    <LRTZ|LRTZ1> osc res damp {EV}
    LRTZ2 osc pls res damp {EV}
    LRTZ3 pls res damp {EV}
  SUBEND
  XYZ
    {/absolute/path/to/coordinates.xyz}
    {natoms
     elem x.xxx y.yyy z.zzz
     elem x.xxx y.yyy z.zzz
     elem x.xxx y.yyy z.zzz
     ...}
  SUBEND
  LCLFLD /absolute/path/to/localfields.txt
    {freq1 freq2 freq3 ..
     .....
    }
  GROUP groupname
    natoms
     elem x.xxx y.yyy z.zzz {pol=pol} (cap=cap} {char=char} {..}
     elem x.xxx y.yyy z.zzz {pol=pol} (cap=cap} {char=char} {..}
     elem x.xxx y.yyy z.zzz {pol=pol} (cap=cap} {char=char} {..}
     ...
  SUBEND
END
```

Element Within the DIMPAR block, you will need a sub-block that defines the parameters for each element that is in your DIM system. You will need to replace 'Element' with the element you are assigning parameters to, as in:

```
Ag
 ...
SUBEND
```

if you are assigning parameters to Ag. Note that the first letter MUST be capitalized and the second MUST be lowercase.

RAD val RAD specifies the atomic radius in the unit defined by the input file. RAD is required for all PIM calculations, all calculations with ALGORITHM options SINGLE or DIRECT, and all frequency-dependent calculations where the AORESPONSE LIFETIME key is given.

POL val POL specifies the polarizability parameter (in a.u.) used in DRF or CPIM.

CAP val CAP specifies the capacitance parameter (in a.u.) used in CPIM.

CHAR val CHAR specifies the atomic charge (in a.u) used in DRF.

OM val OM specifies the resonance frequency (in a.u) used in DRF. This replaces the OM_[HCON] key in the DIMQM block.

OM1 val OM1 specifies the ω_1 parameter (in a.u) used in CPIM.

OM2 val OM2 specifies the ω_2 parameter (in a.u) used in CPIM.

GM1 val GM1 specifies the γ_1 parameter (in a.u) used in CPIM.

GM2 val GM2 specifies the γ_2 parameter (in a.u) used in CPIM.

SIZE val SIZE specifies the size-dependent parameter used in CPIM.

STATIC Calculate the response properties for this atom type as static during a frequency dependent calculation.

EXP /absolute/path/to/experimental/dielectric/file In PIM, the atomic polarizabilities are calculated from the dielectric constant. If you have access to the experimental dielectric constant, this may be supplied directly to DIM/QM. The values will be splined, so it is not necessary to ensure that each frequency at which you are calculating be in the file. DIM/QM expects the file to be formatted with the wavelength (in nm) in the first column, the real part of the dielectric in the second column, and the imaginary part of the dielectric in the third column. All other columns that may exist will be ignored, as well as lines beginning with the hash (#) symbol.

BOUND val A Drude function is typically written as

$$\epsilon_\infty - \frac{\omega_p^2}{\omega(\omega + i\gamma)}$$

with the second term being the Drude function, and the first term accounting for bound electrons. For a conductor with no bound electrons, $\epsilon_\infty = 1$ which is the default value for BOUND. To account for bound electrons you may set BOUND to a value greater than 1. This key only affects PIM.

DRUDE plasma damping {EV} The formula for a Drude function is

$$\epsilon_\infty - \frac{\omega_p^2}{\omega(\omega + i\gamma)}$$

where ϵ_∞ represents the bound electrons (as discussed for BOUND), ω_p (plasma) is the plasma frequency, and γ (damping) is the damping parameter (decay rate). Optionally, EV may be added to specify the values be read in units of electron volts, otherwise they are read in units of a.u. This key only affects PIM.

FERMI val The FERMI key is used to specify a Fermi velocity (in m/s) so that the Drude function may be size-corrected using a modified Drude function:

$$\epsilon_\infty - \frac{\omega_p^2}{\omega(\omega + i(\gamma + v_\mathrm{fermi}/R_\mathrm{eff}))}$$

where +$\mathrm{v_{fermi}}$ is the Fermi velocity and $\mathrm{R_{eff}}$ is the effective nanoparticle radius. This can also be used in conjunction with EXP and DRUDE to size-correct experimental dielectric parameters. This key only affects PIM:

```
<LRTZ|LRTZ1> osc res damp {EV}
LRTZ2 osc pls res damp {EV}
LRTZ3 pls res damp {EV}
```

There are three forms of the Lorentzian function seen in the literature:

$$\sum_n \frac{f_n \Omega_{0,n}^2}{\Omega_{0,n}^2 - \omega^2 - i\Gamma_n\omega}$$

$$\sum_n \frac{f_n \omega_p^2}{\Omega_{0,n}^2 - \omega^2 - i\Gamma_n\omega}$$

$$\sum_n \frac{\Omega_{p,n}^2}{\Omega_{0,n}^2 - \omega^2 - i\Gamma_n\omega}$$

where $\Omega_{0,n}$ (res) is a bound electron resonance frequency, f_n (osc) is a bound electron oscillator strength, Γ_n (damp) is a bound electron excited state decay rate (or damping parameter), ω_p (pls) is the free electron plasma frequency, and $\Omega_{p,n}$ (pls) is the bound electron plasma frequency. You may choose the Lorentzian for against which your parameters were parametrized. The top form is LRTZ1, the middle is LRTZ2, and the bottom is LRTZ3. Because LRTZ1 is the most common, it is also aliased as LRTZ. Optionally, EV may be added to specify the values be read in units of electron volts, otherwise they are read in units of a.u. This key only affects PIM. You may give any of the form of the LRTZ key up to 50 times supply up to 50 Lortenzian functions to make a Drude-Lorentz function.

XYZ The XYZ sub-block is where the DIM atom coordinates are given. Two methods of supplying coordinates are allowed.

In-File Coordinates As an example of how to supply coordinates in-file, imagine you wish to calculate a Au dimer system on the Z-axis. You might define your coordinates as:

```
XYZ
 2
 Au 0.0 0.0 0.0
 Au 0.0 0.0 3.0
SUBEND
```

The first line gives the number of atoms to follow. Every line after that contains the element in the first column (first letter MUST be capitalized, second MUST be lowercase), then the x-component, then the y-component, then the z-component. You may not number the atoms.

External File Coordinates When the DIM system size becomes large, it is often more convenient to keep the DIM coordinates in a separate file. The XYZ block would then look like:

```
XYZ
 /absolute/path/to/coordinates.xyz
SUBEND
```

Note that you MUST include the absolute path to your file and the file name must end with `.xyz`.

The .xyz file is set up identically to the in-file table, except that there is a comment line between the number of atoms and the first atom. The .xyz file for our dimer system would be:

```
2
A gold dimer (this line will be ignored)
Au 0.0 0.0 0.0
Au 0.0 0.0 3.0
```

LCLFLD /absolute/path/to/localfields.txt Text file of the local field induced dipoles that can be read in to reduce calculation time. The file must be in the following format:

```
freq1 freq2 freq3  ...
1
X
Atom1R-DipoleX    Atom1R-DipoleY    Atom1R-DipoleZ
Atom1I-DipoleX    Atom1I-DipoleY    Atom1I-DipoleZ
Atom2R-DipoleX    Atom2R-DipoleY    Atom2R-DipoleZ
Atom2I-DipoleX    Atom2I-DipoleY    Atom2I-DipoleZ
...
Y
Atom1R-DipoleX    Atom1R-DipoleY    Atom1R-DipoleZ
Atom1I-DipoleX    Atom1I-DipoleY    Atom1I-DipoleZ
Atom2R-DipoleX    Atom2R-DipoleY    Atom2R-DipoleZ
Atom2I-DipoleX    Atom2I-DipoleY    Atom2I-DipoleZ
...
```

Here freq1 freq2 freq3 … are the frequencies that are being used in the response calculation and 1 tells the program that the following dipoles are to be used with the first frequency. X and Y tell the program that these are the induced dipoles from perturbations in the X and Y Cartesian direction respectively.

`GROUP groupname` The GROUP sub-block is where the DIM atom coordinates and parameters are given. Combines the XYZ subblock and Element subblock. A (unique) groupname is required (maximum 10 characters).

Example for a water molecule:

```
GROUP water1
3
O  0.00000  0.00000  0.59372  pol=5.7494  char=-0.6690
H  0.00000  0.76544 -0.00836  pol=2.7927  char=0.3345
H  0.00000 -0.76544 -0.00836  pol=2.7927  char=0.3345
SUBEND
```

The first line gives the number of atoms to follow. Every line after that contains the element in the first column (first letter MUST be capitalized, second MUST be lowercase), then the x-component, then the y-component, then the z-component. You may not number the atoms. At the end of the line for each element you can specify the DIM/QM parameters, as in the format in the example. The DIM/QM parameters can be 'POL', 'CAP', 'CHAR', 'OM1', 'OM2', 'GM1', 'GM2', 'SIZE', 'RAD', 'SPILLOUT', 'BOUND', 'DRUDE', 'FERMI', 'LRTZ', or 'EXP'.

4.4.6 QM/FQ: Quantum Mechanics/Fluctuating Charges (and Fluctuating Dipoles)

The Quantum Mechanics/Fluctuating Charges (QM/FQ) and QM/FQ and Fluctuating Dipoles (QM/FQFμ) methods are multiscale models designed to describe the properties of a chemical system perturbed by the presence of its molecular environment, where the latter described using a polarizable classical force-field, see for example Refs.[1][2][3][4][5]. The QM/FQ model has been applied to the calculation of spectroscopic properties of molecules in aqueous solution and different solvents (see[6]), whereas QM/FQFμ has only been applied to aqueous solutions. In QM/FQ each atom of the environment (e.g. a solvation shell) is endowed with a fluctuating charge (FQ) which can vary in response to the electrostatic potential generated by the solute, whereas in QM/FQFμ an additional source of polarization (an induced dipole moment) is placed on each atom. The FQ and FQFμ charges can be constrained such that each molecule within the environment remains electrically neutral. The energy functional of the FQ and FQFμ systems can be written as:

$$F(q,\lambda) = \sum_{\alpha,i} q_{\alpha,i}\chi_{\alpha,i} + 1/2 \sum_{i\alpha j\beta} q_{\alpha,i} T^{qq}_{\alpha i,\beta j} q_{\beta,j} + \sum_{\alpha} \lambda_\alpha \Big(\sum_i q_{\alpha,i} - Q_\alpha\Big)$$

$$F(q,\mu,\lambda) = \frac{1}{2}\sum_{i\alpha j\beta} q_{i\alpha}\mathbf{T}^{qq}_{i\alpha,j\beta} q_{j\beta} + \frac{1}{2}\sum_{ij} \mu^{\dagger}_{i\alpha}\mathbf{T}^{\mu\mu}_{i\alpha,j\beta}\mu_{j\beta} + \sum_{ij} q_{i\alpha}\mathbf{T}^{q\mu}_{i\alpha,j\beta}\mu^{\dagger}_{j\beta} + \sum_{i\alpha} q_{i\alpha}\chi_{i\alpha} + \sum_{\alpha}\lambda_\alpha\left[\sum_i q_{\alpha i} - Q_\alpha\right]$$

where $q_{\alpha,i}$ and $\mu_{\alpha,i}$ are the i-th FQ charge and the i-th dipole belonging to the α-th molecule, $\chi_{\alpha,i}$ is the electronegativity of each atom, λ_α is the Lagrange multiplier associated with the α-th molecule, whose purpose is to fix the total charge to be Q_α. The matrix $T^{qq}_{\alpha i,\beta j}$ describes the interaction kernel between the charges: while the diagonal elements are related to the self-interaction through the chemical hardnesses eta, the off-diagonal elements may be specified through different

---

[1] T. Giovannini, F. Egidi, C. Cappelli, *Molecular spectroscopy of aqueous solutions: a theoretical perspective*, Chemical Society Reviews, 49, 5664 (2020) (https://doi.org/10.1039/C9CS00464E)

[2] C. Cappelli, *Integrated QM/polarizable MM/continuum approaches to model chiroptical properties of strongly interacting solute–solvent systems*, International Journal of Quantum Chemistry, 116, 1532 (2016) (https://doi.org/10.1002/qua.25199)

[3] T. Giovannini, F. Egidi, C. Cappelli, *Theory and algorithms for chiroptical properties and spectroscopies of aqueous systems*, Physical Chemical Chemical Physics, 22, 22864 (2020) (https://doi.org/10.1039/D0CP04027D)

[4] T. Giovannini, A. Puglisi, M. Ambrosetti, C. Cappelli, *Polarizable QM/MM approach with fluctuating charges and fluctuating dipoles: the QM/FQFμ model*, Journal of Chemical Theory and Computation, 15, 2233-2245 (2019) (https://doi.org/10.1021/acs.jctc.8b01149)

[5] T.Giovannini, R. R. Riso, M. Ambrosetti, A. Puglisi, C. Cappelli, *Electronic transitions for a fully polarizable QM/MM approach based on fluctuating charges and fluctuating dipoles: linear and corrected linear response regimes*, The Journal of Chemical Physics, 151, 174104 (2019) (https://doi.org/10.1063/1.5121396)

[6] M.Ambrosetti, S. Skoko, T. Giovannini, C. Cappelli, *Quantum Mechanics/Fluctuating Charge Protocol to Compute Solvatochromic Shifts*, Journal of Chemical Theory and Computation (2021) (https://doi.org/10.1021/acs.jctc.1c00763)

formulations. The $T^{q\mu}_{i\alpha,j\beta}$ and $T^{\mu\mu}_{i\alpha,j\beta}$ matrices describe the interaction kernel between charges and dipoles and between the dipoles, respectively. The electric variables (the FQ charges and the dipoles) can be obtained by minimizing the energy functional with respect to the charges, the dipoles and the Lagrange multipliers, which leads to the following set of linear equations:

$$\left(\begin{array}{cc|c} \mathbf{T}^{qq} & \mathbf{1}_\lambda & \mathbf{T}^{q\mu} \\ \mathbf{1}_\lambda^\dagger & \mathbf{0} & \mathbf{0} \\ \hline -\mathbf{T}^{q\mu^\dagger} & \mathbf{0} & \mathbf{T}^{\mu\mu} \end{array}\right) \left(\begin{array}{c} \mathbf{q} \\ \lambda \\ \hline \mu \end{array}\right) = \left(\begin{array}{c} -\chi \\ \mathbf{Q} \\ \hline \mathbf{0} \end{array}\right) \qquad \Rightarrow \qquad \mathbf{DQ}_\lambda = -\mathbf{C}_Q$$

where $\mathbf{1}_\lambda$ is a rectangular matrix which accounts for the Lagrangians, $\mathbf{C}_Q$ is a vector containing atomic electronegativities and total charge constraints. $\mathbf{Q}_\lambda$ is a vector containing charges, dipoles and Lagrange multipliers.

The coupling of the FQ/FQFμ models with a QM Hamiltonian can be done by introducing the QM/FQ(Fμ) interaction operator as

$$H_{\text{QM/FQ}} = \sum_i^N q_i V[\rho](r_i), H_{\text{QM/FQF}\mu} = \sum_i^N q_i V[\rho](r_i) - \sum_i^N \mu_i E[\rho](r_i),$$

where N is the number of atoms, q_i and μ_i are the i-th FQ charge and dipole located at position r_i, respectively. $V[\rho](r_i)$ and $E[\rho](r_i)$ are the electric potential and field generated by the QM system on the same point. The introduction of the QM/FQ interaction leads to a modified set of linear equations for the FQ charges, i.e.

$$DQ_\lambda = -C_Q - R[\rho]$$

where R is an array collecting the electric potential and field generated by the QM electrons and nuclei at the position of each atom. The FQ charges (and dipoles) are thus determined self-consistently along with the ground-state density. Since the charges (and dipoles) depend on the QM density, explicit terms also appear within response equations that are solved to simulate spectroscopic and excited-state properties of the QM system.

Starting from AMS2021.103 a screening is included for the interaction between MM atoms and the QM density, to avoid unstable results in case numerical integration points are accidentally close to MM atoms. The screened $1/r_{ij}$ has the form:

$$\text{ERF: } = \frac{erf(ar_{ij})}{r_{ij}}$$

$$\text{EXP: } = \frac{(1 - exp(-ar_{ij}))^2}{r_{ij}}$$

$$\text{GAUS: } = \frac{erf(r_{ij}/a)}{r_{ij}}$$

where a can be changed with the QMSCREENFACTOR keyword.

Coupling with FDE

Starting from AMS2022 the coupling between *Frozen Density Embedding (FDE)* (page 127) and FQ is implemented (see Ref.[7]). When a FDE calculation is setup, the FQ(Fμ)vironment atoms respond to both layers of the FDE embedding.

In the case of an excited-state calculation performed with the Time-Dependent DFT method (TDDFT), the atoms belonging to the frozen density layer do not normally respond dynamically to the external field. This can create an imbalance because both the non-frozen QM layer and the fluctuating charge (and dipole) layer both do. As a very cost-effective way to include the response of the frozen-density layer into the calculation is to use the FQ(Fμ) model to estimate it (see Ref.[?] for more details) provided the parameters defining the molecular entities in the frozen layer are available.

This can be accomplished by simply including the FDERESP keyword in the input (see below).

---

[7] F.Egidi, S. Angelico, P. Lafiosca, T. Giovannini, C. Cappelli, *A polarizable three-layer frozen density embedding/molecular mechanics approach*, The Journal of Chemical Physics, 154, 164107 (2021) (https://doi.org/10.1063/5.0045574)

Input options

The below scheme presents all keywords that can be used in a QM/FQ(Fμ) calculation. Further down below you can find a detailed explanation on how to specify the coordinates of the FQ(Fμ) atoms.

```
QMFQ
   AtomType
      Alpha float
      Charge float
      Chi float
      Eta float
      Symbol string
   End
   Coords # Non-standard block. See details.
      ...
   End
   FDERESP Yes/No
   Forcefield [FQ | FQFMU]
   Frozen Yes/No
   Kernel [OHNO | COUL | GAUS]
   MolCharge float
   QMSCREEN [ERF | EXP | GAUS | NONE]
   QMSCREENFACTOR float
End
```

QMFQ

Type Block

Description Block input key for QM/FQ(FMu).

AtomType

Type Block

Recurring True

Description Definition of atomic types in MM environment

Alpha

Type Float

Description Polarizability of FQFMU atom

Charge

Type Float

Description MM fixed charge (non-polarizable only)

Chi

Type Float

Description Electronegativity of FQ atom

Eta

Type Float

Description Chemical Hardness of FQ atom

`Symbol`

Type String

Description Symbol associated with atom type

`Coords`

Type Non-standard block

Description Coordinates and fragment information (FQ only)

`FDERESP`

Type Bool

Default value No

Description In response calculations (TD), the polarization contribution of the FDE part is introduced at the FQ level [See F. Egidi et al. J. Chem. Phys. 2021, 154, 164107].

`Forcefield`

Type Multiple Choice

Default value FQ

Options [FQ, FQFMU]

Description Version of the FQ family of polarizable forcefields

`Frozen`

Type Bool

Default value No

Description Expert option. Do not introduce polarization effect in response calculations.

`Kernel`

Type Multiple Choice

Default value OHNO

Options [OHNO, COUL, GAUS]

Description Expert option. KERNEL can be used to choose the functional form of the charge-charge interaction kernel between MM atoms. Recommended is to use the default OHNO. The COUL screening is the standard Coulomb interaction 1/r. The OHNO choice introduce the Ohno functional (see [K. Ohno, Theoret. Chim. Acta 2, 219 (1964)]), which depends on a parameter n that is set equal to 2. Finally, the GAUS screening models each FQ charge by means of a spherical Gaussian-type distribution, and the interaction kernel is obtained accordingly. For QM/FQFMU only GAUS SCREEN is implemented.

`MolCharge`

Type Float

Default value 0.0

Description Total charge of each fragment (FQ only)

`QMSCREEN`

Type Multiple Choice

Default value GAUS

Options [ERF, EXP, GAUS, NONE]

Description Expert option. QMSCREEN can be used to choose the functional form of the charge-charge interaction kernel between MM atoms and the QM density. The screening types available are ERF (error function), EXP (exponential), GAUS (Gaussian), or NONE. The default is GAUS.

QMSCREENFACTOR

Type Float

Default value 0.2

Description Expert option. Sets the QM/MM interaction kernel screening length. Recommended is to use the default value 0.2 with the GAUS QM/MM screening function.

The FQ(Fμ) region is specified within the Coords block, using the following scheme.

```
Coords
   elem1 x.xxx y.yyy z.zzz N
   elem2 x.xxx y.yyy z.zzz N
   elem3 x.xxx y.yyy z.zzz N
   ...
SubEnd
```

You will need to replace 'elem#' with the element you are assigning parameters to, and after the coordinates you need to include a number N assigning the element to a specific molcule. The latter number is important because the total charge of each environment molecule is constrained to be a specific value (0 by default) which can be specified with the MolCharge keyword if nonzero (see above scheme). For instance, the following block specifies two water molecules, where the oxygen and hydrogen atoms have been given the names OW and HW respectively:

```
Coords
    OW   -0.9957202   0.0160415   1.2422556   1
    HW   -1.4542703  -0.5669741   1.8472817   1
    HW   -0.9377950  -0.4817912   0.4267562   1
    OW   -0.2432343  -1.0198566  -1.1953808   2
    HW    0.4367536  -0.3759433  -0.9973297   2
    HW   -0.5031835  -0.8251492  -2.0957959   2
SubEnd
```

Where parameters must have been assigned for the atom types OW and HW in the respective AtomType blocks.

In case of water recommended is to use the optimized FQ parameters Chi and Eta for O and H in water proposed in Ref.[9]

```
AtomType
  Symbol O
  Chi 0.189194
  Eta 0.523700
SubEnd
AtomType
  Symbol H
  Chi 0.012767
  Eta 0.537512
SubEnd
```

---

[9] T. Giovannini, P. Lafiosca, B. Chandramouli, V. Barone, C. Cappelli, *Effective yet reliable computation of hyperfine coupling constants in solution by a QM/MM approach: Interplay between electrostatics and non-electrostatic effects*, Journal of Chemical Physics 150 (2019) 124102 (https://doi.org/10.1063/1.5080810)

Alternatively, in case of water as solvent, one could use the parameters proposed in Ref.[10]

```
AtomType
  Symbol O
  Chi 0.116859
  Eta 0.584852
SubEnd
AtomType
  Symbol H
  Chi 0.000001
  Eta 0.625010
SubEnd
```

For other solvents, it is recommended to use the parameters reported in Ref.[?].

For QM/FQFµ calculations for water as solvent, the parameters reported in Ref.[?] can be used

```
AtomType
  Symbol O
  Chi   0.290840
  Eta   0.562510
  Alpha 2.218790
SubEnd
AtomType
  Symbol H
  Chi   0.167570
  Eta   0.609320
  Alpha 1.190640
SubEnd
```

4.4.7 FDE: Frozen Density Embedding

The Frozen-Density-Embedding (FDE) option invokes calculation of the effective embedding potential introduced by Wesolowski and Warshel[1] in order to take into account the effect of the environment on the electronic structure of an embedded system. The embedding potential (Eq. 3 in Ref.[5]) depends explicitly on electron densities corresponding to the embedded subsystem (e.g. a solvated molecule) and its environment (e.g. solvent). For a detailed review, see Ref.[3]. The ADF implementation of the method is described in detail in Ref.[24].

A time-dependent linear-response generalization of this embedding scheme was derived in Ref.[6]. Its implementation in an approximate form, which assumes a localized response of the embedded system only (uncoupled FDE), is described in the supplementary material to Ref.[7]. For possible drawbacks and pitfalls in connection with this approximation, see

---

[10] S.W. Rick, S.J. Stuart, B.J. Berne, *Dynamical fluctuating charge force fields: Application to liquid water*, Journal of Chemical Physics 101 (1994) 6141 (https://doi.org/10.1063/1.468398)

[1] T.A. Wesolowski and A. Warshel, *Frozen Density Functional Approach for ab-initio Calculations of Solvated Molecules*, Journal of Physical Chemistry 97, 8050 (1993) (https://doi.org/10.1021/j100132a040)

[5] T.A. Wesolowski and J. Weber, *Kohn-Sham equations with constrained electron density: an iterative evaluation of the ground-state electron density of interacting molecules*, Chemical Physics Letters 248, 71 (1996) (https://doi.org/10.1016/0009-2614(95)01281-8)

[3] T.A. Wesolowski, in: Computational Chemistry: Reviews of Current Trends - Vol. 10, World Scientific, 2006.

[2] J. Neugebauer, C.R. Jacob, T.A. Wesolowski and E.J. Baerends, *An Explicit Quantum Chemical Method for Modeling Large Solvation Shells Applied to Aminocoumarin C151*, Journal of Physical Chemistry A 109, 7805 (2005) (https://doi.org/10.1021/jp0528764)

[4] C.R. Jacob, J. Neugebauer and L. Visscher, *A flexible implementation of frozen-density embedding for use in multilevel simulations*, Journal of Computational Chemistry 29, 1011 (2008) (https://doi.org/10.1002/jcc.20861)

[6] M.E. Casida and T.A. Wesolowski, *Generalization of the Kohn-Sham equations with constrained electron density formalism and its time-dependent response theory formulation*, International Journal of Quantum Chemistry 96, 577 (2004) (https://doi.org/10.1002/qua.10744)

[7] T.A. Wesolowski, *Hydrogen-Bonding-Induced Shifts of the Excitation Energies in Nucleic Acid Bases: An Interplay between Electrostatic and Electron Density Overlap Effects*, Journal of the American Chemical Society 126, 11444 (2004) (https://doi.org/10.1021/ja048846g)

Refs.[?][8][9].

The theory of coupled excited states for subsystems is described in Refs.[35][34], and extended for general response properties in Ref.[10]. This theory (subsystem TDDFT, coupled FDE) allows to treat the mutual response of several subsystems, including the ones that are considered environment.

A generalization of the FDE scheme to the calculation of NMR shieldings has been given in Ref.[11], where also the approximations involved and possible problems are discussed.

With the exception of interaction energies, the current implementation in ADF only allows the calculation of molecular properties that only depend on the electron density and of response properties using TDDFT. For an application to the calculation of several molecular properties in solution and a comparison to the DRF model also available in ADF, see Ref.[?]. For further applications of the ADF implementation, see Ref.[12] (weakly interacting complexes), Refs.[?][?][37][38] (solvent effects), and Refs.[39][40] (other environment effects).

To invoke a frozen-density embedding calculation, two additional specifications in the input are required. First, one or more frozen fragments have to be included in the FRAGMENTS block, and second, the block key FDE has to be included. In the simplest case, this input should look like this:

```
FRAGMENTS
   ...
   FragType FragFile type=FDE
   ...
END

FDE
   PW91K
end
```

In the FRAGMENTS block, for any fragment it is possible to specify the option type=FDE to indicate that the density of this fragment is kept frozen. This density is imported from the file FragFile. The frozen fragments have to be included in addition to the usual, non-frozen fragments. The atoms of the frozen fragments have to be included in the ATOMS block. As with normal fragments, the fragment found in the file will be rotated and translated to its position specified in the ATOMS block. For more details on specifying fragments, see the section ‘fragment files’. In the FDE input block, the recommended PW91k (also known as GGA97) approximant is recommended for the non-additive kinetic energy (the default is the local density approximant). A recommended alternative is NDSD. For all other options the defaults will be used.

---

[8] C.R. Jacob, J. Neugebauer, L. Jensen and L. Visscher, *Comparison of frozen-density embedding and discrete reaction field solvent models for molecular properties*, Physical Chemistry Chemical Physics 8, 2349 (2006) (https://doi.org/10.1039/B601997H)

[9] J. Neugebauer and E.J. Baerends, *Exploring the Ability of Frozen-Density Embedding to Model Induced Circular Dichroism*, Journal of Physical Chemistry A 110, 8786 (2006) (https://doi.org/10.1021/jp0622280)

[35] J. Neugebauer, *Couplings between electronic transitions in a subsystem formulation of time-dependent density functional theory*, Journal of Chemical Physics 126, 134116 (2007) (https://doi.org/10.1063/1.2713754).

[34] J. Neugebauer, *Photophysical Properties of Natural Light-Harvesting Complexes Studied by Subsystem Density Functional Theory*, Journal of Physical Chemistry B 112, 2207 (2008) (https://doi.org/10.1021/jp709956k)

[10] J. Neugebauer, *On the calculation of general response properties in subsystem density functional theory*, Journal of Chemical Physics 131, 084104 (2009) (https://doi.org/10.1063/1.3212883).

[11] C.R. Jacob and L. Visscher, *Calculation of nuclear magnetic resonance shieldings using frozen-density embedding*, Journal of Chemical Physics 125, 194104 (2006) (https://doi.org/10.1063/1.2370947)

[12] C.R. Jacob, T.A. Wesolowski and L. Visscher, *Orbital-free embedding applied to the calculation of induced dipole moments in* CO2··· X *(X=He, Ne, Ar, Kr, Xe, Hg) van der Waals complexes*, Journal of Chemical Physics 123, 174104 (2005) (https://doi.org/10.1063/1.2107567)

[37] J. Neugebauer, M.J. Louwerse, E.J. Baerends and T.A. Wesolowski, *The merits of the frozen-density embedding scheme to model solvatochromic shifts*, Journal of Chemical Physics 122, 94115 (2005) (https://doi.org/10.1063/1.1858411)

[38] J. Neugebauer, M.J. Louwerse, P. Belanzoni, T.A. Wesolowski and E.J. Baerends, *Modeling solvent effects on electron-spin-resonance hyperfine couplings by frozen-density embedding*, Journal of Chemical Physics 123, 114101 (2005) (https://doi.org/10.1063/1.2033749)

[39] M. Zbiri, M. Atanasov, C. Daul, J.-M. Garcia Lastra and T.A. Wesolowski, *Application of the density functional theory derived orbital-free embedding potential to calculate the splitting energies of lanthanide cations in chloroelpasolite crystals*, Chemical Physics Letters 397, 441 (2004) (https://doi.org/10.1016/j.cplett.2004.09.010)

[40] M. Zbiri, C.A. Daul and T.A. Wesolowski, *Effect of the f-Orbital Delocalization on the Ligand-Field Splitting Energies in Lanthanide-Containing Elpasolites*, Journal of Chemical Theory and Computation 2, 1106 (2006) (https://doi.org/10.1021/ct060035a)

Please note that throughout the FDE part of the documentation, the word “approximant” is used instead of the more usual “functional” to emphasize that the exact functional is not known, also in the case of the kinetic energy functional. In the literature one may encounter both words used interchangeably.

By including more than one frozen fragment, it is possible to use a frozen fragment that is a superposition of the densities of isolated molecules (this was possible in the previous version of ADF using the DENSPREP option). For a discussion and tests of the use of such approximate environment densities, see Ref.[?].

There is no restriction on the use of symmetry in FDE calculations, and usually the correct symmetry will be detected automatically. However, in the preparation of frozen fragments that will be rotated and/or translated in the FDE calculation, one has to include the keyword NOSYMFIT for technical reasons.

In the current implementation, only the electron density of the embedded (non-frozen) system is calculated. Therefore, with the exception of interaction energies, only properties that depend directly on the electron density (e.g. dipole moments) are available. In particular, the calculation of energy gradients is not implemented yet. All quantities given in the output refer (unless explicitly specified otherwise) to the non-frozen system only.

To employ the extension of FDE to the calculation of NMR shieldings, the file TAPE10 has to be used in the FDE calculation (by including the option SAVE TAPE10), and subsequently the NMR shielding has to be calculated using the program NMR. The TDDFT extension of the FDE formalism allows the calculation of electronic excitation energies and polarizabilities. This extension is automatically activated if FDE is used in combination with the EXCITATIONS or the RESPONSE key. To allow the mutual response of several subsystems, see the section on [*subsystem TDDFT* (page 135)].

Fragment-specific FDE options

For each frozen fragment, several additional options can be applied. To do this, the fragment specification is used as a subblock key by appending a & sign. The subblock is terminated with SubEnd. This subblock key looks, in the most general form, as follows:

```
FRAGMENTS
  ...
  FragType FragFile type=FDE &
     {FDEOPTIONS [USEBASIS] [RELAX or FREEZEANDTHAW] [OPTIMIZE]}
     {FDEDENSTYPE [SCF | SCFexact | SCFfitted ]}
     {RELAXCYCLES n or FREEZEANDTHAWCYCLES n}
     {XC [LDA | GGA ggapotx ggapotc | MODEL SAOP]}
  SubEnd
  ...
END
```

FDEOPTIONS

FDEOPTIONS USEBASIS If the USEBASIS option is specified, the basis functions of this frozen fragment will be included in the calculation of the embedded subsystem. This allows to expand the density of the embedded subsystem using not only atom-centered basis sets localized in the embedded subsystem but also the ones in the environment Ref.[13]. In large-scale simulations using the embedding potential, this option is recommended to be used in the preparation stage to investigate the basis set dependence of the results (chapter 5.3 in Ref.[?]). This option is also an indispensable element in the procedure introduced in Ref.[?] to test approximants to the kinetic-energy component of the embedding potential introduced by Wesolowski and Warshel.

FDEOPTIONS RELAX or FREEZEANDTHAW If the RELAX option (or equivalent FREEZEANDTHAW option) is specified, the density of this frozen fragment will be relaxed in freeze-and-thaw cycles[?], i.e., the embedded subsystem is frozen, while this fragment is thawed. This is repeated, until convergence is reached or until the maximum number of iterations has been performed. By relaxing frozen fragments, it is possible to

---

[13] T.A. Wesolowski and J. Weber, *Kohn-Sham equations with constrained electron density: The effect of various kinetic energy functional parametrizations on the ground-state molecular properties*, International Journal of Quantum Chemistry 61, 303 (1997) (https://doi.org/10.1002/(SICI)1097-461X(1997)61:2%3C303::AID-QUA13%3E3.0.CO;2-C)

improve a given approximate environment density by including the polarization of the environment due to the embedded system. This option is recommended to be used in the preparation stage of a large-scale numerical simulation. The freeze-and-thaw calculations lead to a pair of electron densities (embedded system and environment) that minimizes the total energy. As a consequence, the electron density of the environment derived from the freeze-and-thaw calculations can be used as a reference to verify the adequacy of the assumed electron density for the environment in a large-scale simulation. Due to technical restrictions, freeze-and-thaw is not possible if an open-shell (unrestricted) fragment is present.

FDEOPTIONS USEBASIS RELAX or FDEOPTIONS USEBASIS FREEZEANDTHAW It is further possible to combine USEBASIS and RELAX or FREEZEANDTHAW. In this case, the basis functions of the non-frozen fragment will be included when the density of the fragment is relaxed. This allows fully relaxed calculations with supermolecular expansion of the electron density of each subsystem. This option is to be used to test approximants to the kinetic-energy component of the embedding potential introduced by Wesolowski and Warshel by means of the procedure introduced in ref.[?].

FDEDENSTYPE The FDEDENSTYPE option can be used to specify which density is read from the fragment file. The possible options are:

FDEDENSTYPE SCF (or FDEDENSTYPE SCFexact) The exact density (not calculated using the fit functions) is used. This is the default.

FDEDENSTYPE SCFfitted The fitted density is used. This is less accurate but can be significantly faster.

RELAXCYCLES n or FREEZEANDTHAWCYCLES n This gives the maximum number of freeze-and-thaw cycles that are performed for this fragment. If the maximum number given in the FDE block is smaller, or if convergence is reached earlier, then fewer cycles are performed. For historical reasons, two equivalent keywords are available.

XC The XC option can be used to select the exchange-correlation potential that is used for this fragment when it is relaxed. By default, the same potential as for the non-frozen system is used, but in some cases it might be preferable to use another approximation for certain fragments. An example is given in Ref.[?].

XC LDA This option selects LDA as exchange-correlation potential for relaxing this fragment.

XC GGA ggapotx ggapotc This selects a GGA potential for relaxing this fragment. The GGA potential is specified by giving the name of the exchange potential, followed by the name of the correlation potential. The available potentials are listed in the documentation for the XC key.

XC MODEL SAOP This selects the model potential SAOP for relaxing this fragment.

Kinetic energy approximants

The approximants to the kinetic energy dependent component of the embedding potential are described here.

```
FDE
  {approximants to the kinetic energy dependent
                component of the embedding potential}
  {CJCORR [rho_cutoff]}
  {GGAPOTXFD exchange approximant}
  {GGAPOTCFD correlation approximant}
end
```

approximants to the kinetic energy dependent component of the embedding potential Several approximants to the kinetic-energy-dependent component of the effective potential given in Eq. (21) of ref.[?] are available. None of them is applicable if the embedded system is covalently bound to its environment. The user is recommended to look at the numerical value of the TSNAD(LDA) parameter which is given in the units of energy and can be considered as a measure of the overlap. The following rule of thumb should be applied: if this parameter is smaller than the estimated interaction energy between the embedded subsystem and the environment, then the available approximants are most probably adequate. If it exceeds this limit, the results can be less reliable.

Printing TSNAD(LDA) is not done by default, as it can be quite time-consuming. Its printing is switched on by including “EXTPRINTENERGY”, and “PRINTRHO2”, and “FULLGRID” in the FDE input block. If no kinetic energy approximant is specified, by default the local-density approximation (Thomas-Fermi approximant) is used. For an assessment of approximants for weakly overlapping pairs of densities see Refs.[?][16][15][17]. Based on these studies, the use of PW91k (= GGA97) is recommended.

APPROXIMANTS TO BE USED IN NORMAL APPLICATIONS

`THOMASFERMI (default)` Local-density-approximation form of vt[rhoA,rhoB][20] derived from Thomas-Fermi expression for Ts[rho][18][19].

`GGA97 (or PW91K)` Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from the Lembarki-Chermette[21] approximant to Ts[rho]. This approximant is currently the recommended one based on the numerical analysis of its accuracy[?][?] and the fact that the used enhancement factor disappears at large reduced density gradients, i.e. where the second-order gradient-expansion approximation fails[?][?].

`NDSD` Similarly to GGA97, the NDSD approximant is constructed by taking into account the asymptotic behavior of the functional vt[rhoA,rhoB] at small density gradients. In the construction of NDSD, the exact property of vt[rhoA,rhoB] at rho_A → 0 and for $\int$ rhoB = 2 given in Eq. A6 of Ref.[22] is also taken into account. The analysis of the accuracy of this potential[?] shows that NDSD is of the same or superior quality as GGA97. NDSD is, therefore, recommended as the successor of GGA97 to be used anywhere where the quality of the results depends directly on the accuracy of the potential vt[rhoA,rhoB], i.e., for obtaining electronic-structure-dependent properties. The analytical form of the corresponding approximant to the functional Ts$^{\text{nad}}$ [rho_A,rho_B]$ exists (Eq. 23 in Ref.[?]). It is not possible, however, to obtain the analytical form of the corresponding parent functional for the kinetic energy Ts[rho]. To reflect this and the fact that, similarly to the GGA approximants to vt[rhoA,rhoB], the numerical values of only first- and second derivatives of density are needed, the label NDSD (Non-Decomposable Second Derivatives) is used.

OBSOLETE APPROXIMANTS (can be used but GGA97 leads usually to a better embedding potential see refs.[?][?])

`LLP91` Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from Lee-Lee-Parr[23] approximant to Ts[rho].

`PW86k` Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from the Fuentealba-Reyes approximant to Ts[rho][24].

`THAKKAR92` Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from the Thakkar approximant to Ts[rho][25].

APPROXIMANTS WHICH MIGHT BE USEFUL ONLY FOR THEORY DEVELOPMENT

---

[16] T.A. Wesolowski, H. Chermette and J. Weber, *Accuracy of Approximate Kinetic Energy Functionals in the Model of Kohn-Sham Equations with Constrained Electron Density: the* $FH\cdots NCH$ *complex as a Test Case*, Journal of Chemical Physics 105, 9182 (1996) (https://doi.org/10.1063/1.472823)

[15] T.A. Wesolowski, *Density functional theory with approximate kinetic energy functionals applied to hydrogen bonds*, Journal of Chemical Physics 106, 8516 (1997) (https://doi.org/10.1063/1.473907)

[17] Y.A. Bernard, M. Dulak, J.W. Kaminski and T.A. Wesolowski, *The energy-differences based exact criterion for testing approximations to the functional for the kinetic energy of non-interacting electrons*, Journal of Physics A 41, 55302 (2008) (https://doi.org/10.1088/1751-8113/41/5/055302)

[20] P. Cortona, *Self-consistently determined properties of solids without band-structure calculations*, Physical Review B 44, 8454 (1991) (https://doi.org/10.1103/PhysRevB.44.8454)

[18] L.H. Thomas, *The calculation of atomic fields*, Mathematical Proceedings of the Cambridge Philosophical Society 23, 542 (1927) (https://doi.org/10.1017/S0305004100011683)

[19] E. Fermi, *Eine statistische Methode zur Bestimmung einiger Eigenschaften des Atoms und ihre Anwendung auf die Theorie des periodischen Systems der Elemente*, Zeitschrift für Physik 48, 73 (1928) (https://doi.org/10.1007/BF01351576)

[21] A. Lembarki and H. Chermette, *Obtaining a gradient-corrected kinetic-energy functional from the Perdew-Wang exchange functional*, Physical Review A 50, 5328 (1994) (https://doi.org/10.1103/PhysRevA.50.5328)

[22] J.M. Garcia Lastra, J.W. Kaminski and T.A. Wesolowski, *Orbital-free effective embedding potential at nuclear cusps*, Journal of Chemical Physics 129, 074107 (2008) (https://doi.org/10.1063/1.2969814)

[23] H. Lee, C. Lee and R.G. Parr, *Conjoint gradient correction to the Hartree-Fock kinetic- and exchange-energy density functionals*, Physical Review A 44, 768 (1991) (https://doi.org/10.1103/PhysRevA.44.768)

[24] D.A. Kirzhnits, Soviet Physics JETP-USSR 5, 64 (1957)

[25] A.J. Thakkar, *Comparison of kinetic-energy density functionals*, Physical Review A 46, 6920 (1992) (https://doi.org/10.1103/PhysRevA.46.6920)

The accuracy of **some** of these approximants was investigated in detail[?][?][?][?]. Each of them was shown to lead to a qualitatively incorrect embedding potential. They shouldn't be used in practical applications.

COULOMB Neglecting completely vt[rhoA,rhoB] (vt[rhoA,rhoB] equals zero) together with the exchange-correlation component of the embedding potential introduced by Wesolowski and Warshel.

TF9W The approximant to vt[rhoA,rhoB][?] derived from the second-order gradient expansion[?] for Ts[rho].

WEIZ The approximant to vt[rhoA,rhoB][?] derived from the von Weizsäcker approximant to Ts[rho][?].

OL91A Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from the first Ou-Yang and Levy approximant to Ts[rho][26].

OL91B Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from the second Ou-Yang and Levy approximant to Ts[rho][?].

E00 Generalized-gradient-approximation form of vt[rhoA,rhoB][27] derived from a kinetic energy functional by Ernzerhof[28] which represents the gradient expansion approximation up to the fourth order.

P92 Generalized-gradient-approximation form of vt[rhoA,rhoB][?] derived from a kinetic energy functional by Perdew[29] which represents the gradient expansion approximation up to the sixth order.

LONG DISTANCE CORRECTIONS TO THE EFFECTIVE POTENTIAL

CJCORR Option to switch on a long-distance correction. By default this option is not used. As was shown in Ref.[30], with the available approximate kinetic-energy approximants, the embedding potential has the wrong form in the limit of a large separation of the subsystems. In particular, it was shown that this can have serious consequences in the case of "supermolecular expansion of electron density of each subsystem" calculations (USEBASIS option). In Ref.[?], a correction is proposed that enforces the correct long-distance limit. (See also this reference for limitations of this correction.)

CJCORR [rho_cutoff] This option switches on the long-distance correction. This option has to be used in combination with one of the above kinetic-energy approximants. By default, a density cut-off of 0.1 is employed.

NONADDITIVE EXCHANGE-CORRELATION APPROXIMANT

GGAPOTXFDGGAPOTCFD Option to specify the non-additive exchange-correlation approximant. By default, in the construction of the effective embedding potential the exchange-correlation approximant that was specified in the XC block is used. It is possible to specify a different approximant with the GGAPOTXFD and GGAPOTCFD options. This is particularly useful in combination with the use of model potentials like SAOP, that can not be used in the embedding potential because of their orbital dependence. (For a discussion, see Ref.[?].)

GGAPOTXFD exchange approximant The exchange approximant is used in the construction of the embedding potential. The same exchange approximants as in the XC key are available.

GGAPOTCFD correlation approximant The correlation approximant is used in the construction of the embedding potential. The same correlation approximants as in the XC key are available.

---

[26] H. Ou-Yang and M. Levy, *Approximate noninteracting kinetic energy functionals from a nonuniform scaling requirement*, International Journal of Quantum Chemistry 40, 379 (1991) (https://doi.org/10.1002/qua.560400309)

[27] A.W. Götz, S.M. Beyhan and L. Visscher, *Performance of Kinetic Energy Functionals for Interaction Energies in a Subsystem Formulation of Density Functional Theory*, Journal of Chemical Theory and Computation 5, 3161 (2009) (https://doi.org/10.1021/ct9001784)

[28] M. Ernzerhof, *The role of the kinetic energy density in approximations to the exchange energy*, Journal of Molecular Structure: THEOCHEM 501-502, 59 (2000) (https://doi.org/10.1016/S0166-1280(99)00414-5)

[29] J.P. Perdew, *Generalized gradient approximation for the fermion kinetic energy as a functional of the density*, Physics Letters A 165, 79 (1992) (https://doi.org/10.1016/0375-9601(92)91058-Y)

[30] C.R. Jacob, S.M., Beyhan and L. Visscher, *Exact functional derivative of the nonadditive kinetic-energy bifunctional in the long-distance limit*, Journal of Chemical Physics 126, 234116 (2007) (https://doi.org/10.1063/1.2743013)

General FDE options

In addition to the fragment-specific options and the kinetic energy approximants, there are also a number of options available in FDE calculations that will be described in the following.

```
FDE
  {FULLGRID}
  {RELAXCYCLES n or FREEZEANDTHAWCYCLES n}
  {RELAXPOSTSCF or FREEZEANDTHAWPOSTSCF}
  {EXTPRINTENERGY}
  {PRINTRHO2}
  {ENERGY}
  {SDFTENERGY}
  {DIPOLE}
end
```

FULLGRID By default, FULLGRID is not used, and in FDE calculations the integration grid is generated as described in Ref.[?] by including only atoms of the frozen subsystem that are close to the embedded subsystem in the generation of the integration grid. The distance cutoff used is chosen automatically, based on the extent of the basis functions of the embedded subsystem. (It can also be chosen manually, see the option qpnear in the INTEGRATION key) This scheme results in an efficient and accurate integration grid. However, it is possible that the default integration scheme is not accurate enough. This can be the case for weakly interacting systems and when the distance between the frozen and the embedded system is large. It is therefore recommended to check the quality of the default integration grid by comparing to results obtained using the full supermolecular grid (FULLGRID option).

If the subkey FULLGRID is included, all atoms of the frozen system are included in the generation of the integration grid. This results in the same grid that would be used in a supermolecular calculation of the combined frozen and embedded system. The integration grid generated by this option might be much larger than the default grid. This option should be used to check the quality of the default integration grid.

RELAXCYCLES n or FREEZEANDTHAWCYCLES n Specifies the maximum number n of freeze-and-thaw iterations[?] that are performed (for frozen fragments with the RELAX) option. If a smaller number of iterations is specified as a fragment-specific option, for this fragment this smaller number is used. Furthermore, if convergence is reached earlier, no more iterations will be performed.

RELAXPOSTSCF or FREEZEANDTHAWPOSTSCF If this option is included, several post-SCF properties will be calculated after each freeze-and-thaw cycle [Ref.[?]. These are otherwise only calculated in the last cycle.

EXTPRINTENERGYPRINTRHO2 If the options EXTPRINTENERGY and PRINTRHO2 are included (both are needed and should be listed on separate lines), several additional quantities will be printed, including TSNAD(LDA). In order to obtain meaningful numbers, also the FULLGRID keyword (see above) has to be used.

ENERGY Option to switch on the calculation of the FDE energy as the sum of the energy E[rhoA] of the active, embedded system and the interaction energy Eint[rhoA,rhoB] of the embedded system with the frozen environment. This relies on the calculation of the total energy for the embedded system and all caveats and restrictions for total energy evaluations apply (see keyword *TOTALENERGY* (page 252)). All energy contributions are evaluated on the grid of the active subsystem. Some contributions to the interaction energy Eint[rhoA,rhoB] require an accurate integration grid in the region of the environment. Thus, in pure embedding calculations (without fragment-specific option RELAX), an accurate calculation of the FDE energy requires a full supermolecular integration grid (FULLGRID option). Details on the implementation and the performance of kinetic energy functionals for interaction energies are documented in Ref.[?]

The calculation of the full, variationally minimized subsystem DFT energy, that is, the sum of the energy of two subsystems E[rhoA] and E[rhoB] and their interaction energy Eint[rhoA,rhoB] in the framework of FDE, is invoked if then the fragment densities are relaxed in freeze-and-thaw cycles (option RELAXCYCLES and fragment-specific FDE option RELAX). In this case the supermolecular integration grid is not required. Instead, in each step of the freeze-and-thaw cycle, the critical energy terms are taken from the previous freeze-and-thaw step of the presently frozen fragment. The convergence of the energy contributions with the number of freeze-and-thaw iterations should be carefully monitored. Due to conceptual problems for the evaluation of the non-additive kinetic energy contribution, only two subsystems, that is, one frozen fragment, is supported for FDE energy calculations with freeze-and-thaw.

SDFTENERGY This is a generalization of the original ENERGY implementation that allows for the evaluation of the FDE/sDFT total energy. Usage of a supermolecular integration grid is recommended (FULLGRID option), because of the non-additive kinetic and XC contributions. EXTPRINTENERGY can be combined with this option to print the subsystem DFT energy after each freeze-and-thaw cycle. For pure embedding calculations, the SDFTENERGY implementation reads the total energy from each frozen fragment's adf.rkf (TAPE21) file. The total energy of frozen fragments is not (re-)evaluated during FDE calculations. It is therefore mandatory to add the *TOTALENERGY* (page 252) keyword to the preparatory isolated fragment calculations.

DIPOLE The dipole of the supersystem is calculated as the sum of analytically integrated fragment dipole moments. Note that the dipole moment becomes origin dependent for charged (sub-)fragments. The current implementation does NOT take care of this.

Frozen Density Embedding with External Orthogonality

An implementation of External Orthogonality (EO) into the FDE framework in ADF can be found in Ref.[31]. In this method in the fragment calculation one need to specify ghost atoms at the positions of the atoms in all the other fragments. Thus each fragment is calculated in the supermolecular basis.

```
AOMat2File
IgnoreOverlap
FDE
  EXTERNALORTHO {factor}
  ...
END
```

EXTERNALORTHO Used to specify the use of external orthogonality (EO) in the FDE block, with an optional factor (default factor is 1e6). Note that the overlap region is defined by ghost atoms, and the general keyword IGNOREOVERLAP is necessary. Additionally, the general keyword AOMAT2FILE is required to save some important fragment AO matrices to adf.rkf for use in an EO calculation. it is recommended that one uses the STOFIT keyword for the STOFIT density fitting method.

FDE and (localized) COSMO

COSMO solvation can be included in combination with FDE. This means that a COSMO cavity will be created that holds both the active and frozen fragments.

For very large systems the solution of the COSMO equations can become the bottleneck of the calculation. A local COSMO variant, which exploits the subsystem nature of the underlying electronic description, is implemented in ADF, see Ref.[32]. This method, called LoCOSMO, is an approximation to regular COSMO for fragment-based electronic structure methods. If a given fragment is active (i.e. only its own density changes over the SCF cycles), the surface charges outside a given radius around this fragment will be kept fixed in magnitude (excluded from the COSMO optimization). However, no interactions are ignored by adding a corresponding constant term to the COSMO solution vector.

```
SOLVATION
  ..
  CHARGED LoCosmo LoCosmoDist
END
```

---

[31] D.V. Chulhai and L. Jensen, *Frozen Density Embedding with External Orthogonality in Delocalized Covalent Systems*, Journal of Chemical Theory and Computation 11, 3080 (2015) (https://doi.org/10.1021/acs.jctc.5b00293).

[32] A. Goez, and J. Neugebauer, *A Local Variant of the Conductor-Like Screening Model for Fragment-Based Electronic-Structure Method*, Journal of Chemical Theory and Computation 11, 5277 (2015) (https://doi.org/10.1021/acs.jctc.5b00832)

`LoCosmo LoCosmoDist` If LoCosmo is included the local COSMO will be used in the calculation. LoCosmoDist is a cutoff radius (Angstrom), which must be specified by the user, if LoCOSMO is included. All surface charges within this radius from any atom of the active fragment are included in the active charge space. A sensible value for LoCosmoDist is 5.0 Angstrom.

To be really useful, Freeze-and-Thaw cycles should be carried out with LoCOSMO. After each fragment calculation, the KF file LOCSURCH is written to disk. If present, this file will be automatically read in during the next (LoCOSMO) fragment calculation. This means that if the internal FT cycles are used by specifying RELAXCYCLES, the code creates and reads in the file in a fully automatic fashion. LOCSURCH is neither created nor read during a regular COSMO calculation.

In case one calculates excitation energies with the EXCITATION block key, and LoCOSMO is specified, only the solvent response due to surface charges within the specified cutoff radius will be taken into account, see also Ref.[33]. It should be noted that in a regular (uncoupled) FDE TDDFT calculation, the response due to the frozen density is neglected.

FDE and QM/FQ(Fμ)

FDE can also be combined with the *QM/FQ(Fμ)* (page 122) method which can be used to embed a quantum system in a polarizable classical environment. The fluctuating classical layer can be included in all stages of an FDE calculation, i.e. when performing the separate calculations for the two isolated fragments as well as the full FDE system. There is also the possibility of using the QM/FQ(Fμ) method to account for the mutual response of the subsystems by endowing the frozen layer's atoms with fluctuating charges (and dipoles), provided a suitable parametrization is available. See the *QM/FQ(Fμ)* (page 122) manual page for more details.

Subsystem TDDFT, coupled FDE

The linear-response subsystem TDDFT code implements the theory of coupled excited states for subsystems as described in Refs.[??]. This theory is based on the FDE extension to excited states[?], which is implemented in ADF in a local response approximation, i.e., neglecting the dynamic response of the environment[?].

The subsystem TDDFT code allows to treat the mutual response of several subsystems, including the ones that are considered environment. A more typical situation would be a system composed of several equivalent chromophores treated as individual subsystems. In this case, the local response approximation leads to uncoupled excited states of the subsystems (hence the acronym FDEu is employed often), while the subsystem TDDFT code couples the monomer excitations to obtain the excited states of the total system (often denoted as coupled frozen density embedding, FDEc). This can be related to excitonic couplings between the monomers[?].

The current implementation is restricted to NOSYM calculations and Singlet-Singlet excitations without frozen core approximation. It makes use of the ALDA kernel (including a Thomas-Fermi part for the contribution arising from the non-additive kinetic energy) for consistency with the uncoupled FDE implementation for excited states. Some features have not or not extensively been tested and should be used with great care, e.g., linear dependencies in the basis set. Details on the calculation of transition moments, oscillator and rotational strengths are described in Ref.[?].

Subsystem TDDFT (FDEc) calculations can be invoked with the SUBEXCI key.

SUBEXCI input

FDEc calculations on coupled excited states first require that an uncoupled FDE-TDDFT calculation has been performed for every subsystem that should be included in the coupled calculation, and that the corresponding adf.rkf files, in which the considered subsystems are "active", have been saved (see the separate FDE input description). This means that it is not possible to use the information on frozen/inactive fragments from a adf.rkf file of a previous uncoupled FDE calculation, which contains all subsystems.

---

[33] A. Goez, and J. Neugebauer, *Including protein density relaxation effects in first-principles embedding calculations of cofactor excitation energies*, Molecular Physics 115, 526 (2016) (https://doi.org/10.1080/00268976.2016.1199823)

Although it is technically possible to use adf.rkf files from non-FDE calculations on the separate subsystems, this would lead to results that are inconsistent with the subsystem TDDFT methodology from Ref.[?]. In any case, a previous TDDFT calculation for each subsystem that should be included in the coupling procedure is necessary. If that is not the case, the subsystem will still be considered in the calculation of the total electron density (needed in the setup of the exchange-correlation kernel), but will not be included in the coupling procedure.

The first subsystem should always be one of the coupled subsystems. The input will then look like the corresponding input for an uncoupled FDE-TDDFT calculation, but in addition should contain the following block:

```
SUBEXCI
  {LOWEST nlowest}
  {OPTSTATES list_of_optstates}
  {CTHRES coupling_threshold}
  {SFTHRES solutionfactor_threshold}
  {COUPLBLOCK}
  {TDA}
  {CICOUPL}
END
```

LOWEST nlowest The selection of the excited states to be coupled consists of two steps. First, a number of reference states are selected. As a default, the nlowest (default: 10) lowest excited states present on the fragment file for the first subsystem are considered.

OPTSTATES list_of_optstates If the keyword OPTSTATES is given, only those excited states of the first subsystem are considered as reference states that are given in the list_of_optstates (numbers of states separated by blanks).

CTHRES coupling_threshold Second, all excitations of all subsystems (present on the fragment adf.rkf files) with an excitation energy that differs by less than coupling_threshold (to be given in units of eV; default: 30 eV) from one of the reference states are selected to be included in the coupling. Note that additional excited states of system 1 may be included here.

COUPLBLOCK If COUPLBLOCK is specified in the input, all couplings between all of these local excited states are included. Otherwise (default), the coupling_threshold is also applied to select pairs of states for which couplings are calculated. I.e., couplings are not calculated if the two particular states to be coupled differ in energy by more than coupling_threshold.

SFTHRES solutionfactor_threshold To reduce the computational effort, it is possible to ignore the effect of orbital pairs with coefficients less than solutionfactor_threshold in the solution factors (TDDFT eigenvectors) of the underlying uncoupled calculation in the construction of the exact trial densities during the calculation of the coupling matrix elements. These orbital pair contributions are not ignored in the subsequent calculation of transition moments, oscillator, and rotational strengths. The default value of 0.00001 typically leads to a precision of the coupled excitation energies of about 0.0001 eV.

TDA TDA specifies the use of the Tamm-Dancoff-Approximation (*Tamm-Dancoff approximation* (page 162)) in the underlying uncoupled FDE-TDDFT calculations (Ref.[36]). Contrary to the full SUBEXCI-TDDFT variant, SUBEXCI-TDA allows for the usage of hybrid functionals in the underlying uncoupled FDE-TDDFT calculations.

CICOUPL Within the Tamm-Dancoff Approximation, the couplings between localized excited states on different subsystems correspond directly to so-called exciton couplings (see Ref.[?]). The CICOUPL keyword, in conjunction with TDA, prints these exciton couplings. It is also possible to use CICOUPL with full FDEc-TDDFT. In that case, the excitonic couplings between monomers are reconstructed from an effective 2x2 CIS-like eigenvalue problem, as e.g. done in Ref.[?].

In addition, the input file may contain either an EXCITATION block or the keyword DIFFUSE. Both options lead to a slight adaption of the integration grid. Apart from this, the EXCITATION block will be ignored.

---

[36] C. König, N. Schlüter. J. Neugebauer, *Direct Determination of Exciton Couplings from Subsystem TDDFT within the Tamm-Dancoff Approximation*, Journal of Chemical Physics 138, 034104 (2013) (https://doi.org/10.1063/1.4774117)

The key ALLOW PARTIALSUPERFRAGS is currently necessary to be able to use subsystem information for only one subsystem from a adf.rkf file of a previous FDE calculation:

```
ALLOW PARTIALSUPERFRAGS
```

Restrictions and pitfalls

In the current implementation, only the electron density of the embedded system is calculated. Therefore, with the exception of interaction energies, only properties that depend directly on the electron density (e.g., dipole moments) are available. In addition, the TDDFT extension allows the calculation of electronic excitation energies and polarizabilities, and NMR shieldings can be calculated.

Warning: EVERYTHING ELSE IS NOT YET IMPLEMENTED. THE RESULTS OBTAINED FOR OTHER PROPERTIES MIGHT BE MEANINGLESS.

Kinetic energy approximant: Although the effective embedding potential is derived from first principles using universal density approximants, the ADF implementation relies on approximations. Currently, two implemented approximations are recommended[?]: PW91k (also known as GGA97) which uses electron densities and the corresponding gradients to express the non-additive kinetic energy component of the embedding potential, or TF (Thomas-Fermi LDA approximant), which does not use gradients at all. Either approximation is applicable only in cases where the overlap between electron densities of the corresponding interactions is small. Note: so far, no approximation has been developed for the strong-overlap case - two subsystem linked by covalent bonds for instance.

4.4.8 3D-RISM: 3D reference Interaction Site Model

Introduction

The three-dimensional reference interaction site model (3D-RISM) provides the solvent structure in the form of a 3D site distribution function, $g_{\gamma}^{UV}(r)$, for each solvent site, γ.

It enables, at modest computational cost, the calculations of thermodynamics, electronic properties and molecular solvation structure of a solute molecule in a given molecular liquid or mixture. Using 3D-RISM, one can study chemical reactions, including reaction coordinates and transition state search, with the molecular solvation described from the first principles based on a molecular-mechanics type description of the solvent. The method yields all of the features available by using other solvation approaches.

The 3D-RISM implementation has been revised and supplemented by a new option that uses the fitted electron density in the interaction between solvent and solute directly. Without the use of point charges, numerical stability has been improved. Using this option, 3D-RISM can now be used for structure optimization, (numerical) frequencies, vertical excitation energies and magnetic properties. The code has also been parallelized. Details can be found in Ref.[1]. Starting from AMS2022, new point charge based options are available to be used in combination with structure optimization, numerical frequencies and vertical excitation energies.

Details on the original implementation of 3D-RISM-KH in ADF can be found in Ref.[2], with applications in Ref.[3]. The

---

[1] M. Reimann and M. Kaupp, *Evaluation of an Efficient 3D-RISM-SCF Implementation as a Tool for Computational Spectroscopy in Solution*, Journal of Physical Chemistry A 124, 7439 (2020) (https://dx.doi.org/10.1021/acs.jpca.0c06322)

[2] S. Gusarov, T. Ziegler and A. Kovalenko, *Self-Consistent Combination of the Three-Dimensional RISM Theory of Molecular Solvation with Analytical Gradients and the Amsterdam Density Functional Package*, Journal of Physical Chemistry A 110, 6083 (2006) (https://doi.org/10.1021/jp05[#ref11]_4t)

[3] D. Casanova, S. Gusarov, A. Kovalenko and T. Ziegler, *Evaluation of the SCF Combination of KS-DFT and 3D-RISM-KH; Solvation Effect on Conformational Equilibria, Tautomerization Energies, and Activation Barriers*, Journal of Chemical Theory and Computation 3, 458 (2007) (https://doi.org/10.1021/ct6001785)

theory of 3D-RISM-KH in combination with DFT can be found in Refs.[4][5][6][7] . A combination of 3D-RISM-KH with FDE (frozen-density embedding) can be found in Ref.[8].

Similar to explicit solvent simulations, 3D-RISM properly accounts for chemical peculiarities of both solute and solvent molecules, such as hydrogen bonding and hydrophobic forces, by yielding the 3D site density distributions of the solvent. Moreover, it readily provides, via analytical expressions, all of the solvation thermodynamics, including the solvation free energy potential, partial molar volume and compressibility. Starting from AMS2023, enthalpic and entropic decomposition of the solvation free energy can be obtained approximately using analytic expression (which are exact for CHRGLVL=MM or pure Lennard-Jones fluids), also including the macroscopic dependence of the dielectric constant on temperature and solvent density.[9] The expression for the solvation free energy (and its derivatives) in terms of integrals of the correlation functions follows from a particular approximation for the so-called closure relation used to complete the integral equation for the direct and total correlation functions.

Solvation free energy

The solvation free energy can be calculated as:

> Solvation Free Energy =
> [(Total Bonding Energy) + (Internal Energy)
>   - T * (Entropy) + (Excess Chemical Potential)]_3D-RISM
> - [(Total Bonding Energy) + (Internal Energy) - T * (Entropy)]_Gas-Phase

If one assumes that the internal energy and the vibrational and rotational entropy of the solute molecule is the same in solution a

> Solvation Free Energy =
> [(Total Bonding Energy) + (Excess Chemical Potential)]_3D-RISM
> - [(Total Bonding Energy)]_Gas-Phase

However, a formally accurate calculation should include the difference between the thermal corrections from frequency calculations produced by ADF in the SCF calculation with 3D-RISM solvation and in gas phase.

Input

When performing 3D-RISM simulations, each atom in the **ATOMS** block in the AMS part of the input must have two Lennard-Jones parameters specified, adf.SigU and adf.EpsU, for example:

```
System
   ATOMS
     C     0.00     0.00     0.00     adf.SigU=3.50    adf.EpsU=0.066    adf.ChgU=-0.
↪4000
     ...
   END
End
```

They can be obtained from a Lennard-Jones force-field parameter set. Atom centered point charges adf.ChgU can optionally be set as well but will only be used in combination with CHRGLVL=MM (see below).

---

[4] A. Kovalenko and F. Hirata, *Potentials of mean force of simple ions in ambient aqueous solution. II. Solvation structure from the three-dimensional reference-interaction site model approach, and comparison with simulations*, Journal of Chemical Physics 112, 10403 (2000) (https://doi.org/10.1063/1.481677)

[5] A. Kovalenko and F. Hirata, *Self-consistent description of a metal-water interface by the Kohn-Sham density functional theory and the three-dimensional reference-interaction site model*, Journal of Chemical Physics 110, 10095 (1999) (https://doi.org/10.1063/1.478883)

[6] A. Kovalenko and F. Hirata, *Potentials of mean force of simple ions in ambient aqueous solution. I. Three-dimensional reference-interaction site model approach*, Journal of Chemical Physics 112, 10391 (2000) (https://doi.org/10.1063/1.481676)

[7] A. Kovalenko, *Three-dimensional RISM theory for molecular liquids and solid-liquid interfaces*, In Molecular Theory of Solvation; Hirata, Fumio, Ed.; Understanding Chemical Reactivity (series); Mezey, Paul G., Series Ed.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2003; Vol. 24, pp 169-275.

[8] J.W. Kaminski, S. Gusarov, A. Kovalenko and T.A. Wesolowski, *Modeling solvatochromic shifts using the orbital-free embedding potential at statistically mechanically averaged solvent density*, Journal of Physical Chemistry A 114, 6082 (2010) (https://doi.org/10.1021/jp100158h)

[9] M. Reimann and M. Kaupp *Reaction Entropies in Solution from Analytical Three-Dimensional Reference Interaction Site Model Derivatives with Application to Redox and Spin-Crossover Processes*, Journal of Physical Chemistry A 126, 3708 (2022) (https://pubs.acs.org/doi/10.1021/acs.jpca.2c02317)

All other 3D-RISM-related input keys are contained in a **RISM** input block. Below, only the mandatory keywords are shown. Optional keywords are described in the next section.

```
RISM title
  RISM1D
    FLUIDPARAM temper=298. DielConst=78.497 UTotDens=1/A3 0.03333
  SUBEND
  SOLVENT ArbitrarySolventName
    UNITS uWeight=g/mol  ULJsize=A  ULJenergy=kcal/mol Ucoord=A Udens=1/A3
    PARAMETERS Weight=weight nAtoms=NSiteTypes
      N1    Z_alpha1   Sigma_alpha1   Eps_alpha1   X1_1 Y1_1 Z1_1
                                                   X1_2 Y1_2 Z1_2
                                                   ...  ...  ...
      N2    Z_alpha2   Sigma_alpha2   Eps_alpha2   X2_1 Y2_1 Z2_1
                                                   X2_2 Y2_2 Z2_2
                                                   ...  ...  ...

      ...
      DENSPE=density
  SUBEND
  SOLUTE ArbitrarySoluteName
    BOXSIZE   sizeX  sizeY  sizeZ
    BOXGRID    npX    npY    npZ
  SUBEND
END
```

The **RISM1D** sub-block contains general parameters for the preceding 1D-RISM calculation of the solvent(s) to create the bulk susceptibility function. Even though all technical RISM1D sub-keys have reasonable defaults, the FLUIDPARAM sub-key will most likely require attention because its default values for density and dielectric constant are only applicable if the solvent is water. Thus, you may need to change these values when modeling a different solvent. Note that even when using all default values from the RISM1D sub-block the sub-block itself must be specified, even if empty. See below for complete description of the RISM1D sub-block.

The **SOLVENT** sub-block can be repeated for each component if the solvent is a mixture. Each SOLVENT sub-block contains parameters for one solvent. First, each solvent has a name, which is specified on the SOLVENT keyword's line and is arbitrary. The first line in the SOLVENT sub-block must contain the UNITS key. Usually, only the Udens part should be changed. Then follow the actual solvent parameters. In principle, each solvent consists of multiple atoms and functional groups (so called sites). For example, in 3D-RISM terms, methanol consists of 3 sites: CH_3 , O, and H. Each such site has a set of three parameters, shared between all sites of the same type, and the coordinates. These parameters follow the PARAMETERS keyword. The line with the PARAMETERS keyword itself must specify the molecular weight of the solvent and the number of site types that follow. The first line for each site type contains, in this order: number of equivalent sites of this type, z_α (the point charge assigned) , σ_α , ϵ_α (the Lennard-Jones parameters assigned), three coordinates for the first site of this type. If there is more than one site of this type then the coordinates for the 2nd and other sites follow on subsequent lines. The SOLVENT sub-block is concluded by the specific density of this solvent, by default, in molecules per cubic angstrom. This number should be equal to the total density for mono-component solvents. Using Udens=MOL, the specific density needs to be given in molar fractions, and should therefore be one for mono-component solvents.

For example, the SOLVENT block for water would typically look as:

```
SOLVENT water
  UNITS      uWeight=g/mol  ULJsize=A  ULJenergy=kcal/mol Ucoord=A Udens=1/A3
  PARAMETERS Weight=18.015   nAtoms=2
    1    -0.8476   3.166    0.1554       0.000000  0.00000 -0.064605
    2     0.4238   1.000    0.0560      -0.816495  0.00000  0.512747
                                         0.816495  0.00000  0.512747
  DENSPE=0.03333
SUBEND
```

The **SOLUTE** sub-block specifies 3D-RISM parameters for your molecule. The BOXSIZE and BOXGRID sub-keys specify dimensions of the simulation box, in Angstrom, and the number of points of grid in each direction. The box should at least be twice as large as the molecule and the BOXGRID values must be even numbers. Due to the 3D-FFT routines, the largest prime factor of the BOXGRID values should be 7 for optimal performance. The size/np ratio defines the grid spacing in each direction and this should be not larger than 0.5 angstrom for energy calculations and not larger than 0.35 angstrom for structure optimizations. When the BOXSIZE and BOXGRID keywords are not specified the grid is calculated based on the GRIDSTEP and BOXEXTENT values, see below. The number of grid points along the Cartesian coordinate *i* will then be determined as $N_i = (range(i) + 2 * BOXEXTENT)/GRIDSTEP + 1$, where *range(i)* is the span of Cartesian coordinate *i* of the solute molecule. The number of grid points may be increased to be a product of prime numbers not larger than 7.

The **optional** 3D-RISM keys for the RISM1D and SOLUTE sub-blocks are listed below together with their defaults.

```
! optional RISM1D subkeys with their default values:
RISM1D Theory=RISM Closure=KH NOSPEEDUP
  FLUIDPARAM temper=298. DielConst=78.497 UTotDens=1/A3 0.03333
  OUTPUT PrintLev=5 File=solvent
  GRID   Nr=8192  dR=0.025 MaxRout=100.0 MaxKout=0.0
  MDIIS  N=20 Step=0.5 Tolerance=1.e-12
  ELSTAT LRsmear=1.0 Adbcor=0.5
  ITER   Ksave=-1 Kshow=1 Max=10000
  DODEPS alpha=-1.0 kappa=-1.0 dedtp=-1.0 dedpt=-1.0
  Restart1D=restartfile
SUBEND
```

RISM1D Theory - Version of 1D-RISM used: RISM - extended RISM, DRISM - dielectrically consistent RISM (should be used for solvents with a static dipole moment);

Closure - Closure for the 1D problem: KH - Kovalenko-Hirata, HNC - hypernetted chain, PSEn - partial series expansion of HNC up to order n (n=1 is identical to KH), PY - Perkus-Yevik, KGK - Kobryn-Gusarov-Kovalenko;

NOSPEEDUP - Switch off the automatic adjustment of the MDIIS step width.

FLUIDPARAM Temper - temperature;

DielConst - dielectric constant (only used when theory=DRISM);

UTotDens - units for total density: G/CM3, KG/M3, 1/A3 are valid units followed by the density value.

OUTPUT PrintLev - print level;

File - common base name for output files;

OutList - which information will be written to additional output files (no additional information will be written by default): G - distribution function, C - direct correlation function, H - total correlation function, U - interaction function, X - ".xvv" file (Only required up to ADF2019), T - thermodynamics, D - density and temperature derivatives of all correlation functions (automatically set, when DERIV option is chosen in SOLUTE block).

GRID Nr - the size of the 1D-RISM grid, must be a power of 2 (max. value: 16384);

dR - mesh size in Angstrom;

MaxRout - plot range in direct space;

MaxKOut - plot range in reciprocal space.

MDIIS N - number of vectors in the DIIS space;

Step - step size;

Tolerance - convergence criterion.

ELSTAT LRsmear - smearing parameters for coulomb potential;

Abdcor - switching parameter for dielectric correction (only used if theory=DRISM).

`ITER` Ksave - save the current solution every Ksave steps (0 - do not save);

Kshow - print convergence progress every Kshow steps;

Max - maximum number of iterations.

`DODEPS` If this block is set, the temperature and density dependence of the dielectric constant is included in the calculation of temperature and density derivatives (OutList=D);

Alpha - isobaric thermal expansion coefficient of the solvent (in 1/K, not set by default!);

Kappa - isothermal compressibility of the solvent (in 1/Pa, not set by default!);

dedtp - temperature derivative of the dielectric constant at constant pressure (in 1/K, not set by default!);

dedpt - pressure derivative of the dielectric constant at constant temperature (in 1/Pa, not set by default!).

Restart1D - use the given restart file (of the RISMDATA type) to restart the 1D-RISM calculation.

```
SOLUTE ArbitrarySoluteName
  outlist=M  closure=KH  xvvfile=solvent.xvv outfile=rism3d
  Nis=10  DELOZ=0.5 TOLOZ=2.0e-6
  Ksave=-1  Kshow=1  Maxste=10000
  Output=4
  CHRGLVL=ZLM
  GRIDSTEP=0.33
  BOXEXTENT=12.0
SUBEND
```

Outlist - output requested: M - detailed output for the excess chemical potential , F - some corrections (GF, PC+, UC, …) to the excess chemical potential , G - distribution function, C - direct correlation function, H - total correlation function;

Closure - closure for the 3D problem: KH – Kovalenko-Hirata, HNC - hypernetted chain approximations, PSEn - partial series expansion of HNC up to order n (n=1,2,…; n=1 is identical to KH), KGK - Kobryn-Gusarov-Kovalenko;

RBC - Use the current closure in combination with the repulsive bridge correction (incompatible with analytical gradients);

UCA - The value of the a-parameter in the universal correction (a*V + b, mandatory to compute the UC correction);

UCB - The value of the b-parameter in the universal correction (a*V + b, mandatory to compute the UC correction);

Xvvfile - name of the file with the results of the 1D-RISM calculation specified in the RISM1D keyword above, with .xvv appended to it (Only used up to ADF2019);

Outfile - name of the output text files;

Output - print level;

CHRGLVL - which charges computed by ADF to use. This can be ZLM - use the electron density (default), CHELPG - use point charges fitted to the electron density in analogy to the CHELPG procedure, RESP - same as CHELPG but with a harmonic restraint to keep the charges small, MM - use constant point charges given in the System block, MDCq, MDCd, MDCm, or EXMDC (labeled EXACT in older versions). The MDCq, MDCd, MDCm or EXMDC levels can not be combined with analytical gradients;

OLDZLM - reproduce the exact ZLM results obtained with ADF2019 and ADF2020;

RESPWEIGHT - pre-factor of the harmonic restraint in the RESP-like procedure, default value: 0.05;

VDWTYPE - functional form of the van der Waals interaction requested: 12-6 (default), 9-6;

SPHINT - calculate the spherical averaged distribution function around the given atom: 1,2,… - atom in input order , 0 - Center of mass;

SOLVPOS - find the most likely positions of a solvent site: 1,2,… - site in input order , 0 - all sites;

OLDCODE - run the old version of the code (ADF2019 and older), should only be used by experts;

DERIV - compute the temperature and density derivative of the excess chemical potential analytically (only exact for CHRGLVL=MM or pure Lennard-Jones fluids). When DODEPS is set in the RISM1D block, enthalpy and entropy of solvation are calculated as well;

NOSPEEDUP - switch off the automatic adjustment of the MDIIS step width;

GRIDSTEP - Grid spacing, in Angstrom;

BOXEXTENT - Extent of the grid, in Angstrom, beyond the space occupied by atoms in each direction.

The Nis, DELOZ, and TOLOZ have the same meaning for 3D-RISM as parameters of the MDIIS keyword of the RISM1D block. Likewise, Ksave, Kshow, and Maxste are analogous to the parameters of the ITER key in RISM1D.

Parameters for some solvents

The following, compiled set of solvent parameters is taken from the literature (see Ref.[?] and references therein for these and additional solvents).

Table 4.2: **Water Weight=18.015 nAtoms=2**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
O	-0.8476	3.166	0.1554	0.000000	0.000000	-0.064605
H	0.4238	1.000	0.0560	0.000000	0.816495	0.512747
				0.000000	-0.816495	0.512747

Table 4.3: **Methanol Weight=32.04 nAtoms=3**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
H	0.435	1.000	0.0560	0.00000	0.00000	0.00000
O	-0.700	3.070	0.1700	0.00000	0.00000	0.94500
CH3	0.265	3.775	0.2070	1.35136	0.00000	1.39716

Table 4.4: **Ethanol Weight=47.07 nAtoms=4**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
H	0.435	1.000	0.0560	0.000000	0.000000	0.000000
O	-0.700	3.070	0.1700	0.000000	0.000000	0.945000
CH2	0.265	3.775	0.2070	1.356103	0.000000	1.398746
CH3	0.000	3.905	0.1750	1.342751	0.000000	2.928687

Table 4.5: **Isopropanol Weight=60.10 nAtoms=4**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
H	0.435	1.000	0.0560	0.000000	0.000000	0.000000
O	-0.700	3.070	0.1700	0.000000	0.000000	0.945000
CH	0.265	3.850	0.0800	1.356103	0.000000	1.398746
CH3	0.000	3.910	0.1600	1.578230	1.268465	2.224915
				1.578230	-1.268465	2.224915

Table 4.6: **Acetonitrile Weight=41.052 nAtoms=3**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
CH3	0.150	3.775	0.2070	1.458000	0.000000	0.000000
C	0.280	3.650	0.1500	0.000000	0.000000	0.000000
N	-0.430	3.200	0.1700	-1.157000	0.000000	0.000000

Table 4.7: **Dimethylsulfoxide Weight=78.13 nAtoms=3**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
O	-0.459	2.930	0.2800	0.000000	0.000000	1.530000
S	0.139	3.560	0.3950	0.000000	0.000000	0.000000
CH3	0.160	3.810	0.1600	0.000000	1.716700	-0.541300
				-1.665300	-0.416800	-0.541300

Table 4.8: **Acetone Weight=58.08 nAtoms=3**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
O	-0.424	2.960	0.2100	0.000000	0.000000	-1.208737
C	0.300	3.750	0.1050	0.000000	0.000000	0.013263
CH3	0.062	3.910	0.1600	0.000000	1.286301	0.798425
				0.000000	-1.286301	0.798425

Table 4.9: **Dichloromethane Weight=84.93 nAtoms=3**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
C	-0.364	3.400	0.1094	0.000000	0.000000	-0.814053
H	0.218	2.293	0.0157	-0.926635	0.000000	-1.406798
				0.926635	0.000000	-1.406798
Cl	-0.036	3.564	0.2550	0.000000	-1.451416	0.177892
				0.000000	1.451416	0.177892

Table 4.10: **Chloroform Weight=119.38 nAtoms=3**

Atom	z_α / a.u.	σ_α / Å	ϵ_α / kcal/mol	X/Å	Y/Å	Z/Å
C	-0.386	3.400	0.1094	0.000000	0.000000	0.463135
H	0.266	2.115	0.0157	0.000000	0.000000	1.563010
Cl	0.040	3.564	0.2550	1.607217	-0.475149	-0.067114
				-0.392118	1.629465	-0.067114
				-1.215099	-1.154317	-0.067114

OPLS-AA Parameters for common atoms and atom group

The table below contains possible sigma and epsilon parameters for some of the most common solvent groups collected from Ref.[10][11][12]. These parameters are kindly provided by Leonardo Costa.

---

[10] W.L. Jorgensen, J.D. Madura and C.J. Swenson, *Optimized intermolecular potential functions for liquid hydrocarbons,* Journal of the American Chemical Society 106, 6638 (1984) (https://doi.org/10.1021/ja00334a030)

[11] A.E. Kobryn and A. Kovalenko, *Molecular theory of hydrodynamic boundary conditions in nanofluidics,* Journal of Chemical Physics 129, 134701 (2008) (https://doi.org/10.1063/1.2972978)

[12] O. Acevedo and W.L. Jorgensen, *Influence of Inter- and Intramolecular Hydrogen Bonding on Kemp Decarboxylations from QM/MM Simulations,* Journal of the American Chemical Society 127, 8829 (2005) (https://doi.org/10.1021/ja051793y)

Table 4.11: **All atoms model**

Atom	Example	σ_α / Å	ϵ_α / kcal/mol	Reference
C (SP3)	methane	3.73	0.294	?
C (SP3)	neopentane	3.80	0.050	?
C (SP2)	isobutene	3.75	0.105	?
C (SP)	acetonitrile	3.40	0.099	?
C (arom)	benzene	3.55	0.07	?
C	chloroform	4.10	0.05	?
H	O—H	0.7	0.046	?
H	hydrocarbons	2.42	0.03	?
O	water	3.166	0.1554	?
O	alcohol	3.07	0.17	?
O	sulfoxide	2.93	0.28	?
N (SP)	acetonitrile	3.3	0.099	?
S	sulfoxide	3.56	0.395	?
Cl	chloroform	3.40	0.300	?

Table 4.12: **United atom model**

Atom	Example	σ_α / Å	ϵ_α / kcal/mol	Reference
CH (SP3)	isobutane	3.850	0.080	?
CH (SP2)	2-butene	3.800	0.115	?
CH (arom)	benzene	3.750	0.110	?
CH2 (SP3)	n-butane	3.905	0.118	?
CH2 (SP2)	1-butene	3.850	0.140	?
CH3	hydrocarbon	3.775	0.207	?
CH3	acetonitrile	3.6	0.38	?
CH3	sulfoxide	3.81	0.16	?
NH2	amine	3.3	0.17	?

4.5 Electric Field: Homogeneous, Point Charges, Polarizability

A homogeneous external electric field and/or the field due to point charges can be included in the Fock operator. Either can be applied in both a Single-Point calculation or in geometry optimization. When applied in geometry optimization, it will allow for the molecule to rotate with respect to point charges or the field vector but not translate. Rigid translation is explicitly disabled to avoid drifting of the molecule in the external electric field.

Homogeneous electric fields, point charges and external dipole moments should be defined in the System block, which is part of the AMS driver input. This section of the AMS driver manual describes the various input options.

Symmetry

The homogeneous electric field and the point charge fields may polarize the electronic charge density. This must be accounted for in the point group symmetry. If symmetry is not specified in input, ADF uses symmetry NOSYM. You can specify a symmetry, but ADF will not check its correctness. An incorrect symmetry may lead to incorrect results.

Bonding energy

The bonding energy is computed as: the energy of the molecule in the field minus the energy of the constituent fragments without the field.

Polarizability and hyperpolarizability

ADF supports a direct calculation of the (hyper) polarizability (see section on Spectroscopic Properties). The static (hyper) polarizabilities could also be computed by applying a small homogeneous field and comparing the results with the field-free data.

CHAPTER

FIVE

STRUCTURE AND REACTIVITY

See also:

ADF-GUI tutorials: geometry optimization, vibrational frequencies, spin coupling in Fe4S4
Examples: *geometry optimization* (page 702), *reactivity* (page 712)

5.1 Single point calculations

A single point calculation calculates one point on the potential energy surface.

- Single point calculations section of the AMS manual.

5.2 Geometry optimization

A geometry optimization tries to find a (local) minimum on the potential energy surface, where the gradients are zero and the Hessian only has positive eigen values. With AMS one can do an optimization using delocalized or Cartesian coordinates. Constraints (also block constraints) and restraints are possible. An initial Hessain can be provided.

- Geometry Optimization section of the AMS manual.

5.3 Transition state search

A transition state corresponds to a saddle point on the potential energy surface, where the gradients are zero and the Hessian has one negative eigen value.

- Transition state search section of the AMS manual.

A reaction coordinate for transition state search can be specified.

5.4 Linear Transit, PES scan

The PES scan task in AMS allows users to scan the potential energy surface of a system along one or multiple degrees of freedom, while relaxing all other degrees of freedom. If only one coordinate is scanned, this kind of calculation is usually just called a linear transit.

- Linear Transit, PES scan section of the AMS manual.

5.5 Nudged Elastic Band (NEB)

The Nudged Elastic Band (NEB) method can be used to find a reaction path and the transition state between a reactant and a product state.

- Nudged Elastic Band (NEB) section of the AMS manual.

5.6 Intrinsic Reaction Coordinate (IRC)

The path of a chemical reaction can be traced from the transition state to the products and/or reactants using the IRC method.

- Intrinsic Reaction Coordinate (IRC) section of the AMS manual.

5.7 Excited state optimizations

ADF can calculate the nuclear gradient for a particular electronically excited state, which makes it possible to do excited state optimizations.

- *Excited state gradient section of the ADF manual* (page 185),
- Excited state optimizations section of the AMS manual.

5.8 Molecular dynamics

Molecular dynamics can be used to simulate the evolution of a system in time.

- Molecular dynamics section of the AMS manual.

CHAPTER

SIX

GRADIENTS, HESSIAN, THERMODYNAMICS

6.1 Gradients

A nuclear gradient is the first derivative of the energy with respect to the nuclear coordinates. The nuclear gradients are not forces, the difference being the sign.

- Nuclear gradients section of the AMS manual.

6.2 Hessian

A Hessian is the second derivative of the energy with respect to the nuclear coordinates.

- Hessian section of the AMS manual.

One can also calculate a partial Hessian. The Hessian calculation is related to the calculation of IR frequencies, see the IR frequencies and normal modes section of the AMS manual.

6.3 PES point character

The AMS driver can quickly, and without calculating the full Hessian, characterize a PES point as a local minimum, a transition state, a higher order saddle point, or a non-stationary point.

- PES point character section of the AMS manual.

6.4 Thermodynamics, gas phase Gibbs free energy

At the end of a completed IR Frequencies (normal modes) calculation, a survey is given of thermodynamic properties: entropy, internal energy, constant volume heat capacity, enthalpy and Gibbs free energy, see:

- IR frequencies section of the AMS manual
  - Thermodynamics
  - Gibbs free energy change for a gas phase reaction

CHAPTER

SEVEN

SPECTROSCOPIC PROPERTIES

See also:

ADF-GUI tutorials: excitation energies, vibrational frequencies

Examples: *IR spectra* (page 742), *excitation energies* (page 767), *response properties* (page 824), *NMR* (page 871), *ESR* (page 890), *EFG* (page 906)

7.1 Vibrational Spectroscopy

See the Vibrational Spectroscopy section of the AMS manual.

7.1.1 IR frequencies and normal modes

See the IR frequencies and normal modes section of the AMS manual.

The Hessian that is needed can be calculated numerically (for almost all XC functionals) or analytically (for a limited amount of XC functionals, mostly LDA or GGA). This can be set in the AMS part of the input.

```
$ADFBIN/ams <<eor
   NormalModes
      Hessian [Auto | Analytical | Numerical]
   End
   ...
eor
```

Analytical Hessian

The analytical second derivatives (Hessian) implementation in ADF is based on Ref.[1][2][3]. The analytical Hessian is as accurate as the numerical Hessian for the same integration accuracy, but can be up to 3 to 5 times quicker to compute, depending on the molecule, integration grid parameters, and choice of basis set. Using the analytical Hessian AMS will calculate the vibrational frequencies and normal modes.

---

[1] A. Bérces, R. M. Dickson, L. Fan, H. Jacobsen, D. Swerhone and T. Ziegler, *An implementation of the coupled perturbed Kohn-Sham equations: perturbation due to nuclear displacements*, Computer Physics Communications 100, 247 (1997) (https://doi.org/10.1016/S0010-4655(96)00120-8)

[2] H. Jacobsen, A. Bérces, D. Swerhone and T. Ziegler, *Analytic second derivatives of molecular energies: a density functional implementation*, Computer Physics Communications 100, 263 (1997) (https://doi.org/10.1016/S0010-4655(96)00119-1)

[3] S.K. Wolff, *Analytical second derivatives in the Amsterdam density functional package*, International Journal of Quantum Chemistry 104, 645 (2005) (https://doi.org/10.1002/qua.20653)

Warning: In previous versions ADF2006-ADF2020 contained a bug for analytical frequencies calculations of spin polarized molecules using PBEc.

Bug PBEc The bug for analytical frequencies calculations of spin-polarized molecules using PBEc affects calculations with the exchange-correlation functionals PBE, OPBE, RPBE, revPBE, and S12g. This bug was introduced in ADF2006 and fixed in AMS2020.103. Analytical frequencies using LibXC were not affected. Workaround for older versions is to use numerical frequencies. For AMS2020 one should use AMS2020.103 or later. For ADF2019.3 one should use ADF2019.307.

Calculating the analytical Hessian requires the solution of the Coupled Perturbed Kohn-Sham (CPKS) equations, which is an iterative process. This part of the process is of order 3 x *number of atoms*, and is generally the main bottle neck in calculating the frequencies. The immediate result of the solution of the CPKS equations is the U1 matrix, the components of which are closely related to the derivatives of the MO coefficients. One of the adjustable parameters in the input of an analytical frequencies calculation can be used to control the accuracy of the U1 matrix components.

One disadvantage in calculating analytical Hessian is that the range of exchange-correlation functionals is limited. This is because derivative formulas have to be derived for each exchange-correlation functional in ADF, which is not an straight forward task. Here are the currently available functionals:

LDA: XONLY, VWN, STOLL, PW92

Exchange GGA: Becke88, OPTx, PBEx, rPBEx, revPBEx, S12g

Correlation GGA: LYP, Perdew86, PBEc

XC GGA shortcuts: BP86, PBE, RPBE, revPBE, BLYP, OLYP, OPBE

Any functional not mentioned above is not implemented, including PW91 and Hartree-Fock.

A note of caution: For accurate frequencies it is especially important to also have an accurately optimized geometry. During a geometry optimization the integration accuracy is set by default to “Normal”, and so the resulting frequencies will also have this level of integration accuracy while it may be desirable to have frequencies computed with a higher accuracy. One might consider using Good NumericalQuality (or BeckeGrid quality) and set the convergence criteria for the geometry optimization tighter.

```
AnalyticalFreq
   B1Size float
   B1Thresh float
   Check_CPKS_From_Iteration integer
   Debug string
   Hessian [reflect | average]
   Max_CPKS_Iterations integer
   Print string
   PrintNormalModeAnalysis Yes/No
   U1_Accuracy float
End
```

`AnalyticalFreq`

Type Block

Description Define options for analytical frequencies.

`B1Size`

Type Float

Description Sparse grid max memory size

`B1Thresh`

Type Float

Default value 1e-10

Description MMGF_DENB1 and MMGF_GRADB1 cutoff values

`Check_CPKS_From_Iteration`

Type Integer

Default value 1

Description Solution of the CPKS equations is an iterative process, and convergence is achieved if the difference between U1 matrix of successive iterations falls below a certain threshold. This key can be used to determine at which iteration the checking should start taking place.

`Debug`

Type String

Description For debugging purposes. Options: fit, hessian, b1, densities, numbers, symmetry, all.

`Hessian`

Type Multiple Choice

Default value reflect

Options [reflect, average]

Description Whether the final Hessian is obtained by reflecting or averaging?

`Max_CPKS_Iterations`

Type Integer

Default value 20

Description Calculating the analytical frequencies requires the solution of the Coupled Perturbed Kohn-Sham (CPKS) equations, which is an iterative process. If convergence is not achieved (a warning will be printed in the output if this is the case) then this subkey can be used to increase the number of iterations, although convergence is not guaranteed. The user required accuracy of the U1 matrix, as well as the ADF integration accuracy, can effect the rates of convergence.

`Print`

Type String

Description Primarily for debugging purposes. Options: eigs, u1, parts. Choosing EIGS results in the print out of the MO eigenvectors, while U1 results in the print out of the U1 matrices. Except for small molecules this will result in a lot of data being output, and so they are not recommended. Choosing PARTS results in the print out of various sub-hessians that add up to give the final analytical hessian.

`PrintNormalModeAnalysis`

Type Bool

Default value No

Description Request ADF to print analysis of the normal modes independently of AMS.

`U1_Accuracy`

Type Float

Default value 5.0

Description Solution of the CPKS equations is an iterative process, and convergence is achieved if the difference between U1 matrix of successive iterations falls below a certain threshold. This subkey can be used to set the threshold. The accuracy of the U1 will be 10**(-x). So, the higher the number the more accurate the U1 will be. While this parameter effects the accuracy of the frequencies, other factors also effect the accuracy of the frequencies, especially the ADF integration accuracy.

7.1.2 (Resonance) Raman

See the (Resonance) Raman section of the AMS manual. As explained there one has two methods for calculating resonance Raman. One method uses the geometrical derivatives of the frequency-dependent polarizability with a finite lifetime, and the other method uses excited-state energy gradients projected onto the ground-state normal modes.

For (resonance) Raman calculations that use the geometrical derivatives of the frequency-dependent polarizability by default AORESPONSE will be used. To use RESPONSE instead of AORESPONSE include the RESPONSE block key in the Engine ADF part of the input:

```
Engine ADF
  Response
  End
EndEngine
```

7.1.3 VROA: (Resonance) vibrational Raman optical activity

See the VROA section of the AMS manual.

7.1.4 VCD: Vibrational Circular Dichroism

New in AMS2020 is that one can calculate VCD also for open-shell systems in a spin-unrestricted calculation.

See the VCD section of the AMS manual.

7.1.5 Vibrational Polarizabilities

Following a vibrational frequency calculation ADF will automatically print the pure vibrational polarizability[4] of the system. The full polarizability tensor is printed in atomic units along with the isotropic average.

See the Vibrational Polarizabilities section of the AMS manual.

---

[4] M. J. Cohen, A. Willetts, R. D. Amos, and N. C. Handy, *Vibrational contributions to static polarizabilities and hyperpolarizabilities*, Journal of Chemical Physics 100, 4467 (1994) (https://doi.org/10.1063/1.474503)

7.2 Time-dependent DFT

Excitation energies, frequency-dependent (hyper) polarizabilities, Van der Waals dispersion coefficients, higher multipole polarizabilities, Raman scattering intensities and depolarization ratios of closed-shell molecules are all available in ADF[1][2] as applications of time-dependent DFT (TDDFT) ; see[3] for a review.

New in ADF2004.01 is the calculation of circular dichroism (CD) spectra, and the calculation of the optical rotation (dispersion).

Starting from the ADF2005.01 version it is possible to calculate excitation energies for open-shell systems with TDDFT, including spin-flip excitation energies. New in ADF2005.01 is the possibility to use time-dependent current-density functional theory (TDCDFT).

New in ADF2006.01 is the possibility to calculate excitation energies for closed-shell molecules including spin-orbit coupling.

New in ADF2008.01 is the possibility to calculate lifetime effects in (dynamic) polarizabilities (AORESPONSE key).

The input description for these properties is split in three parts: (a) general advice and remarks, (b) excitation energies, and (c) frequency-dependent (hyper) polarizabilities (two alternative implementation: RESPONSE key and AORESPONSE key) and related properties.

Tip: See also the ADF Advanced TDDFT tutorial: TDDFT Study of 3 different Dihydroxyanthraquinones

7.2.1 General remarks on the Response and Excitation functionality

`Symmetry` As in calculations without TDDFT the symmetry is automatically detected from the input atomic coordinates and need not be specified, except in the following case: infinite symmetries cannot be handled in the current release (ATOM, C(lin), D(lin)). For such symmetries a subgroup with finite symmetry must be specified in the input. The usual orientation requirements apply. If higher multipole polarizabilities are required, it may also be necessary to use a lower subgroup (the program will stop with an error message otherwise). For verification of results one can always compare to a NOSYM calculation.

`Closed-shell` The current implementation often supports only closed-shell molecules. If occupation numbers other than 0 or 2 are used the program will detect this, (but only at a later stage of the calculation) and abort. All 'RESPONSE' calculations must be spin-restricted.

`Open-shell` Excitation energies can be obtained for open-shell systems in a spin-unrestricted TDDFT calculation. Spin-flip excitation energies can only be obtained in a spin-unrestricted TDDFT calculation.

`Atomic coordinates in a RAMAN calculation` Atomic coordinate displacements in a RAMAN calculation must be Cartesian, not Z-matrix. Furthermore, the current implementation does not yet support constrained displacements, i.e. you must use *all* atomic coordinate displacements.

`Use of diffuse functions` The properties described here may require diffuse functions to be added to the basis (and fit) sets. Poor results will be obtained if the user is unaware of this. As a general rule, diffuse functions are more important for smaller than for larger molecules, more important for hyperpolarizabilities than for normal polarizabilities, more important for high-lying excitation energies (Rydberg states) than for low-lying excitations, more important for higher multipole polarizabilities than for dipole polarizabilities. The user should know when diffuse functions are required and when they are not: the program will not check anything in this respect. For example, in a study on low-lying excitation energies of a large molecule, diffuse functions will usually have little

---

[1] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Implementation of time-dependent density functional response equations*, Computer Physics Communications 118, 119 (1999) (https://doi.org/10.1016/S0010-4655(99)00187-3)

[2] S.J.A. van Gisbergen, *Molecular Response Property Calculations using Time Dependent Density Functional Theory*, in *Chemistry*. 1998, (http://downloads.scm.com/Doc/gisbergen.pdf.tar.gz) Vrije Universiteit: Amsterdam. p. 190.

[3] E.K.U. Gross, J.F. Dobson and Petersilka, in *Density Functional Theory*, R.F. Nalewajski, Editor. 1996, Springer: Heidelberg.

effect, whereas a hyperpolarizability calculation on a small molecule is pointless unless diffuse functions are included. Diffuse even tempered basis sets are included in the ET/ directory in `$AMSHOME/atomicdata/ADF`), for the elements H-Kr. Somewhat older basis sets can be found in the Special/Vdiff directory in `$AMSHOME/atomicdata/ADF`. For other atoms, the user will have to add diffuse basis and fit functions to the existing data base sets. It is not necessary to start from basis V as was done for the basis sets in Special/Vdiff. For example, for heavier elements it may be a good idea to start from the ZORA/QZ4P basis sets. It may be expected that even more extensive basis sets will come available in the future, when usage and experience increase.

`Linear dependency in basis` If large diffuse basis sets are used, or if diffuse functions are used for atoms that are not far apart the calculation may suffer from numerical problems because of (near-) linear dependencies in the basis set. The user should be aware of this danger and use the DEPENDENCY key to check and solve this.

`The LINEARSCALING input keyword` For reasons of numerical robustness and safety rather strict defaults apply for the neglect of tails of basis and fit functions (see the key LINEARSCALING) in a Response or Excitation calculation. This may result in longer CPU times than needed for non-TDDFT runs, in particular for larger molecules. Possibly this precaution is not necessary, but we have not yet tested this sufficiently to relax the tightened defaults.

`Relativistic effects` The Response and Excitations options can be combined with scalar relativistic options (ZORA or Pauli). The one-electron relativistic orbitals and orbital energies are then used as input for the property calculation. Spin-orbit effects have been incorporated only in this part of the code (excitation energies). In case of a ZORA calculation, the so-called 'scaled' orbital energies are used as default.

`Choice of XC potential` For properties that depend strongly on the outer region of the molecule (high-lying excitation energies, (hyper) polarizabilities), it may be important to use a XC potential with correct asymptotic behavior (approaching -1/r as r tends to infinity). Finally, several asymptotically correct XC potentials have been implemented in ADF, like the LB94 potential[4] and the statistical average of orbital potentials SAOP[8][6]. SAOP is recommended. Because of the correct asymptotic behavior the SAOP potential (and the LB94 potential) can describe Rydberg states correctly. The potentials based upon the so-called GRadient regulated seamless connection of model potentials (GRAC) for the inner and the outer region[5][7] have similar performance to SAOP, but have the disadvantage that the ionization energy of the molecule has to be used as input.

With the SAOP and GRAC functionals for the potential (as well as for LB94), the XC potential is computed from the exact charge density for reasons of stability and robustness (whereas for other functions the (cheaper) fit density is used). This implies that computation times may be longer. Another 'side effect' is that, since there is no energy expression corresponding to these potentials, the final (bonding) energy of such calculations uses another GGA and hence the energy result is not (exactly) consistent with the SCF procedure. Note, finally, that these potentials have been found to be not suitable for geometry optimizations because they maybe are not sufficiently accurate in the bonding region, see the discussion of the XC input key. Applications with SAOP to (hyper)polarizabilities and excitation energies, also for Rydberg transitions, can be found in[?] and with SAOP and Becke-Perdew-GRAC in[??].

---

[4] R. van Leeuwen and E.J. Baerends, *Exchange-correlation potential with correct asymptotic behavior*, Physical Review A 49, 2421 (1994) (https://doi.org/10.1103/PhysRevA.49.2421)

[8] O.V. Gritsenko, P.R.T. Schipper and E.J. Baerends, *Approximation of the exchange-correlation Kohn-Sham potential with a statistical average of different orbital model potentials*, Chemical Physics Letters 302, 199 (1999) (https://doi.org/10.1016/S0009-2614(99)00128-1)

[6] P.R.T. Schipper, O.V. Gritsenko, S.J.A. van Gisbergen and E.J. Baerends, *Molecular calculations of excitation energies and (hyper)polarizabilities with a statistical average of orbital model exchange-correlation potentials*, Journal of Chemical Physics 112, 1344 (2000) (https://doi.org/10.1063/1.480688)

[5] M. Grüning, O.V. Gritsenko, S.J.A. van Gisbergen and E.J. Baerends, *Shape corrections to exchange-correlation Kohn-Sham potentials by gradient-regulated seamless connection of model potentials for inner and outer region*, Journal of Chemical Physics 114, 652 (2001) (https://doi.org/10.1063/1.1327260)

[7] M. Grüning, O.V. Gritsenko, S.J.A. van Gisbergen and E.J. Baerends, *On the required shape correction to the LDA and GGA Kohn Sham potentials for molecular response calculations of (hyper)polarizabilities and excitation energies*, Journal of Chemical Physics 116, 9591 (2002) (https://doi.org/10.1063/1.1476007)

Applications with the old LB94 potential to response calculations can be found in refs.[9] (polarizabilities),[20][21][22] (hyperpolarizabilities),[10] (high-lying excitation energies),[11] (multipole polarizabilities and dispersion coefficients).

`XC kernel` If most cases the adiabatic local density approximated (ALDA) kernel is used in the TDDFT functionality. In case of calculating excitation energies with a hybrid functionals the Hartree-Fock percentage times the Hartree-Fock kernel plus one minus the Hartree-Fock percentage times the ALDA kernel is used. For the LibXC range separated functionals, like CAM-B3LYP, starting from ADF2016.102 the kernel consists of range separated ALDA plus the kernel of the range separated exact exchange part. In ADF2016.101 the kernel for LibXC range separated functionals, like CAM-B3LYP, was using a 100% ALDA plus range separated exact exchange kernel (the ALDA part was not range-separated corrected).

For excitation energy calculations in some cases the full (non-ALDA) kernel can be evaluated, see the *full XC kernel description* (page 163). The Full kernel can not be used in combination with symmetry or excited state geometry optimizations.

`COSMO` The COSMO model for solvation is a cheap method to include solvation effects in the TDDFT applications, see the SOLVATION key. Note that in TDDFT calculations one may have to use two dielectric constants. The reason is that the electronic transition is so fast that only the electronic component of the solvent dielectric can respond, i.e., one should use the optical part of the dielectric constant. This is typically referred to as non-equilibrium solvation. The optical dielectric constant can be obtaining from the (frequency dependent) refractive index n of the solvent as: $\epsilon_{\text{opt}} = \text{n}^2$. One can use the argument NEQL in the subkey SOLV of the key SOLVATION to set a value for the optical dielectric constant. No dielectric constant in the response might be closer to the optical dielectric constant than using the full dielectric constant, This can be achieved if one includes the subkey NOCSMRSP in the block key SOLVATION, such that the induced electronic charges do not influence the COSMO surface charges. However, if one does geometry optimization then the full dielectric constant should be used in the TDDFT simulations since the solvent dielectric now has time to fully respond. The default is that the full dielectric constant is used in the TDDFT calculations.

`Accuracy check list` As mentioned before, the TDDFT module is relatively new and not extensively tested for a wide range of applications. Therefore, we strongly recommend the user to build experience about aspects that may affect the accuracy of TDDFT results. In particular we advise to 'experiment' with

- Varying integration accuracy
- Varying the SCF convergence
- Varying the ORTHONORMALITY and TOLERANCE values in an Excitation calculation
- Varying the linearscaling parameters
- Using diffuse functions
- Using the Dependency key
- Applying the ZORA relativistic corrections for molecules containing heavy nuclei
- Using an asymptotically correct XC potential such as SAOP

---

[9] S.J.A. van Gisbergen, V.P. Osinga, O.V. Gritsenko, R. van Leeuwen, J.G. Snijders and E.J. Baerends, *Improved density functional theory results for frequency-dependent polarizabilities, by the use of an exchange-correlation potential with correct asymptotic behavior*, Journal of Chemical Physics 105, 3142 (1996) (https://doi.org/10.1063/1.472182)

[20] S.J.A. van Gisbergen, J.G. Snijders, and E.J. Baerends, *Time-dependent Density Functional Results for the Dynamic Hyperpolarizability of* C_{60}, Physical Review Letters 78, 3097 (1997) (https://doi.org/10.1103/PhysRevLett.78.3097)

[21] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Calculating frequency-dependent hyperpolarizabilities using time-dependent density functional theory*, Journal of Chemical Physics 109, 10644 (1998) (https://doi.org/10.1063/1.477762)

[22] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Accurate density functional calculations on frequency-dependent hyperpolarizabilities of small molecules*, Journal of Chemical Physics 109, 10657 (1998) (https://doi.org/10.1063/1.477763)

[10] M.E. Casida, C. Jamorski, K.C. Casida and D.R. Salahub, *Molecular excitation energies to high-lying bound states from time-dependent density-functional response theory: Characterization and correction of the time-dependent local density approximation ionization threshold*, Journal of Chemical Physics 108, 4439 (1998) (https://doi.org/10.1063/1.475855)

[11] V.P. Osinga, S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Density functional results for isotropic and anisotropic multipole polarizabilities and* C_6 , C_7 , and C_8 *Van der Waals dispersion coefficients for molecules*, Journal of Chemical Physics 106, 5091 (1997) (https://doi.org/10.1063/1.473555)

7.2.2 Analysis options for TDDFT (excitation energies and polarizabilities)

Several options are available to obtain more detailed results than a few bare numbers for excitation energies, oscillator strengths, transition dipole moments, and (hyper)polarizabilities. For a zero-order understanding of which occupied and virtual orbitals play an important in the polarizability or intensity of an absorption peak,

it may be useful to know the values of the dipole matrix elements between (ground-state) occupied and virtual Kohn-Sham orbitals. If these dipole matrix elements are large for a particular occupied-virtual orbital pair, then this pair is almost certainly of great importance for the whole spectrum or polarizability. This information can be obtained by specifying somewhere in the input file (but NOT inside the RESPONSE or EXCITATION block keys):

```
PRINT DIPOLEMAT
```

7.2.3 Time-dependent Current DFT

The time-dependent current-density-functional (TDCDFT) implementation is built entirely upon the normal TDDFT implementation. Therefore all general remarks that are made for the TDDFT part of the program are also valid for TDCDFT. Only the polarizability and excitation energies of closed shell molecules can be calculated with TDCDFT in the present implementation.

If TDCDFT is used together with the ALDA functional (NOVK option) it will give the same results for the polarizability and excitation energies as TDDFT in a complete basis set. TDCDFT in ADF by default uses the VK functional[12][13], since this is the only current dependent functional that is known presently. Many aspects of the functional are still unknown and the functional should therefore be used with caution. The user is referred to the references for more information on when the VK functional gives good results and when not.

For more information on the implementation and applications of the TDCDFT and the VK functional please read the references[23][24][25][26]. For more details on the theory and implementation in ADF see:[14].

To activate TDCDFT and the VK functional one should add the following block key to the input file:

```
CURRENTRESPONSE
END
```

To calculate the polarizability the keyword Response can be used with the following options:

```
RESPONSE
  ALLCOMPONENTS
  Frequencies freq1 freq2 ... freqN
END
```

The block key EXCITATION can be used with all of its options.

---

[12] G. Vignale and W. Kohn, *Current-Dependent Exchange-Correlation Potential for Dynamical Linear Response Theory*, Physical Review Letters 77, 2037 (1996) (https://doi.org/10.1103/PhysRevLett.77.2037)

[13] G. Vignale and W. Kohn, in *Electronic Density Functional Theory: Recent Progress and New Direction*, ISBN13: 9780306458347, J. F. Dobson, G. Vignale, and M. P. Das, Editors. 1998, Plenum: New York.

[23] M. van Faassen, P. L. de Boeij, R. van Leeuwen, J. A. Berger and J. G. Snijders, *Ultranonlocality in Time-Dependent Current-Density-Functional Theory: Application to Conjugated Polymers*, Physical Review Letters 88, 186401 (2002) (https://doi.org/10.1103/PhysRevLett.88.186401)

[24] M. van Faassen, P. L. de Boeij, R. van Leeuwen, J. A. Berger and J. G. Snijders, *Application of time-dependent current-density-functional theory to nonlocal exchange-correlation effects in polymers*, Journal of Chemical Physics 118, 1044 (2003) (https://doi.org/10.1063/1.1529679)

[25] M. van Faassen and P. L. de Boeij, *Excitation energies for a benchmark set of molecules obtained within time-dependent current-density functional theory using the Vignale-Kohn functional*, Journal of Chemical Physics 120, 8353 (2004) (https://doi.org/10.1063/1.1697372)

[26] M. van Faassen and P. L. de Boeij, *Excitation energies of* Π *-conjugated oligomers within time-dependent current-density-functional theory*, Journal of Chemical Physics 121, 10707 (2004) (https://doi.org/10.1063/1.1810137)

[14] M. van Faassen, Time-Dependent Current-Density-Functional Theory for Molecules, (http://downloads.scm.com/Doc/faassen.pdf) PhD thesis, Rijksuniversiteit Groningen, 2004.

In ADF2012 the block keyword AORESPONSE can also be used with the Vignale-Kohn functional. The current-density is generated on the fly but otherwise the computation is based on the time dependent density response.

In default the VK functional will be applied where the NCT parametrization[15] is chosen for the transverse exchange-correlation kernel for the polarizability and singlet excitation energies (giving the best results for the systems studied so far). For triplet excitation energies the only available parametrization will be used[16]. This option is not tested much and the results are in general much worse than ALDA[?]. It is therefore suggested that VK is not used to calculate triplet excitation energies.

In the output the polarizability tensor (in case of an ALLCOMPONENTS calculation) has a different shape, the results are printed in the more intuitive order x, y, z, instead of y, z, x that the TDDFT implementation uses.

The following subkeys are available within the datablock of CURRENTRESPONSE

```
CURRENTRESPONSE
  QIANVIGNALE
  NOVK
END
```

QIANVIGNALE The QV parametrization[?] will be used for the transverse exchange-correlation kernel instead of NCT.

NOVK TDCDFT will be applied with the ALDA functional instead of the VK functional. In a complete basis this will give the same results as a TDDFT calculation.

Magnetic properties within TDCDFT

It is now possible to calculate several magnetic properties via TDCDFT, among which magnetizabilities, rotational g-tensors, NMR shielding constants, specific rotations and circular dichroism. This alternative method is based on the diamagnetic-current sum rule, which consists of rewriting the diamagnetic current in terms of the response functions. This is the magnetic analogue of the conductivity sum rule, which is invalid when an external magnetic field is applied.

Details on the method can be found in these references:[17][18]. Further details regarding the implementation can be found in the following PhD thesis as well:[19].

In order to obtain these properties, the following subkeys can be used:

```
CURRENTRESPONSE
  MAGNET
  GTENSOR
  NMRSHIELDING
  CDSPEC
  DAMPING Damping
END
```

Depending on which property is calculated, the program chooses a reference point for the external vector potential (see references). For the magnetizability, the center of electronic charges is chosen, while for the rotational g-tensor, the center of nuclear charges is picked up. This choice has no influence in the static case, and has no influence neither for the specific rotations nor for the circular dichroism. It actually only affects the dynamical magnetizabilities (the rotational g-tensor being always computed at zero frequency). Below is a brief description of each subkey and how to use them.

---

[15] R. Nifosi, S. Conti and M. P. Tosi, *Dynamic exchange-correlation potentials for the electron gas in dimensionality D=3 and D=2*, Physical Review B 58: p. 12758 (1998) (https://doi.org/10.1103/PhysRevB.58.12758)

[16] Z.X. Qian and G. Vignale, *Dynamical exchange-correlation potentials for the electron liquid in the spin channel*, Physical Review B 68, 195113 (2003) (https://doi.org/10.1103/PhysRevB.68.195113)

[17] N. Raimbault, P.L. de Boeij, P. Romaniello, and J.A. Berger, *Gauge-Invariant Calculation of Static and Dynamical Magnetic Properties from the Current Density*, Physical Review Letters 114, 066404 (2015) (https://doi.org/10.1103/PhysRevLett.114.066404)

[18] N. Raimbault, P.L. de Boeij, P. Romaniello, and J.A. Berger, *Gauge-Invariant Formulation of Circular Dichroism*, Journal of Chemical Theory and Computation 12, 3278 (2016) (https://doi.org/10.1021/acs.jctc.6b00068)

[19] N. Raimbault, Gauge-invariant magnetic properties from the current (https://tel.archives-ouvertes.fr/tel-01235055), PhD thesis, Université Paul Sabatier, LCPQ, Toulouse, 2015

MAGNET The (static or dynamical) magnetizability tensor.

GTENSOR The rotational g-tensor, calculated directly from the magnetizability.

NMRSHIELDING The NMR shielding tensor. Care should be taken when using this keyword, though, as it was observed that the results failed to match experimental and other theoretical studies when using standard basis sets except for the lightest elements. Only when using very large basis sets should the results be trusted.

CDSPEC Calculates both the specific rotation and the circular dichroism. This keyword should be used in combination with DAMPING if one wants to simulate CD spectra. The output spectrum will exhibit relatively sharp peaks. Linear-interpolating the results with a third-tier program is then necessary. This is done on purpose, so that the final spectrum could be broadened easily (only the absolute amplitudes of the peaks will be affected by this broadening).

DAMPING Damping Adds a (fixed) damping factor to the response function. This allows one to avoid divergences that may arise from the response function when close to resonance, hence helping the self-consistent cycle converge. The value for Damping should be given in eV. A value of 0.1 eV seems to be reasonable.

7.3 Excitation energies: UV/Vis, X-ray, CD, MCD

Ultraviolet-visible (UV/Vis) spectroscopy studies electronic excitations of valence electrons, whereas X-ray spectroscopy studies electronic excitations of core electrons. Excitation energies and oscillator strengths are all available in ADF as applications of time-dependent DFT (TDDFT). Excitation energies can be calculated for closed-shell as well as for open-shell molecules. It is also possible to include spin-orbit coupling and to calculate core excitations (X-ray absorption spectra). Circular dichroism (CD) is the differential absorption of left- and right-handed circularly polarized light. Starting from AMS2023 it is also possible to calculate excitation energies and oscillator strengths using the GW-Bethe-Salpeter equation (GW-BSE) formalism.

Tip: See also the ADF Advanced TDDFT tutorial: TDDFT Study of 3 different Dihydroxyanthraquinones

7.3.1 Excitation energies, UV/Vis spectra

You can perform a calculation of singlet-singlet and singlet-triplet excitation energies of a closed-shell molecule by supplying in the input file the block key EXCITATION. See the next sections for settings of technical parameters, the calculation of excitation energies for open shell molecules, inclusion of spin-orbit coupling, and the calculation of CD spectra.

```
EXCITATIONS
 EXACT
   {IRREP1 N1}
   {IRREP2 N2}
 END
 DAVIDSON
   {IRREP3 N3}
   {IRREP4 N4}
 END
 ALLOWED
 ONLYSING
 ONLYTRIP
 LOWEST nlowest
End
```

Several options can be addressed with subkeys in the data block. This functionality is based on TDDFT and consequently has a different theoretical foundation than the SCF techniques described elsewhere in this User's Guide. Two possible ways are available to solve the eigenvalue equation from which the excitation energies and oscillator strengths are obtained, of which the iterative Davidson procedure is the default. In this case, the program needs to know how many excitation energies are needed per irrep, what accuracy is required, and what type of excitation energies are required (singlet-singlet or singlet-triplet). Suitable defaults have been defined for all of these. Each of these points is discussed below.

Exact diagonalization vs. iterative Davidson procedure The most straightforward procedure is a direct diagonalization of the matrix from which the excitation energies and oscillator strengths are obtained. Since the matrix may become very large, this option is possible only for very small molecules. It can be activated by specifying the block EXACT as one of the subkeys in the Excitations data block. The default is the iterative Davidson method. A few of the lowest excitation energies and oscillator strengths are then found within an error tolerance. An advantage of the EXACT option is that additional information is produced, such as the Cauchy coefficients that determine the average dipole polarizability. The EXACT option can not be used in unrestricted calculations.

Singlet versus triplet By default, the singlet-singlet and singlet-triplet excitation energies are both calculated. The singlets are handled first, then the corresponding triplet excitation energies. One can skip one of these two parts of the calculation by specifying either ONLYSING or ONLYTRIP as a subkey in the data block.

In case of a calculation including spin-orbit coupling one can not separate the singlet-singlet and singlet-triplet excitations. The subkeys ONLYSING and ONLYTRIP are misused in this case to do a spin-restricted calculation, or a spin-polarized calculation, respectively. One should in fact only use the results of the spin-polarized calculation.

Dipole-allowed versus general excitations. If you are interested in the optical absorption spectrum, you may not want to compute singlet-triplet excitation energies, nor singlet-singlet excitation energies which, by symmetry, have zero oscillator strengths. This subkey should not be used in case of spin-orbit coupling. The subkey ALLOWED tells ADF to treat only those irreducible representations for which the oscillator strengths will be nonzero. Of course, the oscillator strengths may still be negligibly small. The ALLOWED subkey automatically implies ONLYSING. The simplest, fastest, and recommended way to obtain information about the ten lowest dipole-allowed excitation energies would be:

```
EXCITATIONS
 ALLOWED
 LOWEST 10
END
```

Which excitation energies and how many? The user can specify how many excitation energies per irrep should be calculated. If no pertaining input is available the program determines these numbers from the smallest differences between occupied and virtual Kohn-Sham orbital energies. By default it looks at the 10 lowest orbital energy differences. This number can be modified, by specifying inside the Excitation block key, for example:

```
LOWEST 30
```

One should be aware that this procedure does not guarantee that the lowest 10 (or 30) excitation energies will actually be found, since the orbital energy difference approximation to the excitation energy is rather crude. However, if the program decides on the basis of this procedure to calculate 4 excitation energies in a certain irreducible representation, these 4 excitation energies are certainly the lowest in that particular irrep.

The user has more control when the number of excitations per irrep is explicitly specified within the EXCITATION block key by the Davidson subkey:

```
DAVIDSON
 E'' 5
 T1.u 2
END
```

The DAVIDSON sub block may contain any number of records and must end with a record END. In the subkey data block a list of irreps, followed by the number of requested excitation energies is specified. Note that the irrep name may not be identical to the usual ADF name. For example E” is called EEE in ADF. The Excitation code will skip an irrep if the label is not recognized. For multidimensional irreps, only the first column is treated, because the other would produce identical output. This implies that the oscillator strengths for E-irreps have to be multiplied by 2 and the oscillator strengths for T-irreps by 3. The ALLOWED subkey should not be used if irreps are specified with the Davidson block subkey, however, the subkey ONLYSING (or ONLYTRIP) can be used in this case.

The EXACT sub-block, mentioned already above, can also be used to treat only a few irreps instead of all. The number of excitation energies does not have to be specified then.

Some early applications of the Excitation feature in ADF can be found in Refs.[39][40][41][42][43][44].

In case of closed shell molecules the unrelaxed excited state electric dipole moment is calculated, see Ref.[54]. In case of a TDDFT-gradients calculation also the relaxed excited state electric dipole moment is calculated. For the definition of unrelaxed and relaxed excited state density, see Ref.[53].

Tamm-Dancoff approximation

Excitation energies can be calculated using the Tamm-Dancoff approximation (TDA)[1] if one includes, besides the EXCITATION block key, the key TDA:

```
TDA
```

GW-Bethe-Salpeter equation (GW-BSE)

Starting from AMS2023 it is possible to calculate excitation energies using the GW-BSE method instead of TD-DFT. The implementation in ADF is described in Ref.[55] Currently, the BSE kernel is compatible with all options inside the XC block and also with the TDA and spin-orbit coupling. As GW-BSE uses the results of a GW calculation, GW-BSE should also not be used in combination with solvent models, like COSMO, or other environments. A GW-BSE calculation is performed by including the BSE key in the EXCITATION block,

```
EXCITATIONS
  ...
 BSE
END
```

In a GW-Bethe-Salpeter Equation calculation a GW calculation is performed to calculate single-particle energies and the screened electron-electron interaction which are then used to calculate optical excitation energies. In a hybrid-DFT calculation, KS-DFT single-particle energies and the unscreened electron-electron interaction are used instead to calculate optical excitation energies.

---

[39] S.J.A. van Gisbergen, F. Kootstra, P.R.T. Schipper, O.V. Gritsenko, J.G. Snijders and E.J. Baerends, *Density-functional-theory response-property calculations with accurate exchange-correlation potentials*, Physical Review A 57, 2556 (1998) (https://doi.org/10.1103/PhysRevA.57.2556)

[40] S.J.A. van Gisbergen, A. Rosa, G. Ricciardi and E.J. Baerends, *Time-dependent density functional calculations on the electronic absorption spectrum of free base porphin*, Journal of Chemical Physics 111, 2499 (1999) (https://doi.org/10.1063/1.479617)

[41] A. Rosa, G. Ricciardi, E.J. Baerends and S.J.A. van Gisbergen, *The Optical Spectra of NiP, Nipz, NiTBP, and NiPc. Electronic effects of meso-tetraaza substitution and tetrabenzoannulation*, Journal of Physical Chemistry A 105, 3311 (2001) (https://doi.org/10.1021/jp003508x)

[42] G. Ricciardi, A. Rosa and E.J. Baerends, *Ground and Excited States of Zinc Phthalocyanine studied by Density Functional Methods*, Journal of Physical Chemistry A 105, 5242 (2001) (https://doi.org/10.1021/jp0042361)

[43] S.J.A. van Gisbergen, J.A. Groeneveld, A. Rosa, J.G. Snijders and E.J.Baerends, *Excitation energies for transition metal compounds from time-dependent density functional theory. Applications to* MnO4$^-$, Ni(CO)$_4$ and Mn^2 (CO)$_{10}$, Journal of Physical Chemistry A 103, 6835 (1999) (https://doi.org/10.1021/jp991060y)

[44] A. Rosa, E.J. Baerends, S.J.A. van Gisbergen, E. van Lenthe, J.A. Groeneveld and J. G. Snijders, *Article Electronic Spectra of* M(CO)$_6$ (M = Cr, Mo, W) *Revisited by a Relativistic TDDFT Approach*, Journal of the American Chemical Society 121, 10356 (1999) (https://doi.org/10.1021/ja990747t)

[54] P. Grobas-Illobre, M. Marsili, S. Corni, M. Stener, D. Toffoli, and E. Coccia, Journal of Chemical Theory and Computation XXXX, XXX (2021) (https://doi.org/10.1021/acs.jctc.1c00211)

[53] F. Furche, R. Ahlrichs, *Adiabatic time-dependent density functional methods for excited state properties*, Journal of Chemical Physics 117, 7433 (20002) (https://doi.org/10.1063/1.1508368)

[1] S. Hirata and M. Head-Gordon, *Time-dependent density functional theory within the Tamm-Dancoff approximation*, Chemical Physics Letters 314, 291 (1999) (https://doi.org/10.1016/S0009-2614(99)01149-5)

[55] Arno Förster, Lucas Visscher, *Quasiparticle Self-Consistent GW-Bethe-Salpeter equation calculations for large chromophoric systems*, Journal of Chemical Theory and Computation 18 (11) 6779–6793 (2022) (https://doi.org/10.1021/acs.jctc.2c00531)

By default the G_0W_0 variant of the GW approximation is used in a BSE calculation. This can be changed in the GW block, see the *GW description* (page 227). Almost all options in the GW block are compatible with a BSE calculation. For open-shell systems, partially self-consistent GW options are currently not supported.

The recommended numerical settings for a GW-BSE calculation are the same as for a GW calculation, see the *MBPT description* (page 329). ADF uses `Dependency bas=5e-3` and `RIHartreeFock DependencyThreshold=5e-3` in case of (any variant of) GW and GW-BSE. This can be overwritten but it is not recommended to do so. It case of convergence problems and/or variational collapse it is recommended to switch on the *MODIFYEXCITATION* (page 168) key and restrict the particle-hole basis in which the BSE is solved to particle-hole pairs with energy differences below for instance 2 Hartree.

For instance, one could do the following:

```
EXCITATIONS
 ...
 BSE
END

MODIFYEXCITATION
   UseOccVirtRange 0.0 2.0
END

GW
...
END

MBPT
...
END
```

Full XC kernel

With XCFUN the full (non-ALDA) kernel can be evaluated, see the *XCFUN description* (page 89). To use the non-ALDA kernel the keyword FULLKERNEL should be put in the EXCITATIONS block. FULLKERNEL can be used with GGAs (including hybrids and RS functionals) but not meta-GGAs or meta-hybrids. For some of the LibXC functionals (mostly GGA's and hybrids) it also possible to use the Full (non-ALDA) kernel. FULLKERNEL can not be used in combination with excited state geometry optimizations or other response properties. If symmetry is used, recommended is to check that SYMMETRY NOSYM gives the same results.

```
XC
 ...
 XCFUN
End
EXCITATIONS
 ...
 FullKernel
END
```

```
XC
 LibXC ...
End
EXCITATIONS
 ...
 FullKernel
END
```

Singlet-triplet excitations are now also possible with FULLKERNEL.

Plasmons in Molecules

In Ref.[2] a scaling approach was used to identify plasmons in molecules. To identify plasmons in general molecules, they proposed to analyze the excitations by scaling the electron-electron interaction in the response calculation. While single-particle excitations are only slightly affected by this scaling, plasmonic excitations are sensitive to variations in the scaling parameter λ. More specifically, their squared excitation energies should increase linearly with the scaling parameter λ. Scaling of Coulomb, XC, and HF parts of the response kernel in TDDFT, can be performed using the subkeywords SCALECOUL, SCALEXC, SCALEHF in the EXCITATIONS block.

```
EXCITATIONS
   SCALECOUL scalecoul
   SCALEXC   scalexc
   SCALEHF   scalehf
END
```

SCALECOUL scalecoul Scaling of Coulomb kernel with scale parameter scalecoul.

SCALEXC scalexc Scaling of the XC-kernel (excluding a possible HF-part) with scale parameter scalexc.

SCALEHF scalehf Scaling of the HF part of the kernel with scale parameter scalehf.

Transition dipole moments between excited states

In Ref.[?] transition dipole moments (electric dipole approximation) between different excited states are calculated. In ADF these transition dipole moments between different calculated excited states can be calculated if the subkey ESESTDM is included. Spin-orbit coupling can also be included. Only implemented for closed shell molecules.

```
EXCITATIONS
   ESESTDM
END
```

ESESTDM Compute transition dipole moments between excited states.

Accuracy and other technical parameters

For details regarding the (near linear scaling and parallelized) implementation, please check Refs.[3][4].

A summary of technical parameters with their defaults is:

```
EXCITATIONS
 VECTORS 40
 TOLERANCE 1e-6
 ORTHONORMALITY 1e-8
 ITERATIONS 200
 KFWRITE 3
END
```

VECTORS vectors The maximum number of trial vectors in the Davidson algorithm for which space is allocated. If this number is small less memory will be needed, but the trial vector space is smaller and has to be collapsed more often, at the expense of CPU time. The default if usually adequate.

---

[2] S. Bernadotte, F. Evers, and C.R. Jacob, *Plasmons in Molecules*, Journal of Physical Chemistry C 117, 1863 (2013) (https://doi.org/10.1021/jp3113073)

[3] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Implementation of time-dependent density functional response equations*, Computer Physics Communications 118, 119 (1999) (https://doi.org/10.1016/S0010-4655(99)00187-3)

[4] S.J.A. van Gisbergen, C. Fonseca Guerra and E.J. Baerends, *Towards excitation energies and (hyper)polarizability calculations of large molecules. Application of parallelization and linear scaling techniques to time-dependent density functional response theory*, Journal of Computational Chemistry 21, 1511 (2000) (https://doi.org/10.1002/1096-987X(200012)21:16%3C1511::AID-JCC8%3E3.0.CO;2-C/abstract)

`TOLERANCE tolerance` Specifies the error tolerance in *the square* of the excitation energies in Hartree units. The default is probably acceptable but we recommend that you verify the results against a stricter default (e.g. 1e-8) for at least a few cases. In case of a BSE calculation, the default is 1e-5.

`ORTHONORMALITY orthonormality` The Davidson algorithm orthonormalizes its trial vectors. Increasing the default orthonormality criterion increases the CPU time somewhat, but is another useful check on the reliability of the results.

`ITERATIONS iterations` The maximum number of attempts within which the Davidson algorithm has to converge. The default appears to be adequate in most cases.

`KFWRITE kfwrite` If kfwrite is 0 then do not write contributions, transition densities, and restart vectors to adf.rkf, since this can lead to a huge adf.rkf, especially if many excitations are calculated. Default value kfwrite is 3, which means that contributions, transition densities, and restart vectors are written to adf.rkf.

7.3.2 Excitation energies for open-shell systems

Excitation energies can be obtained for open-shell systems in a spin-unrestricted TDDFT calculation[5]. To perform an open-shell TDDFT calculation one just needs to do an unrestricted SCF calculation and use the EXCITATION keyword. Presently the excitation energies can only be found with Davidson's procedure. In case of spin-orbit coupling, see the section on *approximate spin-orbit coupled excitation energies open shell molecule* (page 171).

The printed symmetry in the output in TDDFT calculations is actually the symmetry of transition density. For closed-shell systems, the symmetry of the excited state is the same as the symmetry of the transition density, while for open-shell systems, the symmetry of the excited states is the direct product between the symmetry of the transition density and the ground state symmetry. Note that the ground state symmetry of an open shell molecule is not necessarily A1.

For degenerate representations such as the 2-dimensional E-representations or the 3-dimensional T-representations, the occupation should be either fully occupied or zero. For example, for an orbital in an E-representation the α and β occupation number should be either 2 or 0. The α occupation number can of course be different from the β occupation number.

As for the spin-state, the general rule is that if the excited state mainly results from transitions from the singly occupied orbitals to virtual orbitals or from fully occupied orbitals to the singly occupied orbitals, the spin state of the excited state should roughly be the same as that of the ground state. However, if the excited state mainly comes from transitions from fully occupied orbitals to virtual orbitals, the spin state of the excited state are usually a mixture since TDDFT can only deal with single excitations within adiabatic approximation for the XC kernel[7]. Sometimes we just suppose the spin state of this kind of excited states to be the same as that of ground state[?]. In the MO → MO transitions part for the excitations of the output file, the spin of each molecular orbitals are also specified to help assign the spin state of the excited states. The transitions are always from α spin-orbital to α spin-orbital or from β spin-orbital to β spin-orbital.

7.3.3 Spin-flip excitation energies

Spin-flip excitation energies[8][9] can only be obtained in a spin-unrestricted TDDFT calculation. This can not be used in case of spin-orbit coupling. At present, the spin-flip excitation energies can only be calculated with Tamm-Dancoff approximation (TDA)[?] and Davidson's method.

To calculate spin-flip excitation energies, one must specify two keys:

---

[5] F. Wang and T. Ziegler, *Excitation energies of some d1 systems calculated using time-dependent density functional theory: an implementation of open-shell TDDFT theory for doublet-doublet excitations*, Molecular Physics 102, 2585 (2004) (https://doi.org/10.1080/0026897042000275080)

[7] Z. Rinkevicius, I. Tunell, P. Salek, O. Vahtras and H. Agren, *Restricted density functional theory of linear time-dependent properties in open-shell molecules*, Journal of Chemical Physics 119, 34 (2003) (https://doi.org/10.1063/1.1577329)

[8] F. Wang and T. Ziegler, *Time-dependent density functional theory based on a noncollinear formulation of the exchange-correlation potential*, Journal of Chemical Physics 121, 12191 (2004) (https://doi.org/10.1063/1.1821494)

[9] F. Wang and T. Ziegler, *The performance of time-dependent density functional theory based on a noncollinear exchange-correlation potential in the calculations of excitation energies*, Journal of Chemical Physics 122, 74109 (2005) (https://doi.org/10.1063/1.1844299)

```
SFTDDFT
TDA
```

anywhere in the input file in addition to the EXCITATION block keyword.

In spin-flip TDDFT, the XC kernel can be calculated directly from the XC potential. To use the LDA potential for the XC kernel, which roughly corresponds to the ALDA in ordinary TDDFT, one must specify the key

```
FORCEALDA
```

anywhere in the input file. Only calculations using the LDA potential in the SCF are fully tested. Using other GGA potentials in the SCF and using the FORCEALDA key at the same time may introduce unreasonable results, while using LB94 or SAOP potential in the SCF without the FORCEALDA key may give unstable results. Unstable results have been reported for the PW91 functional.

For open-shell molecules, spin-flip transition can result in transition to the ground state with a different S_z value, while the symmetry of the transition density is A1. The excitation energy of this transition should be zero and this can be used to test the reliability of spin-flip TDDFT.

The symmetry of the excited states can be determined in the same way as that in spin-unrestricted TDDFT calculations. As for the spin state, similar to that in the spin-unrestricted TDDFT calculations, some states may be more or less pure spin states, others may just be mixtures. The users can interpret the excited state through the transitions that contribute to this state. Note that the transitions are always from α spin-orbital to β spin-orbital in spin-flip calculations, or from β spin-orbital to α spin-orbital.

7.3.4 Select (core) excitation energies, X-ray absorption

Two methods can be used to reduce the computational costs of, for example, core excitation energies, or some other high lying excitation energy. In the state selective method scheme a guess vector for the orbital transition has to be provided. An overlap criterion is used to follow the wanted eigenvector. In this scheme the one-electron excited state configuration space remains complete, see Ref[11]. In the second scheme, the range of excitations that are calculated is modified, which means that the one-electron excited state configuration space is reduced to the interesting part, see Ref.[10]. The calculated excited states are more accurate with the state selective method if convergence is reached, however, the second scheme is more robust, and it is easier to find convergence.

These selection methods can, for example, also be used in case one calculates excitation energies as Kohn-Sham orbital energy differences, see subkey *SINGLEORBTRANS* (page 169) of the key EXCITATIONS.

For X-ray spectra it can be important to include quadrupole intensities.

State selective optimization excitation energies

The state selective method (key SELECTEXCITATION) can be used to reduce the computational costs of, for example, core excitation energies. In this scheme a guess vector for the orbital transition has to be provided. It should be used in combination with the Davidson method to calculate excitation energies. An overlap criterion is used to follow the wanted eigenvector. This method for state selective optimization of excitation energies is based on the method by Kovyrshin and Neugebauer, see Ref.[?]. This key can also be used in case of spin-orbit coupling. The use of the key SELECTEXCITATION is similar as the use of the key MODIFYEXCITATION. However, the key SELECTEXCITATION can not be used in combination with the key MODIFYEXCITATION. In the state selective method (key SELECTEXCITATION) the one-electron excited state configuration space remains complete, whereas it is reduced in case the scheme with the MODIFYEXCITATION key.

---

[11] A. Kovyrshin, J. Neugebauer, *State-selective optimization of local excited electronic states in extended systems*, Journal of Chemical Physics 133, 174114 (2010) (https://doi.org/10.1063/1.3488230)

[10] M. Stener, G. Fronzoni and M. de Simone, *Time dependent density functional theory of core electrons excitations*, Chemical Physics Letters 373, 115 (2003) (https://doi.org/10.1016/S0009-2614(03)00543-8)

The starting guess vector(s) for the excitation energies can be selected, for example by selecting 1 occupied orbital and 1 virtual orbital.

```
SELECTEXCITATION
   OscStrength oscstrength
   UseOccVirtRange elowoccvirt ehighoccvirt
   UseOccVirtNumbers nrlowoccvirt nrhighoccvirt
   UseOccRange elowocc ehighocc
   UseVirtRange elowvirt ehighvirt
   UseOccupied
      irrep orbitalnumbers
      irrep orbitalnumbers
      ...
   SubEnd
   UseVirtuaL
      irrep orbitalnumbers
      irrep orbitalnumbers
      ...
   SubEnd
   UseScaledZORA
end
```

`OscStrength oscstrength` Use only pairs of an occupied and virtual orbital as guess vectors, for which the oscillator strength of the single-orbital transition is larger than oscstrength.

`UseOccVirtRange elowoccvirt ehighoccvirt` Use only pairs of an occupied and virtual orbital as guess vectors, for which the orbital energy difference is between elowoccvirt and ehighoccvirt (in hartree). If one is interested in the lowest excitation energies, use for elowoccvirt a value smaller than the HOMO-LUMO gap, and for ehighoccvirt a value larger than the energy range one is interested in.

`UseOccVirtNumbers nrlowoccvirt nrhighoccvirt` Use only pairs of an occupied and virtual orbital as guess vectors, for which in the sorted list of the orbital energy differences, the number of the single-orbital transition is between nrlowoccvirt and nrhighoccvirt.

`UseOccRange elowocc ehighocc` Use only occupied orbitals in the guess vectors which have orbital energies between elowocc and ehighocc (in hartree).

`UseVirtRange elowvirt ehighvirt` Use only virtual orbitals in the guess vectors which have orbital energies between elowvirt and ehighvirt (in hartree).

`UseOccupied` Use only the occupied orbitals in the guess vectors which are specified.

`UseVirtual` Use only the virtual orbitals in the guess vectors which are specified.

`irrep` The name of one of the irreducible representations (not a subspecies) of the point group of the system. See the Appendix for the irrep names as they are used in ADF.

`orbitalnumbers` A series of one or more numbers: include all numbers of the orbitals in the guess vectors that are to be used. In an unrestricted calculation the same numbers are used for the spin-α orbitals and the spin-β orbitals.

Modify range of excitation energies

The key MODIFYEXCITATION can be used to reduce the computational costs of, for example, core excitation energies. This key can also be used in case of spin-orbit coupling. The use of the key MODIFYEXCITATION is similar as the use of the key SELECTEXCITATION. However, the key MODIFYEXCITATION can not be used in combination with the key SELECTEXCITATION. In the state selective method (key SELECTEXCITATION) the one-electron excited state configuration space remains complete, whereas it is (effectively) reduced in case the scheme with the MODIFYEXCITATION key.

One possibility is to allow only selected occupied orbitals and or selected virtual orbitals in the TDDFT calculations. In this scheme the complete one-electron excited state configuration space is reduced to the subspace where only the core electrons are excited, see Stener et al.[?]. In the actual implementation this is done by artificially changing the orbital energies of the uninteresting occupied orbitals to a large negative value (default -1d6 Hartree), and by by artificially changing the orbital energies of the uninteresting virtual orbitals to a large positive value (default 1d6).

In ADF2010 an extra possibility is added with the new subkey UseOccVirtRange, which restricts the space of excitation energies, by allowing only pairs of occupied and virtual orbitals, for which the difference in orbital energy is between a certain range.

```
MODIFYEXCITATION
   OscStrength oscstrength
   UseOccVirtRange elowoccvirt ehighoccvirt
   UseOccVirtNumbers nrlowoccvirt nrhighoccvirt
   UseOccRange elowocc ehighocc
   UseVirtRange elowvirt ehighvirt
   UseOccupied
      irrep orbitalnumbers
      irrep orbitalnumbers
      ...
   SubEnd
   UseVirtuaL
      irrep orbitalnumbers
      irrep orbitalnumbers
      ...
   SubEnd
   SetOccEnergy esetocc
   SetLargeEnergy epsbig
   UseScaledZORA
end
```

`OscStrength oscstrength` Use only pairs of an occupied and virtual orbital as guess vectors, for which the oscillator strength of the single-orbital transition is larger than oscstrength.

`UseOccVirtRange elowoccvirt ehighoccvirt` Use only pairs of an occupied and virtual orbital, for which the orbital energy difference is between elowoccvirt and ehighoccvirt (in hartree).

`UseOccVirtNumbers nrlowoccvirt nrhighoccvirt` Use only pairs of an occupied and virtual orbital as guess vectors, for which in the sorted list of the orbital energy differences, the number of the single-orbital transition is between nrlowoccvirt and nrhighoccvirt.

`UseOccRange elowocc ehighocc` Use only occupied orbitals which have orbital energies between elowocc and ehighocc (in Hartree).

`UseVirtRange elowvirt ehighvirt` Use only virtual orbitals which have orbital energies between elowvirt and ehighvirt (in Hartree).

`UseOccupied` Use only the occupied orbitals which are specified.

`UseVirtual` Use only the virtual orbitals which are specified.

irrep The name of one of the irreducible representations (not a subspecies) of the point group of the system. See the Appendix for the irrep names as they are used in ADF.

orbitalnumbers A series of one or more numbers: include all numbers of the orbitals that are to be used. In an unrestricted calculation the same numbers are used for the spin-α orbitals and the spin-β orbitals.

SetOccEnergy esetocc All occupied orbitals that have to be used will change their orbital energy to esetocc. In practice only useful if one has selected one occupied orbital energy, and one want to change this to another value. Default: the orbital energies of the occupied orbitals that are used are not changed.

SetLargeEnergy epsbig The orbital energies of the uninteresting occupied orbitals are changed to -epsbig Hartree, and the orbital energies of the uninteresting virtual orbitals are changed to epsbig Hartree (Default: epsbig = 1d6 Hartree).

UseScaledZORA Use everywhere the scaled ZORA orbital energies instead of the ZORA orbital energies in the TDDFT equations. This can improve deep core excitation energies. Only valid if ZORA is used. Default: use the unscaled ZORA orbital energies.

Excitations as orbital energy differences

Instead of the relative expensive TDDFT calculation of excitation energies, sometimes just calculating Kohn-Sham orbital energy differences may already be useful. The subkeyword SINGLEORBTRANS in the block key EXCITATIONS, will calculate excitation energies as Kohn-Sham orbital energy differences. The subkeyword SINGLEORBTRANS in the block key EXCITATIONS replaces the obsolete key KSSPECTRUM. For a given excitation from an occupied orbital to a virtual orbital the oscillator strength is calculated from the the dipole transition moment between this occupied orbital and this virtual orbital. This method is best suited if LDA or a GGA is used in the SCF.

Especially useful for core excitation energy calculations. If SINGLEORBTRANS is used, it is possible to use fractional occupation numbers in the SCF, like is used in the DFT transition state (DFT-TS) scheme, see, for example, Ref.[12]. Note: for fractional occupation numbers, typically an orbital is treated in the excitation calculation as if it is fully occupied if the occupation number is 1.5 or more, and it is treated as if it is fully unoccupied if the occupation number is 0.5 or less. In the transition state procedure for core excitations half an electron is moved from an initial core orbital to final virtual orbital and the SCF KS eigenvalue difference is taken as excitation energy. This is the original Slater formulation and is justified doing a Taylor expansion of total energy with respect to occupation numbers, it can be shown that errors arise from third order terms which are small. This, however, is not very practical: to converge SCF by putting half an electron in virtual orbitals is very hard, usually it works only for the lowest virtuals. In order to avoid this problem one neglects the half an electron in virtual, this scheme (only half an electron removed from occupied orbital) is called Transition Potential (TP) and is widely employed in XAS from K-edges.

```
EXCITATIONS
 SingleOrbTrans
END
```

SingleOrbTrans keyword to use only orbital energy differences

The value for the subkeyword KFWRITE in the EXCITATIONS block key is set by default to 0 in case of SINGLEORBTRANS. If kfwrite is 0 then do not write contributions, transition densities, and restart vectors to adf.rkf, since this can lead to a huge adf.rkf, especially if many excitations are calculated.

---

[12] R. De Francesco, M. Stener, and G. Fronzoni, *Theoretical Study of Near-Edge X-ray Absorption Fine Structure Spectra of Metal Phthalocyanines at C and N K-Edges*, Journal of Physical Chemistry A, 116 2285 (2012) (https://doi.org/10.1021/jp2109913)

Quadrupole intensities in X-ray spectroscopy

For electronic excitations in the ultraviolet and visible range of the electromagnetic spectrum, the intensities are usually calculated within the dipole approximation, which assumes that the oscillating electric field is constant over the length scale of the transition. For the short wavelengths used in hard X-ray spectroscopy, the dipole approximation may not be adequate. In particular, for metal K-edge X-ray absorption spectroscopy (XAS), it becomes necessary to include higher-order contributions. An origin-independent calculation of quadrupole intensities in XAS was implemented in ADF by Bernadotte et al., see Ref.[13]. These quadrupole intensities may also be important for the calculation of X-ray emission spectroscopy, see *section on XES* (page 170). Can not be used in combination with spin-orbit coupling.

Usage

```
EXCITATION
  ...
  XAS
  {ALLXASMOMENTS}
  {ALLXASQUADRUPOLE}
END
```

XAS Use XAS within the EXCITATION block to initiate the calculation of the higher oder multipole moment integrals and the calculation of the quadrupole oscillator strengths. This will only print the total oscillator strength and the excitation energy.

ALLXASMOMENTS Use ALLXASMOMENTS within the EXCITATION block in combination with the XAS keyword. This will print out all the individual transition moments used within the calculation of the total oscillator strength.

ALLXASQUADRUPOLE Use ALLXASQUADRUPOLE within the EXCITATION block in combination with the XAS keyword. This will print out the individual oscillator strength components to the total oscillator strength.

The *MODIFYEXCITATION* (page 168) or *SELECTEXCITATION* (page 167) keyword could be used to select a core orbital.

7.3.5 XES: X-ray emission spectroscopy

For the calculation of XES (X-ray emission spectroscopy) spectra the approach of Ref.[14] is implemented in the ADF program by Atkins et. al., see Ref.[15]. This is a frozen orbital, one-electron ΔDFT approach which uses orbital energy differences between occupied orbitals to model the X-ray emission energies. Even though it is the simplest possible approximation for the calculation of XES spectra, it has been shown to work well for V2C-XES (valence-to-core X-ray emission spectroscopy) spectra of transition metal complexes.

The XES keyword initiates the calculation of X-ray emission energies to a core orbital. In addition to dipole oscillator strengths this keyword also triggers the calculation of the higher order moment integrals and the calculation of the quadrupole oscillator strengths. By default it calculates the emission to the first orbital in the first symmetry, often the deepest core orbital, and only prints the total oscillator strength and the excitation energy. Can not be used in combination with spin-orbit coupling.

```
XES
   {COREHOLE irrep number}
   {ALLXESMOMENTS}
   {ALLXESQUADRUPOLE}
END
```

---

[13] S. Bernadotte, A.J. Atkins, Ch.R. Jacob, *Origin-independent calculation of quadrupole intensities in X-ray spectroscopy*, Journal of Chemical Physics 137, 204106 (2012) (https://doi.org/10.1063/1.4766359)

[14] N. Lee, T. Petrenko, U, Bergmann, F. Neese, and S. DeBeer, *Probing Valence Orbital Composition with Iron Kβ X-ray Emission Spectroscopy*, Journal of the American Chemical Society 132, 9715 (2010) (https://doi.org/10.1021/ja101281e)

[15] A.J. Atkins, M. Bauer, and Ch.R. Jacob, *The chemical sensitivity of X-ray spectroscopy: high energy resolution XANES versus X-ray emission spectroscopy of substituted ferrocenes*, Physical Chemistry Chemical Physics 15, 8095 (2013) (https://doi.org/10.1039/C3CP50999K)

`COREHOLE irrep number` This allows the selection of the acceptor orbital for the calculation of the emission oscillator strengths. For example 'COREHOLE A1 2' calculates oscillator strengths to the orbital 2 in irrep A1.

`ALLXESMOMENTS` Use ALLXESMOMENTS within the XES block. This will print out all the individual transition moments used within the calculation of the total oscillator strength.

`ALLXESQUADRUPOLE` Use ALLXESQUADRUPOLE within the XES block. This will print out the individual oscillator strength components to the total oscillator strength.

7.3.6 Excitation energies and Spin-Orbit coupling

Spin-orbit coupling can be included in the TDDFT calculation of excitation energies for closed-shell molecules. Two methods can be used in ADF. The first one includes spin-orbit coupling as a perturbation to a scalar relativistic calculation of excitation energies, in which spin-orbit coupling matrix elements (SOCMEs) between excited states are calculated. The second one includes spin-orbit coupling self-consistently in the ground state calculation. If spin-orbit coupling is large, the second one is more accurate, but is also more time-consuming.

The results of these spin-orbit coupled TDDFT calculations include the calculation of the zero field splitting (ZFS) of triplet excited states and the calculation of radiative rate constants, which could be used to calculate radiative phosphorescence lifetimes.

Perturbative inclusion of spin-orbit coupling

```
SOPERT
  {NCALC ncalc}
  {ESHIFT eshift}
  {GSCorr Yes/No}
End
Relativity
  Level Scalar
  Formalism ZORA
End
EXCITATIONS
END
```

The perturbative method, which is described in Ref.[16], is an approximate time-dependent density-functional theory (TDDFT) formalism to deal with the influence of spin-orbit coupling effect on the excitation energies for closed-shell systems. In this formalism scalar relativistic TDDFT calculations are first performed to determine the lowest single-group excited states and the spin-orbit coupling operator is applied to these single-group excited states to obtain the excitation energies with spin-orbit coupling effects included. The computational effort of the present method is much smaller than that of the two-component TDDFT formalism. The compositions of the double-group excited states in terms of single-group singlet and triplet excited states are obtained automatically from the calculations. In Ref.[?] it was shown that the calculated excitation energies based on the present formalism affords reasonable excitation energies for transitions not involving 5p and 6p orbitals. For transitions involving 5p orbitals, one can still obtain acceptable results for excitations with a small truncation error, while the formalism will fail for transitions involving 6p orbitals, especially 6p1/2 spinors.

Although this method is not completely correctly implemented for (meta-)hybrids or Hartree-Fock, it still gives reasonable excitation energies, and can thus be useful also in that case. Note that SYMMETRY C(2H) is not implemented for spin-orbit coupled excitations, use SYMMETRY C(S), C(I) or NOSYM, instead.

`NCALC=ncalc` Number of spin-orbit coupled excitation energies to be calculated. Default (and maximum) value: 4 times the number of scalar relativistic singlet-singlet excitations.

---

[16] F. Wang and T. Ziegler, *A simplified relativistic time-dependent density-functional theory formalism for the calculations of excitation energies including spin-orbit coupling effect*, Journal of Chemical Physics 123, 154102 (2005) (https://doi.org/10.1063/1.2061187)

`ESHIFT=eshift` The actually calculated eigenvalues are calculated up to the maximum singlet-singlet or singlet-triplet scalar relativistic excitation energy plus eshift (in Hartree). Default value: 0.2 Hartree.

`GSCORR Yes/No` The singlet ground state is included, which means that spin-orbit coupling can also have some effect on energy of the ground state. The spin-orbit matrix in this case is on basis of the ground state and the singlet and triplet excited states. Default Yes.

Some extra information about the spin-obit matrix is written to the output if one includes

```
SOPERT
  {NCALC ncalc}
  {ESHIFT eshift}
  {GSCorr Yes/No}
End
PRINT SOMATRIX
```

If one includes PRINT SOMATRIX the spin-orbit matrix (in Hartree) on basis of singlet and triplet excited states will be printed. Note that a triplet has three sublevels. The numbering of the basis of singlets and triplets is given above the spin-orbit matrix. The spin-orbit matrix has a real and imaginary part. On the diagonal the singlet or triplet energies is added. This spin-orbit matrix is the one that is diagonalized to get the spin-orbit coupled excitation energies. For example, for the effect of spin-orbit coupling between T1 and S1 one finds 3 complex values, between the 3 sublevels of T1 and the one level of S1. For a magnitude one could take the square root of the (real part squared + imaginary part squared).

Starting from ADF2018 these spin-orbit coupling matrix elements (SOCMEs) are printed in an easier format in the ouput. The SOCME that is printed is calculated as a root mean square: square root of (the sum of squares of spin-orbit coupling matrix elements of all sublevels of the uncoupled states).

Self-consistent spin-orbit coupling

```
Relativity
  Level Spin-Orbit
  Formalism ZORA
End
EXCITATIONS
  {ALSORESTRICTED}
END
```

Starting from the ADF2006.01 version in ADF the relativistic TDDFT formalism, including spin-orbit coupling, is implemented for closed-shell molecules with full use of double-group symmetry[17]. This relativistic time-dependent density-functional theory (TDDFT) is based on the two-component zeroth-order regular approximation (ZORA) and a non-collinear exchange-correlation (XC) functional. This two-component TDDFT formalism has the correct non-relativistic limit and affords the correct threefold degeneracy of triplet excitations.

Starting from AMS2023 spin-orbit coupling can be included in the calculation of CD spectra[28], see *section on CD spectra* (page 179).

In case of a calculation including spin-orbit coupling one can not separate the singlet-singlet and singlet-triplet excitations. By default the spin-polarized excitation energies are calculated (the noncollinear scheme is used for the spin-dependent exchange-correlation kernel). The subkeys ALSORESTRICTED can be used to include also excitation energies in which a spin-restricted exchange-correlation kernel is used. One should in fact only use the results of the spin-polarized calculation, which is based on the noncollinear exchange-correlation (XC) functional. For the same reason, the ALLOWED subkey

---

[17] F. Wang, T. Ziegler, E. van Lenthe, S.J.A. van Gisbergen and E.J. Baerends, *The calculation of excitation energies based on the relativistic two-component zeroth-order regular approximation and time-dependent density-functional with full use of symmetry*, Journal of Chemical Physics 122, 204103 (2005) (https://doi.org/10.1063/1.1899143)

[28] H.D. Ludowieg, M. Srebro-Hooper, J. Crassous, J. Autschbach, *Optical Activity of Spin-Forbidden Electronic Transitions in Metal Complexes from Time-Dependent Density Functional Theory with Spin-Orbit Coupling*, ChemistryOpen 2022, 11, e202200020 (https://doi.org/10.1002/open.202200020)

should not be used if spin-orbit coupling is included. Note that SYMMETRY C(2H) is not implemented for spin-orbit coupled excitations, use SYMMETRY C(S), C(I) or NOSYM, instead.

To perform a spin-orbit coupled TDDFT calculation one just needs to do a spin-orbit coupled SCF calculation and use the EXCITATION keyword. The molecule needs to be closed shell, and should be calculated spin-restricted. Thus do not use the UNRESTRICTED, COLLINEAR, or NONCOLLINEAR keyword. See, however, also next section.

The contribution to the double group excited states in terms of singlet and triplet single group excited states can be estimated through the inner product of the transition density matrix obtained from two-component and scalar relativistic TDDFT calculations to better understand the double group excited states[18]. In order to get this analysis one needs to perform a scalar relativistic TDDFT calculation of excitation energies on the closed shell molecule first, and use the resulting adf.rkf as a fragment in the spin-orbit coupled TDDFT calculation of excitation energies, including the keyword STCONTRIB (Singlet and Triplet CONTRIButions):

```
STCONTRIB
```

This STCONTRIB analysis is not performed for (meta-)hybrids, unless one uses the Tamm-Dancoff approximation (TDA) approximation, but then it may also fail. If one wants a similar analysis for (meta-)hybrids one may consider to the perturbative inclusion of spin-orbit coupling in the calculation of excitation energies.

Note that if hybrids are used, the dependency key is automatically set, and this may effectively reduce the number of excitations, which may give problems in the STCONTRIB analysis. A workaround for these problems is to first calculate the scalar relativistic fragment without the EXCITATIONS keyword. Use the adf.rkf of this calculation as fragment in a scalar relativistic calculation with the EXCITATIONS keyword. Use the adf.rkf of the second calculation as fragment in the spin-orbit coupled calculation, including the STCONTRIB keyword.

Highly approximate spin-orbit coupled excitation energies open shell molecule

Excitation energies can be obtained for open-shell systems in a spin-unrestricted TDDFT calculation including spin-orbit coupling. This approximate method uses a single determinant for the open shell ground state. The Tamm-Dancoff approximation (TDA) is needed and symmetry NOSYM should be used. Best is to use the noncollinear approximation. For analysis it is advised to calculate the molecule also with the scalar relativistic spin-restricted method and use it as fragment in the spin-orbit coupled calculation. This will make it easier to identify the excitations.

```
Unrestricted
Symmetry NOSYM
Relativity
  Level Spin-Orbit
  Formalism ZORA
  SpinOrbitMagnetization NONCOLLINEAR
End
TDA
Excitations
End
```

Note that this approximate method for open shell molecules is not able to show the subtle effects of spin-orbit coupling. Some of the reasons are the approximate nature of the XC functionals for open shell molecules, the single determinant that is used for the open shell ground state, and that only single excitations are included in the excitation. If one does not include spin-orbit coupling the spin-unrestricted TDDFT approach introduces spin-contamination such that the result does not represent transitions between pure spin states. Inclusion of spin-orbit coupling will not simplify this. However, if spin-orbit coupling is large, then this method may help to identify excitations.

Note that the approximations made in this approximate method are much worse than for spin-orbit coupled TDDFT for closed shell systems. In that case one can get a reasonable description of the subtle effects of spin-orbit coupling, for

---

[18] F. Wang and T. Ziegler, *Theoretical study of the electronic spectra of square-planar platinum (II) complexes based on the two-component relativistic time-dependent density-functional theory*, Journal of Chemical Physics 123, 194102 (2005) (https://doi.org/10.1063/1.2104427)

example, for the zero-field splitting of a triplet excited state.

7.3.7 CV(n)-DFT: Constricted Variational DFT

In the constricted nth order variational density functional method (CV(n)-DFT)[19][6][20] the occupied excited state orbitals are allowed to relax in response to the change of both the Coulomb and exchange-correlation potential in going from the ground state to the excited state. This theory is not time-dependent nor is it based on response theory. It is instead variational in nature and has been termed constricted variational DFT or CV(n)-DFT.

Due to bugs in older versions it is important to use ADF2016.105 or later. In ADF2017 the relaxation density is a bit differently calculated than in ADF2016, which will slightly modify the results compared to ADF2016. In ADF2017 singlet-triplet excitations are added[21] There have been different working equations implemented in different modified versions of ADF. In ADF2016 and ADF2017 all CV-DFT excitation energies are calculated consistently, this is why some energies may differ from previously published values.

CV(n)-DFT requires an all electron basis set.

The CVNDFT block key regulates the execution of the CV(n)-DFT code, which calculates the singlet or triplet electronic excitations for the closed shell molecules. Note that one has to choose either singlet-singlet excitations (subkey ONLYSING in EXCITATIONS) or singlet-triplet excitations (subkey ONLYTRIP in EXCITATIONS), in case of CV(n)-DFT, one can not calculate them both in one run. The parameter n in (n) describes the order of this theory. There are two limiting cases implemented in the CV(n)-DFT code: n=2 and n= ∞. Since CV(n)-DFT coincides in the second order with TDDFT, CV(n)-DFT code is designed as an extension of the EXCITATIONS module. Therefore, the EXCITATIONS block has to be present in the input file together with CVNDFT. CVNDFT takes all the information about the number of excitations and their symmetry from the EXCITATIONS block. Moreover, the TDDFT transition density vectors are used as an initial guess for the CV(∞)-DFT calculations.

There are a few approximations within the CV(∞)-DFT approach, therefore there are a few corresponding sub-blocks in the CVNDFT block. In the simplest case the TDDFT transition density U-vector is substituted into the infinite order CV(∞)-DFT excitation energy. This approximation corresponds to the CV_DFT sub-block, so the input fragment would look like this:

```
CVNDFT
  CV_DFT &
  SUBEND
END
EXCITATIONS
  ONLYSING|ONLYTRIP
  ...
END
```

In general, the U-vector has to be optimized with respect to the infinite order CV(∞)-DFT excitation energy. It is accomplished iteratively in the SCF-CV(∞)-DFT method[22], which is invoked when the SCF_CV_DFT sub-block is present. Therefore, there are input parameters that control the SCF procedure, i.e. the total number of iterations and tolerance. The corresponding input fragment would look like this:

---

[19] J. Cullen, M. Krykunov, and T. Ziegler, *The formulation of a self-consistent constricted variational density functional theory for the description of excited states*, Chemical Physics 391, 11 (2011) (https://doi.org/10.1016/j.chemphys.2011.05.021)

[6] M. Krykunov and T. Ziegler, *Self-consistent Formulation of Constricted Variational Density Functional Theory with Orbital Relaxation. Implementation and Applications*, Journal of Chemical Theory and Computation 9, 2761 (2013) (https://doi.org/10.1021/ct300891k)

[20] T. Ziegler, M. Krykunov, I. Seidu, Y.C. Park, *Constricted Variational Density Functional Theory Approach to the Description of Excited States*, Density Functional Methods for Excited States: Topics in Current Chemistry 368, 61 (2016) (https://doi.org/10.1007/128_2014_611)

[21] Y.C. Park, F. Senn, M. Krykunov, and T. Ziegler, *Self-Consistent Constricted Variational Theory RSCF-CV(∞)-DFT and Its Restrictions To Obtain a Numerically Stable Δ SCF-DFT-like Method: Theory and Calculations for Triplet States*, Journal of Chemical Theory and Computation 12, 5438 (2016) (https://doi.org/10.1021/acs.jctc.6b00333)

[22] T. Ziegler, M. Krykunov, J. Cullen, *The implementation of a self-consistent constricted variational density functional theory for the description of excited states*, Journal of Chemical Physics 136, 124107 (2012) (https://doi.org/10.1063/1.3696967)

```
CVNDFT
  ITERATION iter
  TOLERANCE tol
  SCF_CV_DFT &
    DAMPING mix
  SUBEND
END
EXCITATIONS
  ONLYSING|ONLYTRIP
  ...
END
```

`iter` iter is the maximum number of iterations. Default 50.

`tol` tol is the convergence criterion, i.e. the SCF-CV(∞)-DFT procedure stops when the given accuracy is achieved. Default 1e-4.

`mix` mix is the relative weight of the new U-vector that is added to the one from the previous iteration (default value is mix=0.2).

Another feature of CV(∞)-DFT is a possibility to optimize (or relax) the molecular orbitals for the particular excitation. This is accomplished in the R-CV(∞)-DFT method, see for an application Ref.[23]. In the last case, the transition U-vector is frozen, while the orbitals are relaxed. For example, the input fragment which employs the U-vector from TDDFT would look like this:

```
CVNDFT
  ITERATION iter
  TOLERANCE tol
  R_CV_DFT &
    RELAXALPHA start_a_iter
    RELAXBETA  start_b_iter
    DAMPORBRELAX mix_relax
  SUBEND
END
EXCITATIONS
  ONLYSING|ONLYTRIP
  ...
END
```

`start_a_iter` start_a_iter is the SCF cycle number at which the relaxation of α orbitals starts. Default value 1.

`start_b_iter` start_b_iter is the SCF cycle number at which the relaxation of β orbitals starts. Default value 1.

Since there is no optimization of the U-vector in R-CV(∞)-DFT, the relaxation of orbitals (either α or β or both) should be turned on at the very first iteration.

`mix_relax` The mix_relax parameter defines the relative weight of the new relaxation vector that is added to the one from the previous iteration (the default value is mix_relax=0.2).

Note that all tree methods can be used in the input file. The corresponding input fragment would look like this:

```
CVNDFT
  ITERATION iter
  TOLERANCE tol
  CV_DFT &
  SUBEND
  SCF_CV_DFT &
```

(continues on next page)

---

[23] F. Senn, M. Krykunov, *Excited State Studies of Polyacenes Using the All-Order Constricted Variational Density Functional Theory with Orbital Relaxation*, Journal of Physical Chemistry A 119, 10575 (2015) (https://doi.org/10.1021/acs.jpca.5b07075)

(continued from previous page)

```
    DAMPING mix
  SUBEND
  R_CV_DFT &
    RELAXALPHA start_a_iter
    RELAXBETA  start_b_iter
    DAMPORBRELAX mix_relax
  SUBEND
END
EXCITATIONS
  ONLYSING|ONLYTRIP
  ...
END
```

In this case the U-vector will be optimized with respect to the infinite order CV(∞)-DFT excitation energy by the SCF-CV(∞)-DFT code and then supplied to the R-CV(∞)-DFT code.

CVNDFT prints in the output the excitation energies as well as the maximum value of the lambda parameter λ. This number corresponds to the largest singular value of the singular value decomposition of the U matrix/vector (i.e. the U-vector can be considered as a matrix with *nvir* rows and *nocc* columns, where *nvir* is the number of virtual orbitals and *nocc* is the number of occupied orbitals). If this number is close to $\pi/2$ (i.e. $\lambda_{max} \approx 1.57$), then the corresponding electronic excitation can be approximated by a single natural transition orbital (NTO) to another single NTO transition. This type of transitions are obtained in Δ SCF. Therefore, this information can be used, for example, to compare the CV(∞)-DFT results to the Δ SCF excitation energies.

7.3.8 HDA: Hybrid Diagonal Approximation

This method is only relevant if a (meta-)hybrid is used in the SCF. The hybrid diagonal approximation (HDA)[52] is based on utilizing the hybrid exchange only for the diagonal terms in the response equations to calculate excitations. This allows one to limit the computational cost of the TD-DFT simulation while keeping basically the same accuracy as in the full TD-DFT scheme using hybrid xc-functionals. It is furthermore not necessary to correct all the diagonal terms with hybrid exchange. A cutoff parameter can be chosen in the input in order to reduce the number of diagonal terms that have to be calculated, which can speed up the HDA calculations.

```
EXCITATIONS
 HDA
 HDA_CutOff hda_cutoff
End
```

`HDA` Activate the hybrid diagonal approximation (HDA).

`hda_cutoff` Cutoff on difference in energy between virtual and occupied orbitals eps_virt-eps_occ, in order to reduce the number of diagonal terms that are corrected. Default is a huge value, which means that all diagonal terms are corrected.

This method can not be used in excited state geometry optimizations. For spin-orbit coupled HDA excitations one needs to use symmetry NOSYM:

```
relativity
 level spin-orbit
end
Symmetry NOSYM
EXCITATIONS
```

(continues on next page)

---

[52] M. Medves, L. Sementa, D. Toffoli, G. Fronzoni, A. Fortunelli, *An efficient hybrid scheme for time dependent density functional theory*, Journal of Chemical Physics 152, 184104 (2020) (https://doi.org/10.1063/5.0005954)

(continued from previous page)

```
 HDA
 HDA_CutOff hda_cutoff
End
```

7.3.9 TD-DFT+TB, TDA-DFT+TB

The basic idea of TD-DFT+TB[24] is to use the molecular orbitals from a DFT ground state calculation as input to an excited state calculation with TD-DFTB coupling matrices. TB means tight binding. In ADF2023 TDA-DFT+TB is implemented, which is similar as TD-DFT+TB, but now using TDA. Starting from ADF2023 one can also use TD-DFT+TB and TDA-DFT+TB for open shell molecules. If many excitations are required, this method will speed up the calculation drastically in comparison to the standard time needed for TDDFT calculations of excitation energies. This method is best suited if a (meta-)GGA or LDA is used in the SCF. The TDA-DFT+TB method can not be used in excited state geometry optimizations.

For TD-DFT+TB use

```
SYMMETRY NOSYM
EXCITATIONS
 TD-DFTB
END
```

For TDA-DFT+TB use

```
SYMMETRY NOSYM
EXCITATIONS
 TDA-DFTB
END
```

One can use this, for example, in combination with the *MODIFYEXCITATION key* (page 168) to use only single-orbital transitions that have a minimal oscillator strength:

```
SYMMETRY NOSYM
EXCITATIONS
 TD-DFTB
 ONLYSING
END
MODIFYEXCITATION
  OscStrength 0.001
End
```

In TD-DFT+TB reducing the space of single-orbital transitions with the key MODIFYEXCITATION will reduce the memory needed in ADF. Starting from ADF2016.102 for TD-DFT+TB only the lowest (in energy) 100000 single orbital transitions are included by default. One can override this by using a different value for the subkey UseOccVirtNumbers of the key MODIFYEXCITATION.

```
MODIFYEXCITATION
 UseOccVirtNumbers 1 100000
END
```

The value for the subkeyword KFWRITE in the EXCITATIONS block key is set by default to 0 in case of TD-DFT+TB. If kfwrite is 0 then do not write contributions, transition densities, and restart vectors to adf.rkf, since this can lead to a huge adf.rkf, especially if many excitations are calculated.

---

[24] R. Rüger, E. van Lenthe, T. Heine, L. Visscher, *Tight-Binding Approximations to Time-Dependent Density Functional Theory - a fast approach for the calculation of electronically excited states*, Journal of Chemical Physics 144, 184103 (2016) (https://doi.org/10.1063/1.4948647)

7.3.10 sTDA, sTDDFT

The simplified Tamm-Dancoff approach (sTDA)[25] and simplified time-dependent DFT approach (sTDDFT)[26] by Grimme et al. are implemented in ADF. In these approaches the time-dependent DFT equations are simplified by the evaluation of the two-electron integrals as short-range damped Coulomb interactions between (transition) charge density monopoles and a truncation of the single excitation expansion space. These methods are best suited if a (meta-)hybrid or a range-separated-hybrid is used in the SCF. These methods will speed up the calculation drastically in comparison to the standard time needed for TDA or TDDFT calculations of excitation energies for hybrids. Starting from ADF2023 one can use sTDA and sTDDFT for open shell molecules. Theses methods can not be used in excited state geometry optimizations.

An example application of sTDDFT is given in the TDDFT advanced tutorial

For sTDA use:

```
SYMMETRY NOSYM
EXCITATIONS
 sTDA
END
```

For sTDDFT use:

```
SYMMETRY NOSYM
EXCITATIONS
 sTDDFT
END
```

For global hybrids ADF will use the parameters α and β that depend on the amount of exact exchange a_x as:

$$\alpha = \alpha_1 + \alpha_2 a_x$$
$$\beta = \beta_1 + \beta_2 a_x$$

where $\alpha_1 = 1.42$, $\alpha_2 = 0.48$, $\beta_1 = 0.2$, $\beta_2 = 1.83$ are used in ADF. These values are fitted parameters by Grimme?. For range-separated hybrids one should set the parameters α, β and a_x in the input of ADF with the keyword MODIFYEXCITATION. See Ref.[27] for parameters that are used by Risthaus et al. for a few range-separated functionals. However the α and β parameters are mixed up in Ref.?. Thus use:

Functional	a_x	α	β
CAM-B3LYP	0.38	0.90	1.86
WB97	0.61	4.41	8.00
WB97X	0.56	4.58	8.00

For example, for CAM-B3LYP use

```
MODIFYEXCITATION
  GrimmeAlpha 0.9
  GrimmeBeta 1.86
  GrimmeAex 0.38
END
```

---

[25] S. Grimme, *A simplified Tamm-Dancoff density functional approach for the electronic excitation spectra of very large molecules*, Journal of Chemical Physics 138, 244104 (2013) (https://doi.org/10.1063/1.4811331)

[26] C. Bannwarth and S. Grimme, *A simplified time-dependent density functional theory approach for electronic ultraviolet and circular dichroism spectra of very large molecules*, Computational and Theoretical Chemistry 1040-1041, 45 (2014) (https://doi.org/10.1016/j.comptc.2014.02.023)

[27] T. Risthaus, A. Hansen, and S. Grimme, *Excited states using the simplified Tamm-Dancoff-Approach for range-separated hybrid density functionals: development and application*, Physical Chemistry Chemical Physics 16, 14408 (2014) (https://doi.org/10.1039/C3CP54517B)

If one uses the keyword MODIFYEXCITATION one may also set more parameters that are used in the sTDA and sTDDFT approach.

```
MODIFYEXCITATION
  GrimmeAlpha Alpha
  GrimmeBeta Beta
  GrimmeAex Aex
  GrimmeDEmax DEmax
  GrimmeTPmin TPmin
  GrimmePertC|NoGrimmePertC
END
```

`GrimmeAlpha Alpha` To set the α parameters, should be used with GrimmeBeta and GrimmeAex.

`GrimmeBeta Beta` To set the β parameters, should be used with GrimmeAlpha and GrimmeAex.

`GrimmeAex Aex` To set a_x that is used in the sTDA or sTDDFT part of the calculation, should be used with GrimmeAlpha and GrimmeBeta.

`GrimmeDEmax DEmax` Single orbital transitions that have an orbital energy difference less than DEmax (in Hartree) are included. Default value DEmax = 0.4 Hartree. See also the meaning of E_{max} in Grimme[?].

`GrimmeTPmin TPmin` Single orbital transitions that would have a cumulative perturbative energy contribution larger than TPmin is included. Default value TPmin = 10^{-4}. See also the meaning of t_p in Grimme[?].

`GrimmePertC|NoGrimmePertC` In case of GrimmePertC, which is the default, if the cumulative perturbative energy contribution for a single orbital transition is smaller TPmin, this contribution is used to modify the diagonal value of a matrix that is used in the sTDA or the sTDDFT method. If NoGrimmePertC is included such contributions are neglected. See Grimme[?] for more details on the truncation of the single excitation space.

Starting from ADF2016.102 for sTDA or sTDDFT only the lowest (in energy) 10000 single orbital transitions are included by default. One can override this by using a different value for the subkey UseOccVirtNumbers of the key MODIFYEXCITATION.

```
MODIFYEXCITATION
 UseOccVirtNumbers 1 10000
END
```

The value for the subkeyword KFWRITE in the EXCITATIONS block key is set by default to 0 in case of sTDA or sTDDFT. If kfwrite is 0 then do not write contributions, transition densities, and restart vectors to adf.rkf, since this can lead to a huge adf.rkf, especially if many excitations are calculated.

7.3.11 CD spectra

Circular dichroism (CD) is the differential absorption of left- and right-handed circularly polarized light. Starting from ADF2010 Hartree-Fock and hybrids can also be used to calculate CD spectra.

Starting from AMS2023 spin-orbit coupling can be included in the calculation of CD spectra, see Ref.[?], which allows one, for example, to calculate circularly polarized luminescence (CPL) for spin-forbidden transitions such as T1-S0 emission, and spin-forbidden electronic circular dichroism. These are spin-forbidden if one would not include spin-orbit coupling. Phosphorescence dissymmetry factors are calculated.

```
EXCITATIONS
 CDSPECTRUM
 ANALYTICAL
 VELOCITY
End
```

CDSPECTRUM If the subkey *CDSPECTRUM* is included in the key EXCITATIONS the rotatory strengths for the calculated excitations are calculated, in order to simulate Circular Dichroism (CD) spectra[30][31]. Interesting for chiral molecules.

Velocity If the subkey *VELOCITY* is included ADF calculates the dipole-velocity representation of the oscillator strength. If applicable (use of subkey *CDSPECTRUM*) the dipole-velocity representation of the rotatory strength is calculated. Default the dipole-length representation of the oscillator strength and rotatory strength is calculated. Remark: in case of (range separated) (meta-) hybrids and TDA the dipole-length and velocity representation of the oscillator strength are not the same (not even in the basis set limit), see Ref.[29].

7.3.12 MCD

MCD or magnetic circular dichroism is the differential absorption of left and right circularly polarized light in the presence of a magnetic field. MCD intensity is usually described in terms of different contributions called A, B and C terms, see Refs.[32][33]. A further parameter D is often discussed in MCD studies. D is proportional to the intensity of an absorption band and is closely related to the oscillator strength. A and B terms for closed and open-shell molecules and C terms of open-shell molecules induced by spin-orbit coupling can be calculated. Starting from ADF2010 C terms related to spatially degenerate states, i.e. breaking of degeneracies can be calculated. The A term is only calculated in case the molecule has symmetry and then only for excited states that belong to multi-dimensional irreps.

For MCD calculations for molecules that have C(2) or D(2) symmetry use SYMMETRY NOSYM.

Input options

```
EXCITATIONS
   MCD {options}
   ONLYSINGLET
   {SELECT transition number}
   {DTENSOR {Dxx Dxy Dyy Dxz Dyz|D E/D}}
End

ALLPOINTS
{Relativity
   Level Scalar
   Formalism ZORA
 End}
{SOMCD}
{ZFS}
```

MCD If the subkey MCD is included in the key EXCITATIONS the MCD parameters of some or all of the excitations considered in the TDDFT procedure are calculated[45][46][47][35]. This subkey should not be used with spin-orbit coupling (but, see below). Several other keywords could be important.

---

[30] J. Autschbach and T. Ziegler, *Calculating molecular electric and magnetic properties from time-dependent density functional response theory*, Journal of Chemical Physics 116, 891 (2002) (https://doi.org/10.1063/1.1420401)

[31] J. Autschbach, T. Ziegler, S.J.A. van Gisbergen and E.J. Baerends, *Chiroptical properties from time-dependent density functional theory. I. Circular dichroism spectra of organic molecules*, Journal of Chemical Physics 116, 6930 (2002) (https://doi.org/10.1063/1.1436466)

[29] R.A. Harris, *Oscillator Strengths and Rotational Strengths in Hartree-Fock Theory*, Journal of Chemical Physics 50, 3947 (1969) (https://doi.org/10.1063/1.1671653)

[32] S.B. Piepho and P. N. Schatz, *Group Theory in Spectroscopy With Application to Magnetic Circular Dichroism*, (Wiley, New York, 1983).

[33] W.R. Mason, *A Practical Guide to Magnetic Circular Dichroism Spectroscopy*, (Wiley, New Jersey, 2007).

[45] M. Seth and T. Ziegler, *Formulation of magnetically perturbed time-dependent density functional theory*, Journal of Chemical Physics 127, 134108 (2007) (https://doi.org/10.1063/1.2772849)

[46] M. Seth, M. Krykunov, T. Ziegler, J. Autschbach and A. Banerjee, *Application of magnetically perturbed time-dependent density functional theory to magnetic circular dichroism: Calculation of B terms*, Journal of Chemical Physics 128, 144105 (2008) (https://doi.org/10.1063/1.2901967)

[47] M. Seth, M. Krykunov, T. Ziegler and J. Autschbach, *Application of magnetically perturbed time-dependent density functional theory to magnetic circular dichroism. II. Calculation of A terms*, Journal of Chemical Physics 128, 234102 (2008) (https://doi.org/10.1063/1.2933550)

[35] M. Seth, T. Ziegler and J. Autschbach, *Application of magnetically perturbed time-dependent density functional theory to magnetic circular dichroism. III. Temperature-dependent magnetic circular dichroism induced by spin-orbit coupling*, Journal of Chemical Physics 129, 104105 (2008) (https://doi.org/10.1063/1.2976568)

ALLPOINTS: required for an MCD calculation.

ONLYSINGLET: this keyword should be used in combination with a MCD calculation.

RELATIVITY: Scalar Relativistic ZORA: required for a calculation of temperature-dependent C terms. In this case the keyword SOMCD must also be added as a key by itself, and the calculation must be unrestricted. If only A and B terms are calculated then ZORA is not needed but can be included if desired.

ZFS: If the ZFS keyword and MCD with SOMCD are also included then the influence of the calculated zero-field splitting (ZFS) on the temperature-dependent MCD is evaluated. The MCD in the presence of ZFS is described as anisotropic in the output because the Zeeman splitting becomes orientation dependent in the presence of ZFS.

In ADF2010 the temperature-dependent MCD due to the breaking of degeneracies of excited states by spin-orbit coupling can be calculated. Although all temperature-dependent MCD is typically called “C terms”, the parameters associated with the MCD are labeled “CE” to distinguish them from the MCD due to mixing between states caused by spin-orbit coupling that is labeled “C”. The CE terms have a derivative shape like A terms. They have the same temperature-dependence as normal C terms. If they are present, CE terms are calculated automatically along with C terms if the keyword SOMCD is included in the input.

`MCD {options}` Options include NMCDTERM, NMIX, DCUTOFF, MCDOUT, CGOUT, NANAL, NANAL2, FULLOMEGA, NOAB, NODIRECT, NOCG, CONVCG, ITERCG, ITER2CG, BMIN, BMAX, TMIN, TMAX and NTEMP.

`NMCDTERM=nmcdterm` Number of excitations for which MCD parameters are to be calculated. The nmcdterm lowest energy excitations are treated. The default is the number of transitions considered in the TDDFT calculation.

`NMIX=nmix` Number of transitions allowed to mix in a SOS calculation. Default is the number of transitions considered in the TDDFT calculation.

`DCUTOFF=dcutoff` MCD parameters will only be calculated for transitions with sufficient intensity. Each Cartesian component of each transition is considered separately. If the dipole strength D of that component is below dcutoff then the MCD is not calculated. The default is 1.0e-6.

`MCDOUT=mcdout` Number that determines the amount of output to be printed about the MCD calculation. Higher means more output. Possible values are 0, (orientationally averaged and Cartesian components of MCD parameters only) 1 (as for 0 but with the addition of a short analysis) or 2 (as for 1 but with the addition of a lengthy analysis). Theoretical analyses of MCD parameters are presented in several places including Refs.[?????]. The default for MCDOUT is 0.

`CGOUT=cgout` The perturbed transition densities used to evaluate the B and C term parameters can be obtained through an iterative conjugate-gradient procedure. Convergence information of the conjugate-gradient algorithm is printed every cgout iterations. Default is 10.

`NANAL=nanal, NANAL2=nanal2` If MCDOUT is set to 2, a detailed analysis of the B and/or C term parameters in terms of which states mix and how much MCD each mixing causes, is presented. The parameters NANAL and NANAL2 determine how many contributions are included in the analyses. Defaults are 10 for NANAL and 5 for NANAL2.

`FULLOMEGA` A standard TDDFT calculation involves the solution of an eigenvalue equation to obtain the excitation energies and transition densities of interest. ADF can solve this eigenvalue equation two ways: through diagonalization of the full Omega matrix or through the Davidson procedure where Omega is never explicitly constructed. Construction of the complete Omega matrix is generally only feasible for smaller problems. The matrix Omega appears again in the equations solved to obtain MCD. Here again Omega can be built or only the products of Omega with a vector can be used as is the case in the Davidson procedure. The default is to not construct Omega. If the keyword FULLOMEGA is included then Omega is constructed. Note that the choice of FULLOMEGA is completely independent of whether EXACT or DAVIDSON is chosen in the earlier TDDFT calculation.

NOAB If this keyword is included then A and B terms are not calculated. NOAB only makes sense if SOMCD is included in the input otherwise no MCD will be calculated at all.

NODIRECT The perturbed transition density needed to evaluate B and C term parameters is obtained through the solution of a large system of equations. This system of equations is solved in two ways: through a sum-over-states (SOS) type approach where the solution is expanded in a known set of transition densities or through the direct solution of the system of equations by the conjugate gradient procedure. The SOS method is much faster but also less accurate, particularly for larger systems. By default MCD parameters are evaluated through both approaches. If the NODIRECT keyword is included then only the SOS calculation is performed.

NOCG The conjugate gradient procedure is first used in combination with a preconditioner that generally speeds up convergence significantly. If no solution is found in a reasonable number of iterations then the procedure is restarted without the preconditioner. If the NOCG keyword in included then the preconditioner is never used.

CONVCG Convergence criterion for the CG iterative methods. The default value of 0.01 is probably good enough for most applications. This choice seems to produce B and C terms that are converged to 3 significant figures. Except for small systems, it is not recommended that CONVCG be set to a much smaller number as this will probably cause a large number of convergence failures.

ITERCG=itercg Number of iterations before failure in the first (preconditioned) CG solver. This solver either succeeds quickly or not at all so the default value is 30.

ITER2CG=iterc2g Number of iterations before failure in the B of C term parameter calculation of the unconditioned CG solver. This solver is often slow so the default value is 200.

BMIN=bmin, BMAX=bmax, NBFIELD=nbfield, TMIN=tmin, TMAX=tmax, NTEMP=ntemp Temperature dependent MCD intensity often varies nonlinearly with T and B when T is small and/or B is large. It may therefore be of interest to evaluate the MCD intensity over a range of temperatures and/or magnetic fields. This can be achieved through the use of the BMIN, BMAX, NBFIELD, TMIN, TMAX and NTEMP keywords. The MIN and MAX keywords give the maximum values of B or T. NBFIELD and NTEMp indicate how many values are to be considered. Note that magnetic fields are assumed to be given in Tesla and temperatures in Kelvin. For example, BMIN=1, BMAX=5, NBFIELD=5 means that fields of 1,2,3,4 and 5 T will be considered. Defaults are BMIN=BMAX=1, TMIN=TMAX=5 and NBFIELD=NTEMP=1.

SELECT nselect1 nselect2 nselect3... Rather than selecting the first nmcdterm transitions for consideration individual transitions can be selected through the SELECT keyword. The transitions of interest are listed after the SELECT keyword. Note that the numbering follows that given in the summary table at the end of the TDDFT calculation. To consider a degenerate transition only the first component need be included. Note that it makes no sense to use both the SELECT and NMCDTERM keywords together.

DTENSOR Dxx Dxy Dyy Dxz Dyz DTENSOR D E/D As noted earlier, if the ZFS keyword is included with MCD and SOMCD then the influence of zero-field splitting on temperature-dependent MCD will be evaluated. As an alternative to the ZFS keyword the D-tensor parameters can be entered directly through the DTENSOR keyword in the EXCITATIONS block. Two input formats are possible. Five real numbers Dxx Dxy Dyy Dxz Dyz can be entered. These five numbers are sufficient to define the traceless tensor D. Alternatively, the two parameters D and E/D can be entered. In this case the coordinate system chosen to define the molecular geometry must be the the principle axis system of the D-tensor. D, Dxx, Dxy, Dyy, Dxz and Dyz should be given in wavenumbers (cm-1).

Notes

If an MCD calculation is run, the transition densities obtained in the TDDFT calculation are saved to adf.rkf. For large molecules this can result in a very large adf.rkf file.

An MCD calculation relies on the excitation energies and, in particular, the transition densities that result from the preceding TDDFT calculation. If the results of the TDDFT calculation are poor then it is likely that the results of the MCD calculation will be poor. It therefore should be kept in mind that most TDDFT calculations will make use of the Davidson method for finding the eigenvalues and eigenvectors of the TDDFT equation. The Davidson approach involves some approximations that can lead to some variation in results with the applied parameters. The most important example of this is the fact that the results vary depending on how many eigenvalue/eigenvector pairs are calculated, ie how many transitions are selected through the LOWEST keyword. The variation is small for the eigenvalues (excitation energies) but can be significant for the eigenvectors (transition densities). A variation in the transition densities leads to variation in the transition dipoles which can significantly impact calculated MCD parameters. The moral of this story is that when calculating MCD parameters it is best to choose one value of LOWEST and stick with it.

The most time-consuming part of an MCD calculation is the solution of the system of equations through the conjugate-gradient solver. The solver can fail so be aware of warnings concerning convergence in the output. A few hints to improve convergence are: a) choose a value of LOWEST that is at least double the number of transitions for which you desire MCD parameters. This helps to improve the SOS calculation which provides an initial guess for the conjugate gradient solver. The solver is sensitive to the initial guess so changing LOWEST by a small amount may help (or hinder) convergence significantly. Keep the previous note in mind when playing with LOWEST however. b) The preconditioned conjugate gradient solver is usually fast but does not converge monotonically to the correct answer. The unpreconditioned solver is much slower but tends to converge monotonically. If the preconditioned solver fails but leaves a fairly well converged result for the unpreconditioned solver the latter usually converges quickly. If the preconditioned solver does not leave a fairly well converged result it may be worth changing the number of iterations it uses since a few iterations earlier or later may provide a much better converged answer. c) The SELECT keyword can be used to work on the remaining transitions for which converged results have not been obtained.

All MCD parameters are presented in au. To convert A and C terms to the alternative unit D^2 (Debye squared) the value in au should be multiplied by 6.46044. To convert the B term to the alternative unit of D^2 /cm-1 the value in au should be multiplied by 2.94359e-05.

The A, B and C terms are defined through the equation suggested by Stephens (equation 1 in[?] and also see refs.[?][?][34]). This equation assumes that MCD intensity varies linearly with applied magnetic field and that the temperature-dependent component varies linearly with temperature as 1/T. For the most part, these assumptions are reasonable. An exception is that the temperature-dependent part varies from linearity when T is very small. To allow for this situation a temperature and magnetic field dependent multiplicative constant (chi(B,T)) is evaluated whenever temperature-dependent MCD parameters are considered. This constant includes all magnetic field and temperature dependence of the temperature-dependent MCD. Thus chi(B,T)*C can be used in place of B*C/kT in equation 1 of[?] when MCD spectra are to be simulated. Note that, since the g-factor for all states is here approximated by 2.0, chi applies to all transitions.

7.3.13 Analysis

By default the ADF output will show the excitation energies, oscillator strengths from ground state to excited state, and transition dipole moments from ground state to excited state. Also an analysis in terms of single orbital transitions (occupied to virtual) is given, and their contributions to the transition dipole moment. For analysis reasons one might be interested in pure single orbital transition. Such information can be obtained in case one calculates excitation energies as Kohn-Sham orbital energy differences, see subkey *SINGLEORBTRANS* (page 169) of the key EXCITATIONS.

---

[34] A. Bérces and T. Ziegler, *The harmonic force field of benzene. A local density functional study*, Journal of Chemical Physics 98, 4793 (1993) (https://doi.org/10.1063/1.464983)

NTO: Natural Transition Orbitals

Natural Transition Orbitals[36] come from a transformation of the transition density matrix. They are the closest you can get to visualizing an excitation as a one-electron excitation from 1 orbital to another. They give insight in the localization of excitations. Calculation of Natural Transition Orbitals (NTO's) will be performed if the subkey NTO is included in the key EXCITATIONS.

```
EXCITATIONS
  ...
  NTO
End
```

For (meta-)hybrids and range-separated functionals NTO's will be calculated only if one uses the Tamm-Dancoff approximation (TDA).

NTOs can be visualized in AMSspectra as shown in the GUI tutorial .

SFO analysis

In ADF2018 an SFO analysis of the excitation is implemented. This includes an analysis of the transition dipole moment. Especially useful in a fragment calculation, where one only has a few fragments. Not implemented for spin-orbit coupled excitations.

```
EXCITATIONS
  ...
  SFOANALYSIS
  N_SFO NMAXPRINTED
End
```

`SFOANALYSIS` If SFOANALYSIS is included a the SFO analysis of the excitation will be calculated.

`N_SFO NMAXPRINTED` NMAXPRINTED is the maximum number of printed contributions. Default value for NMAXPRINTED is 40.

The SFO - SFO dipole matrix elements can be printed in the output with:

```
PRINT DIPOLEMAT
```

Charge-transfer descriptors

Two methods for calculating charge-transfer descriptors are implemented (also working for spin-orbit coupled excitations).

The first method is the charge-transfer diagnostic overlap quantity LAMBDA, developed by Peach, Tozer, et al.[37] and a hole-electron distance R_HE, see Guido, Adamo, et al.[38].

The second method is the charge transfer descriptor CT, hole-electron distance R_HE, and some other descriptors, de-

---

[36] R.L. Martin, *Natural transition orbitals*, Journal of Chemical Physics 118, 4775 (2003) (https://doi.org/10.1063/1.1558471)

[37] M.J.G. Peach, P. Benfield, T. Helgaker, and D.J. Tozer, *Excitation energies in density functional theory: An evaluation and a diagnostic test*, Journal of Chemical Physics 128, 044118 (2008) (https://doi.org/10.1063/1.2831900)

[38] C.A. Guido, P. Cortona, B. Mennucci, and C. Adamo, *On the Metric of Charge Transfer Molecular Excitations: A Simple Chemical Descriptor*, Journal of Chemical Theory and Computation 9, 3118 (2013) (https://doi.org/10.1021/ct400337e)

veloped by Plasser, Lischka, et al.[48][49][50][51]. For this method a fragment calculation is required. Using a similar method a charge transfer descriptor CT_AT was defined that uses an atomic distance criterion, which will also be calculated in case of atomic fragments.

```
EXCITATIONS
  ...
  DESCRIPTORS
  {Descriptors_CT_AT_Rab Rab}
END
```

`DESCRIPTORS` If DESCRIPTORS is included the charge transfer descriptors are calculated.

`Descriptors_CT_AT_Rab Rab` Rab is the atomic distance criterion used for the calculation of CT_AT (in Angstrom). Default value for Rab is 2 Angstrom.

7.4 Excited state (geometry) optimizations

See also:

- Tutorial UV/Vis spectrum of ethene

It is possible to do excited state geometry optimizations, see Ref.[1]. Gradients for closed-shell singlet-singlet, closed shell singlet-triplet, conventional open shell and spin-flip open-shell TDDFT calculations can be evaluated.

In ADF2023 one can also calculate excited state gradients for the approximate *TD-DFT+TB* (page 177) method in case of closed-shell singlet-singlet and closed shell singlet-triplet excitations, see Ref.[2]. TD-DFT+TB should only be used for LDA and (meta-)GGAs.

With the keyword EXCITEDGO the gradients of the TDDFT excitation energy can be calculated. The *EXCITATIONS* (page 160) block must also be included in the input. The excitation energy gradients will only be calculated if the ground state gradients are calculated.

The gradients of the excitation energy are combined with the ground state gradients to give the gradients of the excited state. These gradients can be used in much the same way as ground state gradients are used. The type of calculation is chosen in the same way as for a ground state calculation. Examples of possible run types are:

- Geometry optimization
- Transition state search
- Frequency analysis with numerical second derivatives: (analytical second derivatives are not possible).

In general, an option that applies to a ground state geometry optimization will also apply to an excited state geometry optimization. A TDDFT geometry optimization will proceed in very much the same way as a ground state geometry optimization. The major difference will be that a TDDFT calculation will take place after the SCF and before the ground state gradients are evaluated. TDDFT gradients are calculated after the ground state gradients.

---

[48] F. Plasser, and H. Lischka, *Analysis of Excitonic and Charge Transfer Interactions from Quantum Chemical Calculations*, Journal of Chemical Theory and Computation 8, 2777 (2012) (https://doi.org/10.1021/ct300307c)

[49] F. Plasser, M. Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. I. Formalism*, Journal of Chemical Physics 141, 024106 (2008) (https://doi.org/10.1063/1.4885819)

[50] F. Plasser, S.A. Bäppler, Wormit, and A. Dreuw, *New tools for the systematic analysis and visualization of electronic excitations. II. Applications*, Journal of Chemical Physics 141, 024107 (2008) (https://doi.org/10.1063/1.4885820)

[51] S.A. Mewes, J.-M. Mewes, A. Dreuw, and F. Plasser, *Excitons in poly(para phenylene vinylene): a quantum-chemical perspective based on high-level ab initio calculations*, Physical Chemistry Chemical Physics 18, 2548 (2016) (https://doi.org/10.1039/c5cp07077e).

[1] M. Seth, G. Mazur, and T. Ziegler, *Time-dependent density functional theory gradients in the Amsterdam density functional package: geometry optimizations of spin-flip excitations*, Theoretical Chemistry Accounts 129, 331 (2011) (https://doi.org/10.1007/s00214-010-0819-2)

[2] S. Havenridge, R. Rüger, C.M. Aikens, *Analytical excited state gradients for time-dependent density functional theory plus tight binding (TDDFT + TB)* The Journal of Chemical Physics 158, 224103 (2023) (https://doi.org/10.1063/5.0142240).

Gradients for closed-shell singlet-singlet, closed shell singlet-triplet, conventional open shell and spin-flip open-shell TDDFT calculations can be evaluated. The FORCEALDA option and TDA options should be used with spin-flip calculations.

Not all functionals can be used in combination with TDDFT gradients. The following should work:

LDA: VWN, XALPHA

GGA: Any allowed combination of the Perdew86, LYP and PBEc correlation functionals and the Becke88, revPBEx, RPBEx, PBEx and OPTx exchange functionals.

Hybrid: B1LYP, B3LYP, B3LYP*, BHANDHLYP, BHANDH, O3LYP, X3LYP, B1PW91, MPW1PW, PBE0, OPBE0.

Most LibXC GGA and hybrid functionals.

RS Hybrid with XCFUN: CAMY-B3LYP and more (starting form ADF2018).

QM/MM TDDFT gradients can be calculated.

COSMO TDDFT gradients can be calculated (starting from ADF2018).

Scalar relativistic effects can be included with the ZORA or mass-velocity-Darwin Hamiltonians.

At this time, gradients involving frozen cores and spin-orbit TDDFT can not be calculated.

TDDFT gradients can take advantage of symmetry but if the point group of interest includes degenerate irreducible representations then all grid points are needed in integration (equivalent to the ALLPOINTS keyword). This situation is detected automatically. This use of the full grid may make it more efficient to use a point group with only one-dimensional irreducible representations where only the symmetry-unique slice is utilized. For TD-DFT+TB gradients symmetry NOSYM is required.

Degenerate excitations can be optimized. However, since in reality such degeneracies will be split by a Jahn-Teller distortion it is recommended that the symmetry of the chosen point group be lowered so that the transition of interest is no longer labeled by a degenerate representation. A Jahn-Teller distortion will not occur when the degeneracy cannot be broken by nuclear motion, e.g. for a diatomic molecule.

The EXCITEDGO block key has the following form:

```
EXCITEDGO
 {STATE Irreplab nstate}
 {SINGLET/TRIPLET}
 {OUTPUT n}
 {CPKS EPS=err PRECONITER=precon NOPRECONITER=noprecon ITEROUT=iter}
 {EIGENFOLLOW}
END
```

`STATE Irreplab nstate` Choose the excitation for which the gradient is to be evaluated.

> **`Irreplab`** Irreplab is the label from the TDDFT calculation. NOTE: the TDDFT module uses a different notation for some representation names, for example, A' is used instead of AA, A" (A two apostrophes) instead of AAA, A1' instead of AA1, A2" instead of AAA2. The excitations output gives the irrep levels that one should use here.
>
> **`nstate`** This value indicates that the nstate-th transition of symmetry Irreplab is to be evaluated. Default is the first fully symmetric transition.
>
> Note that in a numerical FREQUENCIES calculation symmetry is turned off except to reduce the number of points calculated so irrespective of the specified point group Irreplab is A in this case. Care should be taken to ensure that nstate is correct in a frequencies calculation as this number can change when the point group is changed.

`SINGLET/TRIPLET` SINGLET: A singlet-singlet excitation is considered. The default. TRIPLET: A singlet-triplet excitation is considered.

OUTPUT n The amount of output printed. A higher value requests more detailed output. Default: n=0

CPKS EPS=err PRECONITER=precon NOPRECONITER=noprecon ITEROUT=iter Some control parameters for the CPKS(Z-vector) part of the TDDFT gradients calculation.

EPS=err err is a real number that gives the convergence requirement of the CPKS. Default is 0.0001

PRECONITER=precon precon is the maximum number of iterations allowed for the preconditioned solver. Default = 30.

NOPRECONITER=noprecon noprecon is the maximum number of iterations allowed for the unpreconditioned solver. Default=200.

ITEROUT=iter Details of the CPKS calculation are printed every iter iterations. Default is 5.

EIGENFOLLOW This key tries to follow the eigenvector in excited state geometry optimizations. In the initial implementation the target state of an excited state geometry optimization was indicated by a number and a symmetry, e.g. A2g 3 or the 3rd state of A2g symmetry. This approach becomes problematic when states cross and the state you are interested in become the 4th A2g state for example. An eigenvector-following option has been added that attempts to alleviate this problem. This option is off by default. If the subkeyword EIGENFOLLOW in is included, the state of interest in the first iteration is the same as before. In the second and subsequent iterations the state for which gradients are determined is decided on the basis of the overlap between the transition density of the transition from the previous iteration and the transition densities available in the current iteration. The same symmetry is maintained. Note that this method is not full proof. It assumes that the transition density changes only because of the contributions from the various occupied-virtual orbital pairs change but that the orbitals remain unchanged. This is not necessarily the case. Secondly, the sign of the transition density components is not taken into account.

At each iteration of a TDDFT-gradients calculation the (relaxed) excited state electric dipole moment is also calculated.

7.4.1 Nuclear gradients only

In some cases one only needs the nuclear gradients of 1 or more excited state in a single point calculation. This can be calculated by requesting the gradients in the AMS driver part of the input with the EXCITEDGO keyword.

```
Task SinglePoint

Properties
  Gradients Yes
End

...
...

Engine ADF
   ...
   ...

   EXCITEDGO
     AllGradients
     SING_GRADS
        {IRREP1 integer_list}
        {IRREP2 integer_list}
     End
     TRIP_GRADS
        {IRREP3 integer_list}
        {IRREP4 integer_list}
     End
```

(continues on next page)

(continued from previous page)

```
  End
EndEngine
```

AllGradients calculate the nuclear gradient for all excitations that are calculated.

SING_GRADS Calculate the nuclear gradient for the singlet-singlet excitations that are specified. The excitation numbers per irrep should be specified.

TRIP_GRADS Calculate the nuclear gradient for the singlet-triplet excitations that are specified. The excitation numbers per irrep should be specified.

7.5 Vibrationally resolved electronic spectra

See the Vibrationally resolved electronic spectra section of the AMS manual.

7.5.1 AH-FC: Adiabatic Hessian Franck-Condon

See the AH-FC section of the AMS manual.

7.5.2 VG-FC: Vertical Gradient Franck-Condon

See the VG-FC section of the AMS manual.

7.6 (Hyper-)Polarizabilities, ORD, magnetizabilities, Verdet constants

A (frequency dependent) electric field induces a dipole moment in a molecule, which is proportional to the (frequency dependent) molecular polarizability. Van der Waals dispersion coefficients describe the long-range dispersion interaction between two molecules. Optical rotation or optical activity (ORD) is the rotation of linearly polarized light as it travels through certain materials. A (frequency dependent) magnetic field induces a magnetic moment in a molecule, which is proportional to the (frequency dependent) molecular magnetizability. The Faraday effects describes the rotation of the plane-polarized light due to a magnetic field, which is proportional to the intensity of the component of the magnetic field in the direction of the beam of light. The Verdet constant describes the strength of the Faraday effect for a particular molecule. All these properties are available in ADF as applications of time-dependent DFT (TDDFT).

Two keys can be used for calculating these properties:

- RESPONSE
- AORESPONSE

The RESPONSE key has many unique features compared to the AORESPONSE key, like the use of symmetry during the calculation. The AORESPONSE key also has many unique features compared to the RESPONSE key, namely lifetime effects (polarizabilities at resonance), the calculation of (dynamic) magnetizabilities, Verdet constants, the Faraday B terms, and an alternative way to calculate (resonance) Raman scattering factors. Both RESPONSE and AORESPONSE can calculate polarizabilities in case of spin-orbit coupling.

7.6.1 RESPONSE: (Hyper-)Polarizabilities, ORD

RESPONSE: Polarizabilities

The calculation of frequency-dependent (hyper)polarizabilities and related properties (Raman, ORD) is activated with the block key RESPONSE

```
RESPONSE
END
```

In this example only the *zz* component of the dipole polarizability tensor is calculated, at zero frequency. The orientation of the molecule is therefore crucial. Be aware that the program may modify the orientation of the molecule if the input coordinates do not agree with the symmetry conventions in ADF! (This calculation could equivalently be done through a finite field method).

See also the alternative implementation with the AORESPONSE key that offers some unique features like magnetizability, and lifetime options.

The impact of various approximations on the quality of computed polarizabilities has been studied in, for instance, Refs.[1][2][3]. If you are new to this application field, we strongly recommend that you study a few general references first, in particular when you consider hyperpolarizability calculations. These have many pitfalls, technically (which basis sets to use, application of the DEPENDENCY key) and theoretically (how do theoretical tensor components relate to experimental quantities, different conventions used). Please, take a good look both at ADF-specific references[31][32][6][4] and at general references related to this subject: Refs.[5][33][34], the entire issues of Chem.Rev.94, the ACS Symposium Series #628, and further references in the ADF-specific references.

```
RESPONSE
  ALLCOMPONENTS
  Frequencies freq1 freq2 ... freqN [unit]
  ALLTENSOR
  Quadrupole
  Octupole
  Traceless
END
```

`Entire tensor or only one component` You specify the ALLCOMPONENTS subkey to get the entire polarizability tensor, instead of just the *zz* component.

---

[1] S.J.A. van Gisbergen, V.P. Osinga, O.V. Gritsenko, R. van Leeuwen, J.G. Snijders and E.J. Baerends, *Improved density functional theory results for frequency-dependent polarizabilities, by the use of an exchange-correlation potential with correct asymptotic behavior*, Journal of Chemical Physics 105, 3142 (1996) (https://doi.org/10.1063/1.472182)

[2] S.J.A. van Gisbergen, F. Kootstra, P.R.T. Schipper, O.V. Gritsenko, J.G. Snijders and E.J. Baerends, *Density-functional-theory response-property calculations with accurate exchange-correlation potentials*, Physical Review A 57, 2556 (1998) (https://doi.org/10.1103/PhysRevA.57.2556)

[3] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *A Density Functional Theory study of frequency-dependent polarizabilities and van der Waals dispersion coefficients for polyatomic molecules*, Journal of Chemical Physics 103, 9347 (1995) (https://doi.org/10.1063/1.469994)

[31] S.J.A. van Gisbergen, J.G. Snijders, and E.J. Baerends, *Time-dependent Density Functional Results for the Dynamic Hyperpolarizability of* C_{60}, Physical Review Letters 78, 3097 (1997) (https://doi.org/10.1103/PhysRevLett.78.3097)

[32] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Calculating frequency-dependent hyperpolarizabilities using time-dependent density functional theory*, Journal of Chemical Physics 109, 10644 (1998) (https://doi.org/10.1063/1.477762)

[6] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Accurate density functional calculations on frequency-dependent hyperpolarizabilities of small molecules*, Journal of Chemical Physics 109, 10657 (1998) (https://doi.org/10.1063/1.477763)

[4] B. Champagne, E.A. Perpète, S.J.A. van Gisbergen, E.J. Baerends, J.G. Snijders, C. Soubra-Ghaoui, K.A. Robins and B.Kirtman, *Assessment of conventional density functional schemes for computing the polarizabilities and hyperpolarizabilities of conjugated oligomers: An ab initio investigation of polyacetylene chains*, Journal of Chemical Physics 109, 10489 (1998) (https://doi.org/10.1063/1.477731) Erratum: Journal of Chemical Physics 111, 6652 (1999) (https://doi.org/10.1063/1.479106)

[5] A. Willets, J.E. Rice, D.M. Burland and D.P. Shelton, *Problems in comparison of experimental and theoretical hyperpolarizabilities*, Journal of Chemical Physics 97, 7590 (1992) (https://doi.org/10.1063/1.463479)

[33] D.M. Bishop, *Aspects of Non-Linear-Optical Calculations*, Advances in Quantum Chemistry 25, 3 (1994) (https://doi.org/10.1016/S0065-3276(08)60017-9)

[34] D.P. Shelton and J.E. Rice, *Measurements and calculations of the hyperpolarizabilities of atoms and small molecules in the gas phase*, Chemical Reviews 94, 3 (1994) (https://doi.org/10.1021/cr00025a001)

`Frequencies` List of frequencies at which the polarizability is to be calculated. Default unit eV. A range of frequencies with equidistant points can be specified with the boundaries (inclusive) as well as the number of desired subintervals separated by colons. For example, 1.0:1.5:5 is equivalent to 1.0 1.1 1.2 1.3 1.4 1.5.

`Higher multipole polarizabilities` Instead of just calculating the dipole-dipole polarizability, one may address the dipole-quadrupole, quadrupole-quadrupole, dipole-octupole, quadrupole-octupole, and octupole-octupole polarizability tensors. These can all be calculated in a single run, using the subkey ALLTENSOR. If only quadrupole-quadrupole or octupole-octupole tensors are needed, the subkey quadrupole or octupole should be used.

`Traceless` If the subkey TRACELESS is included this ensures the calculation of tensors using the traceless form of the quadrupole operator.

RESPONSE: Accuracy and convergence

```
RESPONSE
  erralf 1e-6
  erabsx 1e-6
  errtmx 1e-6
  ncycmx 30
END
```

`erralf, erabsx, errtmx` The subkeys erralf, erabsx, errtmx determine the convergence criteria for a polarizability calculation. The strict defaults are shown. It is rarely necessary to change the defaults, as these are rather strict but do not lead to many iterations.

`ncycmx` The maximum number of attempts within which the algorithm has to converge. The default appears to be adequate in most cases.

RESPONSE: Hyperpolarizabilities

`Hyperpolarizabilities`

```
RESPONSE
  HYPERPOL LaserFreq
END
```

The first hyperpolarizability tensor β is calculated (in atomic units in the 'theoreticians convention', i.e. convention T=AB in Ref.[?]) if the subkey HYPERPOL is present with a specification of the laser frequency (in Hartree units). If also the subkey ALLCOMPONENTS is specified, all components of the hyperpolarizability tensor will be obtained.

As mentioned before, by default only the static dipole hyperpolarizability tensor is computed. If one is interested in the frequency-dependent hyperpolarizability, the input could look like:

```
RESPONSE
  ALLCOMPONENTS
  HYPERPOL 0.01
  DYNAHYP
END
```

The subkey DYNAHYP has to be added and the main frequency ω has to be specified in Hartrees after the subkey hyperpol. In the output all nonzero components of the tensors governing the static first hyperpolarizability, second harmonic generation, electro-optic Pockels effect, and optical rectification are printed.

Note: Second hyperpolarizabilities are currently not available analytically in case of RESPONSE. Some can however be obtained by calculating the first hyperpolarizability in a finite field.

The effect of using different DFT functionals (LDA, LB94, BLYP) on hyperpolarizabilities in small molecules has been investigated in[?].

RESPONSE: Optical rotation dispersion (ORD)

```
RESPONSE
  OPTICALROTATION
END
```

OPTICALROTATION With the subkey OPTICALROTATION the (frequency dependent) optical rotation[7][8] will be calculated. For correct calculations one should calculate the entire tensor (see also the subkey ALLCOMPONENTS), which is done automatically.

An alternative implementation uses the AORESPONSE key, in which life time effects can be included.

7.6.2 AORESPONSE: Lifetime effects, (Hyper-)polarizabilities, ORD, magnetizabilities, Verdet constants

The AORESPONSE key offers some unique features compared to the RESPONSE key, namely lifetime effects (polarizabilities at resonance), the calculation of (dynamic) magnetizabilities, Verdet constants, the Faraday B terms, and an alternative way to calculate (resonance) Raman scattering factors. Note that the RESPONSE key also has many unique features, like the use of symmetry during the calculation.

AORESPONSE: Polarizabilities

If the block key AORESPONSE is used, by default, the polarizability is calculated.

```
AORESPONSE
  Frequencies freq1 freq2 ... freqN
  LIFETIME width
END
```

Frequencies List of frequencies at which the time-dependent properties are to be calculated. Default unit eV. A range of frequencies with equidistant points can be specified with the boundaries (inclusive) as well as the number of desired subintervals separated by colons. For example, 1.0:1.5:5 is equivalent to 1.0 1.1 1.2 1.3 1.4 1.5.

LIFETIME width Specify the resonance peak width (damping) in Hartree units. Typically the lifetime of the excited states is approximated with a common phenomenological damping parameter. Values are best obtained by fitting absorption data for the molecule, however, the values do not vary a lot between similar molecules, so it is not hard to estimate values. A value of 0.004 Hartree was used in Ref.[9].

The spin-orbit ZORA polarizability code (Ref.[11]) is automatically selected if the AORESPONSE keyword is given in a spin-orbit coupled calculation. In this case a spin-restricted calculation is required, but, unlike the rest of AORESPONSE, also SYMMETRY NOSYM. Spin-polarization terms in the XC response kernel are neglected. In Ref.[?] the imaginary

---

[7] J. Autschbach and T. Ziegler, *Calculating molecular electric and magnetic properties from time-dependent density functional response theory*, Journal of Chemical Physics 116, 891 (2002) (https://doi.org/10.1063/1.1420401)

[8] J. Autschbach, S. Patchkovskii, T. Ziegler, S.J.A. van Gisbergen and E.J. Baerends, *Chiroptical properties from time-dependent density functional theory. II. Optical rotations of small to medium sized organic molecules*, Journal of Chemical Physics 117, 581 (2002) (https://doi.org/10.1063/1.1477925)

[9] L. Jensen, L. Zhao, J. Autschbach and G.C. Schatz, *Theory and method for calculating resonance Raman scattering from resonance polarizability derivatives*, Journal of Chemical Physics 123, 174110 (2005) (https://doi.org/10.1063/1.2046670)

[11] A. Devarajan, A. Gaenko, and J. Autschbach, *Two-component relativistic density functional method for computing nonsingular complex linear response of molecules based on the zeroth order regular approximation*, Journal of Chemical Physics 130, 194102 (2009) (https://doi.org/10.1063/1.3123765)

polarizability dispersion curves (spin-restricted) match well the broadened spin-orbit TDDFT data from Ref.[10]. Thus the corrections from the spin-polarization terms appear to be rather minor. No picture change corrections were applied in the ZORA formalism. AORESPONSE with hybrids in combination with spin-orbit is not implemented.

AORESPONSE: Technical parameters and expert options

```
AORESPONSE
 ...
  SCF        {NOCYC} {NOACCEL} {CONV=conv} {ITER=niter}
  GIAO
  FITAODERIV
  COMPONENTS {XX} {XY} {XZ} {YX} {YY} {YZ} {ZX} {ZY} {ZZ}
  ALDA|XALPHA
  ALPHA
END
```

SCF {NOCYC} {NOACCEL} {CONV=conv} {ITER=niter} Specify CPKS parameters such as the degree of convergence and the maximum number of iterations:

NOCYC - disable self-consistence altogether

NOACCEL - disable convergence acceleration

CONV - convergence criterion for CPKS. The default value is 10^{-6} . The value is relative to the uncoupled result (i.e. to the value without self-consistence).

ITER - maximum number of CPKS iterations, 50 by default.

Specifying ITER=0 has the same effect as specifying NOCYC.

GIAO Include the Gauge-Independent Atomic Orbitals (GIAO). This option should not be used with damping (LIFETIME keyword) and the VELOCITYORD option should be used instead.

FITAODERIV Use fitted AO Derivatives. This will improve the density fitting, can only be used in cae of STO fitting. In case of ZlmFit one can improve the fitting with the ZLMFIT block key.

COMPONENTS {XX} {XY} {XZ} {YX} {YY} {YZ} {ZX} {ZY} {ZZ} Limit the tensor components to the specified ones. Using this option may save the computation time.

ALDA|XALPHA If ALDA is specified the VWN kernel is used. This option is the default. If ALPHA is specified the Xα kernel is used instead of the default VWN one. For functionals that use LYP correlation, like BLYP, always the Xα kernel is used, even if one specified ALDA.

ALPHA Writes perturbed density matrix to TAPE16.

AORESPONSE: Damped First Hyperpolarizabilities

In Ref.[12] an implementation of finite lifetimes into TDDFT for the calculation of quadratic response properties in ADF is described. All β tensor components (27 in total) will be printed in the output.

```
AORESPONSE
  BETA|QUADRATIC
  Frequencies  freq1 freq2
```

(continues on next page)

---

[10] F. Wang, T. Ziegler, E. van Lenthe, S.J.A. van Gisbergen and E.J. Baerends, *The calculation of excitation energies based on the relativistic two-component zeroth-order regular approximation and time-dependent density-functional with full use of symmetry*, Journal of Chemical Physics 122, 204103 (2005) (https://doi.org/10.1063/1.1899143)

[12] Z. Hu, J. Autschbach, and L. Jensen, *Simulation of resonance hyper-Rayleigh scattering of molecules and metal clusters using a time-dependent density functional theory approach*, Journal of Chemical Physics 141, 124305 (2014) (https://doi.org/10.1063/1.4895971)

(continued from previous page)

```
  LIFETIME width
  STATIC|OPTICALR|EOPE|SHG
END
```

BETA Option to calculate the damped first hyperpolarizability (β) using quadratic response theory based on the 2n+1 rule. Two input frequencies are required for this calculation and the property $\beta(-(\omega_1+\omega_2);\omega_1,\omega_2)$ will be the output. Note that one can choose certain values of the two frequencies to calculate different types of β, such as static case $\beta(0;0,0)$, optical rectification $\beta(0;\omega,-\omega)$, electro-optical Pockels effect $\beta(-\omega;\omega,0)$, and second harmonic generation $\beta(-2\omega;\omega,\omega)$. Alternatively, these can be efficiently calculated using the (sub-)keywords STATIC, OPTICALR, EOPE, and SHG, respectively. Note that the needed input frequencies all rely on ω_1 (freq1) when using the (sub)keywords above.

QUADRATIC This option possess the same functionality as BETA, i.e., calculate the damped β, except not adapting the 2n+1 rule. The (sub)keywords STATIC, OPTICALR, EOPE,and SHG are also applicable here. Note that this approach facilitates the direct partitioning of the response into contributions from localized orbitals and is important for natural bond analysis.

Note: Please only use HARTREE or EV as the unit for the input frequencies. The unit option ANGSTROM does not work correctly due to the current implementation structure.

AORESPONSE: Damped Second Hyperpolarizabilities

In Ref.[13] a general implementation for damped cubic response properties is presented using time-dependent density functional theory (TDDFT) and Slater-type orbital basis sets. To directly calculate two-photon absorption (TPA) cross sections, an implementation of a reduced damped cubic response approach is described in Ref.[?]. All γ tensor components (81 in total) will be printed in the output.

```
AORESPONSE
  GAMMA|CUBIC
  Frequencies  freq1 freq2 freq3
  LIFETIME width
  STATIC|EFIOR|OKE|IDRI|EFISHG|THG|TPA
END
```

GAMMA Option to calculate the damped second hyperpolarizability (γ) using cubic response theory based on the 2n+1 rule. Three input frequencies are required for this calculation and the property $\gamma(-(\omega_1+\omega_2+\omega_3);\omega_1,\omega_2,\omega_3)$ will be the output. Note that one can choose certain values of the three frequencies to calculate different types of γ, such as static case $\gamma(0;0,0,0)$, electric field induced optical rectification $\gamma(0;\omega,-\omega,0)$, optical Kerr effect $\gamma(-\omega;\omega,0,0)$, intensity dependent refractive index $\gamma(-\omega;\omega,\omega,-\omega)$, electric field induced second harmonic generation $\gamma(-2\omega;\omega,\omega,0)$, and third harmonic generation $\gamma(-3\omega;\omega,\omega,\omega)$. Alternatively, these can be efficiently calculated using the (sub)keywords STATIC, EFIOR, OKE, IDRI, EFISHG, and THG, respectively. The (sub)keyword TPA can be used to calculate the γ corresponding to the two photon absorption process (i.e., the reduced form of $\gamma(-\omega;\omega,\omega,-\omega)$), however, it can ONLY be used with keyword GAMMA. Note that the needed input frequencies all rely on ω_1 (freq1) when using the (sub)keywords above.

CUBIC This option possess the same functionality as GAMMA, i.e., calculate the damped γ, except not adapting the 2n+1 rule. The (sub)keywords STATIC, EFIOR, OKE, IDRI, EFISHG and THG are also applicable here. Note that this approach facilitates the direct partitioning of the response into contributions from localized orbitals and is important for natural bond analysis.

Note: Please only use HARTREE or EV as the unit for the input frequencies. The unit option ANGSTROM does not work correctly due to the current implementation structure.

---

[13] Z. Hu, J. Autschbach, and L. Jensen, *Simulating Third-Order Nonlinear Optical Properties Using Damped Cubic Response Theory within Time-Dependent Density Functional Theory*, Journal of Chemical Theory and Computation 12, 1294 (2016) (https://doi.org/10.1021/acs.jctc.5b01060)

AORESPONSE: ORD

```
AORESPONSE
  OPTICALROTATION
  VELOCITYORD
  Frequencies freq1 freq2 ... freqN
  LIFETIME width
END
```

`OPTICALROTION` Specify OPTICALROTION to calculate optical rotatory dispersion spectrum instead of polarizabilities.

`VELOCITYORD` This option should be used instead of OPTICALROT with GIAO if the finite lifetime effects need to be taken into account (LIFETIME option).

AORESPONSE: magnetizabilities, Verdet constants, Faraday B term

```
AORESPONSE
  MAGNETICPERT
  MAGOPTROT
  FREQUENCIES freq1 freq2 ... freqN [unit]
  LIFETIME width
END
```

`MAGNETICPERT` Calculate static or time-dependent magnetizability, see also Ref.[14].

`MAGOPTROT` Specify MAGOPTROT to calculate the Verdet constant instead of polarizability, see for the details of the implementation Ref.[16]. When it is specified together with the LIFETIME key the real and imaginary part of the damped Verdet constant will be calculated. Combination of three keys MAGOPTROT, LIFETIME and FREQUENCIES yields the magnetic optical rotatory dispersion and magnetic circular dichroism spectrum (Faraday A and B terms) calculated simultaneously in the range from freq1 to freqN. It is also possible to combine MAGOPTROT, LIFETIME and FREQUENCY. In order to obtain the Faraday B terms from the Verdet constant calculations it is necessary to perform several steps, involving a fit of the imaginary Verdet data to the MCD spectrum. You can request SCM for details on the fitting procedure. For details of the method, see Ref.[15].

AORESPONSE: Raman

```
AORESPONSE
  RAMAN
  Frequencies  freq1 [unit]
  LIFETIME width
END
```

`RAMAN` Calculates the Raman scattering factors. The AORESPONSE-Raman only works with one frequency. If one frequency is specified the Raman scattering factors are calculated at that frequency. The Raman option is compatible with the lifetime option so that resonance Raman scattering can be calculated. For details of this method, see Ref.[?]. To get Raman intensities with AORESPONSE, numerical frequencies need to be calculated using a

---

[14] M. Krykunov and J. Autschbach, *Calculation of static and dynamic linear magnetic response in approximate time-dependent density functional theory*, Journal of Chemical Physics 126, 24101 (2007) (https://doi.org/10.1063/1.2423007)

[16] M. Krykunov, A. Banerjee, T. Ziegler and J. Autschbach, *Calculation of Verdet constants with time-dependent density functional theory. Implementation and results for small molecules*, Journal of Chemical Physics 122, 074105 (2005) (https://doi.org/10.1063/1.1850919)

[15] M. Krykunov, M. Seth, T. Ziegler and J. Autschbach, *Calculation of the magnetic circular dichroism B term from the imaginary part of the Verdet constant using damped time-dependent density functional theory*, Journal of Chemical Physics 127, 244102 (2007) (https://doi.org/10.1063/1.2806990)

FREQUENCIES key in the GEOMETRY input block. Non-resonance Raman intensities can also be obtained using the RESPONSE key or, alternatively, using RAMANRANGE in combination with analytically or numerically pre-calculated frequencies.

Applications of AORESPONSE

It may be useful to consult the following applications of the AORESPONSE key in ADF:

- Calculation of static and dynamic linear magnetic response in approximate time-dependent density functional theory[?]
- Calculation of CD spectra from optical rotatory dispersion, and vice versa, as complementary tools for theoretical studies of optical activity using time-dependent density functional theory[17]
- Calculation of origin independent optical rotation tensor components for chiral oriented systems in approximate time-dependent density functional theory[18]
- Time-dependent density functional calculations of optical rotatory dispersion including resonance wavelengths as a potentially useful tool for determining absolute configurations of chiral molecules[19]
- Calculation of optical rotation with time-periodic magnetic field-dependent basis functions in approximate time-dependent density functional theory[20]
- A Quantum Chemical Approach to the Design of Chiral Negative Index Materials[21]
- Calculation of Verdet constants with time-dependent density functional theory. Implementation and results for small molecules[?]
- Calculations of resonance Raman[?][35]
- Calculations of surface-enhanced Raman scattering (SERS)[36][37]
- Calculation of magnetic circular dichroism spectra from damped Verdet constants[?]
- Calculation of the polarizability in case of spin-orbit coupling[?]

See also:

- advanced tutorial on plasmon enhanced two photon absorption

---

[17] M. Krykunov, M.D. Kundrat and J. Autschbach, *Calculation of CD spectra from optical rotatory dispersion, and vice versa, as complementary tools for theoretical studies of optical activity using time-dependent density functional theory*, Journal of Chemical Physics 125, 194110 (2006) (https://doi.org/10.1063/1.2363372)

[18] M. Krykunov and J. Autschbach, *Calculation of origin independent optical rotation tensor components for chiral oriented systems in approximate time-dependent density functional theory*, Journal of Chemical Physics 125, 34102 (2006) (https://doi.org/10.1063/1.2210474)

[19] J. Autschbach, L. Jensen, G.C. Schatz, Y.C.E. Tse and M. Krykunov, *Time-dependent density functional calculations of optical rotatory dispersion including resonance wavelengths as a potentially useful tool for determining absolute configurations of chiral molecules*, Journal of Physical Chemistry A 110, 2461 (2006) (https://doi.org/10.1021/jp054847z)

[20] M. Krykunov and J. Autschbach, *Calculation of optical rotation with time-periodic magnetic field-dependent basis functions in approximate time-dependent density functional theory*, Journal of Chemical Physics 123, 114103 (2005) (https://doi.org/10.1063/1.2032428)

[21] A. Baev, M. Samoc, P.N. Prasad, M. Krykunov and J. Autschbach, *A Quantum Chemical Approach to the Design of Chiral Negative Index Materials*, Optics Express 15, 5730 (2007) (https://doi.org/10.1364/OE.15.005730)

[35] L. Jensen, L. Zhao, J. Autschbach and G.C. Schatz, *Resonance Raman Scattering of Rhodamine 6G as calculated using Time-Dependent Density Functional Theory*, Journal of Physical Chemistry A 110, 5973 (2006) (https://doi.org/10.1021/jp0610867)

[36] L.L. Zhao, L. Jensen and G.C. Schatz, Pyridine - Ag_{20} *Cluster: A Model System for Studying Surface-Enhanced Raman Scattering*, Journal of the American Chemical Society 128, 2911 (2006) (https://doi.org/10.1021/ja0556326)

[37] L. Jensen, L.L. Zhao and G.C. Schatz, *Size-Dependence of the Enhanced Raman Scattering of Pyridine Adsorbed on* Ag_n (n=2-8,20) *Clusters*, Journal of Physical Chemistry C 111, 4756 (2007) (https://doi.org/10.1021/jp067634y)

7.6.3 PolTDDFT: Damped Complex Polarizabilities

A fast algorithm to solve the Time Dependent Density Functional Theory (TDDFT) equations in the space of the density fitting auxiliary basis set has been developed and implemented[25]. The method, named PolTDDFT, extracts the spectrum from the imaginary part of the polarizability at any given photon energy, avoiding the bottleneck of Davidson diagonalization. The original idea which made the present scheme very efficient consists in the simplification of the double sum over occupied-virtual pairs in the definition of the complex dynamical polarizability, allowing an easy calculation of such matrix as a linear combination of constant matrices with photon energy dependent coefficients. The method has been extended for the calculation of circular dichroism spectra[26].

In AMS2023 several analysis options are added. An analysis of the absorption and CD spectrum in terms of single orbital transitions, using the subkey ANALYSIS. An analysis of the spectra per region (subkey REGIONSFORANALYSIS), using the fragment projection analysis approach, see Ref.[22]. If one is using the GUI or *densf* (page 291) one can visualize the induced (first-order perturbed) density (relevant for polarizability), the transition (first-order perturbed) density (relevant for absorption), and the rotator strength density (relevant for CD, see Ref.[29]). If one is using the GUI, one can visualize the so called transition contribution maps (TCM, see Ref.[23]), individual component maps of oscillatory strength (ICM-OS, see Ref.[24]), and individual component maps of rotatory strength (ICM-RS, see Ref.[?]).

In case a (meta-)hybrid is used in the SCF, the hybrid diagonal approximation (HDA)[38] is used. HDA is based on utilizing the hybrid exchange only for the diagonal terms in the response equations. This allows one to limit the computational cost of the TD-DFT simulation while keeping basically the same accuracy as in the full TD-DFT scheme using hybrid xc-functionals. It is possible to use the density fitting auxiliary basis set to further speed up the HDA (subkey HDA_fitted).

It is very important to use specially made auxiliary fit sets, which are available only for a very limited amount of elements. Symmetry and frozen cores can be used. Should not be used for range-separated functionals. Should not be used with spin-orbit coupling. STOFIT can not be used.

UV/Vis spectra, CD spectra

If one includes the POLTDDFT keyword the (real and imaginary part of the) diagonal of the polarizability tensor and rotatory strengths will be calculated, which can be used to calculate the photoabsorption and circular dichroism (CD) spectra.

```
POLTDDFT
  IRREP
     Irrep1
     Irrep2
     ...
  SUBEND
  KGRID eVgrid
```

(continues on next page)

---

[25] O. Baseggio, G. Fronzoni, and M. Stener, *A new time dependent density functional algorithm for large systems and plasmons in metal clusters*, Journal of Chemical Physics 143, 024106 (2015) (https://doi.org/10.1063/1.4923368)

[26] O. Baseggio, D. Toffoli, G. Fronzoni, M. Stener, L. Sementa, and A. Fortunelli, *Extension of the Time-Dependent Density Functional Complex Polarizability Algorithm to Circular Dichroism: Implementation and Applications to* Ag_8 *and* $Au_{38}(SC_2H_4C_6H_5)_{24}$, Journal of Physical Chemistry C 120, 24335 (2016) (https://doi.org/10.1021/acs.jpcc.6b07323)

[22] L. Sementa, G. Barcaro, O. Baseggio, M. De Vetta, A. Dass, E. Aprà, M. Stener, and A. Fortunelli, *Ligand-Enhanced Optical Response of Gold Nanomolecules and Its Fragment Projection Analysis: The Case of* $Au_{30}(SR)_{18}$, Journal of Physical Chemistry C 121, 10832 (2017) (https://doi.org/10.1021/acs.jpcc.6b12029)

[29] L. Chang, O. Baseggio, L. Sementa, D. Cheng, G. Fronzoni, D. Toffoli, E. Aprà, M. Stener, and A. Fortunelli, *Individual Component Map of Rotatory Strength and Rotatory Strength Density Plots As Analysis Tools of Circular Dichroism Spectra of Complex Systems*, Journal of Chemical Theory and Computation 14, 3703 (2018) (https://doi.org/10.1021/acs.jctc.8b00250)

[23] S. Malola, L, Lehtovaara, J. Enkovaara, and H, Häkkinen, *Birth of the Localized Surface Plasmon Resonance in Monolayer-Protected Gold Nanoclusters*, ACS Nano 7, 10263 (2013) (https://doi.org//10.1021/nn4046634)

[24] S. Theivendran, L. Chang, A. Mukherjee, L. Sementa, M. Stener, A. Fortunelli, and A. Dass, *Principles of Optical Spectroscopy of Aromatic Alloy Nanomolecules:* $Au_{36\text{-}x}Ag_x(SPh\text{-}tBu)_{24}$, Journal of Physical Chemistry C 122, 4524 (2018) (https://doi.org/10.1021/acs.jpcc.8b00556)

[38] M. Medves, L. Sementa, D. Toffoli, G. Fronzoni, A. Fortunelli, *An efficient hybrid scheme for time dependent density functional theory*, Journal of Chemical Physics 152, 184104 (2020) (https://doi.org/10.1063/5.0005954)

(continued from previous page)

```
  NGRID ngrid
  FREQRANGE eVi eVf
  NFREQ nfreq
  LIFETIME eVwidth
  CUTOFF eVcutoff
  LAMBDA lambda
  VELOCITY
  HDA_fitted
  ANALYSIS
  REGIONSFORANALYSIS region_names
END
```

`IRREP` Subblock key for selecting which symmetry irreps of the excitations to calculate (all excitations by default). In the subkey data block list the symmetry irrep labels, like B1, for example.

`KGRID eVgrid` Keyword KGRID is used to discretize the energy scale for calculating the complex dynamical polarizability. Only pairs of an occupied and virtual orbital are included, for which the orbital energy difference is lower than eVgrid (9 eV by default).

`NGRID ngrid` Ngrid is the number of points within the energy grid (180 by default).

`FREQRANGE eVi eVf` Keyword FREQRANGE is used to specify the equally spaced points in the spectrum for which one would like to calculate the complex dynamical polarizability. The first point is eVi (0 eV by default) and the last one is eVf (5 eV by default).

`NFREQ nfreq` The total number of points in the spectrum is nfreq (100 by default).

`LIFETIME eVwidth` Specify the resonance peak width (damping) in eV units. Typically the lifetime of the excited states is approximated with a common phenomenological damping parameter. Values are best obtained by fitting absorption data for the molecule, however, the values do not vary a lot between similar molecules, so it is not hard to estimate values. Default value is 0.1 eV.

`CUTOFF eVcutoff` For a given point in the spectrum, only include pairs of an occupied and virtual orbital, for which the orbital energy difference is lower than the energy of the point in the spectrum plus eVcutoff. The default value for eVcutoff is 4 eV.

`LAMBDA lambda` Jacob's scaling factor[27] for the study of plasmonic resonances. This factor, 0<lambda<1 (default lambda=1), turns on the coupling matrix K.

`VELOCITY` If the subkey VELOCITY is included ADF calculates the dipole moment in velocity gauge. Default the dipole-length representation is used.

`HDA_fitted` If the subkey HDA_fitted is included the density fitting auxiliary basis set is used to calculate HDA. HDA is only relevant if (meta-)hybrids are used.

`ANALYSIS` An analysis of the absorption and CD spectrum in terms of single orbital transitions.

`REGIONSFORANALYSIS region_names` Analysis per region, using the fragment projection analysis approach, see Ref.[?]. Will split the absorption and CD spectrum in region_i -> region_j terms. The region_names are the names of the regions that should be used. *Example: Damped complex polarizabilities with POLTDDFT: Au10* (page 858) shows how to use this option.

---

[27] S. Bernadotte, F. Evers, and C.R. Jacob, *Plasmons in Molecules*, Journal of Physical Chemistry C 117, 1863 (2013) (https://doi.org/10.1021/jp3113073)

Reduced fit set

For PolTDDFT it very important is to use specially made auxiliary fit sets. These are available[39] for most elements, except lanthanides and actinides, using a large frozen core. These special basis sets can be found in the atomic database directories, with a directory name POLTDDFT, for example, the directory $AMSHOME/atomicdata/ADF/POLTDDFT/TZP. They require relativistic ZORA to be used, since the frozen core description is relativistic ZORA. An example:

```
Basis
  Type POLTDDFT/DZ
  PerAtomType Symbol=Au File=POLTDDFT/TZP/Au.4f
End
```

Table 7.1: Available basis sets in PolTDDFT (May 2021)

Element	ae or fc	DZ	DZP	TZP
H-He (Z=1-2)	ae	Yes	Yes	Yes
Li-Ne (Z=3-10)	.1s	Yes	Yes	Yes
Na-Ar (Z=11-18)	.2p	Yes	Yes	Yes
K-Ca (Z=19-20)	.3p	Yes	Yes	Yes
Sc-Zn (Z=21-30)	.3p	Yes		Yes
Ga-Kr (Z=31-36)	.3d	Yes	Yes	Yes
Rb-Cd (Z=37-48)	.4p	Yes		Yes
In-Ba (Z=49-56)	.4d	Yes		Yes
Hf-Hg (Z=72-80)	.4f	Yes		Yes
Tl (Z=81)	.5p	Yes		Yes
Pb-Ra (Z=82-88)	.5d	Yes		Yes

7.6.4 Van der Waals dispersion coefficients

```
RESPONSE
  ALLCOMPONENTS
  VANDERWAALS NVanderWaals
  {ALLTENSOR}
END
```

`Dispersion coefficients` Simple dispersion coefficients (the dipole-dipole interaction between two identical molecules, the C_6 coefficient) are calculated in a single ADF calculation. General dispersion coefficients are obtained with the auxiliary program DISPER, which uses two output files (file named TENSOR) of two separate ADF runs as input. See the Properties and the Examples documents. To get the dispersion coefficients one has to calculate polarizabilities at imaginary frequencies between 0 and infinity. The ADF program chooses the frequencies itself. The user has to specify the number of frequencies, which in a sense defines the level of accuracy, as an argument to the subkey VanDerWaals.

`NVanderWaals` One can specify the number of frequencies with NVanderWaals. Ten frequencies is reasonable. Without the key ALLTENSOR only dipole-dipole interactions are considered. If ALLTENSOR is specified, higher dispersion coefficients are also calculated. This ADF calculation generates a file with name TENSOR, which contains the results of multipole polarizabilities at imaginary frequencies. This TENSOR file has to be saved. Similarly, the TENSOR file for the second monomer has to be saved. The files have to be renamed to files 'tensorA' and 'tensorB' (case sensitive) respectively. Then the program DISPER has to be called in the same directory where the 'tensorA' and 'tensorB' files are located. DISPER needs no further input.

---

[39] M. Medves, G. Fronzoni, M. Stener, *Optimization of density fitting auxiliary Slater-type basisfunctions for time-dependent density functional theory*, Journal of Computational Chemistry 43, 1923 (2022) (https://doi.org/10.1002/jcc.26992)

DISPER program: Dispersion Coefficients

The DISPER program was originally written by V.Osinga[28]. The original documentation was written by S.J.A. van Gisbergen.

Van der Waals dispersion coefficients

The program DISPER computes Van der Waals dispersion coefficients up to C_{10} for two arbitrary closed-shell molecules. ADF itself can already compute some C_6 and C_8 coefficients between two identical closed-shell molecules. These coefficients describe the long-range dispersion interaction between two molecules. It requires previous ADF-TDDFT calculations for the polarizability tensors at imaginary frequencies for the two interacting molecules. Each such ADF calculation produces a file TENSOR (if suitable input for ADF is given). The TENSOR files must be renamed tensorA and tensorB, respectively and must be present as local files for DISPER. The DISPER program takes no other input and prints a list of dispersion coefficients.

A schematic example, taken from the set of sample runs, for the usage of DISPER is the following:

Step1: run ADF for, say, the HF molecule. In the input file you specify the RESPONSE data block:

```
RESPONSE
  MaxWaals 8    ! Compute dispersion coefficients up to C8
  ALLTENSOR     ! This option must be specified in the ADF calc for a
                ! subsequent DISPER run
  ALLCOMPONENTS ! Must also be specified for DISPER
End
```

At the end of the run, copy the local file 'TENSOR' to a file 'tensorA'. For simplicity, we will now compute the dispersion coefficients between two HF molecules. Therefore, copy 'tensorA' to 'tensorB'.

Now run DISPER (without any other input). It will look for the local files 'tensorA' and 'tensorB' and compute corresponding dispersion coefficients to print them on standard output.

```
$AMSBIN/disper -n1 << eor
eor
```

The output might look something like this:

```
DISPER 2000.02 RunTime: Apr04-2001 14:14:13
  C-COEFFICIENTS
  n   LA  KA  LB  KB  L  coefficient(Y)    coefficient(P)
  6   0   0   0   0   0   28.29432373         28.29432373
  6   2   0   0   0   2    7.487547697         3.348533127
  8   0   0   0   0   0  416.1888455         416.1888455
  8   0   0   2   0   2    0.4323024202E-05    0.1933315197E-05
  8   2   0   0   0   2  402.3556946         179.9389368
  8   2   0   2   0   4    0.4238960180E-05
  8   4   0   0   0   4  -36.67895539        -12.22631846
  8   4   0   2   0   6   -0.2000286301E-05
```

The n-value in the first column refers to the long-range radial interaction. The case n=6 refers to the usual dipole-dipole type interaction related to a $1/R^6$ dependence in the dispersion energy. The n=7 case relates to a dipole-quadrupole polarizability on one system and a dipole-dipole polarizability on the other (this is not symmetric!). The n=8 term may contain contributions from a quadrupole-quadrupole polarizability on one system in combination with a dipole-dipole polarizability on the other as well as contributions from two dipole-quadrupole polarizabilities.

---

[28] V.P. Osinga, S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Density functional results for isotropic and anisotropic multipole polarizabilities and* C_6 , C_7 , and C_8 *Van der Waals dispersion coefficients for molecules*, Journal of Chemical Physics 106, 5091 (1997) (https://doi.org/10.1063/1.473555)

Terms which are zero by symmetry are not printed. In the example above, this is the case for all n=7 terms, because the systems (apparently) are too symmetric to have a nonzero dipole-quadrupole polarizability. The best known and most important coefficients are the isotropic ones, determining the purely radial dependence of the dispersion energy. They are characterized by the quantum numbers: 6 0 0 0 0 0 (or 8 0 0 0 0 0 etc.) Other combinations of quantum numbers refer to different types of angular dependence. The complete set determines the dispersion energy for arbitrary orientations between the two subsystems A and B.

The complete expressions are rather involved and lengthy. We refer the interested reader to the paper[?] which contains a complete description of the meaning of the various parts of the output, as well as references to the earlier literature which contain the mathematical derivations. In particular, a useful review, which was at the basis of the ADF implementation, is given in[30]. Of particular significance is Eq.(8) of the JCP paper mentioned above, as it defines the meaning of the calculated coefficients $C_n^{L_A,K_A,L_B,K_B,L}$ as printed above.

For highly symmetric systems, a different convention is sometimes employed. It is based on Legendre polynomials (hence the 'P' in the final column) instead of on the spherical harmonics (the 'Y' in the column before the last). The 'P' coefficients are defined only for those coefficients that are nonzero in highly symmetric systems and never contain additional information with respect to the 'Y' coefficients. They are defined [Eq. (14) in the mentioned J. Chem. Phys. paper] in terms of the 'Y' coefficients by:

$$C_n^L = (-1)^L C_n^{L,0,0,0,L}/\sqrt{2L+1}$$

Because the quality of the dispersion coefficients is determined by the quality of the polarizabilities that are the input for DISPER, it is important to get good polarizabilities from ADF. For that it is important, in the case of small systems, to use an asymptotically correct XC potential (several choices are available in ADF, such as SAOP or GRAC) and a basis set containing diffuse functions. We refer to the ADF User's Guide for details.

7.7 Ligand Field and Density Functional Theory (LFDFT)

7.7.1 Introduction

Hans Bethe and John Hasbrouck van Vleck have introduced the Ligand Field (LF) model simultaneously in the 1930s. This theory often plays a central role for molecules containing d- and/or f-elements with open shells. It is a semi-empirical model with adjustable parameters. Twenty years later Christian Klixbüll Jorgensen and Klaus Erik Schäffer proposed the Angular Overlap Model (AOM) that is a transcription of LF theory using molecular orbitals, still using adjustable parameters. Starting from the very beginning of the twenty first century until nowadays, a new, non-empirical, Density Functional Theory (DFT) based Ligand Field model[1][2] was developed, called LFDFT. LFDFT emerged through the collaboration of many European scientists during these years.

The key feature of the LFDFT approach is the explicit treatment of near degeneracy correlation using ad hoc Configuration Interaction (CI) within the active space of Kohn-Sham (KS) orbitals with dominant p-, d- or f-character. The calculation of the CI-matrices is based on a symmetry decomposition in the full rotation group and/or on a ligand field analysis of the energies of all single determinants (micro-states) calculated according to Density Functional Theory (DFT) for frozen KS-orbitals corresponding to the averaged configuration, possibly with fractional occupations, of the p-, d- or f-orbitals. In the literature one can find cases where this procedure yields multiplet energies with an accuracy of a few hundred wave numbers and fine structure splitting accurate to less than a tenth of this amount. This procedure was used to calculate different molecular properties, e.g. Zero Field Splitting (ZFS)[3], Zeeman interaction[4], Hyper-Fine Splitting

---

[30] A. van der Avoird, P.E.S. Wormer, F. Mulder, R.M. Berns, Topics in Current Chemistry 93, 1 (1980)

[1] M. Atanasov, C.A. Daul, C. Rauzy, *New insights into the effects of covalency on the ligand field parameters: a DFT study*, Chemical Physics Letters 367, 737 (2003) (https://doi.org/10.1016/S0009-2614(02)01762-1)

[2] M. Atanasov, C.A. Daul, C. Rauzy, *A DFT Based Ligand Field Theory*, Structure & Bonding 106, 97 (2004) (https://doi.org/10.1007/b11308)

[3] A. Borel, C.A. Daul, L. Helm, *Hybrid ligand-field theory/quantum chemical calculation of the fine structure and ZFS in lanthanide(III) complexes*, Chemical Physics Letters 383, 584 (2004) (https://doi.org/10.1016/j.cplett.2003.11.082)

[4] M. Atanasov, E.J. Baerends, P. Beattig, R. Bruyndockx, C. Daul, C. Rauzy, *The calculation of ESR parameters by density functional theory: the g- and A-tensors of Co(acacen)*, Chemical Physics Letters 399, 433 (2004) (https://doi.org/10.1016/j.cplett.2004.10.041)

(HFS)[5], magnetic exchange coupling?[6], and shielding constants[7].

Strictly speaking, LF theory is nothing but the consideration of active p-, d- and/or f-electrons moving in the potential of a passive chemical environment. A rigorous formulation of a passive chemical environment could be given using a frozen density embedding (FDE) scheme. However, in practice, it turns out that the method described in Ref.[8] using an effective Hamiltonian is more advantageous and yields more insights to the experimentalists. The LFDFT model was extended to two-open-shell systems[9]. This has relevance for inter-shell transitions in lanthanides, important for understanding the optical and magnetic properties of rare-earth materials. It could also be used, for example, to calculate multiplet effects in X-ray absorption spectroscopy. Theory and applications for f → d transitions can be found in Refs.?[10][11].

7.7.2 Input description

Average of Configuration run

From a defined atomic structure (for example, obtained from geometry optimization or crystal .cif structure file), run a DFT calculation representing the electron configuration system from which the metal ion belongs. This DFT run is achieved by the Average of Configuration (AOC) calculation with fractional occupations of s, p, d or f orbitals, i.e. for $3d^7$ electron configuration of Co^{2+}, 7 electrons are evenly distributed in the molecular orbitals having dominant cobalt character. Choosing which orbitals have to be fractionally occupied may not always be trivial, since there may be several close-lying levels with different character. This may lead to problematic SCF convergence. One should check that the final fractionally occupied molecular orbitals indeed have the expected metal character, since otherwise the subsequent LFDFT calculation will be meaningless. It is required that the metal ion is placed as first atom in the coordinate system. Besides, the calculation implies a single-point spin restricted SCF without symmetry constraint (C1 point group = SYMMETRY NOSYM). A scalar relativistic ZORA Hamiltonian should be used.

For a Co^{2+} d^7 electron configuration the input for ADF could be something like:

```
$AMSBIN/ams << eor

System
  Atoms
    Co 0.0 0.0 0.0
    ...
  End
End

...

Engine ADF
  ...
  IrrepOccupations
    A 56 1.4 1.4 1.4 1.4 1.4
```

(continues on next page)

---

[5] M. Atanasov, C.A. Daul, *A DFT based ligand field model for magnetic exchange coupling in transition metal dimmer complexes: (i) principles*, Chemical Physics Letters 379, 209 (2003) (https://doi.org/10.1016/S0009-2614(03)01325-3)

[6] M. Atanasov, C.A. Daul, *A DFT based ligand field model for magnetic exchange coupling in transition metal dimer complexes:: (ii) application to magnetic systems with more than one unpaired electron per site*, Chemical Physics Letters 381, 584 (2003) (https://doi.org/10.1016/j.cplett.2003.10.024)

[7] F. Senn, C.A. Daul, *Calculation of* ^{59}Co *shielding tensor* σ *using LF-DFT*, Journal of Molecular Structure: THEOCHEM 954, 105 (2010) (https://doi.org/10.1016/j.theochem.2010.02.027)

[8] C. Daul, *Non-empirical Prediction of the Photophysical and Magnetic Properties of Systems with Open d- and f-Shells Based on Combined Ligand Field and Density Functional Theory (LFDFT)*, Chimia 68 (2014) (https://doi.org/10.2533/chimia.2014.633)

[9] H. Ramanantoanina, W. Urland, F. Cimpoesu, and C. Daul, *Ligand field density functional theory calculation of the* $4f^2 \rightarrow 4f^1\ 5d^1$ *transitions in the quantum cutter* $Cs_2KYF_6{:}Pr^{3+}$, Physical Chemistry Chemical Physics 15, 13902 (2013) (https://doi.org/10.1039/C3CP51344K).

[10] H. Ramanantoanina, W. Urland, A. Garcia-Fuente, F. Cimpoesu, and C. Daul, *Ligand field density functional theory for the prediction of future domestic lighting*, Physical Chemistry Chemical Physics 16, 14625 (2014) (https://doi.org/10.1039/C3CP55521F).

[11] H. Ramanantoanina, M. Sahnoun, A. Barbiero, M. Ferbinteanu, F. Cimpoesu, *Development and applications of the LFDFT: the non-empirical account of ligand field and the simulation of the f–d transitions by density functional theory*, Physical Chemistry Chemical Physics 17, 18547 (2015) (https://doi.org/10.1039/C5CP02349A).

(continued from previous page)

```
  End
  relativity level=scalar
  SYMMETRY NOSYM
EndEngine
eor
```

LFDFT atomic database

The LFDFT atomic database is *not* included in the ADF distribution.

The easiest way to install it is using the ADF-GUI and open AMSinput and go to the LFDFT section (**Properties → Ligand Field DFT**). Then it will automatically be installed via AMSpackages if needed. You can also install it from the command line using `$AMSBIN/amspackages install lfdft`, or with the AMSpackages gui `$AMSBIN/amspackages gui`.

Alternatively, you can download the LFDFT atomic database LDFDT from http://downloads.scm.com/Downloads/lfdft/LFDFT.zip, and save the unzipped file in the a different location. If you save this LFDFT atomic database in a different location, make sure to point the environment variable SCM_LFDFT to this directory.

Note that at present available electron configurations for the LFDFT atomic database are:

- s^n, n=0,..,2
- p^n, n=0,..,6
- d^n, n=0,..,10
- f^n, n=0,..,14
- p^5 d^n, n=0,..,10
- f^n d^1, n=1,..,13
- d^9 f^n, n=1,..,14

LFDFT run

The adf.rkf (TAPE21) file of the average of configuration (AOC) run should be an input file for the program lfdft. Results of the LFDFT calculation will be put on adf.rkf. The input for lfdft is keyword oriented and is read from the standard input. Spin-orbit coupling is calculated using the ZORA equation by default. In cases where this is an underestimation, the old approximate method (a simple core potential) and/or an extra multiplication factor can be included. A finite magnetic field can be included.

```
$AMSBIN/lfdft << eor
ADFFILE adffile
NSHELL nshell
NLVAL1 nval1 lval1
{NLVAL2 nval2 lval2}
MOIND1 MO#1 MO#2 ... MO#(2*lval1+1)
{MOIND2 MO#1 MO#2 ... MO#(2*lval2+1)}
SOC soc1 {soc2}
SOCTYPE
   shell1 type1
   {shell2 type2}
End
BField bx by bz
```

(continues on next page)

(continued from previous page)

```
DegeneracyThreshold threshold
eor
```

`ADFFILE adffile` Path to adf.rkf (TAPE21) file from which lfdft reads data and to which lfdft writes data. Default TAPE21.

`NSHELL nshell` nshell is the number of shells in the electron configuration system under consideration: for Co^{2+} in $3d^7$ electron configuration, nshell = 1; for Fe^{2+} in $2p^5$ $3d^7$, nshell = 2. nshell should be equal to 1 or 2.

`NLVAL1 nval1 lval1` nval1, and lval1, are the main quantum number n and l for shell 1, in case of all-electron calculations on the metal atoms. For 5d this is '5 2'. For frozen core calculations the number n should be reduced with the number of core levels with angular momentum l that are in the core. For example for 5d, with 3d and 4d in the core, one should use '3 2'.

`NLVAL2 nval2 lval2` If nshell=2, nval2, and lval2, are the main quantum number n and l for shell 2.

`MOIND1 MO#1 MO#2 ... MO#(2*lval1+1)` MO#1, …, are the indices of the molecular orbitals which have the most dominant nval1 lval1 character, for example the most dominant 3d character. They should match the fractionally occupied orbitals that are used in the AOC run for shell 1.

`MOIND2 MO#1 MO#2 ... MO#(2*lval1+1)` If nshell=2, MO#1, …, are the indices of the molecular orbitals which have the most dominant nval2 lval2 character. They should match the fractionally occupied orbitals that are used in the AOC run for shell 2.

`SOC soc1 {soc2}` soc1 indicates if spin-orbit coupling is considered (1) or not (0) for shell 1. If nshell=2, same applies for soc2 for shell 2. soc1 and soc2 can also be real numbers, in case one wants to scale the calculated approximate spin-orbit coupling(s) by LFDFT. By default, the spin-orbit coupling is included (soc1 = 1, soc2 = 1).

```
SOCTYPE
  shell1 type1
  {shell2 type2}
End

type1 and type2 can be either 'zora' or 'core' and indicate whether the spin-orbit␣
↪coupling for the given shell is calculated using the corresponding term in the zora␣
↪equation (type = zora) or using the old approximation with the core potential only␣
↪(type = core). The default is set to zora.
```

`BField bx by bz` Include a finite magnetic field (Tesla). For MCD calculations include a magnetic field in the z-direction. For example, `BField 0.0 0.0 1.0` is a magnetic field of 1.0 Tesla in the z-direction. Default no magnetic field is included. The DegeneracyThreshold should be small to see the splitting of levels due to the magnetic field.

`DegeneracyThreshold threshold` Energy difference threshold (eV) to determine degenerate levels.

For example for a $3d^7$ electron configuration the input for lfdft could be

```
$AMSBIN/lfdft <<eor
ADFFile ams.results/adf.rkf
NSHELL 1
NLVAL1 3 2
MOIND1 29 30 31 32 33
SOC    1
SOCTYPE
   shell1 zora
End
eor
```

EPR (ESR) g-tensor

LFDFT can be used to calculate the EPR (ESR) g-tensor, see Ref.[?]. In the LFDFT implementation in ADF the EPR (ESR) g-tensor will be calculated only for Kramers doublet states. Results of the calculations should normally only be used for an odd number of electrons. Spin-orbit coupling should be included, but a finite magnetic field should not be included, and no other keywords are needed. One should be careful when interpreting the g-tensor if 2 or more Kramer doublets are close in energy. In that case in the effective Hamiltonian used to interpret EPR (ESR) experiments an effective spin higher than S=1/2 might be used, which is not taken into account in the calculations.

LFDFT intensities

One can calculate excitation energies and oscillator strengths between two atomic multiplet states which come from different electron configurations of the same molecule with the module lfdft_tdm. Note that electronic transitions between two multiplet states which come from the same electron configuration are not dipole allowed. lfdft_tdm can calculate excitation energies and oscillator strengths (in the dipole approximation) from the calculated ground state multiplet of one electron configuration to all multiplet states that can be made for the other electron configuration. The calculated transition dipole moments and oscillator strengths are in arbitrary units. They are averaged over the degeneracy of the ground state as well as over the degeneracy of the excited state multiplet. The oscillator strength has to be multiplied with the degeneracy of the excited state multiplet.

The input for lfdft_tdm is keyword oriented and is read from the standard input.

```
$AMSBIN/lfdft_tdm << eor
STATE1 file1
STATE2 file2
eor
```

`STATE1 file1` Filename file1 should be a result adf.rkf of a lfdft calculation, that contains the ground state electron configuration.

`STATE1 file2` Filename file2 should be a result adf.rkf of a lfdft calculation, that contains the excited state electron configuration. Results of the lfdft_tdm calculation will be put on file2.

For example for Pr $4f^2$ → Pr $4f^1$ $5d^1$ the input could be something like:

```
$AMSBIN/lfdft_tdm <<eor
STATE1 f2.results/adf.rkf
STATE2 f1d1.results/adf.rkf
eor
```

In this case lfdft_tdm will calculate the excitations from the ground state of Pr $4f^2$ to all multiplet states that can be made with the Pr $4f^1$ $5d^1$ electron configuration.

MCD

LFDFT can be used to calculate Magnetic circular dichroism (MCD) effect in X-ray absorption spectroscopy, see Ref.[12]. By using the same methodology as for an X-ray absorption calculation, one needs 2 LFDFT calculations for two atomic multiplet states which come from different electron configurations of the same molecule. Additionally, in both calculations the same finite magnetic field in the z-direction (for example, Bfield 0 0 1) should be included. Next the module lfdft_tdm can be used to calculate besides the total intensity, also the polarized part that is perpendicular to the magnetic axis, the circular-left and circular-right part.

---

[12] H. Ramanantoanina, M. Studniarek, N. Daffé, J. Dreiser, *Non-empirical calculation of X-ray magnetic circular dichroism in lanthanide compounds*, Chemical Communications 55, 2988 (2019) (https://doi.org/10.1039/C8CC09321K).

7.8 NMR

NMR chemical shifts and NMR spin-spin couplings can be calculated. Effects due to spin-orbit coupling can be included. All electron basis sets can be used.

Worked out examples on chemical shift- and spin-spin-coupling calculations can be found in the **tutorials section**:

- Chemical shifts and nuclear spin-spin coupling constants w. NBO analysis: Analysis of NMR parameters with Localized Molecular Orbitals
- Chemical shifts and nuclear spin-spin coupling constants w. visualization of the spectrum in the GUI: H-NMR spectrum with spin-spin coupling

7.8.1 NMR Chemical Shifts

NMR Chemical shifts have been implemented[1][2][3][4][5] in a separate property program NMR. It requires the adf.rkf (TAPE21) result file from an ADF calculation. The NMR module can be combined with the ZORA treatment for relativistic effects and with Spin-Orbit effects, making it suitable for treatment of heavy elements. See also the general review on relativistic computations of NMR parameters[6].

Two GUI tutorials on NMR Chemical shift calculations are currently available:

- Chemical shifts and nuclear spin-spin coupling constants w. NBO analysis: Analysis of NMR parameters with Localized Molecular Orbitals
- Chemical shifts and nuclear spin-spin coupling constants w. visualization of the spectrum in the GUI: H-NMR spectrum with spin-spin coupling

The NMR chemical shift δ_i is defined as the difference of the isotropic NMR chemical shielding of a reference compound σ_{ref} minus that of the interesting compound σ_i:

$$\delta_i = \sigma_{ref} - \sigma_i$$

The NMR program calculates NMR chemical shieldings σ_i.

---

[1] G. Schreckenbach and T. Ziegler, *The calculation of NMR shielding tensors using GIAO's and modern density functional theory*, Journal of Physical Chemistry 99, 606 (1995) (https://doi.org/10.1021/j100002a024)

[2] G. Schreckenbach and T. Ziegler, *The calculation of NMR shielding tensors based on density functional theory and the frozen-core approximation*, International Journal of Quantum Chemistry 60, 753 (1996) (https://doi.org/10.1002/(SICI)1097-461X(1996)60:3%3C753::AID-QUA4%3E3.0.CO;2-W)

[3] G. Schreckenbach and T. Ziegler, *Calculation of NMR shielding tensors based on density functional theory and a scalar relativistic Pauli-type Hamiltonian. The application to transition metal complexes*, International Journal of Quantum Chemistry 61, 899 (1997) (https://doi.org/10.1002/(SICI)1097-461X(1997)61:6%3C899::AID-QUA3%3E3.0.CO;2-R)

[4] S.K. Wolff and T. Ziegler, *Calculation of DFT-GIAO NMR shifts with inclusion of spin-orbit coupling*, Journal of Chemical Physics 109, 895 (1998) (https://doi.org/10.1063/1.476630)

[5] S.K. Wolff, T. Ziegler, E. van Lenthe and E.J. Baerends, *Density functional calculations of nuclear magnetic shieldings using the zeroth-order regular approximation (ZORA) for relativistic effects: ZORA nuclear magnetic resonance* , Journal of Chemical Physics 110, 7689 (1999) (https://doi.org/10.1063/1.478680)

[6] J. Autschbach and S. Zheng, *Relativistic computations of NMR parameters from first principles: Theory and applications*, Annual Reports on NMR Spectroscopy 67, 1 (2009) (https://doi.org/10.1016/S0066-4103(09)06701-5)

Important notes

adf.rkf (TAPE21) and TAPE10

NMR requires an ASCII input file and adf.rkf (TAPE21) and TAPE10 result files from an ADF calculation on the molecule to be analyzed. The ADF result files adf.rkf (TAPE21) and TAPE10 must be present with names TAPE21 and TAPE10 in the directory where you execute NMR, or alternatively one can use the keys ADFFile and TAPE10File to point to an adf.rkf file and TAPE10 file. Use the keywords SAVE TAPE10 in the adf calculation in order to obtain a TAPE10 result file.

Recalculation of TAPE10 by NMR

Warning: the NMR property program will not always give the correct result for every SCF potential in the ADF calculation, like for example the SAOP potential, or if one uses COSMO in the ADF calculation, if one lets the NMR recalculate TAPE10. This is due to the GIAO method used in this program, which requires the calculation of the SCF potential, which is not done correctly for potentials, other than the standard LDA and GGA potentials. To obtain correct results one should, in addition to the use of TAPE21, also use TAPE10 that ADF generates, using the keywords SAVE TAPE10, and use it as input for the NMR property program. On TAPE10 the SCF potential is written, which is read in by the NMR program.

Atomic calculation

NMR calculations on 1 atom must have symmetry NOSYM.

Spin-orbit coupling

NMR calculations on systems computed by ADF with Spin Orbit relativistic effects included must have used NOSYM symmetry in the ADF calculation. NMR can also be combined with ADF ZORA calculations. The NMR program reads from adf.rkf (TAPE21) the relativistic option that is used in the ADF calculation, and will use the same relativistic option in the NMR calculations.

Kernel used in spin-orbit coupling

One can include an improved exchange-correlation kernel, as was implemented by J. Autschbach[7], but this is not used by default, see subkey USE. This is an important option that should be seriously considered and has been advocated in Ref.[7].

Bug spin-orbit part ADF2008 - ADF2013

In the ADF2008.01 a bug was introduced in the spin-orbit part of the calculated chemical shielding, which caused the calculated chemical shielding to be gauge dependent. This bug is relevant for spin-orbit coupled calculations for ADF versions ADF2008-ADF2013. In ADF2014 this bug has been fixed.

Unscaled ZORA default ADF2014

There is gauge dependence if the scaled ZORA method is used in the calculation of NMR chemical shieldings. Therefore the default method for NMR chemical shielding calculations is changed in ADF2014 to use the unscaled ZORA method.

SAOP

The use of the model SAOP potential leads to isotropic chemical shifts which are substantially improved over both LDA and GGA functionals, and of similar accuracy as results with a self-interaction-corrected functional (SIC), see[8]. SAOP is computationally expedient and routinely applicable to all systems, requiring virtually the same computational effort as LDA and GGA calculations.

NICS

The Nucleus-Independent Chemical Shift (NICS) can be calculated at any point in the molecule.

Hybrids

---

[7] J. Autschbach, *The role of the exchange-correlation response kernel and scaling corrections in relativistic density functional nuclear magnetic shielding calculations with the zeroth-order regular approximation*, Molecular Physics 111, 2544 (2013) (https://doi.org/10.1080/00268976.2013.796415)

[8] J. Poater, E. van Lenthe and E.J. Baerends, *Nuclear magnetic resonance chemical shifts with the statistical average of orbital-dependent model potentials in Kohn.Sham density functional theory*, Journal of Chemical Physics 118, 8584 (2003) (https://doi.org/10.1063/1.1567252)

Starting from ADF2009.01 Hartree-Fock and the hybrid potentials can used in combination with NMR chemical shielding calculations. see Refs.[9][10]. Use SAVE TAPE10 in the ADF calculation. The use of frozen cores and hybrids gives gauge dependent results for the NMR chemical shieldings, therefore the NMR program will stop in this case.

Meta-GGA's and meta-hybrids

Meta-GGA's and meta-hybrids should not be used in combination with NMR chemical shielding calculations. The results are wrong due to an incorrect inclusion of GIAO terms.

Input options

The input file for NMR uses the block key NMR, with several (optional) subkeys, each having a series of options. For analysis a separate block key can be used.

```
$AMSBIN/nmr << eor
ADFFile adffile
TAPE10File tape10file
ZSOAO2007
RECALCULATETAPE10
NMR
End
Analysis
End
eor
```

ADFFile adffile

Path to adf.rkf (TAPE21) file from which nmr reads data and to which nmr writes data. Default TAPE21.

TAPE10File tape10file

Path to TAPE10 file from which nmr reads data. Default TAPE10.

ZSOAO2007 keyword

In the ADF2008.01 a bug was introduced in the spin-orbit part of the calculated chemical shielding, which caused the calculated chemical shielding to be gauge dependent. This bug is relevant for spin-orbit coupled calculations for ADF versions ADF2008-ADF2013. Workaround in ADF versions ADF2008-ADF2013 is to include the keyword ZSOAO2007 in the NMR part of the input, which causes a one-center approximation to be used. The bug has been fixed in ADF2014 by introducing an extra gauge-correction term for the spin-orbit coupled part. One can still get the (slightly) incorrect results in ADF2014 by using the keyword WRONGSOGAUGE, and not including ZSOAO2007.

REACALCULATETAPE10 keyword

If there is no TAPE10 present the NMR program will stop. One can use the key REACALCULATETAPE10 such that TAPE10 will be recalculated by the NMR module. Not recommended to be used. Better use 'SAVE TAPE10' in the ADF calculation, and use this TAPE10 as input for NMR.

NMR block key

```
NMR
  Out  OutOptions
  Calc CalcOptions
  Use UseOptions
```

(continues on next page)

---

[9] M. Krykunov, T. Ziegler and E. van Lenthe, *Hybrid density functional calculations of nuclear magnetic shieldings using Slater-type orbitals and the zeroth-order regular approximation*, International Journal of Quantum Chemistry 109, 1676 (2009) (https://doi.org/10.1002/qua.21985)

[10] M. Krykunov, T. Ziegler and E. van Lenthe, *Implementation of a hybrid DFT method for calculating NMR shieldings using Slater-type orbitals with spin-orbital coupling included. Applications to* ^{187} Os, ^{195} Pt and ^{13} C *in heavy metal complexes*, Journal of Physical Chemistry A 113, 11495 (2009) (https://doi.org/10.1021/jp901991s)

(continued from previous page)

```
  U1K  U1KOptions
  Nuc  NucOptions
  Atoms  AtomsOptions
  Ghosts  GhostsOptions
  Analysis  AnalysisOptions
End
```

`Out OutOptions` The subkey Out controls printed output. Its options specify the details by their (optional) presence. The following OutOptions are recognized (Default ISO TENS):

- **`All`** Implies all the other options
- **`ISO`** Isotropic shielding constants
- **`Tens`** Shielding tensors
- **`Eig`** Eigenvectors
- **`U1`** The U1 matrix
- **`F1`** The first order change in the Fock matrix
- **`S1`** The first order change in the Overlap matrix
- **`AOP`** The paramagnetic AO matrix (= the matrix in the representation of elementary atomic basis functions)
- **`AOD`** The diamagnetic AO matrix
- **`AOF`** The Fermi-contact AO matrix
- **`REFS`** Literature references
- **`INFO`** General information

`Calc CalcOptions` The subkey Calc controls what is actually calculated. The following options are available (Default ALL):

- **`All`** Implies all of the other options to this key
- **`Para`** The paramagnetic part
- **`Dia`** The diamagnetic part
- **`FC`** The Fermi-contact part in case of the Pauli Hamiltonian
- **`SO`** The Fermi-contact part in case of the ZORA Hamiltonian

`Use UseOptions` The subkey Use controls some optional options (default none)

- **`FXC`** Improves the exchange-correlation kernel used, as was implemented by J. Autschbach[?]. Important only in case of spin-orbit coupled calculations. This may give some (small) gauge dependent results when using this. Important option that should be seriously considered and has been advocated in Ref.[?].
- **`SCALED`** Implies the scaled ZORA method, which gives (slightly) gauge dependent results, as was shown in Ref.[?]. Note that in case of the ZORA Hamiltonian default the unscaled ZORA method is used. For chemical shifts, only compare results with the same options.
- **`SO1C`** Before ADF2008.01 in the the spin-orbit term a 1-center approximation was used, which does not suffer from gauge dependence. This 1-center approximation can be used with USE SO1C.

`U1K U1KOptions` The subkey U1K determines which terms are included in the calculation of the U1 matrix (first order changes in MO coefficients). Options (Default Best):

Best The best (recommended) options for each relativistic option are included for this subkey. Implies 'None' for non-relativistic and scalar relativistic ZORA, 'SO' + 'SOFULL' for spin-orbit coupled ZORA, and 'MV + 'Dar' for the Pauli Hamiltonian.

None Implies none of the other options to this key.

All Implies all the other options to this key.

MV The mass-velocity term .

Dar The Darwin term.

ZMAN The Spin-Zeeman term. Can be included only in case of spin-orbit coupled Pauli Hamiltonian.

SO ZORA spin-orbit part.

SOFULL ZORA spin-orbit part.

Note: for chemical shifts, only compare results with the same options. If the subkey U1K is used with the option ALL in the ZORA calculation, then the scaled ZORA orbital energies are used in the making of the U1 matrix, which is not recommended. Recommended is to use 'U1K Best' in all cases, which uses plain ZORA orbital energies in the making of the U1 matrix.

NUC NucOptions This subkey NUC specifies for which nuclei the NMR shielding is calculated. Default all nuclei are calculated, i.e. as for omitting the subkeys ATOMS and NUC. Else you may use this options by simply typing Nuc in the NMR block (without any further data); this means: for no nuclei at all. Alternatively you may type the index of the atom(s) you want to see analyzed. Default all nuclei are calculated, i.e. as for omitting this subkey.

Example:

```
NUC 2 1
```

The numbers refer to the internal numbering of the nuclei as it appears somewhere early in the general ADF output. This internal numbering is also the internal NMR numbering, but it is not necessarily the same as the input ordering. Use the subkey ATOMS to specify the nuclei according to this input ordering in the ADF calculation.

Note that the number of nuclei has a significant consequence for the total CPU time.

Atoms AtomsOptions This subkey ATOMS specifies for which nuclei the NMR shielding is calculated. Default all nuclei are calculated, i.e. as for omitting the subkeys ATOMS and NUC.

Example:

```
ATOMS 2 1
```

The numbers refer to the input ordering in the ADF calculation. Use the subkey NUC to specify the nuclei according to the internal NMR numbers of the atoms.

GHOSTS The subkey GHOSTS is a block type subkey. The format is:

```
Ghosts
  xx1 yy1 zz1
  xx2 yy2 zz2
  ...
SubEnd
```

With this key, the user can specify ANY point(s) within the molecule at which the shielding is to be calculated (whatever the physical meaning of this shielding is). One can think of those points as neutrons within the molecule. There is a publication by P. Schleyer et al. using a similar feature (J. Am. Chem. Soc. **118**, 6317, 1996). They call it NICS, Nucleus-Independent Chemical Shift. Note that the NICS value is minus 1 times the isotropic part of the shielding tensor that is calculated at these points:

```
xx1 yy1 zz1
```

real numbers that specify the Cartesian coordinates of 'ghost' 1, etc.

The coordinates have to be specified in Angstrom. The same set of coordinates has to be specified as 'point charges with charge zero' using the subsubkey MultipolePotential in the ElectrostaticEmbedding subkey of the key System in the AMS block. This is necessary in order to allow the appropriate distribution of integration points around the ghosts.

E.g., if you want to have two 'ghosts' with the coordinates xx1 yy1 zz1 and xx2 yy2 zz2 then you **must** also have in the input the key MultipolePotential key as follows:

```
System
  ...
  ElectrostaticEmbedding
    MultipolePotential
      Coordinates
        xx1 yy1 zz1   0.0
        xx2 yy2 zz2   0.0
      End
    End
  End
End
```

(the last number is the charge at these coordinates - zero).

Eventually, this step should be programmed internally but for now the procedure outlined above works. No check is done to verify whether those 'point charges' are taken care of or not, but their omission leads to unpredictable results.

Only Cartesian coordinates are possible for ghosts, even if the atoms were originally specified using internal coordinates. This shouldn't be a problem, though (e.g., one could start an ADF run of the molecule of interest, and get very soon the Cartesian coordinates of the atoms in the output. This run would then be aborted, and restarted with the ghosts specified as desired.) The ghosts are numbered in the output as NNUC+1, NNUC+2 … where NNUC is the total number of nuclei in this molecule. Default: no ghosts.

`Analysis AnalysisOptions` The subkey Analysis controls the MO analysis. After the word (subkey) Analysis you type an integer, which then specifies that the first so many MOs are to be analyzed. Default no Analysis. The value of this analysis subkey in the block key NMR is somewhat limited. The separate ANALYSIS block key can give more analysis of the NMR chemical shielding.

Analysis block key

The NMR shielding tensor, can be analyzed in detail, see Refs.[12][13]?. For the analysis option with the ANALYSIS block key there are some restrictions. In the ADF calculation all electron basis sets should have been used, and SYMMETRY NOSYM. Can not be used in case of non-relativistic calculations. The ADF calculation should use relativistic scalar ZORA or relativistic spinorbit ZORA. In case of scalar relativistic ZORA the keyword FAKESO should be added to the NMR input (outside of the NMR or Analysis block keys). The analysis utilizes the ZORA spin-orbit branch of the NMR code. For scalar ZORA, the NMR analysis contributions will appear in equivalent pairs for spin-orbitals even if the ADF calculation is closed-shell spin restricted. The MO numbering then also reflects this doubling of MOs. In the analysis, canonical MOs number 1 and 2 are the alpha and beta spin ADF MO 1, canonical MOs number 3 and 4 correspond to the alpha and beta spin ADF MO 2, and so on. In case of spinorbit relativistic ZORA the keyword FAKESO should not be included. Curly brackets are not part of the key. For an NBO analysis of NMR, see the *section on NBO analysis* (page 263).

---

[12] J. Autschbach, *Analyzing NMR shielding tensors calculated with two-component relativistic methods using spin-free localized molecular orbitals*, Journal of Chemical Physics 128, 164112 (2008) (https://doi.org/10.1063/1.2905235)

[13] J. Autschbach and S. Zheng, *Analyzing Pt chemical shifts calculated from relativistic density functional theory using localized orbitals: The role of Pt 5d lone pairs*, Magnetic Resonance in Chemistry 46, S45 (2008) (https://doi.org/10.1002/mrc.2289)

```
Analysis
  print threshold
  canonical
  {components}
End
{FakeSO}
```

`print threshold` The print keyword selects printout of contributions relative to the total diamagnetic, paramagnetic. For example in case of 'print 0.01' only contributions greater than 1% are printed. Set to zero to print ALL contributions.

`canonical` It enables an analysis of the shielding in terms of the canonical MOs.

`components` The components keyword is optional and enables an analysis not only of the isotropic shielding but also of the diagonal Cartesian components of the tensor XX, YY, and ZZ). In order to analyze the principal shielding tensor components with canonical MOs you can calculate the shielding tensor first with the NMR code, rotate the molecule such that the principal axes system aligns with the Cartesian coordinate system, and then repeat the NMR calculation with the analysis features switched on.

7.8.2 Paramagnetic NMR Chemical Shifts

Knowledge of the g-tensor and hyperfine A-tensor can be used in the prediction and analysis of paramagnetic NMR shifts, see Refs.[11][14][15]. Because of the dependence on the g-tensor, prediction of paramagnetic NMR (pNMR) shifts is not straightforward, as the dependence of the pNMR shift on excess α or β electron spin density at a nucleus is to be combined with the sign and magnitude of the isotropic g-value. Ref.[?] describes in detail how to calculate pNMR contact chemical shifts and pNMR pseudo-contact chemical shifts, using the ADF program. In ADF2024 (part of) this procedure can be included more automatically.

For doublets (effective spin S=1/2), or for effective spin S higher than 1/2 where one neglects effects due to zero-field splitting (ZFS, ESR D-tensor) the paramagnetic NMR shielding σ can be written as [Ref.[16]]:

$$\sigma = \sigma_{orb} - \frac{\mu_B}{g_N \mu_N} \frac{S(S+1)}{3k_B T} \mathbf{g} \cdot \mathbf{A}$$

Here σ_{orb} is the so called orbital dependent part of the NMR chemical shielding, μ_B is the Bohr magneton, μ_N is the nuclear magneton, g_N is the nuclear magnetic g-factor, k_B is the Boltzmann constant, and T is the temperature in K (Kelvin). In addition S is the effective spin, **g** is the g-tensor, and **A** is the A-tensor that are used in the effective spin Hamiltonian, which is commonly used for the interpretation of ESR experiments.

For open shell molecules in ADF the so called orbital dependent part of the NMR chemical shielding σ_{orb} can only be calculated without taken into account spin-orbit coupling.

```
# ADF calculation on the molecule

$AMSBIN/ams << eor
  ...
  Engine ADF
```

(continues on next page)

---

[11] J. Autschbach, S. Patchkovskii, and B. Pritchard, *Calculation of Hyperfine Tensors and Paramagnetic NMR Shifts Using the Relativistic Zeroth-Order Regular Approximation and Density Functional Theory*, Journal of Chemical Theory and Computation 7, 2175 (2011) (https://doi.org/10.1021/ct200143w)

[14] S. Moon, and S. Patchkovskii, *First-principles calculations of paramagnetic NMR shifts*, in Calculation of NMR and EPR parameters, ISBN13: 9783527307791, M. Kaupp, M. Bühl, V.G. Malkin, Editors, (Wiley, Weinheim, 2004).

[15] P. Hrobárik, Ro. Reviakine, A.V. Arbuznikov, O.L. Malkina, V.G. Malkin, F.H. Köhler, and M. Kaupp, *Density functional calculations of NMR shielding tensors for paramagnetic systems with arbitrary spin multiplicity: Validation on 3d metallocenes*, Journal of Chemical Physics 126, 024107 (2007) (https://doi.org/10.1063/1.2423003)

[16] B.Martin, J. Autschbach, *Temperature dependence of contact and dipolar NMR chemical shifts in paramagnetic molecules*, Journal of Chemical Physics 142, 054108 (2015) (https://doi.org/10.1063/1.4906318)

(continued from previous page)

```
    SpinPolarization value
    Unrestricted Yes
    Relativity
      Level Scalar
    End
    Basis
      Core None
    End
    Symmetry NOSYM
    SAVE TAPE10
  EndEngine
eor

# Calculation of the ESR A-tensors

$AMSBIN/cpl << eor
  adffile path/to/adf.rkf
  tape10file path/to/TAPE10
  hyperfine
    atoms add_all_atoms
  END
eor
rm logfile TAPE15

# Calculation of the ESR g-tensor

$AMSBIN/nmr << eor
  adffile path/to/adf.rkf
  tape10file path/to/TAPE10
  nmr
    gfactors
    out iso tens
    u1k best
    calc all
  END
eor
rm logfile TAPE15

# Calculation of the orbital dependent part of the NMR chemical shielding

$AMSBIN/nmr << eor
  ADFFile path/to/adf.rkf
  TAPE10File path/to/TAPE10
  ALLINONE
  nmr
     u1k best
     calc all
     out iso tens
  end
  Temperature temperature
eor
```

Note that one can only do this at the scalar relativistic ZORA level, one needs symmetry NOSYM, one needs all electron basis sets, and one needs to include the key ALLINONE in the input for NMR. If one does not specify the temperature, a default temperature of 298.15 K is used.

See also:

Examples: *pNMR chemical shielding* (page 882)

7.8.3 NMR spin-spin coupling constants

The NMR spin-spin coupling constants[1][2] have been implemented in a separate program CPL. It can be combined with ZORA and Spin-Orbit treatment of relativistic effects to study heavy elements. The original version of this part of the User's Guide was written by Jochen Autschbach, primary author of the CPL code.

Two GUI tutorials on spin-spin coupling constant calculations are currently available:

- Chemical shifts and nuclear spin-spin coupling constants w. NBO analysis: Analysis of NMR parameters with Localized Molecular Orbitals
- Chemical shifts and nuclear spin-spin coupling constants w. visualization of the spectrum in the GUI: H-NMR spectrum with spin-spin coupling

Introduction

The CPL code of the Amsterdam Density Functional program system allows the user to calculate *Nuclear Spin-spin Coupling Constants* (NSSCCs)[??]. NSSCCs are usually observed in NMR (Nuclear Magnetic Resonance) spectroscopy and give rise to the splitting of the signals of the NMR spectrum in multiplets. They contain a wealth of information about the geometric and electronic structure of the compound being investigated.

The calculation needs a standard adf.rkf (TAPE21) ADF output file. CPL reads also an input key and optional settings from stdin (usually from an input file). Technical parameters such as the maximum memory usage can be set here as well.

One of the key features of the program is its ability to treat heavy nuclei with the ZORA relativistic formalism. We refer the reader to the literature for details about our implementation[??], and the general review on relativistic computations of NMR parameters[3]. Please use the information printed in the output header of the CPL program in order to provide references of this work in scientific publications.

The development of the CPL program started in 2000. CPL provides the main functionality in order to evaluate NSSCCs based on DFT, as well as a number of additional features in order to provide an analysis of the results. Several analysis features for the coupling constant have been added, see the CONTRIBUTIONS sub key. Please report bugs or suggestions to SCM at support@scm.com.

Theoretical and technical aspects

Within the non-relativistic theory of nuclear spin-spin coupling, there are four terms contributing to the NSSCC between two nuclei A and B: the paramagnetic and diamagnetic orbital terms (OP and OD, respectively), and the electron-spin dependent Fermi-contact (FC) and spin-dipole term (SD). In the literature, the OP and OD terms are often named PSO and DSO (for paramagnetic and diamagnetic spin-orbital). In the more general ZORA formulation, very similar operators are responsible for the NSSCC, therefore we use the same terminology for the individual contributions. In general, the interpretation of the results for a heavy atom system is basically equivalent to a non-relativistic situation.

In most cases, the FC term yields the most important contribution to the NSSCC. However, many exceptions are known for which one or each of the other terms can be non-negligible or even dominant. We therefore suggest that you always check, at least for a smaller but similar model system, or by using a smaller basis set, which of the four terms are negligible and which are dominant.

---

[1] J. Autschbach and T. Ziegler, *Nuclear spin-spin coupling constants from regular approximate density functional calculations. I. Formalism and scalar relativistic results for heavy metal compounds*, Journal of Chemical Physics 113, 936 (2000) (https://doi.org/10.1063/1.481874)

[2] J. Autschbach, and T. Ziegler, *Nuclear spin-spin coupling constants from regular approximate relativistic density functional calculations. II. Spin-orbit coupling effects and anisotropies*, Journal of Chemical Physics 113, 9410 (2000) (https://doi.org/10.1063/1.1321310)

[3] J. Autschbach and S. Zheng, *Relativistic computations of NMR parameters from first principles: Theory and applications*, Annual Reports on NMR Spectroscopy 67, 1 (2009) (https://doi.org/10.1016/S0066-4103(09)06701-5)

By default, the CPL program computes the FC coupling between the first and all other nuclei of the molecule, respectively. Other couplings or the computation of the OP, OD and SD terms can be requested by input switches (see the 'Running CPL' section of this document for details).

All contributions to the NSSCC are evaluated with the help of the numerical integration scheme implemented into ADF. In general, the computation of the OD term is computationally very cheap, since only integrals involving the electron density have to be evaluated. The next expensive term is the OP term. For this contribution, the first-order perturbed MOs have to be computed. With the available density functionals in ADF, the OP term does not cause a change in the Kohn-Sham potential, and the first-order MOs can be computed directly (i.e. without an iterative procedure). This is equivalent to the approach that has been implemented in the NMR code for ADF.

Both the FC and the SD terms induce electron spin-density to first-order as a perturbation. Equivalent to the iterative solution of the unperturbed Kohn-Sham equations, the first-order MOs depend on that first-order spin-density, which in turn depends on the first-order MOs. Therefore, in order to evaluate the FC and SD NSSCC contributions, the CPL program carries out a SCF cycle. In the scalar or non-relativistic case, the computational cost for the FC term is comparable to an ADF single point calculation with a local density functional. The evaluation of the SD term is more expensive. The current implementation utilizes the CPL spin-orbit code to compute the combined FC+SD contribution and therefore leaves some room for future speed-ups. In most cases, the SD term yields a negligible NSSCC and the much faster code for the scalar- or non-relativistic FC term can be used. However, it is very important to include the SD term in the computation if coupling anisotropies are to be evaluated.

In the case where the NSSCC computation is based on spin-orbit coupled relativistic two component ZORA MOs, the SD term causes only a marginal increase in computational time as compared to the FC term alone. Generally, in this case the computational cost for the FC term is already approximately one order of magnitude higher than in the scalar or non-relativistic case, since the 3 (x, y, z) components of the spin-density with respect to 3 components of the perturbation, respectively, have to be determined self-consistently. The additional presence of the SD term only shows up in a somewhat more costly evaluation of the matrix elements of the perturbation operator. However, CPL spends most of its computational time in the SCF cycle. Therefore, in spin-orbit computations the computation of the FC+SD terms is the default. The OP term has to be evaluated self-consistently, too, in this case and is added as a perturbation in the SCF cycle upon request.

We use the terminology 'perturbing' and 'responding nucleus' within the CPL output. The 'perturbing' nucleus is the one, for which the first-order MOs have to be computed (self-consistently), while the NSSCC is then determined by these first-order MOs and the FC, SD, and OP matrix elements of the second, 'responding' nucleus. For the OD term, this distinction makes no sense but is used in the output for reasons of consistency.

Experimental NSSCCs between two nuclei A and B are usually reported as J(A,B) in Hertz. From a computational point of view, the so-called reduced NSSCCs K(A,B) are more convenient for comparisons. CPL outputs both. The J's are set to zero in case the nuclear magneto-gyric ratio of one of the nuclei A or B is not available at run time.

Further technical aspects and current limitations

In order to facilitate the future computation of rather large molecules, all matrix elements of the perturbation operators FC, SD, and OP are evaluated in the Slater AO basis that is specified as input in the CREATE runs of ADF. The AO matrix elements are further transformed to the basis of MOs and the calculation proceeds within the MO basis. This allows for a convenient analysis of the results in terms of contributions from individual occupied and virtual MOs. Such an analysis can be requested by input.

The matrix elements themselves as well as the first-order contributions to the potential are evaluated by numerical integration. The CPL code, which is parallelized, can use multiple processors for these steps of the computation. The accuracy setting for the numerical integration is of high importance to obtain accurate matrix elements. Furthermore, the basis set being employed needs to be flexible enough to describe the perturbation correctly. This means usually that modified basis sets have to be used in particular for heavy element calculations.

The first-order potential is currently approximated by the VWN functional. The X:math:*alpha* potential is available as an alternative but usually leads to less accurate results. In ADF2009.01 the first order potential of the PBE family of GGA functionals and the hybrid PBE0 functional can be used.

Currently, only spin-restricted computations for systems with an even number of electrons are supported. Further, the calculation does not make use of symmetry and must be based on an ADF run with input SYMMETRY=NOSYM. Non-Aufbau configurations are not supported. The atom input list must not contain dummy atoms.

With the present version of CPL, the SD term and the FC/SD cross term cannot be evaluated separately. Either, the sum of FC + SD + cross terms, or the FC term individually, are computed.

CPL is restartable after various time-consuming steps of the computation.

In ADF2009.01 the hybrid PBE0 functional can be used in combination with NMR spin-spin coupling calculations, see the documentation for the extra keys that are needed. However, other hybrid functionals and Hartree-Fock can not or should not be used in combination with NMR spin-spin coupling calculations.

In ADF2009.01 the effects of a finite size of a nucleus on the spin-spin couplings can be calculated. A finite size of the nucleus can be set with the NUCLEARMODEL key in the input for the ADF calculation.

Bug fix in case more than 1 perturbing atom and DSO or PSO

In the ADF2006.01b version a bug in the CPL module is fixed that gave problems in ADF2006.01 and older versions. The problem in ADF2006.01 and older versions is: In case there is more than 1 perturbing atom and the DSO or PSO term is calculated, only the results of the spin-spin couplings for the first perturbing atom are correct, but the results of the other spin-spin couplings may be incorrect.

Input file for CPL: adf.rkf

In order to run the CPL code, you need the general ADF output file adf.rkf named TAPE21 being present in the directory where CPL is running. Alternatively one can use the key ADFFile to point to an adf.rkf file. Most of the computation's specific settings will be taken from adf.rkf (TAPE21), such as the integration accuracy, the basis set, the density functional being employed, nuclear coordinates, and so on. *That also means that nearly all of the aspects that affect the quality of CPL's results are already determined in the input for the ADF run*. Five aspects are of particular importance here:

1. The numerical integration accuracy: the perturbation operators are large in the vicinity of the nuclei. Therefore, you have to make sure that the integration grid is fine enough in the atomic core regions.
2. The basis set: NSSCCs are sensitive chemical probes, and therefore flexible basis sets have to be employed in order to yield a valid description of the MOs that determine the NSSCCs. We have found that it is imperative to use at least a TZ2P basis set. Additional polarization functions in the valence shell may be necessary. Furthermore, the FC perturbation usually requires additional steep 1s functions (i.e. with exponents much higher than the nuclear charge) for a proper description. In the relativistic heavy element case, the use of additional steep basis functions as compared to the ZORA/TZ2P basis is mandatory. The use of steep functions is only of high importance for those nuclei, for which the NSSCC is to be evaluated. In ADF2017 specialized large all electron basis sets have been added in the directory $AMSHOME/atomicdata/ADF/ZORA/TZ2P-J and $AMSHOME/atomicdata/ADF/ZORA/QZ4P-J, suitable for NSSCCs, where steep basis functions have been added, which is especially important for the FC term. Previously in ADF2009.01 for a limited number of elements basis sets suitable for NSSCCs have been added to the ADF basis set directory, in the directory $AMSHOME/atomicdata/ADF/ZORA/jcpl. These specialized basis sets are especially important for the heavy NMR nuclei. For the nuclei for which NSSCCs are to be evaluated, it is necessary to use all-electron basis sets. This is not a restriction due to the implementation, but we have found that, with the available frozen core basis sets, the flexibility of the basis in the vicinity of the nuclei is not sufficient. It is possible to use frozen core basis sets if you add enough basis functions in the core region such that the basis approaches the flexibility of at least a double-zeta all-electron basis there[7]. In that sense, the savings in computational time due to usage of a frozen core basis are not as pronounced as in standard ADF computations. Unless reliable frozen-core basis sets for the NSSCC computation are available we strongly discourage the use of frozen core basis sets with the CPL program!
3. The finite size of a nucleus: typically, the isotropic J-couplings are reduced in magnitude by about 10 to 15 % for couplings between one of the heaviest NMR nuclei and a light atomic ligand, and even more so for couplings

between two heavy atoms, see Ref.[4]. However, one should have really large basis sets with tight basis functions to observe this effect in calculations, see the previous point about basis sets. The TZ2P-J, QZ4P-J, and jcpl basis sets are suitable for finite nucleus calculations. A finite size of the molecule can be set in the ADF program with the key NUCLEARMODEL:

```
NuclearModel Gaussian
```

4. The density functional: the results of the CPL code depend mostly on the shape of the MOs that have been determined by ADF, and their orbital energies. Both, in turn, depend on the density functional or Kohn-Sham potential that has been chosen for the ADF run (and the basis set quality). It is difficult to give a general advice here concerning the NSSCCs. So far we have found that the use of GGAs improves the NSSCCs with respect to experiment in most cases in comparison to LDA. Different GGAs often yield very similar results. Further, in particular for those cases for which the OP term is large or even dominant, both standard LDAs and GGAs sometimes do not provide an accurate enough description of the orbitals, and deviation of the CPL results as compared to experiment can be substantial. Future developments of density functionals might be able to cure these problems. For the time being, we recommend that you base the CPL run on different choices of density functionals in the ADF run, and investigate the convergence of the result with respect to basis set and integration accuracy. Note that CPL itself uses the VWN functional by default to determine the first-order perturbed MOs. There are enough indications to believe that this is a reasonable approximation for NMR purposes. In ADF2009.01 the first order potential of the PBE family of GGA functionals and the first order potential of the hybrid PBE0 functional can be used. See Refs.[6][7] for applications of such first order potentials. However, other hybrid functionals and Hartree-Fock can not or should not be used in combination with NMR spin-spin coupling calculations.

5. Modeling the experimental setup: computing such sensitive numbers as NMR chemical shifts and in particular NSSCCs can result in substantial deviations from experimental data. The simple reason might be that the isolated system that has been computed at zero temperature is not at all a good approximation to the system that has been studied experimentally. We[8][9] and other authors have found that in particular solvent effects can contribute very substantially to the NSSCC. In case you are comparing CPL results to experimental data obtained in strongly coordinating solvents we suggest that you consider solvent effects as a major influence. We have found that even weakly coordinating solvents can cause sizable effects on the NSSCCs for coordinatively unsaturated metal complexes. Other sources of errors can be the neglect of vibrational corrections to the NSSCCs (usually in the range of a few percent).

If the parameters of the underlying ADF computation are carefully chosen and the density functional is able to provide an accurate description of the molecule under investigation, it is possible to compute NSSCCs by means of DFT with very satisfactory accuracy (please note that for properties as sensitive as NSSCCs, agreement with experimental results within about 10% error can be regarded as quite good). Further, chemical trends will be correctly reproduced for a related series of molecules in most cases. However, due to the inherent approximate character of the density functionals currently available with ADF, and necessary basis set limitations, great care should be taken that the results are reliable. *CPL assumes Aufbau configurations. Please make sure that there are no empty orbitals with energies below the highest occupied MO (HOMO). In addition, the SYMMETRY NOSYM key has to be used in the ADF computation.*

---

[4] J. Autschbach, *Magnitude of Finite-Nucleus-Size Effects in Relativistic Density Functional Computations of Indirect NMR Nuclear Spin-Spin Coupling Constants*, ChemPhysChem 10, 2274 (2009) (https://doi.org/10.1002/cphc.200900271)

[6] J. Autschbach, *Two-component relativistic hybrid density functional computations of nuclear spin-spin coupling tensors using Slater-type basis sets and density-fitting techniques*, Journal of Chemical Physics 129, 094105 (2008) (https://doi.org/10.1063/1.2969100), Erratum: Journal of Chemical Physics 130, 209901 (2009) (https://doi.org/10.1063/1.3131724)

[7] D.L. Bryce and J. Autschbach, *Relativistic hybrid density functional calculations of indirect nuclear spin-spin coupling tensors . Comparison with experiment for diatomic alkali metal halides*, Canadian Journal of Chemistry 87, 927 (2009) (https://doi.org/10.1139/V09-040)

[8] J. Autschbach and T. Ziegler, *A Theoretical Investigation of the Remarkable Nuclear Spin.Spin Coupling Pattern in* [(NC)$_5$ Pt-Tl(CN)]$^-$, Journal of the American Chemical Society 123, 5320 (2001) (https://doi.org/10.1021/ja003866d)

[9] J. Autschbach and T. Ziegler, *Solvent Effects on Heavy Atom Nuclear Spin.Spin Coupling Constants: A Theoretical Study of Hg.C and Pt.P Couplings*, Journal of the American Chemical Society 123, 3341 (2001) (https://doi.org/10.1021/ja003481v)

Running CPL

Main input switches

With the ADF output TAPE21 present in the current working directory, the CPL code is invoked by:

```
$AMSBIN/cpl < input_file
```

where input_file contains the input for CPL. Best is supply also the TAPE10 of the ADF calculation in addition to TAPE21 (specify SAVE TAPE10 in the ADF input file).

input_file must contain at least one block-type input key in order to start the CPL run. The input key is

```
$AMSBIN/cpl << eor
  NMRCOUPLING
  END
eor
```

This represents a minimal input file for CPL. The NMRCOUPLING key hosts all optional keys that are relevant for the NSSCCs themselves. In addition to the mandatory NMRCOUPLING key, CPL recognizes the following input switches:

```
$AMSBIN/cpl << eor
  ADFFile adffile
  TAPE10File tape10file
  ..
eor
```

adffile is the path to an adf.rkf (TAPE21) file from which nmr reads data and to which nmr writes data. Default TAPE21. tape10file is the path to a TAPE10 file from which nmr reads data. Default TAPE10.

```
$AMSBIN/cpl << eor
  GGA
  ...
eor
```

See the separate section for this key, which influences the first order potential that is used.

```
$AMSBIN/cpl << eor
  RESTART /full/path/to/restart_file
  ...
eor
```

restart the computation from file restart_file (full pathname should be given). This is the TAPE13 produced during a CPL run. By default, TAPE13 is deleted after a successful completion of CPL. As with ADF restarts, you can not use the name TAPE13 for restart_file but you have to rename it, e.g. to tape13.restart.

```
$AMSBIN/cpl << eor
  SAVE TAPE13
  ...
eor
```

keep the restart file even after a successful completion of CPL. TAPE13 is currently the only file that is meaningful as a parameter to SAVE.

NMRCOUPLING subkeys

The available switches within a NMRCOUPLING/END block control the computation of the NSSCCs. By default, the program will evaluate the FC coupling contribution for the first nucleus being the perturbing nucleus and all remaining nuclei responding.

Please **note** that the ordering of atoms in CPL is generally different from the ADF input. The ordering of atoms is the one being stored in TAPE21 and it is grouped by fragment types. In case you are in doubt about the ordering of atoms, you can run CPL for a few seconds. It will print a list of atoms with their coordinates. The ordering is currently the same as required the NMR program in the ADF program system. On the other hand, note that for the subkeys ATOMPERT and ATOMRESP the number of the atoms refer to the input ordering in the ADF calculation. Curly brackets are not part of a key.

Available subkeys are:

```
$AMSBIN/cpl << eor
  NMRCOUPLING
    NUCLEI {npert nresp1 nresp2}
    ATOMPERT {npert1 npert2 npert3}
    ATOMRESP {nresp1 nresp2 nresp3}
    GAMMA {nnuc gamma}
    DSO
    PSO
    SD
    FC
    SCF {ITERATIONS=25 | NOCYCLE | CONVERGE=1e-4 }
    XALPHA
    CONTRIBUTIONS {1E19} {LMO, SFO, LMO2, SFO2}
  END
  ...
eor
```

`NUCLEI {npert nresp1 nresp2}` Use nucleus no. npert as the perturbing nucleus, and nuclei nresp1, nresp2, etc as responding nuclei. You can supply more than one NUCLEI keys, in which case CPL evaluates the first-order MOs for each perturbing nucleus that is specified and computes the NSSCCs between all specified responding nuclei. For each NUCLEI line in the input, CPL has to perform an SCF cycle. Note: for the numbers of the atoms the internal CPL numbering should be used.

`ATOMPERT {npert1 npert2 npert3}` See next subkey ATOMRESP.

`ATOMRESP {nresp1 nresp2 nresp3}` ATOMPERT: use nucleus no. npert1, npert2, etc. as the perturbing nuclei. ATOMRESP: use nucleus no. nresp1, nresp2, etc. as the responding nuclei. You can supply more than one ATOMPERT and (or) ATOMRESP key. CPL computes the NSSCCs for all pairs of combinations of perturbing atoms and responding atoms. For each perturbing atom CPL has to perform an SCF cycle, which is the expensive part in the calculation. Note: the numbers refer to the input ordering in the ADF calculation. Use the subkey NUCLEI to specify the nuclei according to the internal CPL numbers of the atoms.

`GAMMA {nnuc gamma}` Input a non-default magneto-gyric ratio of g = gamma for nucleus no. nnuc, in units of rad/(T s). Note that one should include the the typical 10^7 factor. CPL normally uses the g value of the most abundant NMR active isotope for a nucleus of a given charge by default. With the GAMMA keyword you can override this value or supply a value if CPL does not know about it. A list of g's that is used in the computation is printed in the output. You have to provide the GAMMA key for each nucleus you want to specify.

`DSO` Compute the diamagnetic orbital term for each NSSCC that is requested (not default)

`PSO` Compute the paramagnetic orbital term for each NSSCC (not default)

`SD` Compute the SD term for each NSSCC. This is only default for spin-orbit ADF runs. The output will contain the sum of the FC and SD contributions. *Please note that requesting this option results in a greatly increased computational cost in scalar or non-relativistic runs.* The option NOSD will turn the SD computation off in spin-orbit runs and has no effect otherwise.

`FC` Compute the FC contribution to the NSSCCs. This is the default option. Please note that it is currently not possible to compute the SD term without the FC term. Consult the 'practical aspects' section for instructions how to estimate the FC/SD cross term. The option NOFC will disable both the FC and SD computation.

SCF {ITERATIONS=25 | NOCYCLE | CONVERGE=1e-4 } Settings related to the SCF cycle that is carried out by CPL. Valid options are (with default values if applicable):

ITERATIONS 25 maximum number of iterations

NOCYCLE perform no cycle, equivalent to ITERATIONS 0

CONVERGE 1e-4 convergence criterion, an input of e corresponds approximately to a convergence of log(-e) digits, i.e. the results will be converged to about four significant digits by default. The measurement for the convergence is based on the sum *S* of the magnitudes of all occupied-virtual matrix elements of the induced first-order exchange potential. Note that the actual convergence criterion being used in the computation is e times *S* of the first cycle, i.e. the convergence criterion is set relative to the initial value of *S*.

XALPHA Use first-order Xalpha potential instead of VWN potential (default). This will usually decrease the accuracy for couplings involving hydrogen, and does not have a large effect for couplings between heavier nuclei (not default). The key is mainly intended to ensure compatibility with our previously published results.

CONTRIBUTIONS {1e19} {LMO, SFO, LMO2, SFO2} Print contributions from individual orbitals to the FC and OP term of the NSSCCs that are larger in magnitude than a certain threshold. The threshold refers to the reduced coupling constant K in SI units (not default). Additionally, an analysis in terms of Boys localized MOs (see User's Guide and SFOs. At present, either each key LMO, SFO, LMO2, SFO2 can be used individually, or grouped as {LMO, SFO2} or {SFO2, LMO}. If you need all analyses or different combinations, it is recommended to restart the CPL calculation from TAPE13, and to specify 0 iterations in the SCF. This way, the only additional computational cost should be the analysis itself.

The equation and an application for the analyses due to the LMO and SFO keys is described in Refs.[14][15]. The other analysis is based on the same equation as in Ref.[5]. For an NBO analysis of the spin-spin couplings, see the *section on NBO analysis* (page 263).

In order for the LMO-based analyses to work, the MO → LMO transformation matrix needs to be stored on TAPE21. In the ADF input, you can achieve this with the option "STORE" to the LOCORB key, i.e.

```
LOCORB STORE
   ... options
END
```

GGA key

```
$AMSBIN/cpl << eor
  GGA
  ...
eor
```

GGA Use first-order GGA potential instead of the first-order VWN potential. Should only be used for the PBE family of GGA exchange-correlation functionals and for the hybrid functional PBE0. See Refs.[??] for applications of calculating spin-spin couplings with PBE0. However, other hybrid functionals and Hartree-Fock can not or should not be used in combination with this key GGA. For consistency reasons of the first-order potential one should use the keyword USESPCODE in the ADF calculation. An example input for ADF for the hybrid PBE0 would then contain:

```
USESPCODE
XC
```

(continues on next page)

---

[14] J. Autschbach, C.D. Igna, T. Ziegler, *A theoretical investigation of the apparently irregular behavior of Pt-Pt spin-spin coupling constants* Journal of the American Chemical Society 125, 1028 (2003) (https://doi.org/10.1021/ja027931q)

[15] B.L. Guennic, K. Matsumoto, J. Autschbach, *On the NMR properties of platinum thallium bonded complexes: Analysis of relativistic density functional theory results*, Magnetic Resonance in Chemistry 42, S99 (2004) (https://doi.org/10.1002/mrc.1450)

[5] J. Khandogin, T. Ziegler, *A density functional study of nuclear magnetic resonance spin-spin coupling constants in transition metal systems*, Spectrochimica Acta Part A 55, 607 (1999) (https://doi.org/10.1016/S1386-1425(98)00265-0)

(continued from previous page)

```
 hybrid PBE0
End
```

Practical Aspects

Minimal input

The default settings for CPL are invoked by the simple minimal content of the input file:

```
$AMSBIN/cpl << eor
  NMRCOUPLING
  END
eor
```

This is equivalent to

```
$AMSBIN/cpl << eor
  NMRCOUPLING
   NUCLEI 1 2 3 4 5 6 7 8 ..(up to number of atoms)
   SCF CONVERGE 1e-4 ITERATIONS 25
   FC
  END
eor
```

Restarts

CPL is restartable after the computation of each the complete set of FC or FC/SD and OP matrix elements, and after their transformation to the MO basis. Further, in spin-orbit runs or in scalar- or non-relativistic computations involving the SD term, CPL is restartable after each SCF cycle. As with ADF restarts, you need to supply a proper input file for a restarted computation, and the restart file TAPE13 (which needs to be renamed). Changing the input of a calculation for a restart is not supported. In restarted runs, the program will automatically continue at the latest possible point before the execution stopped, and changing the input between restarts can cause inconsistencies that may lead to a crash.

Unless you are computing a very large molecule, the most likely need for a restart will probably occur during a computation of the FC/SD SCF cycle. We have already mentioned that this is a very time consuming part of the computation, and for this reason CPL can be restarted after each completed SCF cycle. The convergence of the results should not be affected by a restart. You can, e.g., use this in order to complete a lengthy CPL computation in case you have tight time limits in your queuing system, or after a power loss.

How to avoid the unnecessary computation of many SCF cycles

As already mentioned, once the first-order MOs with respect to the perturbation by one of the nuclear spins have been determined, the NSSCC between this and all other nuclei can be computed rather quickly. For each nucleus that participates in at least one of the coupling constants to be determined, the matrix elements of the FC, SD, and OP operators have to be evaluated once (unless the computation of the respective terms is disabled).

You can use this information in order to minimize the number of nuclei for which an SCF cycle has to be performed. This can lead to a great speedup of the computation. The final result, the NSSCC between A and B, does not depend on which nucleus has been chosen as the ‘perturbing’ one, and which as the ‘responding’ one (convergence has to be good enough, though). Suppose you want to compute the NSSCCs in the water molecule, with O being nucleus no. 1. In that case,

```
NUCLEI 1 2 3
NUCLEI 2 3
```

yields the same O-H and H-H coupling constants as the input

```
NUCLEI 2 1
NUCLEI 1 3
NUCLEI 3 2
```

but with less computational effort due to the fact that only 2 instead of 3 SCF cycles will be performed. The example chosen here is trivial, but in other cases it can be worthwhile to consider different sequences of computations.

Alternatively you can use the ATOMPERT and ATOMRESP subkeys:

```
ATOMPERT 1 2
ATOMRESP 2 3
```

which will calculate the spin-spin coupling of the nuclei 1-2, 1-3, and 2-3 (skips 2-2, since the nuclei are the same), which are the same O-H and H-H couplings as before.

Note: The numbers of the nuclei for the subkeys ATOMPERT and ATOMRESP refer to the input ordering in the ADF calculation, whereas the numbers of the nuclei for the subkey NUCLEI refer to the internal CPL numbers of the atoms.

Computing individual terms in the coupling tensor

As we have mentioned before, the FC, OP and OD terms can be calculated individually, but not the SD term. In case the SD input option is given, the FC+SD contribution is evaluated instead. This is NOT equal to the sum of the individual FC and SD contributions since there is a cross term between these two. Due to computational simplicity and efficiency, CPL evaluates either the matrix elements for the FC operator, or the combined ones for FC+SD. The final result therefore contains either FC only, or FC, SD plus the cross terms. Only the latter, in addition to the OP and OD contributions, should be compared to experimental results. We will implement the computation of the individual SD term in a future version of CPL in order to assist the analysis of the CPL results.

Likewise, in a spin-orbit based relativistic computation, there exists a cross term between the spin-dependent FC and SD terms, and the OP term. In the scalar- or non-relativistic limit, this contribution is always zero. With the PSO option present, CPL computes the FC, SD and OP terms including all cross contributions. Even though the output suggests that the individual OP and FC+SD terms are printed, they contain additional cross terms if spin-orbit coupling is large. You can run CPL with the options

```
$AMSBIN/cpl << eor
  NMRCOUPLING
   NOFC
   NOSD
   PSO
  END
  ...
eor
```

in order to evaluate the individual OP contribution(s). In a second run, you can then compute just the FC+SD contributions. The differences between these two CPL runs and a third one with all three terms present yields the relativistic (FC+SD)/OP cross term.

Two-bond and more-bond couplings

CPL does not discriminate between one-bond and two-bond couplings etc. in any technical sense. Even though we[????] have validated the code mostly for one-bond NSSCCs, the coupling between any pair of nuclei in the molecule can be computed. See Ref.[?] for an example.

Principal axis system, the whole coupling tensor

CPL evaluates the complete 3x3 coupling tensor with respect to the Cartesian input coordinate system. Depending on the orientation of the molecule, and the local symmetry, the coupling tensor has in fact often only a small number of independent components. CPL evaluates the ‘principal components’ by the following procedure: the 3x3 matrix is transformed into the basis of the eigenvectors of its symmetric part. This diagonalizes the symmetric part of the coupling tensor. A set of eigenvectors (= 'principal axis system') is also printed.

Main references[?,?,?,?,?,?,?,?].

See also refs.[10,11,?,12,13].

7.9 ESR/EPR

The EPR (ESR) g-tensor, hyperfine interaction (A-tensor), nuclear quadrupole interaction (Q-tensor), and zero-field splitting (ZFS, D-tensor) can be calculated. Effects due to spin-orbit coupling can be included. All electron basis sets can be used.

The separate program EPR/NMR ($AMSBIN/epr) program is no longer documented, since most of its capabilities are implemented in newer modules. See for the documentation of the EPR/NMR module in the old ADF 2010 Properties document (http://downloads.scm.com/Doc/Doc2010/ADF/Properties/Properties.pdf), and Refs.[1,2].

7.9.1 ESR/EPR g-tensor and A-tensor

A-tensor, no spin-orbit coupling

```
$AMSBIN/ams << eor
...
Engine ADF
  ESR
  END
  SpinPolarization spinpolarization
  unrestricted Yes
  {Relativity
     level scalar
     formalism ZORA
  End}
  {NUCLEARMODEL gaussian}
  ...
EndEngine
eor
```

If spin-orbit coupling is neglected, the spin in the effective spin Hamiltonian, which is commonly used for the interpretation of ESR experiments, is the real electronic spin of the paramagnetic molecule. In the spin-unrestricted DFT calculations one then uses eigenfunctions of S_z . The A-tensor can then simply be calculated as expectation value of the corresponding operator, see Ref.[3].

---

[10] N.F. Ramsey, *Electron Coupled Interactions between Nuclear Spins in Molecules*, Physical Review 91, 303 (1953) (https://doi.org/10.1103/PhysRev.91.303)

[11] R.M. Dickson, T. Ziegler, *NMR Spin-Spin Coupling Constants from Density Functional Theory with Slater-Type Basis Functions*, Journal of Physical Chemistry 100, 5286 (1996) (https://doi.org/10.1021/jp951930l)

[12] D.L. Bryce, R. Wasylishen, *Indirect Nuclear Spin-Spin Coupling Tensors in Diatomic Molecules: A Comparison of Results Obtained by Experiment and First Principles Calculations*, Journal of the American Chemical Society 122, 3197 (2000) (https://doi.org/10.1021/ja9942134)

[13] G. Schreckenbach, S.K. Wolff, T. Ziegler, *Covering the Entire Periodic Table: Relativistic Density Functional Calculations of NMR Chemical Shifts in Diamagnetic Actinide Compounds*, in Modeling NMR chemical shifts, ACS Symposium Series, Vol 732, J.C. Facelli, A.C. de Dios, Editors (American Chemical Society, Washington DC, 1999), Chapter 7 (https://doi.org/10.1021/bk-1999-0732.ch007)

[1] G. Schreckenbach and T. Ziegler, *Calculation of the G-tensor of electron paramagnetic resonance spectroscopy using Gauge-Including Atomic Orbitals and Density Functional Theory*, Journal of Physical Chemistry A 101, 3388 (1997) (https://doi.org/10.1021/jp963060t)

[2] S. Patchkovskii and T. Ziegler, *Calculation of the EPR g-Tensors of High-Spin Radicals with Density Functional Theory*, Journal of Physical Chemistry A 105, 5490 (2001) (https://doi.org/10.1021/jp010457a)

[3] E. van Lenthe, A. van der Avoird and P.E.S. Wormer, *Density functional calculations of molecular hyperfine interactions in the zero order regular approximation for relativistic effects*, Journal of Chemical Physics 108, 4783 (1998) (https://doi.org/10.1063/1.475889)

The A-tensor will be calculated for all nuclei. Terms due to the spin-polarization density at the nucleus are included in the evaluation of the A-tensor. For an accurate evaluation of the spin-polarization density at the nucleus it is important to use an all-electron basis set for the nuclei that one is interested in, avoiding the frozen core approximation. For heavy elements the incorporation of a Gaussian finite nucleus model can be important. However, one should have really large basis sets with tight basis functions to observe this effect in calculations. One possibility is to use the $AMSHOME/atomicdata/ADF/ZORA/QZ4P basis set, although even this large basis set is not large enough sometimes. The basis sets in the directories ZORA/TZ2P-J, ZORA/Z4P-J, and ZORA/jcpl are suitable for finite nucleus calculations.

In case one uses a finite nuclear model for the charge distribution, starting from ADF2013 ADF also uses a finite distribution of the nuclear magnetic dipole moment for the calculation of the A-tensor.

A-tensor, perturbative inclusion spin-orbit coupling

```
$AMSBIN/ams << eor
...
Engine ADF
  SpinPolarization spinpolarization
  unrestricted Yes
  Relativity
    Level Scalar
    Formalism ZORA
  End
  Symmetry NOSYM
  ...
EndEngine
eor

$AMSBIN/cpl << eor
adffile ams.results/adf.rkf
hyperfine
 atoms 1 2 :: calculates A-tensor for atom 1 and 2, input order
 SCF Converge=1e-7 {Iterations=25}
end
...
eor
```

The calculation of A-tensors is implemented in the CPL program as a second derivative property (spin-orbit coupling and nuclear magnetic field as perturbation) within the two-component relativistic zeroth-order regular approximation (ZORA), see Ref.[4]. This implementation allows for hybrid (only PBE0) DFT calculations, but not metaGGA's and not metahybrids.

Note that the CPKS convergence in CPL has to be set tightly (1e-7 or 1e-8) to get converged PSOSO terms for the A-tensor. For hyperfine calculations the default value is 1e-7.

g-tensor, perturbative inclusion spin-orbit coupling

```
$AMSBIN/ams << eor
...
Engine ADF
  SpinPolarization spinpolarization
  unrestricted Yes
  Relativity
    level scalar
    formalism ZORA
  End
```

(continues on next page)

---

[4] J. Autschbach, S. Patchkovskii, and B. Pritchard, *Calculation of Hyperfine Tensors and Paramagnetic NMR Shifts Using the Relativistic Zeroth-Order Regular Approximation and Density Functional Theory*, Journal of Chemical Theory and Computation 7, 2175 (2011) (https://doi.org/10.1021/ct200143w)

(continued from previous page)

```
  Symmetry NOSYM
  ...
EndEngine
eor

$AMSBIN/nmr << eor
adffile ams.results/adf.rkf
nmr
 gfactors
 u1k best
 calc all
 out iso tens
end
end input
eor
```

The calculation of g-tensors is implemented in the NMR program as a second derivative property (spin-orbit coupling and external magnetic field as perturbation) within the two-component relativistic zeroth-order regular approximation (ZORA), see Ref.[5]. This implementation allows for hybrid (B3LYP, PBE0, etc) DFT calculations, but not metaGGA's and not metahybrids. This implementation requires the use of all electron basis sets.

For an older implementation of this method, see the EPR/NMR module documentation in the old ADF 2010 Properties document (http://downloads.scm.com/Doc/Doc2010/ADF/Properties/Properties.pdf), and Refs.??[8][9].

g-tensor and A-tensor, self consistent spin-orbit coupling

```
$AMSBIN/ams << eor
...
Engine ADF
  ESR
  END
  unrestricted Yes
  Relativity
    Level Spin-Orbit
    Formalism ZORA
    SpinOrbitMagnetization Collinear
  End
  Symmetry NOSYM
eor
```

In a spin-orbit coupled spin unrestricted relativistic ZORA calculation and the ESR block key, the g-tensor and the nuclear magnetic dipole hyperfine interaction (A-tensor) will be calculated, see also Refs.?[10]. In such a calculation degenerate perturbation theory is used with the external magnetic field or nuclear magnetic field as perturbation. The calculation must use the collinear approximation, and symmetry must be NOSYM. This implementation does allow for metaGGA, and (meta-)hybrid DFT calculations, but then GIAO's are not used. There may be more than one unpaired electron. Terms due to the spin-polarization density at the nucleus are included in the evaluation of the A-tensor. However, one can not set the number of unpaired electrons, the 'spinpolarization' argument of the key CHARGE will be ignored.

Note: in a spin-orbit coupled spin restricted relativistic ZORA calculation and the ESR block key, ADF will also calculate and print the nuclear magnetic dipole hyperfine interaction, but the terms due to the spin-polarization density at the nucleus

---

[5] J. Autschbach and B. Pritchard, *Calculation of molecular g-tensors using the zeroth-order regular approximation and density functional theory: expectation value versus linear response approaches*, Theoretical Chemistry Accounts 129, 453 (2011) (https://doi.org/10.1007/s00214-010-0880-x)

[8] S. Patchkovskii and G. Schreckenbach in Calculation of NMR and EPR parameters, ISBN13: 9783527307791, M. Kaupp, M. Bühl, V.G. Malkin, Editors, (Wiley, Weinheim, 2004).

[9] S. Patchkovskii, R.S. Strong, C.J. Pickard and S. Un, *Gauge invariance of the spin-other-orbit contribution to the g-tensors of electron paramagnetic resonance*, Journal of Chemical Physics 122, 214101 (2005) (https://doi.org/10.1063/1.1917840)

[10] E. van Lenthe, A. van der Avoird and P.E.S. Wormer, *Density functional calculations of molecular g-tensors in the zero order regular approximation for relativistic effects*, Journal of Chemical Physics 107, 2488 (1997) (https://doi.org/10.1063/1.474590)

are absent. Furthermore, if there is more than one unpaired electron, the computed results will simply be incorrect, without any warning from the program. On the other hand, in case of one unpaired electron, and very large effects of spin-orbit coupling, the spin-restricted calculation may be of interest, since it uses Kramer's symmetry exact.

7.9.2 ESR/EPR Q-tensor

For the calculation of the ESR Q-tensor see the key *QTENS* (page 226).

7.9.3 ESR/EPR Zero-field splitting (D-tensor)

With the keyword ZFS the zero-field splitting (ZFS) of the ground state can be calculated.

```
ZFS
{HARTREEFOCK}
```

Zero-field splitting is the breaking of degeneracies of the ground state that is not described by a standard non-relativistic Hamiltonian. ZFS as calculated by ADF is that exhibited by molecules whose ground state has spin S>1/2 and no spatial degeneracy. This type of ZFS has two contributions, second-order spin-orbit coupling and direct electron spin-spin coupling. The calculation of ZFS with DFT is described in Ref.[6][7][11][12]. With the keyword ZFS the spin-orbit coupling term is included. If one also wants to calculate the direct electron spin-spin term, one has to include the key HARTREEFOCK, which calculates the Coulomb and (Hartree-Fock like) exchange contributions to the direct electron spin-spin term. In the spin-orbit coupling term no Hartree-Fock like exchange contributions are included.

```
HARTREEFOCK
```

ZFS can be calculated in combination with LDA and GGAs but not hybrid or meta-GGA functionals. In order to calculate ZFS, The `Relativity -> level` should be set to `Scalar` and `Relativity -> Formalism` to `ZORA` (these are the default values).

Just the simple keyword ZFS is needed in order to calculate zero-field splitting. Several optional additional keywords can also be included. The complete list is:

```
ZFS {PEDERSON|NEESE} {ANALYSIS|FULLANALYSIS}
```

`PEDERSON|NEESE` PEDERSON: The available approaches for calculating ZFS with DFT each differ subtly from the others. We believe that the method proposed by van Wüllen and coworkers[?][?] is the most theoretically complete but it may be that for certain systems the other approaches are more accurate. The van Wüllen formulation is the default but if the PEDERSON keyword in included then the equation proposed by Pederson and Khanna[?] is used. NEESE: If the NEESE keyword is included then the equation for ZFS proposed by Neese[?] is used.

`ANALYSIS|FULLANALYSIS` ANALYSIS: Neese has presented some interesting analyses of ZFS[?]. If the ANALYSIS keyword is invoked then the contributions to the ZFS is divided into terms from alpha-beta, alpha-alpha, beta-beta and beta-alpha one-electron excitations. FULLANALYSIS: The output requested by the ANALYSIS keyword is further extended to analyze each of the alpha-beta, alpha-alpha, beta-beta and beta-alpha contributions in terms of the individual one-electron excitations.

---

[6] M.R. Pederson, S.N. Khanna, *Magnetic anisotropy barrier for spin tunneling in* Mn_{12} O_{12} *molecules*, Physical Review B 60, 9566 (1999) (https://doi.org/10.1103/PhysRevB.60.9566)

[7] F. Neese, *Calculation of the zero-field splitting tensor on the basis of hybrid density functional and Hartree-Fock theory*, Journal of Chemical Physics 127, 164112 (2007) (https://doi.org/10.1063/1.2772857)

[11] C. van Wüllen, *Magnetic anisotropy from density functional calculations. Comparison of different approaches:* Mn_{12} O_{12} *acetate as a test case*, Journal of Chemical Physics 130, 194109 (2009) (https://doi.org/10.1063/1.3134430)

[12] S. Schmitt, P. Jost, C. van Wüllen, *Zero-field splittings from density functional calculations: Analysis and improvement of known methods*, Journal of Chemical Physics 134, 194113 (2011) (https://doi.org/10.1063/1.3590362)

7.10 Nuclear Quadrupole Interaction (EFG)

```
QTENS
```

This key activates the computation of the Nuclear Electric Quadrupole Hyperfine interaction. It can be applied to open-shell and to closed-shell systems. QTENS gives you the Nuclear Electric Quadrupole Hyperfine interaction (Q-tensor)[1]. The latter is directly related to the Electric Field Gradient (EFG) . The Q-tensor elements (in MHz) equal the the electric field gradient tensor elements (in a.u.) times 234.9647 times the nuclear quadrupole moment (NQM in barn units, 1 barn = 10-28m^2 = 10-24cm^2) and divided by 2I(2I-1), where I is the nuclear spin. The Nuclear Quadrupole Coupling Constant (NQCC) (in MHz) is the largest value of the principal values of the EFG (in a.u.) times 234.9647 times the nuclear quadrupole moment (in barn units). The electric field gradient tensor is printed next to the Q-tensor.

In the case of ZORA the program will calculate the EFG in the so called ZORA-4 approximation, which includes a small component density ("picture-change correction"), see[?]. If you also want to calculate the EFG using the ZORA density, thus in the Foldy-Wouthuysen picture, include the keywords:

```
PRINT FWPICTURE
```

In case QTENS is used for ^{57}Fe, ^{119}Sn, ^{125}Te, ^{193}Ir, and ^{197}Au, quadrupole splittings are written in units of mm/s, used in Mössbauer spectroscopy.

Analysis of the EFG

With the EFG keyword in AOResponse a Mulliken type analysis of the EFG principal components, and an analysis in terms of canonical MOs, can be performed. Required is symmetry NOSYM. This not implemented in case of spin-orbit coupling. For an NBO analysis of the EFG, see the *section on NBO analysis* (page 263). For an explanation of the output and a general usage tutorial, see[2]. Further references and recommended citations, see[3].

```
Symmetry NOSYM
Aoresponse
 efg NUC
end
```

`efg NUC` Here NUC is the number of the nucleus at which the EFG is to be computed (ADF internal atom ordering). Availiable for one nucleus at the time.

7.11 Mössbauer spectroscopy

Isomer shifts

By default the electron density at the nuclei is calculated, no input key is required. In the implementation in ADF, the electron density is not calculated exactly at the center of the nucleus, however, at points on a small sphere around the center of a nucleus. The printed electron density in the output of ADF is the average electron density on these points. The radius of the sphere is an approximated finite nuclear radius. The electron density at the nuclei could be used for the interpretation of isomer shifts in Mössbauer spectroscopy. Typically one needs to perform a fit of the experimentally measured isomer shifts and calculated electron densities, like, for example, is done in Ref.[1].

---

[1] E. van Lenthe and E.J. Baerends, *Density functional calculations of nuclear quadrupole coupling constants in the zero-order regular approximation for relativistic effects*, Journal of Chemical Physics 112, 8279 (2000) (https://doi.org/10.1063/1.481433)

[2] J. Autschbach, S. Zheng, and R.W. Schurko, *Analysis of Electric Field Gradient Tensors at Quadrupolar Nuclei in Common Structural Motifs*, Concepts in Magnetic Resonance Part A 36A, 84 (2010) (https://doi.org/10.1002/cmr.a.20155)

[3] A.J. Rossini, R.W. Mills, G.A. Briscoe, E.L. Norton, S.J. Geier, I. Hung, S. Zheng, J. Autschbach, and R.W. Schurko, *Solid-State Chlorine NMR of Group IV Transition Metal Organometallic Complexes*, Journal of the American Chemical Society 131, 3317 (2009) (https://doi.org/10.1021/ja808390a)

[1] W.-G. Han, T. Liu, T. Lovell and L. Noodleman, *DFT calculations of* ^{57}Fe *Mössbauer isomer shifts and quadrupole splittings for iron complexes in polar dielectric media: Applications to methane monooxygenase and ribonucleotide reductase*, Journal of Computational Chemistry 27, 1292 (2006) (https://doi.org/10.1002/jcc.20402)

One should use all electron basis sets for the Mössbauer active elements. Important is to use the same basis set, same exchange correlation functional, same integration accuracy, and same nuclear model (see key NUCLEARMODEL), if electron densities at nuclei in different molecules are compared. Note that the absolute electron density at a nucleus heavily depends on the accuracy of the basis set in the core region of this nucleus, especially if relativistic effects are included.

In case of ZORA the electron density at the nuclei is calculated with the ZORA-4 method, which includes a small component density. Previously it was using the ZORA density, which does not include a small component density. If you also want to calculate this ZORA density at the nuclei, include the keywords:

```
PRINT FWPICTURE
```

Instead of SAPA (the Sum of neutral Atomical potential Approximation) MAPA is used by default for ZORA in ADF2017. The MAPA (the Minimum of neutral Atomical potential Approximation) at a point is the minimum of the neutral Atomical potentials at that point. Advantage of MAPA over SAPA is that the gauge dependence of ZORA is reduced. The ZORA gauge dependency is small for almost all properties, except for the electron density very close to a heavy nucleus.

A ZORA calculation gives a too large density at the nucleus compared to a relativistic Dirac calculation, mostly due to the 1s-orbital. If one performs a fit of the experimentally measured isomer shifts against calculated electron densities the fact that the ZORA electron density is too large is not so important. If an absolute value of the electron density at the nucleus is important one may use the relativistic X2C method, see key *RELATIVITY* (page 97). Note that one then needs large basis sets with tight STO 1s-functions to get accurate results (for example the basis sets in the TZ2P-J or QZ4P-J directory).

Quadrupole splittings

For the calculation of Mössbauer quadrupole splittings see the key *QTENS* (page 226).

Nuclear resonance vibrational spectroscopy (NRVS)

The nuclear resonance vibrational spectroscopy (NRVS) experiment can be thought of as Mösbauer spectroscopy with vibrational sidebands. The NRVS experiment provides the complete set of bands corresponding to modes that involve motion of the Mössbauer active atoms. In order to calculate this with the ADF program a partial vibrational Density-Of-States (PVDOS) has been implemented. A PVDOS factor for a given atom is the ratio of this atom nucleus kinetic (vibrational) energy to the total vibrational energy of all nuclei, for a given mode. PVDOS factors for every atom and every mode are written to adf.rkf if *IR Frequencies* (page 151) are calculated. To visualize the calculated PVDOS use the AMSspectrum program: select the PVDOS spectrum type. Next select one or more atoms to get the PVDOS spectrum generated by the selected atoms. This is useful for analysis of NRVS spectra in bioinorganic chemistry for NRVS-active nuclei.

7.12 GW

7.12.1 General

Warning: In AMS2023.102 (or bug fixed AMS2023 binary with r112249 or higher) a bug was fixed in ADF regarding GW. An incorrect Fermi energy was used. Especially important fix in case the Fermi energy is not close to zero.

This page describes the basic procedure, usage and scope of a GW calculation. Technical details of the algorithm can be tweaked in the *MBPT input block* (page 331). In AMS2020 G_0W_0 is implemented. In AMS2021 evGW and $evGW_0$ are implemented. In AMS2022 qsGW, $qsGW_0$, and G3W2 are implemented. In AMS2023 GW (G_0W_0, evGW, $evGW_0$, qsGW, $qsGW_0$, and G3W2) with spin-orbit coupling is implemented.

See also:

Tutorial: Accurate Ionization Potential and Electron Affinity with GW

Examples GW (page 912)

The GW method is a relatively accurate method to obtain information about so-called charged excitations, or single-particle excitations. We refer to them as Quasiparticle energies. These are especially important to interpret and predict the outcome of direct and inverse photo-emission spectroscopy and can be used to obtain e.g. very accurate ionization potentials and electron affinities which gives access to the so-called fundamental gap (not to be confused with the optical gap).

GW can be used with scalar and spin-orbit coupled relativistic effects within the `ZORA`, `X2C`, or `RA-X2C` formalism.

In practice, fully self-consistent GW is rarely used for molecular systems. Instead, perturbative approximations, so called quasiparticle GW methods are used since they are cheaper and also more accurate than fully self-consistent GW. The most popular approach is G_0W_0[1], in which quasiparticle energies are obtained as a one-shot perturbative correction to KS eigenvalues. A downside of this approach is the rather pronounced starting-point dependence. This can be overcome to a large extent in eigenvalue-only self-consistent GW (evGW), in which the quasiparticle energies (but not the density) are iteratively updated until self-consistency is reached. In quasiparticle self-consistent GW, also the density is updated in each iteration, so that the results become completely starting point independent. The partially self-consistent GW variants usually converge within 6-12 iterations which make these approaches a factor of 6 to 12 more expensive than G_0W_0.

A GW calculation as implemented in ADF proceeds in five steps as has been described in detail in a series of papers[?][5][6]. The technique is also known as the as GW space-time method. Our implementation is closely related to the scheme outlined in[4]. A GW implementation in ADF for closed shell molecules that includes spin-orbit coupling is described in Ref.[11].

- A DFT single-point calculation is performed. This can be an LDA, GGA or hybrid calculation. At the moment, ADF does not support the use of XCfun, but libXC is supported. Valid choices could be LDA, PBE, BLYP, PBE0, BHandHLYP and many more. The default is LDA. As usual, the functional to be used during the SCF is requested in the *XC input block* (page 72) block. The following requests a G_0W_0 calculation with default settings using a PBE reference:

```
XC
   GGA PBE
END

GW
End
```

- From the KS orbitals and orbital energies, a Green's function (G) is evaluated and from the Green's function the so-called independent-particle polarizability is calculated. This is done in imaginary time.
- The polarizability is Fourier transformed to the imaginary frequency axis where the screened Coulomb potential (W) is evaluated using the Coulomb potential and the polarizability.
- The screened Coulomb potential is Fourier transformed to imaginary time. Here, the self-energy is calculated using G and W (that why it is called GW) which gives access to spectroscopic properties.

---

[1] Arno Förster, Lucas Visscher, *Low-order scaling G0W0 by pair atomic density fitting*, Journal of Chemical Theory and Computation 16 (12), 7381–7399 (2020) (https://doi.org/10.1021/acs.jctc.0c00693)

[5] Arno Förster, Lucas Visscher, *GW100: A Slater-Type Orbital Perspective*, Journal of Chemical Theory and Computation 17 (8), 5080–5097 (2021) (https://doi.org/10.1021/acs.jctc.1c00308)

[6] Arno Förster, Lucas Visscher, *Low-Order Scaling Quasiparticle Self-Consistent GW for Molecules*, frontiers in Chemistry 9, 736591 (2021) (https://doi.org/10.3389/fchem.2021.736591)

[4] J. Wilhelm, D. Golze, L. Talirz, J. Hutter, C.A. Pignedoli, *Toward GW Calculations on Thousands of Atoms*, Journal of Physical Chemistry Letters 9 (2), 306-312 (2012) (https://doi.org/10.1021/acs.jpclett.7b02740)

[11] A. Förster, E. van Lenthe, E. Spadetto, L. Visscher, *Two-Component GW Calculations: Cubic Scaling Implementation and Comparison of Vertex-Corrected and Partially Self-Consistent GW Variants*, Journal of Chemical Theory and Computation XXX, XXX (2023) (https://doi.org/10.1021/acs.jctc.3c00512)

- The self-energy is transformed to the molecular orbital basis from where it is Fourier transformed to the imaginary frequency axis from where it is analytically continued to the complex plane. Along the real frequency axis, the quasiparticle energies are evaluated.
- In case of evGW (evGW$_0$), the input KS eigenvalues are replaced by the quasiparticle energies from the previous iteration and the scheme is iterated until self-consistency in the quasiparticle energies is reached.
- In case of qsGW (qsGW$_0$), A non-local, hermitian, and static exchange-correlation potential is constructed from the self-energy. This exchange-correlation potential replaces the KS exchange-correlation potential. Diagonalization gives a new set of single-particle orbitals and quasiparticle energies. The procedure is repeated until self-consistency is reached.
- By default, the DIIS algorithm is used to accelerate and stabilize convergence of the self-consistent GW schemes. A linear-mixing scheme can also be used.

The GW space-time method has the distinct advantage that it can be very fast, while a full frequency, conventional GW calculation scales to the 6th power of the system size and is prohibitive for systems larger than a few tens of atoms. ADF used advanced density fitting options to accelerate the space-time method further and in practice nearly quadratic scaling can be reached. This enables the routine application of the method to systems of several hundreds of atoms. A G$_0$W$_0$ calculation (without the preceding SCF) is usually not much slower than a hybrid calculation. The downside of the approach is that the analytical continuation technique produces large errors (up to several eV) for core states which are in example important in X-ray spectroscopy. Thus, the GW implementation in ADF should only be used to predict quasiparticle energies for states in the valence-region. In fact, we have only tested it for HOMO and LUMO states (which are arguably most important)

The states of interest can be requested in the GW block

```
GW
  nStates 5
End
```

is the default and calculates 5 occupied and five unoccupied states.

```
GW
  nStates 1
End
```

calculates the HOMO and LUMO quasiparticle energy only.

7.12.2 Levels of self-consistency

Eigenvalue-only self-consistent GW

An evGW calculation is requested by

```
GW
  SelfConsistency evGW
End
```

One can also only iterate G by keep W fixed which reduces the cost of each iteration by roughly 50 %. This is requested by

```
GW
  SelfConsistency evGW0
End
```

On the other hand, much of the starting point dependence of the G_0W_0 method remains in the $evGW_0$ method and it is generally not recommended.

quasiparticle self-consistent GW

An qsGW calculation is requested by

```
GW
  SelfConsistency qsGW
End
```

One can also only iterate G by keep W fixed which reduces the cost of each iteration by roughly 50 %. This is requested by

```
GW
  SelfConsistency qsGW0
End
```

On the other hand, much of the starting point dependence of the G_0W_0 method remains in the $qsGW_0$ method and it is generally not recommended.

More options:

QPHamiltonian In quasiparticle self-consistent GW, the frequency-dependent self-energy is mapped to a static exchange-correlation in each iteration. The mapping is not unique and different schemes have been suggested. They differ in the way the frequency dependence of the self-energy is treated. In ADF, three variants can be used. KSF1 and KSF2 are from the paper by Kotani et al.[7]. KSF1 refers to eq. 10 therein and is most commonly implemented, KSF2 refers to eq. 11 therein. KSF2 is the default in ADF since it is numerically more stable. The variant denoted as Kutepov has been suggested bu Kutepov et al.[8] and uses a first-order expansion around the chemical potential.

FixedGrids Per default, the imaginary frequency and imaginary time grids used in the GW calculation are updated in each iteration in a qsGW calculation since this is necessary to make the results strictly starting point dependent. This can also be turned off, and fixed grids are used throughout. This might be helpful in case of convergence problems.

Convergence

The self-consistency can be controlled by a few parameters: For example

```
GW
  Converge HOMO=5e-3
End
```

requests that the evGW(0) calculation is considered converged if the difference between the HOMO quasiparticle energies of 2 consecutive iterations does not change by more than 5 meV. The default is 1 meV which is in practice usually a little too tight. We recommend to adjust this parameter according to your requirements, for example the experimental resolution you would like to match.

For qsGW ($qsGW_0$), the change in the norm of the density matrix is used as an additional criterion to control convergence. In evGW ($evGW_0$) it is ignored.

---

[7] A.L. Kutepov, V.S. Oudovenko, G. Kotliar, *Linearized self-consistent quasiparticle GW method: Application to semiconductors and simple metals*, Computer Physics Communications 407-414 (2017) (http://dx.doi.org/10.1016/j.cpc.2017.06.012)

[8] Takao Kotani, Mark Van Schilfgaarde, Sergey V. Faleev, *Quasiparticle self-consistent GW method: A basis for the independent-particle approximation*, Physical Review B 76 (16) 1-24 (2007) (https://doi.org/10.1103/PhysRevB.76.165106)

```
GW
  Converge Density=1e-07
End
```

is the default. For very large systems and when QZ basis sets are used, it is recommended to decrease that value, for example to

```
GW
  Converge Density=1e-08
End
```

```
GW
  linearmixing 0.2
End
```

turns of DIIS and request to use linear mixing instead with a mixing parameter of 0.2. This can be useful if for some reason, convergence using DIIS cannot be achieved. However, it is usually better to adjust the number of vectors in the DIIS expansion. This is achieved by (for example)

```
GW
  DIIS 5
End
```

The default are 10 expansion vectors. In case of difficulties converging

7.12.3 Second-order self-energy

Starting from ADF2022, it is possible to go beyond the GW approximation for the self-energy (expansion in screened interaction to first order) and also take into account the next term (expansion in the screened interaction to second order). The algorithm has been described in this paper. ..[10]

- This can be used with all variants of self-consistency.
- The second-order self-energy is always applied as a perturbative correction of the GW quasiparticle energies, using a statically screened interaction W. More precisely,

$$\Sigma^{GW} = G(\omega) * W(\omega)$$

$$\Sigma^{GW+G3W2} = G(\omega) * W(\omega) + G(\omega) * W(\omega = 0) * G(\omega) * G(\omega) * W(\omega = 0)$$

- The second-order self-energy is activated by typing

```
GW
  selfenergy G3W2
End
```

- It has been shown that the second-order correction is especially accurate for electron affinities and HOMO-LUMO gaps.[?] Since it effectively scales as the fourth power of the system size, it should not be used for systems much larger than 50 atoms.

---

[10] A. Förster, L. Visscher, *EGW100: A Slater-Type Orbital Perspective*, Journal of Chemical Theory and Computation 17(8), 5080-5097 (2021) (https://doi.org/10.1021/acs.jctc.1c00308)

7.12.4 Embedding

The GW method can be combined with both the COSMO and DIM embedding methods. This coupling is only availabe for the G_0W_0 level of approximation and is highly experimental, therefore users should excercise caution in interpreting the results. The implementation is based on the work by Duchemin et al.[12] on the combination of GW with the polarizable continuum model, which we further extended to DIM method.

To add embedding effects in a GW calculation, simply include the corresponding keywords in the input file as is done for regular DFT calculations.

7.12.5 Recommendations

Basis sets

The recommended numerical settings depend strongly on the basis set. The recommended basis set depends on system size and the property of interest. The following are recommendations which should be seen as guidelines and not as definite. We recommend to always verify the basis set convergence of for the property of interest. Details about basis set convergence can be found in this paper[?]

Basis sets

All-electron: All-electron calculations are always recommended.
Basis set size: Larger basis sets are needed to achieve the same accuracy as in a DFT calculation
Augmented basis sets: Should only be used if absolutely necessary. They are however often necessary for an accurate description of the electron affinity for molecules with LUMO above the vacuum level.

Ionization Potentials: TZ2P or larger
Electron affinities, bound LUMO: Corr/TZ3P or larger
Electron affinities, unbound LUMO: AUG/ATZ2P or larger
Fundamental Gaps: Triple-zeta quality basis sets are typically sufficient.

The GW-BSE excitation energies and the screened Coulomb interaction do only depend on the QP gaps and not on their absolute values. Therefore, triple-zeta quality basis sets are typically sufficient.

For **highly accurate results** we recommend to perform an extrapolation to the complete basis set limit: For this, perform 2 calculations using the Corr/TZ3P and Corr/QZ6P. The basis set limit is then calculated according to

$$\epsilon_n^{CBS} = \epsilon_n^{QZ} - \frac{1}{N_{bas}^{QZ}} \frac{\epsilon_n^{QZ} - \epsilon_n^{TZ}}{\frac{1}{N_{bas}^{QZ}} - \frac{1}{N_{bas}^{TZ}}}$$

The values for N_{bas}^{TZ} and N_{bas}^{QZ} can be found in the adf.rkf file in the `GW` section under the entry `nBas`. Additional explanations can be found in the references[3] and[?]

---

[12] I.Duchemin, D. Jacquemin, X. Blase, *Combining the GW formalism with the polarizable continuum model: A state-specific non-equilibrium approach*, The Journal of Chemical Physics 144, 164106 (2016) (http://dx.doi.org/10.1063/1.4946778)

[3] M.J. Van Setten, F. Caruso S. Sharifzadeh X. Ren M. Scheffler F. Liu J. Lischner L. Lin J. Deslippe S.G. Louie C. Yang F. Weigend J.B. Neaton F. Evers P. Rinke, *GW100: Benchmarking G0W0 for Molecular Systems*, Journal of Chemical Theory and Computation 11 (12), 5665-5687 (2015) (https://doi.org/10.1021/acs.jctc.5b00453)

Numerical aspects

- According to the choice of basis set, the recommended numerical settings can differ. For a discussion, see the *MBPT page* (page 331).
- qsGW has the highest requirements on the numerical parameters than evGW and G_0W_0. Usually, it is necessary to use a larger value in the `Dependency` key, see the *MBPT page* (page 331). Note that starting from AMS2022 ADF will use a `Dependency bas=5e-3` and `RIHartreeFock DependencyThreshold=5e-3` in case of (any variant of) GW. One can override these values in the input.
- The implemented GW algorithm is very sensitive to numerical noise and depending on the numerical settings and/or the underlying exchange-correlation functional, (occupied) quasiparticle energies from partially self-consistent GW calculations performed on different hardware can differ by a few meV. The discrepancies are generally more pronounced for core states for which the analytical continuation technique is rather inaccurate.
- In our experience, these differences only occur with large basis sets (QZ4P or larger) and when Minnesota functionals (we tested M06, M06-2X and M06-HF) are used to calculate the KS reference. When the DIIS algorithm is used to converge the quasiparticle energies, the number of iterations needed for convergence can then differ as well.
- For optimal accuracy, all electron basis sets hould be used. Note, that frozen cores do not lead to large computational savings in case of GW calculations.

Choosing the KS reference

- As for ground-state properties, it is far from trivial to recommend a universal functional. However, for G_0W_0 calculations, a few general guidelines can be offered. A good discussion is found in[2]. It should be noted, that GGA functional should not be used, even though a PBE starting point is a popular choice. We rather recommend to use a hybrid functional. The relevant parameter in the choice of the hybrid is usually the fraction of exact exchange. In our experience, PBE0 with 40-50 % exact exchange is a good choice. You might also use a range-separated hybrid via LibXC, for example LRC-wPBEH.
- If in doubt, one of the partially self-consistent schemes should be used. evGW is almost starting point independent and qsGW is completely starting point independent. Therefore, for qsGW, it is in principle irrelevant what starting point is chosen. Convergence properties can in principle be affected, although the number of iterations until convergence is reached is also more or less independent of the starting point[7]. Also for evGW, starting from a hybrid functional or a range-separated functional usually results in the highest accuracy.

7.12.6 GW key

```
GW
   nStates integer
   SelfEnergy [HF | GW | G3W2 | SOSEX | GWGamma | G3W2dynamic]
   SelfConsistency [G0W0 | EVGW0 | EVGW | QSGW0 | QSGW]
   QPHamiltonian [KSF1 | KSF2 | SRG | LQSGW]
   nIterations integer_list
   Converge
      Density float_list
      HOMO float
   End
   FixedGrids Yes/No
```

(continues on next page)

---

[2] F. Bruneval, M.A.L., Marques, *Benchmarking the starting points of the GW approximation for molecules*, Journal of Chemical Theory and Computation 9 (1), 324-329 (2013) (https://dx.doi.org/10.1021/ct300835h)

(continued from previous page)

```
   preconditionQSGW Yes/No
   OffDiagonalEFermi Yes/No
   LinearMixing float_list
   AdaptiveMixing float_list
   DIIS integer
   LinearizeQPequations Yes/No
   ScissorShift Yes/No
End
```

GW

Type Block

Description Instruct ADF to perform a G0W0 calculation.

nStates

Type Integer

Default value 5

GUI name N states

Description Number of Quasiparticle States to be printed to output.

The default is 5 states which in this case means that min(5, Number of particle states) occupied and min(5, Number of hole states) hole states are printed. The whole list of states can be printed by setting this parameter to -1'

SelfEnergy

Type Multiple Choice

Default value GW

Options [HF, GW, G3W2, SOSEX, GWGamma, G3W2dynamic]

Description Controls the form of the self-energy.

GW is the default and corresponds to the standard GW calculation.

G3W2 is a GW calculation plus a perturbative second-order statically screened exchange correction (second order expansion in the self-energy). Note, that there the self-energy is always static.

SelfConsistency

Type Multiple Choice

Default value G0W0

Options [G0W0, EVGW0, EVGW, QSGW0, QSGW]

Description Sets the level of self-consistency in a GW calculation.

G0W0 calculates a one-shot, perturbative correction to the KS eigenvalues.

In evGW and evGW0, the quasi-particle energies are updated until self-consistency is reached.

evGW0 requests that the Green's function is evaluated self-consistently but not the screened interaction.

In qsGW, the density is updated as well, however, the self-energy is mapped to a static effective potential and the Dyson equation is solved by diagonalization instead of inversion. The results of a qsGW are independent of the choice of the underlying exchange-correlation functional and are usually the most accurate ones.

The same is done in qsGW0, but the screened interaction is not updated.

`QPHamiltonian`

Type Multiple Choice

Default value KSF2

Options [KSF1, KSF2, SRG, LQSGW]

Description The quasi-particle Hamiltonian can be constructed in different ways.

KSF1 refers to the original construction by Kotani, Van Schilfgaarde anf Faleev (KSF) which is also implemented in TURBOMOLE.

KSF2 refers to an alternative construction by KSF.

KSF1 is not recommended since it is numerically less stable than KSF2. The results are typically very similar.

The QP energies at which the matrix elements are evaluated can be tweaked further, see the two subsequent keys: However, KSF2 is recommended since it typically leads to QP energies with the best agreement with experiment.

Ignored when not a quasi-particle self-consistent GW calculation is performed

`nIterations`

Type Integer List

Default value [10]

GUI name Number of iterations

Description The maximum number of iterations within the (partially or fully) self-consistent GW calculation has to converge.

Ignored when Formalism is set to G0W0

`Converge`

Type Block

Description Sets convergence criteria for the GW calculation in self-consistent case

`Density`

Type Float List

Default value [1e-08, 1e-05]

Description First Criterion for self-consistency procedure to terminate.

Criterion is the trace of the density matrix. Ignored in non-selfconsistent Calculation and in eigenvalue self-consistent GW

It is possible to run a qsGW calculation with an inner SCF loop which updates the static part of the elf-energy only. This can be useful to accelerate the convergence in case linear mixing is used. It is not recommended to use linear mixing, so it is also not recommened to use that inner loop as well. The second number in this list specifies the convergence criterion for the inner SCF loop.

`HOMO`

Type Float

Default value 0.003

Unit eV

GUI name HOMO energy convergence

Description Criterion for self-consistency procedure to terminate.

The self-consistent GW calculation terminates, when the difference between the HOMO QP energies between 2 consecutive iterations is below this number.

The LUMO energy converged faster than the HOMO energy so when the HOMO energy is converged according to this criterion, the LUMO energy will be converged as well.

In non-selfconsistent Calculation, this criterion is ignored.

`FixedGrids`

Type Bool

Default value No

Description In a self-consistent GW calculation, do not recalculate Grids. Can be useful in case of convergence problems. Only relevant for qsGW and qsGW0. In case of evGW and evGW0, the grids are always kept fixed.

`preconditionQSGW`

Type Bool

Default value No

Description If true, the QSGW equations are solved but prior to each diagonalization, i.e. a G0W0 calculation is performed to find the optimal QP energies at which to analytically continue the self-energy.

This is in principle a more consistent construction than KSF1 or KSF2 since the diagonal elements are consistent with G0W0.

In KSF1 and KSF2, the diagonal elements are evaluated at the QP energies from the previous iteration which is equivalent to a zeroth-order Taylor expansion of the diaognal elements around the previous QP energies.Enabling this option typically leads to slightly lower QP energies.

`OffDiagonalEFermi`

Type Bool

Default value No

Description Analytically continue the off-diagonal elements of the KSF2 qsGW Hamiltonian at the Fermi-energy instead of omega=0. Typically leads to slightly lower QP energies, i.e. higher ionization potentials. The HOMO-LUMO gaps are typically not affected.

`LinearMixing`

Type Float List

Description Requests to use linear mixing instead of DIIS and sets the mixing parameter for linear mixing of Green's function in case of self-consistency.

It is ignored in non-selfconsistent calculation and overwritten by DIIS when DIIS is also present.

`AdaptiveMixing`

Type Float List

Description Requests to use adaptive mixing instead of DIIS and sets the staring mixing parameter for mixing of Green's function in case of self-consistency.

Adapative mixing is recommenened in case a qsGW calculation does not converge with DIIS.

It is ignored in non-selfconsistent calculation and overwritten by DIIS when DIIS is also present.

`DIIS`

Type Integer

Default value 10

Description Requests to use DIIS. This is the Default. Number of expansion coefficients can be requested as well. Ignored in non-selfconsistent calculation

`LinearizeQPequations`

Type Bool

Default value No

Description Instead of solving the non-linear QP equations in a G0W0 (or evGW calculation) by bisection exacly, linearize them by first-order Taylor expansion.

This is not recommended since it does not save computational time when used together with analytical continuation (as implemented in AMS). It might however be useful for benchmarking or for validating results.

If the results os the linearization differ by a lot (for instance, more than 0.1 eV in frontier QP energies) from the non-linearized results, this might indicate that the GW calculation is not reliable.

`ScissorShift`

Type Bool

Default value No

Description Only calculate the HOMO and LUMO QP energies and shift the remaining QP energies by the same amount.

This is a rather crude approximaiton and not recommended.

It might again be useful for benchmarking purposes.

CHAPTER

EIGHT

TRANSPORT PROPERTIES

See also:

Examples: *transport properties* (page 921)

8.1 Charge transfer integrals (transport properties)

ADF can provide input parameters, such as charge transfer integrals, that are needed in approximate methods that model charge transport properties. ADF has the unique feature that it can (also) calculate such transfer integrals based on the direct method by the use of its unique fragment approach.

In theoretical models of charge transport in organic materials, see Refs.[6][7][8], the whole system is divided into fragments, in which an electron or hole is localized on a fragment, and can hop from one fragment to another. In the tight-binding approximation that is used in these models the electron or hole is approximated with a single orbital, and it is assumed that only the nearest neighboring fragments can couple. The models require accurate values of electronic couplings for charge transfer (also referred to as charge transfer integrals or hopping matrix elements) and site energies (energy of a charge when it is localized at a particular molecule) as a function of the geometric conformation of adjacent molecules. Charge transfer integrals for hole transport can be calculated from the energetic splitting of the two highest-occupied molecular orbitals (HOMO and HOMO-1) in a system consisting of two adjacent molecules, also called “energy splitting in dimer” (ESID) method. For electron transport these can be calculated from the two lowest-unoccupied orbitals (LUMO and LUMO+1) in this ESID method. ADF can also calculate transfer integrals based on the direct method by the use of its unique fragment approach. see Refs.[??]. ADF allows one to use molecular orbitals on individual molecules as a basis set in calculations on a system composed of two or more molecules. The charge transfer integrals obtained in this way differ significantly from values estimated from the energy splitting between the highest occupied molecular orbitals in a dimer. The difference is due to the nonzero spatial overlap between the molecular orbitals on adjacent molecules. Also, ADF’s methods are applicable in cases where an orbital on one molecule couples with two or more orbitals on another molecule.

---

[6] M.D. Newton, *Quantum chemical probes of electron-transfer kinetics: the nature of donor-acceptor interactions*, Chemical Reviews 91, 767 (1991) (https://doi.org/10.1021/cr00005a007).

[7] K. Senthilkumar, F.C. Grozema, F.M. Bickelhaupt, and L.D.A. Siebbeles, *Charge transport in columnar stacked triphenylenes: Effects of conformational fluctuations on charge transfer integrals and site energies*, Journal of Chemical Physics 119, 9809 (2003) (https://doi.org/10.1063/1.1615476).

[8] K. Senthilkumar, F.C. Grozema, C. Fonseca Guerra, F.M. Bickelhaupt, F.D. Lewis, Y.A. Berlin, M.A. Ratner, and L.D.A. Siebbeles, *Absolute Rates of Hole Transfer in DNA*, Journal of the American Chemical Society 127, 14894 (2005) (https://doi.org/10.1021/ja054257e)

8.1.1 Charge transfer integrals with the TRANSFERINTEGRALS key

In this method the matrix elements of the molecular Kohn-Sham Hamiltonian H^{KS} in the basis of fragment orbitals is used to calculate site energies and charge transfer integrals. Likewise the overlap integrals between fragment orbitals are calculated. No explicit electrons are removed or added in this method. For electron mobility calculations the fragment LUMO's are considered. For hole mobility calculations the fragment HOMO's are considered.

To calculate the charge transfer integrals, spatial overlap integrals and site energies, include the key TRANSFERINTEGRALS in the input for ADF. Symmetry NOSYM should be used. The molecular system typically should be build from 2 fragments. In the fragment calculation full symmetry can be used.

```
TRANSFERINTEGRALS
Symmetry NOSYM
Fragments
 frag1 frag1.results/adf.rkf
 frag2 frag2.results/adf.rkf
End
```

By default, integrals are calculated only for the HOMO (LUMO) of the fragments, and possibly HOMO-1, HOMO-2 (LUMO+1, LUMO+2) if the energy of those fragment orbitals are close to the HOMO (LUMO) of that fragment. To calculate the matrix elements and overlap integrals based on all fragment orbitals one can use the key:

```
PRINT FMATSFO
```

The method described here to calculate charge transfer integrals is more approximate than the next method that uses FDE. The major difference is how effects of a localized charge are included.

If 2 fragments are used the electronic coupling V (also known as effective (generalized) transfer integrals J_eff) for hole transfer or electron transfer is calculated as V = (J-S(e1+e2)/2)/(1-S^2). Here e1, e2, are the site energies of fragment 1 and 2, respectively. J is the charge transfer integral, and S the overlap integral.

$$\begin{aligned} e_1 &= \langle\phi_1|H^{KS}|\phi_1\rangle \\ e_2 &= \langle\phi_2|H^{KS}|\phi_2\rangle \\ J &= \langle\phi_1|H^{KS}|\phi_2\rangle \\ S &= \langle\phi_1|\phi_2\rangle \\ V &= \frac{J - S(e1+e2)/2}{1-S^2} \end{aligned}$$

In case of electron mobility calculations ϕ_1 is the LUMO of fragment 1 and ϕ_2 is the LUMO of fragment 2. In case of hole mobility calculations ϕ_1 is the HOMO of fragment 1 and ϕ_2 is the HOMO of fragment 2. The electronic coupling between the HOMO of the donor fragment and the LUMO of the acceptor fragment and vice-versa is also calculated, which represent the probability of a charge recombination process.

If there is (near) degeneracy in the fragment HOMO and/or LUMO multiple electronic couplings V_i are printed. A total electronic coupling is calculated as

$$V_{tot} = \sqrt{\sum_{deg} {V_i}^2}$$

```
TIDegeneracyThreshold float
```

`TIDegeneracyThreshold`

Type Float

Default value 0.1

Unit eV

Description If the orbital energy of the fragment MO is within this threshold with fragment HOMO or LUMO energy, then this fragment MO is included in the calculation of the transfer integrals. Relevant in case there is (near) degeneracy.

8.1.2 Charge transfer integrals with FDE

Overview

The ELECTRONTRANSFER keyblock invokes the calculation of Hamiltonian (site energies and couplings) and overlap matrix elements with FDE-derived localized states. Two FDE calculations are (not strictly) needed before running the ELECTRONTRANSFER calculation. The calculated matrix elements are theoretically similar to the ones obtained with the TRANSFERINTEGRALS keyword. A recent review casts ELECTRONTRANSFER in the state-of-the-art of methods for computing charge transfer couplings, and provides a step-by-step guide for computing such couplings with ADF[1].

Features

- Effects of orbitals relaxation due to localized charges, Refs.[9][10].
- Effects of polarization due to molecules in the environment, Ref.[2].
- ELECTRONTRANSFER is linear scaling in the number of fragments when the system is composed by more than one fragment.
- The code can tackle hole transfer, electron transfer, charge separation and charge recombination processes[3].
- Can compute transfer integrals from diabats obtained with the Constrained DFT method (i.e., invoked by using the experimental CDFT keyword of ADF). This includes calculation of couplings between diabats made of a single fragment.
- It is possible to include the effect of the environment on CDFT diabats by coupling it with an FDE calculation[4].
- The performance of ElectronTransfer in the evaluation of the hole transfer coupling, was benchmarked against wave functions methods, with an error below 7%. PBE, PW91, B3LYP and PBE0 functionals, with PW91k for the non-additive component of the kinetic energy and TZP basis function, are recommended to obtain the FDE-derived localized states[5].
- The code was parallelized in ADF2018.

Limitations

- Hybrid functionals are not yet supported
- adf.rkf (TAPE21) files with charge or spin-localized states with specific names are needed

---

[1] P. Ramos, M. Mankarious, M. Pavanello, *A critical look at methods for calculating charge transfer couplings fast and accurately*, in Practical Aspects of Computational Chemistry IV, Jerzy Leszczynski and Manoj Shukla (eds.), 2016, Springer (https://doi.org/10.1007/978-1-4899-7699-4_4)

[9] M. Pavanello and J. Neugebauer, *Modelling charge transfer reactions with the frozen density embedding formalism*, Journal of Chemical Physics 135, 234103 (2011) (https://doi.org/10.1063/1.3666005)

[10] M. Pavanello, T. Van Voorhis, L. Visscher, and J. Neugebauer, *An accurate and linear-scaling method for calculating charge-transfer excitation energies and diabatic couplings*, Journal of Chemical Physics 138, 054101 (2013) (https://doi.org/10.1063/1.4789418)

[2] M. Pavanello, T. Van Voorhis, L. Visscher, and J. Neugebauer, *An accurate and linear-scaling method for calculating charge-transfer excitation energies and diabatic couplings*, Journal of Chemical Physics 138, 054101 (2013) (https://doi.org/10.1063/1.4789418)

[3] A. Solovyeva, M. Pavanello, J. Neugebauer, *Describing Long-Range Charge-Separation Processes with Subsystem Density-Functional Theory*, J. Chem. Phys. 140, 164103 (2014) (https://doi.org/10.1063/1.4871301)

[4] P. Ramos, M. Pavanello, *Constrained Subsystem Density Functional Theory*, Physical Chemistry Chemical Physics 18, 21172 (2016) (https://doi.org/10.1039/C6CP00528D).

[5] P. Ramos, M. Papadakis, M. Pavanello, *Performance of Frozen Density Embedding for Modeling Hole Transfer Reactions*, Journal of Physical Chemistry B 119, 7541 (2015) (https://doi.org/10.1021/jp511275e)

These two limitations do not apply to the method with the TRANSFERINTEGRALS key, but the TRANSFERINTEGRALS has many other limitations. Like the method with the TRANSFERINTEGRALS key fragments are needed.

ELECTRONTRANSFER input

The minimum input for the ELECTRONTRANSFER key is:

```
FRAGMENTS
 frag1 FragFile1
 ...
 fragN FragFileN
END
ELECTRONTRANSFER
 NumFrag N
END
```

where frag1 … fragN are the labels of the fragments in the calculation, and FragFile1 … FragFileN are the adf.rkf (TAPE21) files of spin RESTRICTED calculations of the isolated fragments, N is the total number of fragments employed in the calculation.

Files and file names

The fragment files to be used in the ELECTRONTRANSFER calculation are generally different from the adf.rkf (TAPE21) files used in the FRAGMENTS key block. Two types of fragment adf.rkf (TAPE21) files are needed by the calculation:

- The isolated closed-shell TAPE files for the FRAGMENTS keyblock
- The adf.rkf (TAPE21) files of the charge or spin localized states (which can be obtained with an FDE calculation as done in the example below)

There are 2 charge localized states. They are labeled with A and B. The respective adf.rkf (TAPE21) files must be names as follows:

```
fragA1.t21, fragA2.t21, ... , fragAN.t21 (for state A)
fragB1.t21, fragB2.t21, ... , fragBN.t21 (for state B)
```

The above files should be copied to the working folder of the ADF calculation prior to executing ADF.

Options

```
ELECTRONTRANSFER
 NumFrag N
 {Joint|Disjoint}
 {Debug}
 {Print EIGS|SAB}
 {FDE}
 {INVTHR threshold}
 {CDFT}
 {KNADD}
END
```

`Joint|Disjoint` The default is “Joint”. Joint is always recommended. The “Disjoint” formalism is described in Ref.[?] and is much faster than the “Joint” formalism when more than 2 fragments are considered. Joint and Disjoint are equivalent for systems composed of only 2 fragments. Disjoint should only be used if the fragment files are obtained in an FDE calculation (see FDE below).

`Debug` The code performs additional checks (determinants, diagonalizations, inversions, traces, etc.). Substantial increase in the output should be expected.

Print If EIGS, it will print the (unformatted) matrix of the MO coefficient in the AO representation. If SAB, it will print the (unformatted) matrix of the diagonal and transition overlap matrix in the MO representation.

FDE An FDE calculation including more than 2 fragments must include the following key block:

```
ELECTRONTRANSFER
  FDE
END
```

and the numerical integration precision in the *last* FDE calculation for every subsystem should be set to no less than:

```
BeckeGrid
   Quality Good
End
```

if in the subsequent ELECTRONTRANSFER calculation the DISJOINT subkey is used.

Invthr threshold Default 1.0e-3, is a threshold for the Penrose inversion of the transition overlap matrix. If warnings about density fitting are printed, invthr may be increased up to 1.0e-2. Larger invthr might affect the quality of the calculated couplings and excitation energies.

CDFT If disjoint is selected, this keyword must be selected if the evaluation of the electronic coupling is sought between diabats located on the same CDFT fragment. In this case, the CDFT fragment has to be always first under the ATOMS keyword.

KNADD If disjoint is selected, this allows the kinetic energy of each fragment to be obtained locally without using the full grid. This keyword is recommended when there are many tens of subsystems, such as systems with several solvent molecules.

Output

The output of the example in $AMSHOME/examples/adf/ElectronTransfer_FDE_H2O is discussed here. This example involves the calculation of electronic coupling, site energies and charge-transfer excitation energy for the hole transfer in a water dimer.

```
============  Electron Transfer RESULTS ===================

Electronic Coupling =         0.000000 eV
Electronic Coupling =        -0.003569 cm-1
H11-H22             =        -1.396546 eV
Excitation Energy   =         1.396546 eV
Overlap             =         0.000000
H11 H22 H12 =  -152.443000816341  -152.391678701092  -151.743979368040 Eh
S11 S22 S12 =     0.981795415192     0.981006454450    -0.000000023700


=========== END Electron Transfer RESULTS ================
```

Due to symmetry, the overlap is almost diagonal (Overlap = 0.00), thus the transition density is evaluated with one less electron as explained in Ref.[?].

The electronic coupling between the state with a positive charge localized on one water molecule and another with the charge localized on the other water molecule is given by “Electronic Coupling” and is reported in eV and cm^-1.

“H11-H22” is the difference of the site energies in eV. Values of the site energies are given by the first two values of “H11 H22 H12” in atomic units.

“Excitation Energy” reports the value of the transfer excitation energy as calculated by diagonalization of the 2X2 generalized eigenvalue problem in the basis of the charge-localized states, see Refs.[??].

"S11 S22 S12" are the values of the non-normalized overlaps.

8.2 GREEN: Non-self-consistent Green's function calculation

green is an auxiliary program which can be used to calculate the density of states (DOS) and transmission of molecules connected to semi-infinite contacts. The transmission is the electron transmission through a molecule connected to semi-infinite contacts. The calculation is based on the non-self-consistent Green's function method, which is an approximation to the non-equilibrium Green's function (NEGF) method. The details of this method can be found in chapter 2 and appendix C of the PhD thesis of Jos Seldenthuis (2011) (http://downloads.scm.com/Doc/Seldenthuis2011.pdf). See Ref.[1] for more details on the applicability of the wide-band limit approximation. Suggestion is to consider to use NEGF implementation in the periodic program BAND program, see the BAND manual, since it might give better results.

8.2.1 Introduction

The utility program *green* calculates the density of states (DOS) and zero-bias transmission of molecule connected to two semi-infinite contacts. A typical calculation consists of two parts. The first is the calculation of the effect of the semi-infinite contacts, contained in the so-called self-energy matrix. The second is the calculation of the desired properties of the molecule with the self-energies.

Self-energy

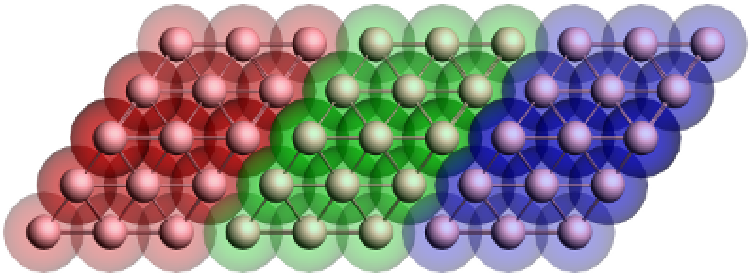


Figure 1: Geometry of the gold contact used in the calculation of the self-energy. The lead consists of two surface layers, left (red) and right (blue), and a bulk layer (green). Each principal layer in turn consists of three atomic layers. This should be sufficient to ensure that the Hamiltonian of the central (green) layer is a bulk Hamiltonian.

Since the contacts are semi-infinite, the calculation of their self-energy is effectively a bulk calculation. Since ADF only works with systems of finite size, approximations have to be made. Fig. 1 shows the typical geometry used in the calculation of a gold contact. The geometry consists of three parts, the so-called principal layers. These layers should be large enough that the atoms on one side are not influenced by whatever is attached to the other side. Three atomic layers usually suffice. The green region is the bulk layer. The red and blue regions are the surface layers. Note that the blue region corresponds to the left contact of a molecule and the red region to the right contact.

To calculate the self-energy, we first need to do a single-point calculation of a principal layer. This layer is then used as a fragment in the following calculations. Note that all ADF calculations have to be performed with SYMMETRY NOSYM. We then build up the contact geometry from three copies of the layer fragment as in Fig. 1 and perform another single-point calculation. This results in a Hamiltonian describing the three contact layers and the coupling between them.

From the adf.rkf (TAPE21) file *green* can now calculate the self-energy matrices with the SURFACE key. This has to be done once for every energy for which we want to calculate the DOS or transmission. For the left contact of the molecule, *green* needs the blue and green fragments. The self-energy is calculated by taking the (blue) surface layer and iteratively adding more (green) bulk layers until matrices converge to the semi-infinite result. The self-energy of the right contact is similarly calculated from the red and green fragments. Since the self-energy described the effect of an infinite chain

---

[1] C.J.O. Verzijl, J.S. Seldenthuis, and J.M. Thijssen, *Applicability of the wide-band limit in DFT-based molecular transport calculations*, Journal of Chemical Physics 138, 094102 (2013) (https://doi.org/10.1063/1.4793259)

of (green) bulk regions on a (red or blue) surface layer, this calculation does not depend on whatever is attached to the contacts. The self-energy matrices can therefore be reused for different molecules.

DOS and transmission

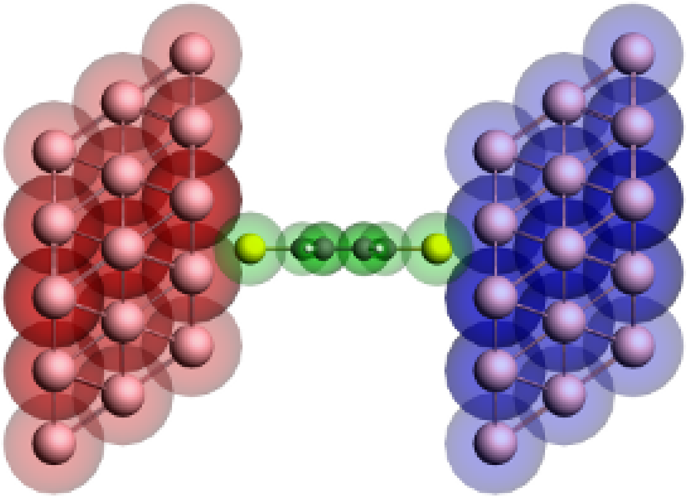


Figure 2: Geometry of the extended molecule used in the calculation of a benzenedithiol junction. The molecule is shown in green, while the left and right contact regions are shown in red and blue, respectively. Note that the red region corresponds to the blue surface layer in Figure 1 and vice versa.

Once the self-energy matrices have been calculated for the desired energies, we can compute the DOS and transmission of a molecule. However, since the self-energy matrices couple to the surface layers of the contacts, we need to include those surface layers in the calculation of the molecule (see Fig. 2). We therefore first perform a single-point calculation with ADF of the isolated molecule. The result is then used as a fragment and combined with the fragments of the surface layers to construct the so-called extended molecule. We then perform another single-point calculation of the final geometry.

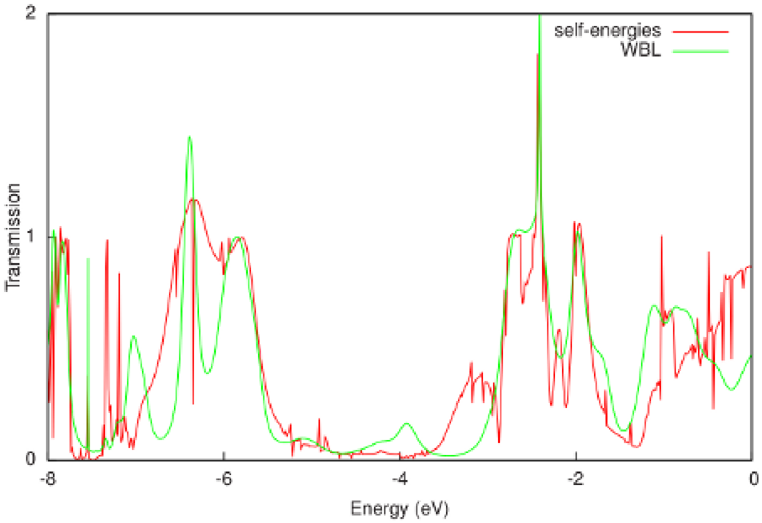
From the self-energies of the contacts and the adf.rkf file of the extended molecule, *green* can now compute the DOS and transmission. This calculation is non-self-consistent since the ADF calculations are all performed on finite instead of semi-infinite systems. This will result in certain artifacts in the DOS and transmission spectra, but those can be made arbitrarily small by choosing the principal layers large enough.

8.2.2 Wide-band-limit

In the wide-band limit (WBL) the coupling to the leads is assumed to be independent of energy. Therefore one does not need to calculate any self-energies. This also means that the eigenspace of the Green's function is independent of energy. It can therefore be diagonalized in advance, greatly speeding up the calculation of the DOS and the transmission. See Ref.[?] for more details on the applicability of the wide-band limit in DFT-based molecular transport calculations.

In the example $AMSHOME/examples/adf/green_Al/green_WBL.run of *green*, the transmission of *benzenedithiol junction* (page 979) in the wide-band limit (WBL) is calculated. In order to model the molecule-metal interface, we do need to include a few gold layers in the calculation. However, unlike *before* (page 967), only a single atomic layer as the principal layer is used. Because a single atomic layer is an unnatural configuration for gold, a minor amount of smearing is necessary to make the calculation converge. The molecule is sandwiched in between the electrodes just like before (see Fig. 2 in *the example for benzenedithiol* (page 979)). However, this time each atomic layer of gold gets its own fragment. The reason for this configuration is that if the WBL is used on the entire gold contact the result is an an unphysical coupling to the leads; even the gold atoms contacting the molecule would have a direct coupling to the environment. A much better result can be obtained by only using the WBL on the back-most atomic layer and letting the electrons propagate naturally through the rest of the contact. Because the WBL is computationally so inexpensive, we can easily calculate the DOS and transmission for 10,000 points instead of 1000.

A comparison of the resulting transmission with the calculation with self-energies is shown in the following figure:



The WBL shows good agreement with the non-WBL transmission around the Fermi energy (-0.195 Hartree or -5.306 eV). Note that the quality of the WBL depends on the choice of the coupling (ETA). For this particular contact geometry we obtain good agreement for ETA = 0.02 Hartree, but a better value may be found for other electrodes. Finally, the WBL can be incrementally improved by adding more gold layers to the extended molecule. For many layers it converges to the calculation with full self-energies.

8.2.3 Input options

The input for *green* is keyword oriented and is read from the standard input. *green* is typically first used to calculate the self-energy matrices of the left and right contacts (with the SURFACE key), and then to calculate the density of states (DOS) and transmission (with the DOS and TRANS keys, respectively), using those self-energies. The only keyword required to be present in all calculations is the EPS keyword, which specifies the energy range.

```
$AMSBIN/green << eor
 EPS mineps maxeps numeps
 {ETA eta}
 {SO sh sl {moc}}
 {SURFACE filename
     FRAGMENTS f1 f2
  END}
 {DOS filename}
 {TRANS
  LEFT filename
     FRAGMENT fragment
     ETA eta
  END
  RIGHT filename
     FRAGMENT fragment
     ETA eta
  END}
eor
```

`EPS mineps maxeps numeps` The energy range for which either the self-energy matrices or the DOS and transmission have to be calculated. The range consists of numeps (≤ 1) points running from mineps to maxeps inclusive.

`(optional) ETA eta` The imaginary energy, or the distance from the real axis, in the calculation of the Green's function. The value needs to be a small positive number to prevent singularities in the calculation. The default value (10^{-6} Hartree) is sufficient for most calculations.

(optional) SO sh sl {moc} The shifts for the scissors operator. All occupied orbitals (HOMO and below) are shifted by sh, while the unoccupied orbitals (LUMO and above) are shifted by sl. Orbitals are considered occupied if their (possibly fractional) occupation is larger than moc (default 0). The scissor operator can partially remedy the underestimation of the HOMO-LUMO gap in DFT. The sh and sl shifts generally have the same magnitude, but opposite sign (with sh usually being negative and sl positive). A good estimate for the magnitude is the sum of the ionization potential and the energy of the HOMO of the *free* molecule. This can be improved by including image charge effects. For more details, see Ref.[2]. By default, sh=sl=0.

(optional) SURFACE

```
SURFACE filename
   FRAGMENTS f1 f2
END
```

The SURFACE block key enables the calculation of the self-energy matrices. The filename specifies the adf.rkf file resulting from an ADF calculation of the contacts. This calculation has to be performed with SYMMETRY NOSYM. The FRAGMENTS key is used to specify the two principal layers between which the surface is defined. The resulting self-energy matrices (one for every energy point given by EPS) is stored in a binary KF file named SURFACE.

(optional) DOS filename The DOS key enables the calculation of the density of states. The filename specifies the adf.rkf file containing the result of an ADF calculation of the extended molecule (performed with SYMMETRY NOSYM). Two text files will be generated: DOS_A and DOS_B, containing, respectively, the DOS of the spin-A and spin-B electrons. In the case of a spin-unrestricted calculation, DOS_A and DOS_B might differ. If only the DOS of the spin-A electrons is required, the calculation can be sped up by specifying NOSAVE DOS_B. The DOS key requires the presence of the LEFT and RIGHT keys.

(optional) TRANS The TRANS key enables the calculation of the transmission. The filename specifies the adf.rkf file containing the result of an ADF calculation of the extended molecule (performed with SYMMETRY NOSYM). Two text files will be generated: TRANS_A and TRANS_B, containing, respectively, the transmission of the spin-A and spin-B electrons. In the case of a spin-unrestricted calculation, TRANS_A and TRANS_B might differ. If only the transmission of the spin-A electrons is required, the calculation can be sped up by specifying NOSAVE TRANS_B. The TRANS key requires the presence of the LEFT and RIGHT keys.

LEFT/RIGHT

```
LEFT filename
   FRAGMENT fragment
   ETA eta
END
```

The LEFT and RIGHT block keys specify the left and right self-energies used in a calculation of the DOS and transmission. If a filename is specified, the self-energy matrices are read from that file. The energy range of the self-energies has to be consistent with the range specified by the EPS keyword. The FRAGMENT key is used to denote the fragment in the extended molecule (given by the argument to the DOS or TRANS key) to which the self-energy couples. If no filename is specified, the wide-band limit is used. The ETA key can then be used to specify the magnitude of the coupling (10^{-3} Hartree by default).

---

[2] S.Y. Quek, L. Venkataraman, H.J. Choi, S.G. Louie, M.S. Hybertsen and J.B. Neaton, *mine-Gold Linked Single-Molecule Circuits: Experiment and Theory*, Nano Letters 7, 3477 (2007) (https://doi.org/10.1021/nl072058i)

8.2.4 Output

After a successful calculation of the self-energy matrices, *green* produces a binary KF file named SURFACE containing two sections. The Surface section contains the energy range:

contents of Surface	comments
mineps	start of the energy range
maxeps	end of the energy range
numeps	number of points

The Sigma section contains the real and imaginary parts of the self-energy matrices:

contents of Sigma	comments
nfo	number of fragment orbitals (dimension of the self-energy matrices)
Re(Sigma_%d)	the real part of the %d self-energy matrix (numbered from 1 up to numeps)
Im(Sigma_%d)	the imaginary part of the %d self-energy matrix (numbered from 1 up to numeps)

A successful calculation of the density of states (DOS) or transmission results in the text files DOS_A and DOS_B, and TRANS_A and TRANS_B, respectively. The suffixes _A and _B denote the different spins. The text files the DOS and transmission for every energy point and can be plotted with, for example, gnuplot.

8.2.5 GREEN with ADF-GUI

In ADF2017 the ADF-GUI can be used to setup a GREEN calculation, and look at the results. Note, however, that not all GREEN options are supported with the ADF-GUI. For example, one can not use unrestricted or wide-band-limit calculations with the ADF-GUI. GUI tutorials exist for NEGF with DFTB or BAND. Suggestion is to consider to use NEGF implementation in the periodic program BAND program, see the BAND manual, since it might give better results.

CHAPTER

NINE

ANALYSIS

See also:

ADF-GUI tutorial: all ADF tutorials, fragment analysis

Examples: *analysis* (page 986)

9.1 Molecules built from fragments

ADF analyzes the results in terms of user-specified subsystems from which the total system is built. The program tells you how the 'Fragment orbitals' (FO's) of the chemically meaningful sub-units mix with FO's on other fragments to combine to the final molecular orbitals.

ADF builds a molecule from user-defined fragments, which may be single atoms or larger moieties, for example, ligands, functional groups, or complete molecules in a donor-acceptor complex. In practice, this means that the results of the ADF calculation on a fragment are saved on a file and that the fragment files are then used in setting up the calculation on the overall system. The fragment orbitals (FOs), i.e., the MOs from the calculations on the fragments, are employed as basis functions in the new calculation. This does not imply a basis set truncation or contraction because the virtual FOs are included: the FOs constitute only a transformation of the basis set. If there are symmetry-equivalent fragments, for example, the six CO molecules in octahedral $Cr(CO)_6$, the program generates symmetry combinations of the FOs and uses the symmetrized fragment orbitals (SFOs) as basis functions. The SFOs transform as the irreducible representations (irreps) of the molecule, allowing a symmetry-driven analysis of the results. In absence of any symmetry the SFOs are identical to the FOs.

The fragment approach offers considerable advantages. It enhances the interpretative power of ADF as it leads to a more transparent picture of bonding, which reduces from a complicated mixing of many primitive basis functions (possessing little physical relevance) to a few key interactions between meaningful fragment (frontier) orbitals. The fragment approach also improves the numerical precision. In ADF, energies are calculated directly, with respect to the fragments, by one single numerical integral of the difference energy density $\epsilon[\rho, \mathbf{r}] - \sum_A \epsilon_A[\rho_A, \mathbf{r}]$ between the overall molecule and the constituting fragments.

$$\Delta E[\rho] = \int d\mathbf{r} \left(\epsilon[\rho, \mathbf{r}] - \sum_A \epsilon_A[\rho_A, \mathbf{r}] \right)$$

In other words, we evaluate the energy of the overall molecule, $E[\rho] = \int d\mathbf{r}\epsilon[\rho, \mathbf{r}]$, and the energies of each of the fragments, say the atoms that constitute the overall molecule, $E_A[\rho] = \int d\mathbf{r}\epsilon_A[\rho_A, \mathbf{r}]$, in the same numerical integration grid. This provides more accurate relative energies than subtracting total energies from separate calculations, because the same relative numerical integration error applies to a much smaller quantity, yielding, in turn, a much smaller absolute error.

Note that the user has the freedom to make his own choice of fragments. This is, however, not a matter of plain arbitrariness, and it does not make the analysis tools less meaningful. On the contrary, this freedom simply reflects the many perspectives from which a particular chemical phenomenon can be viewed.

In practice, many calculations are performed using as fragments the so-called basic atoms, which are the smallest possible building blocks in ADF. The basic atoms are not necessarily physically realistic objects - indeed, usually they are not, as they must be spin-restricted and spherically symmetric. The computed (bonding) energy w.r.t. basic atoms, then, does not yield quantities that can be compared to experimental data directly. Rather, one must correct for the true ground state of the isolated single atoms.

Text is mostly taken from: *Chemistry with ADF* G. te Velde, F.M. Bickelhaupt, E.J. Baerends, C. Fonseca Guerra, S.J.A. van Gisbergen, J.G. Snijders, T. Ziegler Journal of Computational Chemistry 22, 931 (2001) (https://doi.org/10.1002/jcc.1056).

Link: *How to make molecular fragments* (page 50)

Tutorial: ADF fragment analysis

Examples: *analysis options* (page 986)

9.2 Bond energy analysis

No special input keys are required. However, if one has open shell fragments, see key UNRESTRICTEDFRAGMENTS or key FRAGOCCUPATIONS.

ADF calculates various chemically meaningful terms that add up to the bond energy, with an adaptation of Morokuma's bond energy decomposition to the Kohn-Sham MO method. The individual terms are chemically intuitive quantities such as electrostatic energy, Pauli repulsion, and orbital interactions. The latter are symmetry decomposed according to the Ziegler transition state method. For a discussion of bonding energy decompositions and applications see e.g. ref.[1][2][3][12][13][14][15][16][17][18]. For a discussion of forming electron pair bonding between open shell molecules see Ref.[4]. For an ETS-NOCV analysis of the orbital interaction, see Ref.[5].

In ADF2012 the calculation of the Pauli repulsion for metaGGA's and metahybrids is implemented. Note that for hybrids this was already implemented before in case of closed shell fragments. In ADF2012 for hybrids, metaGGA's, and metahybrids the calculation of the Pauli repulsion is also implemented if one is simulating an unrestricted fragment with the key FRAGOCCUPATIONS.

---

[1] T. Ziegler and A. Rauk, *On the calculation of Bonding Energies by the Hartree Fock Slater method. I. The Transition State Method*, Theoretica Chimica Acta 46, 1 (1977) (https://doi.org/10.1007/BF00551648)

[2] T. Ziegler and A. Rauk, *A theoretical study of the ethylene-metal bond in complexes between copper(1+), silver(1+), gold(1+), platinum(0) or platinum(2+) and ethylene, based on the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1558 (1979) (https://doi.org/10.1021/ic50196a034)

[3] F.M. Bickelhaupt, N.M. Nibbering, E.M. van Wezenbeek and E.J. Baerends, *The Central Bond in the Three CN* Dimers NC_CN, CN-CN, and CN-NC: Electron Pair Bonding and Pauli Repulsion Effects*, Journal of Physical Chemistry 96, 4864 (1992) (https://doi.org/10.1021/j100191a027)

[12] K. Kitaura and K. Morokuma, *A new energy decomposition scheme for molecular interactions within the Hartree-Fock approximation*, International Journal of Quantum Chemistry 10, 325 (1976) (https://doi.org/10.1002/qua.560100211)

[13] T. Ziegler and A. Rauk, *Carbon monoxide, carbon monosulfide, molecular nitrogen, phosphorus trifluoride, and methyl isocyanide as sigma donors and pi acceptors. A theoretical study by the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1755 (1979) (https://doi.org/10.1021/ic50197a006)

[14] H. Fujimoto, J. Osamura and T. Minato, *Orbital interaction and chemical bonds. Exchange repulsion and rehybridization in chemical reactions*, Journal of the American Chemical Society 100, 2954 (1978) (https://doi.org/10.1021/ja00478a004)

[15] S. Wolfe, D.J. Mitchell and M.-H. Whangbo, *On the role of steric effects in the perturbational molecular orbital method of conformational analysis*, Journal of the American Chemical Society 100, 1936 (1978) (https://doi.org/10.1021/ja00474a055)

[16] A.J. Stone and R.W. Erskine, *Intermolecular self-consistent-field perturbation theory for organic reactions. I. Theory and implementation; nucleophilic attack on carbonyl compounds*, Journal of the American Chemical Society 102, 7185 (1980) (https://doi.org/10.1021/ja00544a003)

[17] F. Bernardi, A. Bottoni, A. Mangini and G. Tonachini, *Quantitative orbital analysis of ab initio SCF=MO computations : Part II. Conformational preferences in H2N—OH and H2N—SH*, Journal of Molecular Structure: THEOCHEM 86, 163 (1981) (https://doi.org/10.1016/0166-1280(81)85082-8)

[18] P.J. van den Hoek and E.J. Baerends, *Chemical bonding at metal-semiconductor interfaces*, Applied Surface Science 41/42, 236 (1989) (https://doi.org/10.1016/0169-4332(89)90063-9)

[4] F.M. Bickelhaupt, M. Solà, C. Fonseca Guerra, *Highly polar bonds and the meaning of covalency and ionicity – structure and bonding of alkali metal hydride oligomers*, Faraday Discussions 135, 451 (2007) (https://doi.org/10.1039/B606093E)

[5] M. Mitoraj, A. Michalak and T. Ziegler, *A Combined Charge and Energy Decomposition Scheme for Bond Analysis*, Journal of Chemical Theory and Computation 5, 962 (2009) (https://doi.org/10.1021/ct800503d)

In ADF2012 for hybrids the exact exchange contribution to the Pauli term is isolated and the contributions to the orbital term are divided among orbital symmetries.

9.2.1 Bond energy details

In the framework of Kohn-Sham MO theory and in conjunction with the fragment approach, one can decompose the bond energy between the fragments of a molecular system - say, a base and a substrate for E2 elimination - into contributions associated with the various orbital and electrostatic interactions. In ADF, we follow a Morokuma-type energy decomposition method. The overall bond energy ΔE is divided into two major components. In the first place, the preparation energy ΔE_{prep} corresponding to the amount of energy required to deform the separated fragments, A and B say, from their equilibrium structure to the geometry they acquire in the overall molecule ($\Delta E_{\text{prep,geo}}$), and to excite them to their valence electronic configuration ($\Delta E_{\text{prep el}}$). In the second place, the interaction energy ΔE_{int} between the prepared fragments.

$$\Delta E = \Delta E_{\text{prep}} + \Delta E_{\text{int}} = \Delta E_{\text{prep,geo}} + \Delta E_{\text{prep,el}} + \Delta E_{\text{int}}$$

In the following step, the interaction energy ΔE_{int} is further decomposed into three physically meaningful terms, which are printed in the ADF output file.

$$\Delta E_{\text{int}} = \Delta V_{\text{elst}} + \Delta E_{\text{Pauli}} + \Delta E_{\text{oi}} = \Delta E^0 + \Delta E_{\text{oi}}$$

The term ΔV_{elst} corresponds to the classical electrostatic interaction between the unperturbed charge distributions of the prepared fragments as they are brought together at their final positions, giving rise to an overall density that is simply a superposition of fragment densities $\rho_A + \rho_B$. (Note that we use the convention that energy terms containing potential energy only, kinetic energy only, or both kinetic and potential energy are indicated by V, T, and E, respectively.) For neutral fragments, ΔV_{elst} is usually attractive. The Pauli repulsion ΔE_{Pauli} arises as the energy change associated with going from $\rho_A + \rho_B$ the wave function $\Psi^0 = NA[\Psi_A \Psi_B]$ that properly obeys the Pauli principle through explicit antisymmetrization (*A* operator) and renormalization (*N* constant) of the product of fragment wave functions. It comprises the destabilizing interactions between occupied orbitals, and is responsible for any steric repulsion. In case of neutral fragments, it can be useful to combine ΔV_{elst} and ΔE_{Pauli} in a term ΔE_0 which, in the past, has been conceived as the steric interaction. However, we prefer to reserve the designation steric interaction or repulsion for ΔE_{Pauli} because that is, as already mentioned, the only source of net repulsive interactions between molecular fragments. Finally, the wavefunction is allowed to relax from Ψ^0 to the fully converged wave function Ψ. The associated orbital interaction energy ΔE_{oi} accounts for electron pair bonding, charge transfer (e.g., HOMO-LUMO interactions) and polarization (empty/occupied orbital mixing on one fragment due to the presence of another fragment). In a symmetric molecule this can be further decomposed into the contributions from the distinct irreducible representations Γ of the interacting system using the extended transition state method. In systems with a clear σ/π separation, this symmetry partitioning proves to be very informative.

$$\Delta E_{\text{oi}} = \sum_{\Gamma} \Delta E_{\text{oi},\Gamma}$$

An alternative, which can also be used in asymmetric molecules, is the ETS-NOCV analysis, which decomposes the bonding interactions in the context of natural orbitals for chemical valence, see Refs[?][6][7].

The pure orbital interaction effect of forming electron pair bonding between open shell molecules can approximately be calculated with a bond energy analysis in which all virtual orbitals are removed from the fragments, see Ref.[?].

An extensive discussion of the physical meaning of all the terms in the energy decomposition is given in F.M. Bickelhaupt and E.J. Baerends, *Kohn-Sham Density Functional Theory: Predicting and Understanding Chemistry*, In: Rev. Comput. Chem.; Lipkowitz, K. B. and Boyd, D. B., Eds.; Wiley-VCH: New York, 2000, Vol. 15, 1-86.

---

[6] M. Mitoraj, A. Michalak and T. Ziegler, *On the Nature of the Agostic Bond between Metal Centers and Beta-Hydrogen Atoms in Alkyl Complexes. An Analysis Based on the Extended Transition State Method and the Natural Orbitals for Chemical Valence Scheme (ETS-NOCV)*, Organometallics 28, 3727 (2009) (https://doi.org/10.1021/om900203m)

[7] M. Mitoraj, and A. Michalak, *Natural orbitals for chemical valence as descriptors of chemical bonding in transition metal complexes*, Journal of Molecular Modeling 13, 347 (2007) (https://doi.org/10.1007/s00894-006-0149-4)

Text is mostly taken from: *Chemistry with ADF*, G. te Velde, F.M. Bickelhaupt, E.J. Baerends, C. Fonseca Guerra, S.J.A. van Gisbergen, J.G. Snijders, T. Ziegler Journal of Computational Chemistry 22, 931 (2001) (https://doi.org/10.1002/jcc.1056).

9.2.2 Notes on double-hybrid functionals and MP2

Since ADF2019.3, double-hybrid density functionals have been implemented in ADF. These include contributions from virtual orbitals from a *MP2 calculation* (page 331). The evaluation of MP2 energies requires orbital energies as input, which, however, are not available for orthogonal fragments. Thus, whenever a double-hybrid calculation is carried out, the Bonding energy with respect to the individual fragments is calculated and printed out. However, all terms which rely on the orthogonal fragment density (e.g. Pauli repulsion terms) do not include the MP2 part of the double-hybrid.

9.2.3 Total energy evaluation

ADF normally does not calculate the total energy of a system (the energy wrt bare nuclei and free electrons). However, ADF calculates the energy of the system with respect to fragment energies. By default, these fragments are the spherical spin-restricted neutral atoms, but one can also use larger fragments. For this reason total energies from other programs could not be compared to ADF directly. Note, however, that only energy difference comparisons are meaningful. These are the only energies that play a role in chemistry of course, and for this one does not need total energies.

If you really want to calculate the total energies, there are two options in ADF

Total energy by adding the binding energy of the atoms

There is a work-around to calculate the total energy of a system: calculate the total energies of the atomic fragments and add them to the bonding energy. Because total energy of an atom is, by definition, the energy difference between the atom and the (nucleus+free electrons) system one can calculate it by calculating a single atom with the charge equal to the number of electrons. ‘Bonding energy’ of such an ‘atom’ will then be equal to negative of the total energy of the atomic fragment. Care should be taken to apply this procedure to frozen-core fragments. In this case, it only makes sense to remove the valence electrons and leave the frozen core.

TOTALENERGY keyword

The total energies have not been tested extensively and should therefore be used with caution. In addition to bond energies it is now possible to compute total energies with ADF by including the keyword TOTALENERGY in the input. This work is in progress.

```
TOTALENERGY
```

The total energy will be computed for the chosen XC functional (LDA, GGA, hybrid functionals, or Hartree-Fock). MetaGGA functionals, (ZORA) scalar relativistic and relativistic spin-orbit calculations, electric fields and QM/MM are not supported yet.

In particular the requirements to the integration accuracy are somewhat higher than for bond energies. It is recommended to use an integration grid (BeckeGrid) of quality “Good”. If in doubt, a convergence test with respect to the integration accuracy is recommended.

9.2.4 Interacting Quantum Atoms (IQA)

Note: This part of the code has been completely rewritten and it is now parallelized. The use of **symmetry keyword** is now allowed (however, all the points of the grid will be used) and the code has been extended to **open-shell systems** (AMS 2023). Relativistic calculations are not allowed. More precisely, in case of a relativistic calculation, components based on the electron density would be correct (such as Coulomb interactions) but those depending explicitly on the wavefunction would be wrong.

The interacting quantum atoms approach (IQA) has been developed by Angel Martín Pendás and coworkers in the framework of real-space partitions of the molecular space[8][9][19]. This can be seen as an alternative to the Bickelhaupt-Baerends' bond energy decomposition scheme but for the **total energy** of the molecular system. A full IQA decomposition is proposed as the default procedure, including both intra-atomic and inter-atomic interactions.

In a wavefunction context, the IQA QTAIM partition of the molecular energy leads exactly to:

$$E = \sum_A \left(T^A + E^{AA}_{Ne} + E^{AA}_{eeCl} + E^{AA}_{eeXC}\right) + \frac{1}{2}\sum_{A\neq B}\left(E^{AB}_{NN} + E^{AB}_{eN} + E^{AB}_{Ne} + E^{AB}_{eeCl} + E^{AB}_{eeXC}\right)$$

The first four terms of the right-hand side in the equation above correspond to intra-atomic (also coined 'self') terms for each atom 'A': the atomic kinetic energy (T^A), the interaction energy between the nucleus and the electron density inside basin A (E^{AA}_{Ne}), and the repulsion energy between electrons in A, decomposed into classical electrostatic (E^{AA}_{eeCl}) and exchange-correlation contributions (E^{AA}_{eeXC}).

The last sum in the equation corresponds to the interactions between each atom pair 'AB' (bonded or not bonded by a bond path): the repulsion energy between nuclei in A and B (E^{AB}_{NN}), the attraction of the electrons in A by the nucleus in B (E^{AB}_{eN}) the attraction of the electrons in B by the nucleus in A (E^{AB}_{Ne}), and the repulsion energy between electrons in A with those in B, which can be split into a classical electrostatic contribution (E^{AB}_{eeCl}) and an exchange-correlation one (E^{AB}_{eeXC}).

Within a Kohn-Sham DFT framework, the implementation of this partition is not straightforward (see ref.[10] for more details). For any atom-atom pair 'AB', we evaluate:

$$E^{AB}_{inter} = E^{AB}_{NN} + E^{AB}_{eN}[\rho] + E^{AB}_{Ne}[\rho] + E^{AB}_{eeCl}[\rho] + E^{AB}_{eeX}\left[\left\{\psi_i^{KS}\right\}\right]$$

where $E^{AB}_{eeX}\left[\left\{\psi_i^{KS}\right\}\right]$ corresponds to the full exact exchange ('HF-like') interacting energy between the two atoms. Tognetti and Joubert have shown that despite underlying approximations, this expression can be safely used at least for semi-quantitative purposes[11].

Three energetic terms are printed for each atom pair. The first one corresponds to the *total* interaction energy between the two atoms (E^{AB}_{inter}). This energy is then split into two contributions: the 'covalent part' corresponds to the exchange energy between the two atomic basins ($E^{AB}_{eeX}\left[\left\{\psi_i^{KS}\right\}\right]$), while the 'non covalent one' is the sum of all remaining electrostatic contributions ($E^{AB}_{NN} + E^{AB}_{eN} + E^{AB}_{Ne}[\rho] + E^{AB}_{eeCl}[\rho]$). This last term also includes a dispersion correction (if and only if dispersion is a posteriori requested in the DFT calculation). It should be noticed that here, *non covalent* is ti be understood as the sum of all contributions that are not covalent (and not only London-type interactions). To summarize, when no dispersion correction of the Grimme's type is added, the *non covalent* is equal to the classical electrostatic part (previously coined *ionic*), while when Grimme's D3 is used, *non covalent* is the sum of this ionic part and of the dispersion correction.

As already mentioned, a full IQA decomposition is now automatically performed, including intra-atomic terms. Choosing the **verbose mode** allows to give more details about the energy decomposition (all terms are detailed in this case and

---

[8] M.A. Blanco, A.M. Pendás and E. Francisco, *Interacting Quantum Atoms: a correlated energy decomposition scheme based on the quantum theory of atoms in molecules*, Journal of Chemical Theory and Computation 1, 1096 (2005) (https://doi.org/10.1021/ct0501093).

[9] A.M. Pendás, MA. Blanco and E. Francisco, *A molecular energy decomposition scheme for atoms in molecules*, Journal of Chemical Theory and Computation 2, 90 (2006) (https://doi.org/10.1021/ct0502209).

[19] J.M. Guevara-Vela, E. Francisco, T. Rocha-Rinza and A.M. Pendás, *Interacting Quantum Atoms - A review*, Molecules 25, 4028 (2020) (https://doi.org/10.3390/molecules25174028)

[10] V. Tognetti and L. Joubert, *On Atoms-in-Molecules Energies from Kohn–Sham calculations*, ChemPhysChem 18, 2675 (2017) (https://doi.org/10.1002/cphc.201700637).

[11] V. Tognetti and L. Joubert, *On the physical role of exchange in the formation of an intramolecular bond path between two electronegative atoms*, Journal of Chemical Physics 138, 024102 (2013) (https://doi.org/10.1063/1.4770495).

described in the output itself). This is now the default option.

Tip: If you need to extract relevant IQA terms along a PES Scan, see also the REG/IQA tutorial

---

You can activate the full IQA analysis via the IQA input block:

```
IQA
   Enabled Yes/No
   Print [normal | verbose]
   AtomsToDo integer_list
End
```

`IQA`

Type Block

Description Total energy decomposition based on the interacting quantum atoms (IQA) approach and using QTAIM real-space partition.

`Enabled`

Type Bool

Default value No

GUI name Calculate: Interacting Quantum Atoms (IQA)

Description Calculate the total energy decomposition using the interacting quantum atoms (IQA) approach and the QTAIM real-space partitioning.

`Print`

Type Multiple Choice

Default value verbose

Options [normal, verbose]

Description Minimal output (default) or verbose mode (detailed energy decomposition)

`AtomsToDo`

Type Integer List

GUI name Include atoms

Description Define a subset of atoms for which the IQA atom-atom interactions are calculated (no intra-atomic terms).

If left empty, all atoms will be included (full IQA).

9.3 Localized Molecular Orbitals

ADF provides the Boys-Foster method and Pipek-Mezey method for localization of Molecular Orbitals[1][2][3]. This implies a unitary transformation of the occupied molecular orbitals as computed in the SCF procedure, with the objective to obtain

---

[1] J. Autschbach and H.F. King, *Analyzing molecular static linear response properties with perturbed localized orbitals*, Journal of Chemical Physics 133, 044109 (2010) (https://doi.org/10.1063/1.3455709)

[2] C. Edmiston and K. Rudenberg, *Localized Atomic and Molecular Orbitals*, Reviews of Modern Physics 35, 457 (1963) (https://doi.org/10.1103/RevModPhys.35.457)

[3] J.M Foster and S.F. Boys, *Canonical Configurational Interaction Procedure*, Reviews of Modern Physics 32, 300 (1960) (https://doi.org/10.1103/RevModPhys.32.300)

a (transformed) set of orbitals that represent exactly the same charge density but with molecular orbitals that are more localized in space than the original MOs.

These localization methods can not be used in case of spin-orbit coupling.

The goal of orbital-localization lies in analysis: the localized orbitals provide an easier-to-interpret picture. The localized molecular orbitals can be visualized with the ADF-GUI.

Orbital localization procedures require a measure of the localization of the orbitals which can then be optimized in the space of the allowed unitary transformations. Methods advocated in the literature differ in the definition of this measure. The Boys-Foster method minimizes the mean extension of the occupied orbitals around their center of gravity; see the literature for details. The Pipek-Mezey localization maximizes the sum of orbital-dependent partial charges on the nuclei, see literature for details. Both Boys-Foster localization method and Pipek-Mezey localization method have been implemented in ADF, see Ref.[?].

Occasionally it is useful to apply the localization only to a subset of the MOs, with the objective to expose certain features better. This is accomplished by performing the localization in a number of distinct steps, where at each step the localization is restricted by keeping a subset of the MOs frozen. A case is worked out in the Examples document.

The computation of localized orbitals is controlled with the block-type key. By default (if the key is not supplied in input) no orbital localization is carried out.

```
LOCORB {nopop store}
  {Criterion <BOYS|PM>}
  Spintype FrozenMOs
  Spintype FrozenMOs
  ...
end
```

`nopop` Specifies that no SFO population analysis is to be carried out on the localized MOs. By default this population analysis will be printed in the output file.

`store` Specifies that the transformation from MOs to localized MOs is stored on adf.rkf (TAPE21).

`Criterion <BOYS|PM>` By default the Boys-Foster localization method is used if Criterion is not specified and in case 'Criterion BOYS' is specified. If 'Criterion PM' is specified, the Pipek-Mezey localization method is used.

`Spintype` Must be either alfa or beta (not case sensitive) and refers to spin-A and spin-B orbitals respectively. In a spin-restricted run beta records are meaningless and must not be used.

`FrozenMOs` A list (possibly empty) of integers, referring to a list of MOs from the SCF, and/or labels of irreducible representations. The integers and/or labels may be given in any order.

Each record Spintype FrozenMOs in the data block defines a localization *cycle* in which the localization procedure is carried out on all orbitals (of the indicated spin), except those indicated by the FrozenMOs.

For either spin at least one localization cycle is carried out. If no data record for that spin is found in the data block, a full localization is performed, without any MOs excluded.

The data block may be completely empty (but the record end must be supplied since the key is block-type) and would be equivalent with specifying two records, one for either spin, without any FrozenMOs:

```
LOCORB {nopop}
end
```

is equivalent with

```
LOCORB {nopop}
  alfa
  beta
end
```

The integers in FrozenMOs refer to an overall list of SCF MOs consisting of all valence MOs in each symmetry representation up to and including the highest non-empty one. So, when for instance in the first irrep MO #4 is the highest non-empty one and in the second irrep mo #2 is the highest non-empty one, then in the overall list the first 4 are the orbitals of the first irrep, the no.s 5 and 6 are from the second irrep, et cetera.

Each symmetry label in FrozenMOs collectively denotes in one stroke all molecular orbitals of that representation up to and including the highest occupied one (in that symmetry). The label may be the name of an irreducible representation or of a subspecies. In the former case all partner representations are denoted collectively. In an atom symmetry for instance, specifying P would be equivalent to P:x P:y P:z.

Note that if the final SCF has in any symmetry representation empty orbitals *below* the highest non-empty orbital in that symmetry - violating the Aufbau principle - then these empty orbitals are included in the above-defined overall list and hence a FrozenMOs specification is necessary, namely to avoid mixing MOs with different occupation numbers in the localization.

Note: It is imperative that in a particular localization cycle only MOs from the SCF are combined that have identical occupation numbers. If this is violated the program will carry out the localization without error message, but the results are incorrect in the sense that the density defined by the localized orbitals is *not* the same anymore as the SCF density.

So, if any of the MOs in the overall list defined above is not fully occupied (open shell, excited state, …) you need to define precisely the localization cycles - localizing in each cycle only MOs with identical occupations and freezing all others - in order to obtain sensible results.

In the output file the localized MOs are printed as expansions in SFOs and (optionally) a population analysis is given, again in terms of the SFOs. Furthermore, each localized MO has associated with it an energy value and an occupation number. The energy is the expectation value of the Fock operator for the orbital. The occupation number is obtained as a weighted sum from the SCF MOs that were combined into the localized orbital. As mentioned before one should combine only SCF MOs with identical occupations into a localized orbital, in which case its occupation number will be the same. The printout of the occupation number of the localized orbital allows therefore a verification that a correct localization procedure has been carried out.

9.3.1 Perturbed Localized Molecular Orbitals

Perturbed localized molecular orbitals, correct to first order in an applied field, can be calculated in case of AORESPONSE. Can be used if the applied field changes the density in first order. Ref.[?] describes this functionality and the implementation in ADF in detail, and gives some examples. The perturbed localized molecular orbitals can be visualized with the ADF-GUI. In addition to the AORESPONSE and the LOCORB key one has to specify the PERTLOC key in order to calculate these perturbed localized molecular orbitals. The optional subkeywords of the block key PERTLOC are described below.

```
LOCORB ..
  ..
End
AORESPONSE
  ..
End
PERTLOC
  efield
  bfield
  alfa
  gprime
  beta
  <static|dynamic>
```

(continues on next page)

(continued from previous page)

```
  <diag|fulltens>
End
```

efield The perturbation is an electric field (default).

bfield The perturbation is a magnetic field. Should be consistent with AORESPONSE.

alfa Analyze the static or dynamic polarizability.

gprime Analyze the G' (gyration) tensor, for optical rotation dispersion. Requires a frequency dependent perturbation field, with a frequency (omega) unequal to zero.

beta Analyze the optical rotation parameter beta. The relation to G' is beta = -G'/omega. The optical rotation parameter beta is calculated directly[4] and has a well-defined static limit, i.e. omega can be zero or non-zero.

<static|dynamic> The static or dynamic (frequency dependent) subkeyword should be consistent with what is used in AORESPONSE. Static is default, should be used for a static field. Dynamic should be used for a frequency dependent perturbation field.

<diag|fulltens> Diag is default, which will only analyze the diagonal of the response tensor. If fulltens is specified the full tensor is analyzed.

9.4 Advanced charge density and bond order analysis

In addition to Mulliken charge analysis, ADF calculates several atomic charges that do not share the flaws of Mulliken (strong basis set dependence). The multipole-derived charge analysis exactly reproduces dipole and higher multipole moments of the molecule. Other charge analysis methods ('Voronoi deformation density' and 'Hirshfeld' provide atomic charges that agree well with chemical intuition. Nalewajski bond orders can be calculated and show good agreement with experimental trends and chemical intuition, even for transition metal compounds.

Note that the amount of data can be regulated with the keys PRINT, NoPrint, EPrint and Debug.

9.4.1 Charges, Populations

Mulliken populations

See the input key *EPRINT* (page 277). See also the *section on Mulliken populations* (page 285).

Hirshfeld charges, Voronoi deformation density (VDD)

No special input key required, except for 'PRINT VDDANALYSIS' for a symmetry decomposed VDD analysis. See also the *section on Hirshfeld charges, Voronoi deformation density* (page 285).

Multipole derived charges

No special input key required. See also the *section on MDC* (page 287).

Charge model 5 (CM5)

Charges calculated with CM5 activated by keyword

---

[4] S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *A Density Functional Theory study of frequency-dependent polarizabilities and van der Waals dispersion coefficients for polyatomic molecules*, Journal of Chemical Physics 103, 9347 (1995) (https://doi.org/10.1063/1.469994)

```
CM5
```

See also the *section on CM5* (page 287).

Only one of the charges above can be returned to the AMS driver, in case charges are needed at the driver level or explicitly requested with `Properties%Charges` in the AMS driver input. The choice for which charge analysis scheme is used for the returned charges in made with the `AtomicChargesTypeForAMS` keyword.

```
AtomicChargesTypeForAMS [Mulliken | Hirshfeld | CM5 | Voronoi | MDC-M | MDC-D | MDC-Q␣
↪|
                         QTAIM]
```

AtomicChargesTypeForAMS

Type Multiple Choice

Default value Mulliken

Options [Mulliken, Hirshfeld, CM5, Voronoi, MDC-M, MDC-D, MDC-Q, QTAIM]

GUI name Atomic charges for AMS

Description Type of atomic charges to be used by AMS.

Note that some of these atomic charges are computed and printed by default in ADF.

Hirshfeld charges are available only for default atomic fragments.

9.4.2 Bond orders

The calculation of bond orders can be requested at the AMS driver level (see the Properties section of the AMS driver manual):

```
$AMSBIN/ams << eor
...
Properties
   BondOrders Yes
End

Engine ADF
   ...
EndEngine
eor
```

By default, bond orders are computed using the Nalewajski-Mrozek[3][4][37][38][39] method. There exist three alternative definitions of the valence and bond order indices within the Nalewajski-Mrozek approach. By default the values obtained from partitioning of Tr(P Δ P) (i.e. Nalewajski-Mrozek-3) are calculated and printed in the output. For more information on alternative Nalewajski-Mrozek bond order indices see (see also: *section on bond order analysis* (page 287)).

In the `Engine ADF` part of the input, you can specify the following options:

---

[3] A. Michalak, R.L. De Kock and T. Ziegler, *Bond Multiplicity in Transition-Metal Complexes: Applications of Two-Electron Valence Indices*, Journal of Physical Chemistry A 112, 7256 (2008) (https://doi.org/10.1021/jp800139g)

[4] R.F. Nalewajski, J. Mrozek and A. Michalak, *Two-electron valence indices from the Kohn-Sham orbitals*, International Journal of Quantum Chemistry 61, 589 (1997) (https://doi.org/10.1002/(SICI)1097-461X(1997)61:3%3C589::AID-QUA28%3E3.0.CO;2-2)

[37] R.F. Nalewajski and J. Mrozek, *Modified valence indices from the two-particle density matrix*, International Journal of Quantum Chemistry 51, 187 (1994) (https://doi.org/10.1002/qua.560510403)

[38] R.F. Nalewajski, J. Mrozek and A. Michalak, *Exploring Bonding Patterns of Molecular Systems Using Quantum Mechanical Bond Multiplicities*, Polish Journal of Chemistry 72, 1779 (1998) (http://ichf.edu.pl/pjch/pj-1998/pj07s98.htm#1779)

[39] R.F. Nalewajski, J. Mrozek and G. Mazur, *Quantum chemical valence indices from the one-determinantal difference approach*, Canadian Journal of Chemistry 74, 1121 (1996) (https://doi.org/10.1139/v96-126)

```
BondOrders
   PrintAll Yes/No
   PrintTolerance float
   TypeForAMS [Nalewajski-Mrozek-1 | Nalewajski-Mrozek-2 | Nalewajski-Mrozek-3 |␣
↪Mayer |
               Gopinathan-Jug]
End
```

`BondOrders`

Type Block

Description Options for the calculation of bond orders. Note: the calculation of bond orders should be requested via the Properties%BondOrders input option in the AMS driver input.

`PrintAll`

Type Bool

Default value No

Description If 'Yes', all five types of bond orders (i.e. Nalewajski-Mrozek-1,2 & 3, Mayer and Gopinathan-Jug) will be printed to the output. Otherwise only the Nalewajski-Mrozek-3 and the type requested in BondOrders%TypeForAMS will be printed.

`PrintTolerance`

Type Float

Default value 0.2

Description Only bond orders larger than this threshold will be printed in the output (this treshold applies only to the printing in the 'BOND-ORDER ANALYSIS' section of the ADF output.

`TypeForAMS`

Type Multiple Choice

Default value Nalewajski-Mrozek-3

Options [Nalewajski-Mrozek-1, Nalewajski-Mrozek-2, Nalewajski-Mrozek-3, Mayer, Gopinathan-Jug]

GUI name Bond order type for AMS

Description The type of bond order that will be saved, printed and used by AMS.

Nalewajski-Mrozek-1,2: bond orders calculated from two-electron valence indices based on partitioning of tr(Delta_P^2) using 3-index set or 4-index set respectively. Nalewajski-Mrozek-3: bond-orders calculated from valence indices based on partitioning of tr(P*Delta_P). Inter-atomic bond orders are not defined with non-atomic fragments.

The bond order analysis is based on SFOs. The symmetry used in the calculation should be `NOSYM`. The bond analysis may be used also for multi-atomic fragments. The fragment-fragment bond orders are printed in such a case. Note that in the present implementation all fragment types should be different.

Mayer bond orders and Mulliken atom-atom populations per l-value can be enabled using the `ExtendedPopan` key:

```
ExtendedPopan Yes/No
```

`ExtendedPopan`

Type Bool

Default value No

GUI name : Extended population analysis

Description Calculate the Mayer bond orders and Mulliken atom-atom populations per l-value

See also the *section on bond order analysis* (page 287).

9.4.3 ETS-NOCV: Natural Orbitals for Chemical Valence

With the ETS-NOCV charge and energy decomposition scheme the deformation density is partitioned into the different components (σ, π, δ) of the chemical bond. The energy contributions to the total bond energy is calculated for each specific orbital interactions between fragments, giving insight in the orbital interactions also for non-symmetric molecules. The ETS-NOCV analysis offers a compact quantitative picture of the chemical bond, which is also qualitatively attractive to chemists.

Tip: See also the EDA-NOCV tutorial

The Natural Orbitals for Chemical Valence (NOCV) approach has been derived from the Nalewajski-Mrozek valence theory??. From the mathematical point of view, each NOCV ψ_{i} is defined as an eigenvector of the deformation density matrix in the basis of fragment orbitals.

$$\Delta P \Psi_i = \nu_i \Psi_i$$

Thus, the deformation density $\Delta\rho$ can be expressed in the NOCV representation as a sum of pairs of complimentary eigenfunctions (ψ_{-k}, ψ_k) corresponding to eigenvalues -ν_{k} and ν_{k} with the same absolute value but opposite signs:

$$\Delta\rho(r) = \sum \Delta\rho_k(r) = \sum \nu_k \left(-\Psi_{-k}^2(r) + \Psi_k^2(r)\right)$$

here, k goes over the pairs of NOCV's.

In the combined ETS-NOCV scheme the orbital interaction term ΔE_{orb} is expressed in terms of NOCV's as[5][6]:

$$\Delta E_{\text{orb}} = \sum \Delta E_k^{\text{orb}} = \sum \nu_k \left(-F_{-k}^{\text{TS}} + F_k^{TS}\right)$$

here, $-F_{-k}^{\text{TS}}$ and F_k^{TS} are diagonal transition-state Kohn-Sham matrix elements corresponding to NOCV's with eigenvalues -ν_{k} and ν_{k} , respectively. The advantage of this expression is that usually only a few complimentary NOCV pairs significantly contribute to the total ΔE_{orb}. Another advantage of this approach is that not only can each $\Delta\rho_k(r)$ be visualized but there is also a well defined bonding energy contribution ΔE_k^{orb} corresponding to it.

Further decomposition of ΔE_k^{orb} into ΔT_k^{orb} and ΔV_k^{orb} components is possible as the Fock matrix can be decomposed into kinetic and potential energy matrices[53]:

$$\Delta E_{\text{orb}} = \sum \nu_k \left(-F_{-k}^{\text{TS}} + F_k^{\text{TS}}\right) = \sum \nu_k \left(-T_{-k}^{\text{TS}} - V_{-k}^{\text{TS}} + T_k^{\text{TS}} + V_k^{\text{TS}}\right) = \sum \left(\Delta T_{\text{orb}} + \Delta V_{\text{orb}}\right)$$

where T_{-k}^{TS} and V_{-k}^{TS} are diagonal elements of T and V matrices corresponding to F^{TS} matrix.

Remarks

The ETS-NOCV analysis is often not very useful when atomic fragments are used. No symmetry must be used in the final calculation, thus, use a Symmetry NOSYM keyword if your molecule is symmetric. The analysis is not completely implemented for meta-GGA's and meta-hybrids.

---

[5] M. Mitoraj, A. Michalak and T. Ziegler, *A Combined Charge and Energy Decomposition Scheme for Bond Analysis*, Journal of Chemical Theory and Computation 5, 962 (2009) (https://doi.org/10.1021/ct800503d)

[6] M. Mitoraj, A. Michalak and T. Ziegler, *On the Nature of the Agostic Bond between Metal Centers and Beta-Hydrogen Atoms in Alkyl Complexes. An Analysis Based on the Extended Transition State Method and the Natural Orbitals for Chemical Valence Scheme (ETS-NOCV)*, Organometallics 28, 3727 (2009) (https://doi.org/10.1021/om900203m)

[53] F. Sagan and M. Mitoraj, *Kinetic and Potential Energy Contributions to a Chemical Bond from the Charge and Energy Decomposition Scheme of Extended Transition State Natural Orbitals for Chemical Valence*, Journal of Physical Chemistry A 123, 4616 (2019) (https://doi.org/10.1021/acs.jpca.9b01420)

Improvements in ADF2012 to both the ETS and NOCV analysis with hybrids. ETS: Now the exact exchange contribution to the Pauli term is isolated and the contributions to the orbital term are divided among orbital symmetries. NOCV: The exact exchange contribution to the Fock operator is included when calculating energy contributions. These changes do not apply to meta-hybrids.

Usage

In order to perform the ETS-NOCV analysis, the following two keywords must be specified at the same time:

```
ETSNOCV
   {RHOKMIN rhokmin}
   {EKMIN ekmin}
   {ENOCV enocv}
   {TVANALYSIS}
End
PRINT {ETSLOWDIN | ETSLOWDIN-Unrestricted}
```

ETSNOCV The ETSNOCV keyword specifies thresholds for printing of NOCV-related information. All three arguments are optional and when all three are omitted only the NOCV's corresponding to eigenvalues abs $(\nu_k) \geq 0.05$ are included in the analysis.

RHOKMIN The threshold for population analysis of each deformation density contribution in terms of individual SFO's.

EKMIN The threshold for orbital interaction energy contributions corresponding to deformation density components originating from each NOCV-pairs.

ENOCV The threshold for NOCV-eigenvalues.

TVANALYSIS Enable T/V decomposition of energy contributions.

PRINT {ETSLOWDIN | ETSLOWDIN-Unrestricted} Only one of the two PRINT options is supposed to be used to activate printing of ETS-NOCV results. The choice depends on the bonding situation.

ETSLOWDIN If one is interested in a description of bonding between closed-shell molecular fragments, then 'PRINT ETSLOWDIN' keyword must be used. In such a case one set of NOCV's originating from the total deformation density matrix $\Delta P = (\Delta P_\alpha + \Delta P_\beta)$ will be printed out. See the example of carbene bonding between closed shell CH2 and Cr(CO)5.

ETSLOWDIN-Unrestricted If, however, one is interested in a description of bonding between open-shell molecular fragments then the 'PRINT ETSLOWDIN-Unrestricted' keyword must be used. In this case two sets of NOCV's originating from ΔP_α and ΔP_β will be printed out. See the example of CH3-CH3 bonding between two CH3 radicals with opposite spins. This option must also be used when one wants to analyze bonding in a molecule with unpaired electrons.

PRINT NOCVHirshfeld The NOCVs can be integrated per fragment using the Hirshfeld partitioning scheme. Using these integrals, one can distinguish between inter- and intra-fragment NOCVs. An inter-fragment NOCV has non-zero integral Hirshfeld ΔQ value and corresponds to a charge transfer between fragments. An intra-fragment NOCV has a small ΔQ value and corresponds to polarization of the fragments. When 'PRINT NOCVHirshfeld' is specified, the $\int \Delta\rho = -\Delta Q$ values per NOCV are printed to the output file in the "Hirshfeld partitioning" table at the end of the ETS-NOCV section.

9.4.4 Adfnbo, gennbo: NBO analysis

- `(PDF) NBO manual`

Dr. Autschbach, SCM, and Prof. Weinhold have collaborated to prepare a simple in put file generator, called adfnbo, for the GENNBO program of Prof. Weinholds Natural Bond Orbital (NBO) package. In ADF2013 and later the NBO 6.0 version is supported http://nbo6.chem.wisc.edu.

The GENNBO executable is included in the ADF distribution and can be enabled via the license file for all those who buy an NBO license from SCM (info@scm.com). An extensive documentation of GENNBO is part of the NBO manual. The application of ADFNBO to frozen-core basis sets needs to be further tested. Usage can be found below and in the Examples Document.

Next a brief summary of the capabilities of GENNBO is given (by Prof. Weinhold). GENNBO implements most capabilities of the full NBO 6.0 program suite as described on the NBO website: http://nbo6.chem.wisc.edu These include determination of natural atomic orbitals (NAOs), bond orbitals (NBOs), and localized MOs (NLMOs), as well as the associated NPA (atomic charges and orbital populations) and NRT (resonance structures, weightings, bond orders) valence descriptors, for a wide variety of uncorrelated and correlated (variational, perturbative, or density functional) theoretical levels. GENNBO-supported options include all keywords except those explicitly requiring interactive communication with the host electronic structure system (viz., $DEL deletions, NEDA, NCS, NJC). The GENNBO program typically sits conveniently on the PC desktop, ready to analyze (or re-analyze at will, with altered options) the final results of a complex ADF calculation performed on a remote cluster.

GENNBO "communicates" with the original ADF calculation through an archive file (JOB.47 file, preserving all necessary details of the final density) that is initially generated by ADF and subsequently becomes the input file for GENNBO. The .47 file contains a standard $NBO … $END keylist that can be edited with a standard word processor or text editor to include chosen NBO keyword options, just as though they might have appeared in the original input stream of an interactive ADFNBO run. The stand-alone GENNBO program therefore allows many alternative NBO analysis options to be explored at leisure, without costly re-calculation of the wave function.

In ADF2018 the NBO 6.0 version6 in the ADF distribution is updated to the March 2017 version, see also http://nbo6.chem.wisc.edu: "The 14-Mar-2017 distribution includes a number of other features of interest to general NBO users: The default NAO search algorithm has been modified to better preserve core-valence separation in rare cases where numerical near-degeneracies can lead to unphysical core-valence mixing. The numerical effects are generally negligible except where they become necessary. The "OLDNAO" keyword restores the legacy NAO algorithm for comparison purposes."

This new NAO method may change the NBO analysis of some properties (EFG, NMR) quite substantially, especially individual contributions.

Usage

ADF needs to write some data to file, which is done by including FULLFOCK, AOMAT2FILE, SAVE TAPE15, and SYMMETRY NOSYM in the input file for ADF. An all electron basis set is needed. Use the key 'spherical' in the adfnbo input. A file named FILE47 is generated by adfnbo which is an input file for the general NBO program gennbo6. Thus the usage is like

```
$AMSBIN/ams <<eor
...
Engine ADF
   Basis
     ...
     Core None
   End
   FullFock
   AOMat2File
   Save TAPE15
   Symmetry Nosym
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

$AMSBIN/adfnbo <<eor
   write
   fock
   spherical
   end input
eor

$AMSBIN/gennbo6 FILE47
```

Remark: if the key TESTJOB in included in the input for adfnbo, NRT will be included in FILE47. The calculation of NRT (natural resonance theory) can be very expensive for larger systems (or may fail to converge). Thus it is not recommended to include the key TESTJOB in the input for adfnbo for larger systems.

NBO analysis of EFG, NMR chemical shifts, NMR spin-spin coupling

For certain molecular properties it is possible to perform detailed analyses in terms of Natural Bond Orbitals (NBOs) and Natural Localized Molecular Orbitals (NLMOs). These features generally require a sequence of ADF and/or property code runs. An initial non-relativistic or scalar relativistic ADF run, followed by the generation of NBO and NLMO data, is required, and the resulting data files need to be present in subsequent property calculations, along with a keyword indicating that the NBO analysis is requested in the property module.

We have noted in the past some slight loss of numerical accuracy of the results after going through the various orbital transformations in the NBO - NLMO sequence. It is important that the user verifies in each case that the total contributions from the analysis are in agreement with the total calculated property, within the numerical integration accuracy limits. In order to assist the user with this, the analysis program always print the total analysis contributions, including small non-printed values.

Moreover, there appears to be a problem with the analysis of the Fock matrix in the NBO program in conjunction with ADF calculations. Therefore please do NOT use the Fock matrix second order perturbation theory analysis in NBO at this time. We will remove this disclaimer once the issue has been fixed. Applications of the NBO-NLMO property analysis codes have so far given no indication that the Fock matrix issue interferes with the analysis.

Important note: If properties are analyzed from within spin-orbit relativistic computations, the NBO/NLMO analysis is performed in terms of scalar (spin-free) relativistic orbitals, as detailed in the technical references. The results from these analyses are exact in the sense that they fully reproduce the final spin-orbit property result, and they allow to dissect the property in terms of more intuitive one component real scalar relativistic localized orbitals. Typically, the property analysis in a spin-orbit calculation involves contributions from unoccupied scalar NLMOs, whereas there are no such contributions if a non-relativistic or scalar relativistic property is analyzed.

Available properties for NBO analysis: EFG, NMR chemical shifts and NMR spin-spin coupling.

NBO analysis of EFG

EFGs: non-relativistic and scalar ZORA, in ADF/AOResponse. Requires initial ADF run with

```
AOresponse
 donothing
End
```

in order to generate orbitals that re equivalent to those generated in the subsequent ADF run where the EFG is calculated. Alternatively, simply calculate the EFG twice, once before the NBO generation step, and once afterward.

The next step (see below) is to create the NBOs and the required data files for the analysis. Afterward, in the second ADF run, use

```
Aoresponse
 efg NUC nbo
end
```

efg NUC nbo Here NUC is the number of the nucleus at which the EFG is to be computed (ADF internal atom ordering). Example: efg 1 nbo.

The threshold for printing the EFG-NBO contributions to output can be adjusted via the `Tresh` options. The default is 0.05, which means that only orbitals with absolute value contribution larger than 5% of the total EFG are printed. To increase the number of contributions printed, specify a smaller threshold. For example:

```
Aoresponse
   efg 1 nbo Tresh 0.01
end
```

In addition to the optional NBO analysis, the EFG program in AOResponse prints a Mulliken type analysis of the EFG principal components, and an analysis in terms of canonical MOs.

WARNING: The ordering of the principal components is lowest to highest including the sign. That is, we have $V11 \leq V22 \leq V33$. This does not conform to the usual convention of $|V11| \leq |V22| \leq |V33|$. Please make sure you select the right component for your analysis.

Example job: $AMSHOME/examples/adf/AlCl3_efgnbo. For an explanation of the output and a general usage tutorial, see[7]. Further references and recommended citations, see[8].

NBO analysis of NMR Chemical shift

An implementation is currently available for spin-orbit ZORA computations. If scalar ZORA calculations are to be analyzed, provide the input keyword

```
FAKESO
```

in the NMR input (outside of the 'nmr' keyword). If this feature is requested one should restrict the calculation to a single shielding tensor per NMR run. It would be good practice to check the results against regular NMR calculations where the analysis feature is not requested. No ZORA scaling is applied in the analysis results. The data should be equivalent to a regular computation in the NMR input with

```
NMR
 u1k best
 calc all
END
```

Depending on whether scalar or spin-orbit calculations are to be analyzed, the sequence of calculations is different:

scalar:

1. ADF, scalar ZORA, symmetry NOSYM, closed shell
2. generate NBOs and required data files for analysis
3. NMR with FAKESO and analysis keywords, use adf.rkf (previously known as TAPE21), TAPE10 from step 1.

spin-orbit:

1. ADF, scalar ZORA, closed shell
2. generate NBOs and required data files for analysis

---

[7] J. Autschbach, S. Zheng, and R.W. Schurko, *Analysis of Electric Field Gradient Tensors at Quadrupolar Nuclei in Common Structural Motifs*, Concepts in Magnetic Resonance Part A 36A, 84 (2010) (https://doi.org/10.1002/cmr.a.20155)

[8] A.J. Rossini, R.W. Mills, G.A. Briscoe, E.L. Norton, S.J. Geier, I. Hung, S. Zheng, J. Autschbach, and R.W. Schurko, *Solid-State Chlorine NMR of Group IV Transition Metal Organometallic Complexes*, Journal of the American Chemical Society 131, 3317 (2009) (https://doi.org/10.1021/ja808390a)

3. delete adf.rkf, TAPE10, TAPE15
4. ADF, spin-orbit ZORA, symmetry NOSYM, closed shell
5. NMR with analysis keywords, using adf.rkf, TAPE10 from step 4

In the NMR run, in addition to the NMR keyword, provide the following

```
analysis
 print 0.01
 canonical
 nbo
 components
end
```

The optional canonical keyword can be used independently from the NBO analysis features. It enables an analysis of the shielding in terms of the canonical MOs. The components keyword is optional and enables an analysis not only of the isotropic shielding but also of each principal component of the tensor. The print keyword selects printout of contributions relative to the total diamagnetic, paramagnetic. In the example, only contributions greater than 1% are printed. Set to zero to print ALL contributions.

Example job: $AMSHOME/examples/adf/CH4_nmrnbo. References[40][41][42].

NBO analysis of NMR spin-spin coupling (J-coupling)

Non-relativistic, scalar ZORA, spin-orbit ZORA

The sequence of jobs is similar to those in the NMR section.

scalar or **non-relativistic**:

1. ADF, scalar ZORA or nonrel.
2. generate NBOs and required data files for analysis
3. CPL with analysis keyword, use adf.rkf (previously known as TAPE21), TAPE10 from step 1.

spin-orbit:

1. ADF, scalar ZORA
2. generate NBOs and required data files for analysis
3. delete adf.rkf, TAPE10, TAPE15
4. ADF, spin-orbit ZORA
5. CPL with analysis keyword, using adf.rkf, TAPE10 from step 4

In the CPL run provide the following ‘contributions’ keyword to enable the analysis

```
nmrcoupling
   ... other options
   contributions 1E19 nbo
end
```

The numerical value selects a print threshold in SI units of T**2/J for the analysis. Increase the value to obtain less detail in the analysis. By default, ‘contributions’ triggers an analysis of the J-coupling in terms of canonical MOs. The nbo keyword enables in addition the NBO-NLMO analysis.

---

[40] J. Autschbach, *Analyzing NMR shielding tensors calculated with two-component relativistic methods using spin-free localized molecular orbitals*, Journal of Chemical Physics 128, 164112 (2008) (https://doi.org/10.1063/1.2905235)

[41] J. Autschbach and S. Zheng, *Analyzing Pt chemical shifts calculated from relativistic density functional theory using localized orbitals: The role of Pt 5d lone pairs*, Magnetic Resonance in Chemistry 46, S45 (2008) (https://doi.org/10.1002/mrc.2289)

[42] J. Autschbach and S. Zheng, *Relativistic computations of NMR parameters from first principles: Theory and applications*, Annual Reports on NMR Spectroscopy 67, 1 (2009) (https://doi.org/10.1016/S0066-4103(09)06701-5)

Please note that due to the history of how the program was developed the output from the scalar/nrel. analysis and from the spin-orbit calculations differs somewhat. The qualitative content is the same.

In scalar ZORA or non-relativistic CPL calculations without the SD term an orbital based analysis is only performed for the Fermi-contact mechanism. If you also need an analysis for the PSO and SD mechanisms but do not want to run a spin-orbit calculation with ADF please use the SD or NOSD keywords which will cause the spin-orbit branch of the CPL code to be used. In ZORA spin-orbit calculations the FC, SD, PSO, and cross terms are analyzed together by default. You can selectively switch them on or off in order to get individual mechanism analyses. The DSO mechanism is often negligible. An analysis tool for this mechanism has therefore not yet been developed.

Example job: $AMSHOME/examples/adf/CPL_CH3OH_NBO. References NMR spin-spin couplings with NBO analysis?[43][44][45]:

Generation of NBOs

How to generate the NBOs, NLMOs, and the data files needed for these calculations (step 2 below is step 2 in the examples above):

1. run ADF with scalar ZORA or non-relativistic options, and keep adf.rkf and TAPE15.

```
AMS_JOBNAME=Scalar $AMSBIN/ams <<eor
  ...
eor
```

2.

```
# run adfnbo in WRITE  mode to create the gennbo input file FILE47
# and one of the required property analysis files, adfnbo.kf

$AMSBIN/adfnbo << eor
 ADFfile Scalar.results/adf.rkf
 TAPE15file Scalar.results/TAPE15
 write
 spherical
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48
$AMSBIN/gennbo6 << FILE47

# run adfnbo in COPY mode to create the second property analysis
# file, adfnbo2.kf

$AMSBIN/adfnbo <<  eor
 ADFfile Scalar.results/adf.rkf
 TAPE15file Scalar.results/TAPE15
 spherical
 copy
eor

# run adfnbo in READ mode: prepare locorb on TAPE21

```

(continues on next page)

---

[43] J. Autschbach, *Analyzing molecular properties calculated with two-component relativistic methods using spin-free Natural Bond Orbitals: NMR spin-spin coupling constants* Journal of Chemical Physics 127, 124106 (2007) (https://doi.org/10.1063/1.2768363)

[44] J. Autschbach and B. Le Guennic, *Analyzing and interpreting NMR spin-spin coupling constants from molecular orbital calculations*, Journal of Chemical Education 84, 156 (2007) (https://doi.org/10.1021/ed084p156)

[45] A.M.A. Boshaala, S.J. Simpson, J. Autschbach and S. Zheng, *Synthesis and Characterization of the Trihalophosphine Compounds of Ruthenium* [RuX$_2$ (η^6 -cymene)(PY$_3$)] (X = Cl, Br, Y = F, Cl, Br) and the Related PF$_2$ (NMe$_2$) and P(NMe$_2$)$_3$ Compounds; Multinuclear NMR Spectroscopy and the X-ray Single Crystal Structures of [RuBr$_2$ (η^6 -cymene)(PF$_3$)], [RuBr$_2$ (η^6 -cymene)(PF$_2$ {NMe$_2$ })], and [RuI$_2$ (η^6 -cymene)(P{NMe$_2$ }$_3$)], Inorganic Chemistry 47, 9279 (2008) (https://doi.org/10.1021/ic800611h)

(continued from previous page)

```
$AMSBIN/adfnbo <<  eor
 ADFfile scalar.results/adf.rkf
 TAPE15file scalar.results/TAPE15
 spherical
 read
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48

# keep the adf.rkf after this sequence in order to
# be able to plot the NBOs and NLMOs with amsview

# clean up, keep adfnbo*.kf for any NBO property analyses.
```

9.4.5 QTAIM: Atoms in Molecules

One can calculate local and atomic properties using a real-space partition of the electronic density. It is based on the quantum theory of atoms in molecules (QTAIM) developed by Richard Bader[9]. Another possibility for Bader analysis is to use the adf2aim utility with a third party program such as Xaim.

Local, atomic, and non-local properties

The QTAIM input block replaces the *Bader* keyword. The former *Bader Reactivity* option now corresponds to the new *ConceptualDFT* block (see below).

The QTAIM functionality encompasses two different but related features: a topological analysis of the electron density gradient field[10], and a calculation of condensed atomic properties by integrating various local functions over atomic basins[46][47]. The topological analysis is performed at all levels of the analysis (normal, extended or full) and it produces a list of the molecule's electron density critical points and bond paths that can be visualized by the AMSview GUI module. Besides, the following properties at the critical points are calculated: electron density, its Laplacian and Hessian matrix, ellipticity, Jenkins' metallicity index[12], reduced density gradient, DFT energy density variation rates[13], and Abramov's local energy components[14]. At the *Extended* analysis level the following condensed atomic properties are calculated: charges, density Laplacian, dipole and quadrupole moments, Shannon and Renyi entropies, spatial distribution of the molecular orbitals over atomic basins. If the *Energy* keyword is specified, the atomic energies[11] are also calculated. At the *Full* level, the atom-atom matrix elements of the localization (LI) and delocalization (DI) indices[48][49] are computed. In ADF 2023, we also added the (optional) calculation of the Source Function introduced by Gatti and Bader[54][55]. This

---

[9] P.L.A. Popelier, *Atoms in Molecules An Introduction*, Pearson Education, Harlow, 2000.

[10] J.I. Rodríguez, *An Efficient Method for Computing the QTAIM Topology of a Scalar Field: The Electron Density Case*, Journal of Computational Chemistry 34, 681 (2013) (https://doi.org/10.1002/jcc.23180)

[46] J.I. Rodríguez, A.M. Köster, P.W. Ayers, A. Santos-Valle, A. Vela and G. Merino, *An efficient grid-based scheme to compute QTAIM atomic properties without explicit calculation of zero-flux surfaces*, Journal of Computational Chemistry 30, 1082 (2009) (https://doi.org/10.1002/jcc.21134)

[47] J.I. Rodríguez, R.F.W. Bader, P.W. Ayers, C. Michel, A.W. Götz and C. Bo, *A high performance grid-based algorithm for computing QTAIM properties*, Chemical Physics Letters 472, 149 (2009) (https://doi.org/10.1016/j.cplett.2009.02.081)

[12] P.W. Ayers, S. Jenkins, *Bond metallicity measures*, Comput. Theor. Chem. 1053, 112 (2015) (https://doi.org/10.1016/j.comptc.2014.10.040)

[13] V. Tognetti, L. Joubert, *Density functional theory and Bader's atoms-in-molecules theory: towards a vivid dialogue*, Phys. Chem. Chem. Phys. 16, 14539 (2014) (https://doi.org/10.1039/C3CP55526G)

[14] Y.A. Abramov, *On the Possibility of Kinetic Energy Density Evaluation from the Experimental Electron-Density Distribution*, Acta Cryst. A53, 264 (1997) (https://doi.org/10.1107/S010876739601495X)

[11] J.I. Rodríguez, .W. Ayers, A.W. Götz, and F.L. Castillo-Alvarado, *Virial theorem in the Kohn-Sham density-functional theory formalism: Accurate calculation of the atomic quantum theory of atoms in molecules energies*, Journal of Chemical Physics 131, 021101 (2009) (https://doi.org/10.1063/1.3160670)

[48] X. Fradera, M.A Austen, *The Lewis model and beyond*, R.F.W. Bader, J. Phys. Chem. A 103, 304 (1999) (https://doi.org/10.1021/jp983362q).

[49] J. Poater, M. Solà, M. Duran, X. Fradera, *The calculation of electron localization and delocalization indices at the Hartree–Fock, density functional and post-Hartree–Fock levels of theory*, Theor. Chem. Acc. 107, 362 (2002) (https://doi.org/10.1007/s00214-002-0356-8).

[54] R.F.W. Bader, C. Gatti, *A Green's function for the density*, Chem. Phys. Lett. 287, 233 (1998) (https://doi.org/10.1016/S0009-2614(97)01457-7)

[55]

function is evaluated for each bond or ring critical point of the molecule, allowing a decomposition of the values of the electron density at these points as a sum of atomic contributions. Since this calculation requires the integration of the laplacian of the electron density, we strongly recommend a numerical quality set (at least) to 'good' (or higher).

`QTAIM`

Type Block

Description This block is used to request a topological analysis of the gradient field of the electron density, also known as the Bader's analysis. If this block is specified without any sub-key, only local properties are calculated.

`AnalysisLevel`

Type Multiple Choice

Default value Normal

Options [Normal, Extended, Full]

Description Set the level of the QTAIM analysis:

Normal - topology analysis and properties at the density critical points,

Extended - same as Normal plus condensed atomic descriptors,

Full - same as Extended plus non-local descriptors.

`Source`

Type Bool

Default value No

GUI name Source Function

Description Calculate the Source Function at BCPs and RCPs.

`AtomsToDo`

Type Integer List

GUI name Include atoms

Description List of atoms for which condensed descriptors are to be calculated. By default all atoms are included.

`Spacing`

Type Float

Default value 0.5

Unit Bohr

Description Specifies spacing of the initial Cartesian grid when searching for critical points. It may be useful to specify a smaller value than the default if some critical points are missed. This will result in a more accurate but slower calculation.

Note that the accuracy of condensed atomic properties can be estimated by integrating the Laplacian of the electron density over a given basin, which, ideally, must vanish. The accuracy of the method can be improved by using a larger integration grid (NumericalQuality). Usually, the default grid is sufficient to achieve the accuracy of 10^{-3} a.u. (differences of milliHartree in the energies). Unfortunately, the convergence of the electron density Laplacian integral with the grid

---

C. Gatti, F. Cargnoni, L. Bertini, *Chemical information from the source function*, J. Comput. Chem. 24, 422 (2003) (https://doi.org/10.1002/jcc.10205) size is not monotonous. Therefore this type of Bader atomic property calculation should be considered for applications where computational efficiency is critical and moderate accuracy is sufficient.

Note: The *Energy* keyword is now obsolete (AMS 2023). The (virial) atomic energies are now automatically calculated at the *Extended* or *Full* level except for relativistic calculations. However, the other QTAIM properties *are* correct for such calculations (they depend on the sole electron density; see reference[15]). Please note that the atomic energies are correct only at equilibrium geometries (the virial theorem is used).

ADF2AIM

The ADF utility adf2aim (original name rdt21) developed by Xavi Lopez, Engelber Sans and Carles Bo converts an ADF adf.rkf (formerly known as TAPE21) to WFN format (for Bader analysis)

The program rdt21 is now called adf2aim and is part of the ADF package.

The WFN file is an input file for the third party program Xaim (see http://www.quimica.urv.es/XAIM for details), which is a graphical user interface to programs that can perform the Bader analysis. Usage of adf2aim can be found in the Examples Document.

Aromaticity index with QTAIM

The cyclic delocalization of mobile electrons in two or three dimensions is probably one of the key aspects that characterize aromatic compounds. However, aromaticity cannot be measured directly by any physical or chemical experiment because it is not a well-defined magnitude. One can define indicators of aromaticity, some of which are implemented in ADF. The matrix of localization/delocalization indices (LI-DI) and the following from it Iring (ring index) and MCI (multi center index) aromaticity indices (see Ref.[16]) can be computed for rings specified by the user. The rings are specified using the AROMATICITY input block, one line per ring. For example, for a benzene molecule where atoms 1-6 are carbon atoms the input would look like this:

```
AROMATICITY
  1 2 3 4 5 6
END
```

The program does not check whether the atoms in the AROMATICITY input block are connected so it is responsibility of the user to specify them correctly. Moreover, for the Iring index to be calculated correctly the atoms must be listed in the order they are connected in the ring.

Calculation of the aromaticity indices invokes the QTAIM analysis automatically and uses its results so all accuracy considerations for the QTAIM key above are also valid here. This feature requires that the calculation is performed without symmetry (Symmetry NOSYM).

Results of the calculation are printed to the output under the “Aromaticity Index” header. The first table presents the total number of electrons per atom and their breakdown in “localized” and “delocalized”. Note: the total number of electrons per atom $N_{total} = N_{loc} + N_{deloc}/2$ since delocalized electrons are counted twice. The second table presents the Iring and MCI ring indices for each ring specified in the AROMATICITY block. These results (and the original LiDi matrix from which they are computed) are stored in the Properties section in the t21 file.

---

[15] J.S.M. Anderson, J.I. Rodriguez, P.W. Ayers, and A.W. Götz, *Relativistic (SR-ZORA) Quantum Theory of Atoms in Molecules Properties*, Journal of Computational Chemistry 82, 81 (2017) (https://doi.org/10.1002/jcc.24520)

[16] F. Feixas, E. Matito, J. Poater and M. Sola, *Quantifying aromaticity with electron delocalisation measures*, Chem. Soc. Rev. 44, 6434 (2015) (https://doi.org/10.1039/c5cs00066a)

9.4.6 Conceptual DFT

Global, atomic, and non-local descriptors

The *ConceptualDFT* block replaces and extends the set of features behind the *Bader Reactivity* keyword found in the ADF versions before ADF2019. The frontier molecular orbital (FMO) approximation is used for all calculations. Orbital degeneracy is taken into account following Martinez[17]. All these descriptors are described in Ref.[18]. Which exactly descriptors are calculated depends on the *AnalysisLevel* key:

- *Normal:* global conceptual DFT descriptors, i.e. electronic chemical potential, electronegativity (its opposite), hardness and softness, hyperhardness[19], electrophilicity index[20], dissociation energies (nucleofuge and electrofuge, see Ref.[21]), electrodonating and electroaccepting powers[22] and the net electrophilicity[23]. Also, new global descriptors called *global dual descriptors* $\Delta f+$ and $\Delta f-$ are calculated[24], which correspond to the spatial integral of the positive (electrophilic) or negative (nucleophilic) regions of the original dual descriptor, respectively. These descriptors allow for comparing total reactivity of molecules, in contrast with Fukui functions or original dual descriptor that integrate over the whole space to 1 or 0 for all molecules.
- *Extended:* both global and atomic properties, including condensed (QTAIM) descriptors in both canonical and grand canonical ensembles. More precisely, the following properties are computed: condensed Fukui functions[26], dual descriptor[25][27] and some of their composite functions[50][51][52], as well as condensed local electrophilicity and nucleophilicity[28].
- *Full:* all properties are calculated, including non-local ones, such as the condensed linear response function[29] and the softness kernel[30].

Please note that atomic electronegativities[31] are calculated separately with an optional *Electronegativity* keyword. In contrast with other properties, it requires all-electron basis sets and triggers the *TotalEnergy* keyword.

---

[17] J. Martínez, *Local Reactivity Descriptors from Degenerate Frontier Molecular Orbitals*, Chem. Phys. Lett. 478, 310-322 (2009) (https://doi.org/10.1016/j.cplett.2009.07.086)

[18] G. Hoffmann, V. Tognetti and L. Joubert, *Can molecular and atomic descriptors predict the electrophilicity of Michael acceptors?*, J. Mol. Model. 24, 281 (2018) (https://doi.org/10.1007/s00894-018-3802-9).

[19] C. Morell, A. Grand, A. Toro-Labbé, *Is hyper-hardness more chemically relevant than expected ?*, J. Mol. Model. 19, 2893 (2013) (https://doi.org/10.1007/s00894-013-1778-z)

[20] R.G. Parr, L.v. Szenpály, S. Liu, *Electrophilicity index*, J. Am. Chem. Soc. 121, 1922 (1999) (https://doi.org/10.1021/ja983494x)

[21] P.W. Ayers, J.S.M. Anderson, J.I. Rodrigueza, Z. Jaweda, *Indices for predicting the quality of leaving groups*, Phys. Chem. Chem. Phys. 7, 1918 (2005) (https://doi.org/10.1039/b500996k)

[22] J.L. Gázquez, A. Cedillo, A. Vela, *Electrodonating and electroaccepting powers*, J. Phys. Chem. A 111, 1966 (2007) (https://doi.org/10.1021/jp065459f)

[23] P.K. Chattaraj, A. Chakraborty, S. Giri, *Net electrophilicity*, J. Phys. Chem. A 113, 10068 (2009) (https://doi.org/10.1021/jp904674x)

[24] F. Guégan, P. Mignon, V. Tognetti, L. Joubert, C. Morell, *Dual descriptor and molecular electrostatic potential: complementary tools for the study of the coordination chemistry of ambiphilic ligands*, Phys. Chem. Chem. Phys. 16, 15558 (2014) (https://doi.org/10.1039/c4cp01613k)

[26] R.G. Parr, W.T. Yang, *Density functional approach to the frontier-electron theory of chemical reactivity*, J. Am. Chem. Soc. 106, 4049 (1984) (https://doi.org/10.1021/ja00326a036)

[25] F. Zielinski, V. Tognetti, L. Joubert, *Condensed descriptors for reactivity: A methodological study*, Chem. Phys. Lett. 527, 67 (2012) (https://doi.org/10.1016/j.cplett.2012.01.011)

[27] C. Morell, A. Grand, A. Toro-Labbé, *New Dual Descriptor for Chemical reactivity*, J. Phys. Chem. A 109, 205 (2005) (https://doi.org/10.1021/jp046577a)

[50] A. Toro-Labbé, P. Jaque, J.S. Murray and P. Politzer, *Connection between the average local ionization energy and the Fukui function*, Chem. Phys. Lett. 407, 143 (2005) (https://doi.org/10.1016/j.cplett.2005.03.041)

[51] J. Padmanabhan, R. Parthasarathi, M. Elango, V. Subramanian, B.S. Krishnamoorthy, S. Gutierrez-Oliva, A. Toro-Labbé, D. R. Roy, P. K. Chattaraj, *Multiphilic descriptor for chemical reactivity and selectivity*, J. Phys. Chem. A 111, 130 (2007) (https://doi.org/10.1021/jp0718909)

[52] P.K. Chattaraj, B. Maiti, U. Sarkar, *Philicity: A unified treatment of chemical reactivity and selectivity*, J. Phys. Chem. A 107, 4973 (2003) (https://doi.org/10.1021/jp034707u)

[28] C. Morell, J.L. Gázquez, A. Vela, F. Guégan, H. Chermette, *Revisiting electroaccepting and electrodonating powers: proposals for local electrophilicity and local nucleophilicity descriptors*, Phys. Chem. Chem. Phys. 16, 26832 (2014) (https://doi.org/10.1039/c4cp03167a)

[29] P. Geerlings, S. Fias, Z. Boisdenghien, F. De Proft, *Conceptual DFT: chemistry from the linear response function*, Chem. Soc. Rev. 43, 4989 (2014) (https://doi.org/10.1039/C3CS60456J)

[30] S. Fias, F. Heidar-Zadeh, P. Geerlings, P.W. Ayers, *Chemical transferability of functional groups follows from the nearsightedness of electronic matter*, PNAS 114, 11633 (2017) (https://doi.org/10.1073/pnas.1615053114)

[31] V. Tognetti, C. Morell, L. Joubert, *Atomic electronegativities in molecules*, Chem. Phys. Lett. 635, 111 (2015) (https://doi.org/10.1016/j.cplett.2015.05.057)

`ConceptualDFT`

Type Block

Description Conceptual DFT Properties

`AnalysisLevel`

Type Multiple Choice

Default value Normal

Options [Normal, Extended, Full]

Description Set the level of the ConceptualDFT analysis:

Normal - global descriptors only,

Extended - both global and condensed (QTAIM) local descriptors,

Full - all descriptors including non local ones.

`Electronegativity`

Type Bool

Default value No

GUI name Atomic electronegativities

Description Calculate atomic electronegativities. Requires an all-electron calculation (no frozen core), triggers the TotalEnergy and increases the [AnalysisLevel] to at least Extended.

`AtomsToDo`

Type Integer List

GUI name Include atoms

Description Define a subset of atoms for which properties are calculated. If the [Domains] block is present then this list specifies which atoms are used to define the domains bounding box.

For more information about *Conceptual DFT*, we suggest the reading of these excellent reviews: Refs.[33] and[32].

Domains of the dual descriptor

In a recent paper[34], Tognetti et al. proposed a real-space partitioning scheme of the dual descriptor (DD) into chemically meaningful *reactivity domains*. Basically, a predominantly electrophilic (nucleophilic) DD domain constitutes a region of space where this function remains positive (negative). They can be visualized using the GUI and quantitative information can be obtained using the additional *Domains* key block.

Note: This analysis requires that the calculation is performed without symmetry (Symmetry NOSYM)

`ConceptualDFT`

Type Block

Description Conceptual DFT Properties

---

[33] H. Chermette, *Chemical reactivity indexes in density functional theory*, J. Comput. Chem. 20, 129 (1999) (https://doi.org/10.1002/(SICI)1096-987X(19990115)20:1%3C129::AID-JCC13%3E3.0.CO;2-A)

[32] P. Geerlings, F. de Proft, W. Langenaecker, *Conceptual Density Functional Theory*, Chem. Rev. 103, 1793 (2003) (https://doi.org/10.1021/cr990029p)

[34] V. Tognetti, C. Morell, L. Joubert, *Quantifying electro/nucleophilicity by partitioning the dual descriptor*, J. Comput. Chem. 36, 649 (2015) (https://doi.org/10.1002/jcc.23840)

`Domains`

Type Block

Description Calculate integrated properties for the domains (same sign) of the dual descriptor.

`Border`

Type Float

Default value 7.0

Unit Bohr

Description Set the extent of the Cartesian grid. Extent is the distance between a face of the grid's bounding box and the most outlying atom in the corresponding direction. If the [AtomsToDo] key is present, the bounding box is created around the specified atoms.

`Display`

Type Float

Default value 0.005

Description Domains for which the integrated DD value is smaller (in magnitude) than the specified value are omitted from the printed output.

`Enabled`

Type Bool

Default value No

GUI name Properties of reactivity domains

Description Calculate properties of reactivity domains.

`Ensemble`

Type Multiple Choice

Default value Canonical

Options [Canonical, GrandCanonical]

Description Statistical ensemble for DD domains. Canonical: DD values are calculated using the statistical canonical ensemble. GrandCanonical: DD values are calculated using the statistical grand canonical ensemble. The grand canonical DD corresponds to (S^2 f(2) - (gamma/eta^3) f^0), where f(2) is the canonical DD, gamma and eta - the hyper-hardness and hardness of the chemical system, respectively, and f^0 is the mean Fukui function. This statistical ensemble is a natural choice when comparing two chemical systems with a different number of electrons.

`Radius`

Type Float

Default value 0.0

Description This option adds a sphere around each nucleus, excluding all points inside it. This can help to separate domains around an atom or to exclude core electrons. Be careful when using this option. In particular, the radius of the sphere should exceed two or three times the [Spacing] value to be effective. By default, no spheres are added.

`Spacing`

Type Float

Default value 0.1

Unit Bohr

Description Specifies spacing (distance between neighboring points) of the rectangular Cartesian grid used when searching for DD domains. It may be useful to specify a smaller value (or increase the size of the grid, see [Border] key) if a substantial part of the electronic density is accounted for.

Threshold

Type Float

Default value 0.001

Description Arbitrary value of dual descriptor used to separate DD domains (values below this threshold are ignored).

9.4.7 adf2damqt: DAMQT interface

Interface (adf2damqt) to the 3rd party DAMQT 2.0 package[35], which can analyze the electron density in molecules, and related quantities. DAMQT includes the method of deformed atoms in molecules (DAM). The DAMQT 2.0 package is described in Ref.[7]. In the ADF calculation TAPE15 and adf.rkf need to be saved. The executable adf2damqt can be run with up to three optional arguments. If no argument is supplied, "ADF" is chosen as default root name (fname) for files generated by the interface, and files containing electron density matrix (fname.den) and molecular orbitals (fname.SLorba and, eventually, fname.SLorbb) will be created in a format suitable to be read by DAMQT.

```
$AMSBIN/ams << eor
...
Engine ADF
   ...
   SAVE TAPE15
EndEngine
eor
mv ams.results/adf.rkf TAPE21
mv ams.results/TAPE15 TAPE15
$AMSBIN/adf2damqt {fname {SPIN} {NOORBITALS}}
```

fname If a specific name is desired for the files (fname), it must be supplied as first optional argument, provided that the name does not coincide with any of the two additional options SPIN and NOORBITALS.

SPIN SPIN: for storing spin density matrix in fname.den file (instead of total electron density, which is the default).

NOORBITALS NOORBITALS: to prevent generation of files with molecular orbitals (by default orbitals are generated).

SPIN and NOORBITALS are case insensitive and can be given in any order (but always after optional fname when required).

---

[35] R. López, J.F. Rico, G. Ramírez, I. Ema, D. Zorrilla, *DAMQT 2.0: A new version of the DAMQT package for the analysis of electron density in molecules*, Computer Physics Communications 192, 289 (2015) (https://doi.org/10.1016/j.cpc.2015.02.027)

9.4.8 FOD: fractional orbital density

Following the analysis method by Grimme and Hansen (Ref.[36]) a fractional occupation number weighted electron density (FOD) can be plotted with AMSview. The scalar field is obtained by finite-temperature DFT calculations with predefined electronic temperature T_{el}, typically T_{el} =5000 K, and for hybrids T_{el} =20000*(amount of HF exchange)+5000, see Ref.[?]. This analysis tool can be used to determine whether static electron-correlation effects are important.

```
OCCUPATIONS ELECTRONICTEMPERATURE=5000
```

The NFOD (integrated FOD) is written in the output of ADF. The FOD can be visualized with AMSview. The contour surface value in AMSview should be set to 0.005 ($e/(bohr)^3$).

9.5 Controlling printed Output

The amount of printed output is regulated with the keys Print, NoPrint, EPrint and Debug. (No)Print and Debug are simple keys, EPrint is a block type key.

Many print options pertain to debugging situations and are included here only for completeness. This section is intended to give a survey of all possibilities. Some items may be mentioned again in other sections where the subject of a particular print switch is discussed.

9.5.1 Print / NoPrint

```
PRINT Argumentlist
Print Argumentlist
NoPrint Argumentlist
```

`Argumentlist` A sequence of names separated by blanks or commas. The keys Print and NoPrint may occur any number of times in the input file. The names in the argument list may refer to various items. For some of them printing is normally on, and you can turn them off with NoPrint. For others the default is not printing; use Print to override that.

Follows a list of the recognized items that are applicable in the argument lists, with a short explanation and defaults. Item names must be used exactly as given in the table - abbreviated or elongated forms will not be recognized - but they are not case sensitive.

Items

`Atdist` Default: No

Inter-atomic distance matrix at each new geometry (in an optimization)

`Bas` Default: Yes

General control of output related to elementary basis functions (bas).

`Character-Table` Default: No

Table of characters for the irreducible representations of the point group symmetry.

`Computation` Default: Yes

Reports progress of the computation, with (concise) info about each SCF cycle.

---

[36] S. Grimme and A. Hansen, *A Practicable Real-Space Measure and Visualization of Static Electron-Correlation Effects*, Angewandte Chemie International Edition 54, 12308 (2015) (https://doi.org/10.1002/anie.201501887)

`Core` Default: No

Description of the frozen core: frozen core expansion functions (corbas) and the expansion coefficients for the frozen orbitals. This printing can only be activated if Functions is also *on*, otherwise it is ignored.

`CoreOrt` Default: No

The valence basis set contains auxiliary Core Functions. They are not degrees of freedom but are used solely to ensure orthogonalization of the valence set to the frozen Core Orbitals. The orthogonalization coefficients and some related overlap matrices are printed.

`CoreTable` Default: No

Internally the charge density and potential of the atomic frozen cores are processed as tables with values for a sequence of radial distances. A few initial and a few final values from these tables are printed, along with the (radial) integral of the core density, which should yield the number of core electrons.

`EKin` Default: No

At the end of SCF: Kinetic energy of each occupied MO.

`EPauli` Default: Yes

The repulsive Pauli term in the bonding energy (also called exchange repulsion) with its decomposition in density functional (lda and nl) and Coulomb terms.

`Fit` Default: Yes

General control of output related to the density fitting.

`Fmat` Default: No

Fock matrix computed at each cycle of the SCF.

`FmatSFO` Default: No

Fock matrix (and overlap matrix) in the basis of symmetrized fragment orbitals (SFOs). This option requires the FULLFOCK and ALLPOINTS keyword to be present in the input. The matrix is printed only at the last SCF cycle. Use 1 iteration in the SCF for the Fock matrix at the first SCF cycle.

`Frag` Default: No

General control of output related to build-molecule-from-fragments.

`Functions` Default: Yes

List of employed Slater-type exponential basis functions and fit functions.

`Group-Operators` Default: No

3*3 matrices of point group symmetry operators, with the axis and angle of rotation

`Irrep-Matrices` Default No

Irreducible representation matrices

`Logfile` Default: Yes

At the end of the calculation a copy of the log file is appended to standard output

`low` Default: No

Construction of the LOW basis from the elementary BAS functions and from the SFOs: combination coefficients

`lowMO` Default: No

MOs are printed in the LOW (Lowdin) representation, in the RESULTS section

`OvlBAS` Default: No

overlap matrices processed during the construction of the LOW basis. Only printed in case OLDORTHON is used in input.

`Pmat` Default: No

The density matrix (in Lowdin representation) in each cycle of the SCF.

`QMpot` Default: Yes

At the end of the SCF for each atom the electrostatic potential at its nucleus (excluding its own contribution of course).

`SCF` Default: Yes

Controls the information about progress of the SCF procedure. Applies only if the print switch computation is on.

`sdiis` Default: No

Expansion coefficients applied by the DIIS procedure during the SCF.

`sdiismat` Default: No

Turns on sdiis(see above) *and* prints the *error vector* constructed by the DIIS routine (this is the commutator of the Fock matrix and the Density matrix). This is used to determine the DIIS expansion coefficients and to assess convergence.

`SFO` Default: depends on system size

General control of SFO-related output (if SFO subkey of key EPRINT is used). If turned off, (almost) all such output is suppressed. If on, such printing is controlled by the eprint subkey SFO. The default depends on the system size: if the number of primitive STOs < 1000, the default is Yes, else No.

`SFOSiteEnergies` Default: No

The Site energy of a SFO is defined as the diagonal Fock matrix element of the Fock matrix of the full complex in SFO representation.

`Smat` Default: No

Overlap matrix of BAS functions.

`Smearq` Default: No

Smear parameter - if and when applied - used in the determination of electronic occupation numbers for the MOs, with details of how it works out at every cycle of the SCF. For debugging purposes.

`SpinOrbit` Default: No

detailed information about how double-group symmetry representations are related to the single group representations

`Tails` Default: No

In each block of integration points (see Blocks) the evaluation of (Slater-type) exponential functions (basis, fit) is skipped when the function has become negligible for all points in that block due to the distance of those points from the atom where the function is centered. The relative savings due to this distance screening is printed at the first geometry cycle (use debug for printing at all cycles).

`TechPar` Default: Yes

Technical parameters such as maximum vector length in vectorized numerical integration loops, SCF parameters.

Arguments for the keys PRINT and NOPRINT.

For print switches Frag, Fit, Repeat, SCF, SFO, TF, see the key EPRINT below.

9.5.2 Debug

The key DEBUG is used to generate extensive output that is usually only relevant for debugging purposes. It operates exactly like the PRINT key but there is no converse: nodebug is not recognized; it would be irrelevant anyway because by default all debug print switches are off.

A list of the possible items for the DEBUG key is given below.

All items of the print list can also be used with the debug key. If they are not mentioned in table III, the meaning is the same as for the print key, but the corresponding output may be generated more often, for instance at every SCF cycle rather than at the last one only.

Table 9.1: Table III. Arguments for the print key DEBUG. All debug switches are by default off.

Item	Explanation
Basis	Construction of the orthonormal LOW basis from elementary (BAS) and fragment (FO) basis.
Core	Core Orthogonalization procedure
Ekin	Kinetic energy matrices. (compare the print switches EKIN)
Fit	Construction of the symmetry adapted fit functions
Fitint	Construction of integrals used in the Fit procedure.
Gradients	The gradients split out in parts.
Pmat	P-matrix (density matrix) during SCF
	and in the ETS analysis program in the BAS representation.
Rhofih	Computation of fit coefficients during the SCF.
SCF	Extensive output during the SCF procedure about many different items.
	See also EPRINT, subkey SCF.
SDIIS	All data concerning the DIIS as used during the SCF.
TransitionField	The Transition State procedure to compute and analyze certain terms in the bonding energy.
	The distinct components, the involved transition field Fock matrices, etc.

9.5.3 Eprint

The key EPRINT is an extended version of the (no)print key, employed for print switches that require more specification than just off or on.

Contrary to what is the case for the keys print and noprint, the key EPRINT must occur only once in the input file; any subsequent occurrences are incorrect and ignored or lead to abort.

```
EPRINT
  subkey
  subkey
  ...
end
```

`subkey` A subkey-type structure: it consists of a keyword followed by data, so that it functions as a simple (sub)key, *or* it is a keyword followed by a data *block* which must then end with the word subend.

The subkeys used in the EPRINT data block are called Eprint keys. A complete list of them is given below. All available EPRINT keys are discussed in the schemes below. The enclosing records EPRINT and end are omitted in these schemes.

Table 9.2: Table IV. List of EPRINT subkeys.

EPRINT subkeys	**Subject**
AtomPop (page 285)	Mulliken population analysis on a per-atom basis
BASPop (page 282)	Mulliken population analysis on a per-bas-function basis
Eigval (page 282)	One-electron orbital energies
Fit (page 278)	Fit functions and fit coefficients
Frag (page 279)	Building of the molecule from fragments.
FragPop (page 282)	Mulliken population analysis on a per fragment basis
OrbPop (page 279)	(Mulliken type) population analysis for individual MOs
OrbPopEr (page 279)	Energy Range (ER) in hartree units for the OrbPop subkey
Repeat (page 280)	repetition of output in Geometry iterations (SCF, optimization, …)
SCF (page 280)	Self Consistent Field procedure
SFO (page 281)	Information related to the Symmetrized Fragment Orbitals and the analysis
TF (page 281)	Transition Field method.

9.5.4 Eprint subkeys vs. Print switches

Several EPRINT subkeys are merely shortcuts for normal (no)print switches. All such simple subkeys are used in the following way:

```
EPRINT
  ESUBKEY argumentlist
END
```

Esubkey One of the following EPRINT subkeys: Fit, Frag, Repeat, SCF, sdiis, SFO, TF.

argumentlist A sequence of names, separated by delimiters. Each of these names will be concatenated with the esubkey and the combination will be stored as a normal print switch. Example: Frag rot, SFO will be concatenated to fragrot and fragsfo and both will be stored as print switches. All such combinations can also be specified directly with the key PRINT. The example is therefore exactly equivalent with the input specification: print FragRot, Fragsfo****

If any of the names starts with the two characters no, the *remainder* of the name will be concatenated with the EPRINT, but now the result will be stored and treated as a noprint switch. Items that are on by default can in this way be turned off. Example:

```
EPRINT
  FRAG noRot Eig
END
```

This turns Rot *off* and Eig *on* for the EPRINT subkey Frag. Equivalent would be:

```
NOPRINT FragRot
Print FragEig
```

Follows a description of all simple EPrint subkeys:

Fit

The subkey fit controls output of how the elementary fit functions are combined into the symmetric (A1) fit functions. It controls also printing of the initial (start-up) and the final (SCF) fit coefficients.

```
EPRINT
  FIT list
END
```

`list` A list of items, separated by blanks or commas. The following items are recognized: Charge, Coef, Comb.

`Charge` The amount of electronic charge contained in the fit (start-up), total and per fragment.

`Coef` The fit coefficients that give the expansion of the charge density in the elementary fit functions.

`Comb` The construction of the totally symmetric (A1) fit function combinations from the elementary fit functions.

By default all options are off.

Frag

The subkey frag controls output of how the molecule is built up from its fragments.

```
EPRINT
  FRAG list
END
```

`list` A list of items, separated by blanks or commas. The following items are recognized: Eig, Fit, Rot, SFO.

`Eig` The expansion coefficients in elementary functions (bas) of the fragment Molecular Orbitals as they are on the fragment file.

`Rot` The rotation (and translation) required to map the master fragment (i.e. the geometrical data on the fragment file) onto the actual fragment which is part of the current molecule. N.B.: if eig and rot are both *on*, the rotated fragment orbitals are printed also.

`Fit` The fit coefficients that describe the fitted charge density of the fragments after the rotation from the *master* fragment on file to the actual fragment. These are the molecular fit coefficients that are used (by default) to construct the total molecular start-up (fitted) charge density and hence the initial Coulomb and XC potential derived from it.

`SFO` The Symmetry-adapted combinations of Fragment Orbitals that are used in the current calculation. This feature ensures that the definition of the SFOs is printed. This will happen anyway whenever the EPRINT subkey SFO itself is activated. By default all options are off. Remark: SFO analysis in a Spin-Orbit relativistic calculation is implemented only in the case there is one scalar relativistic fragment, which is the whole molecule.

OrbPop

Specifies that (Mulliken type) population analysis should be printed for individual MOs, both on a per-SFO basis and on a per-bas function basis. The format of the subkey is as follows:

```
EPRINT
  ORBPOP TOL=X Nocc Nunocc
  SUBEND
END
```

X is the threshold for the SFO coefficient value to include in the listing for the per-SFO analysis. Nocc is the number of the highest occupied and Nunocc is the number of the lowest unoccupied orbitals to analyze.

OrbPopEr

Specifies the energy range for the MOs to which the OrbPop key applies. The default range is from -0.7 below the HOMO to 0.2 Hartree above the LUMO. Usage:

```
EPRINT
  OrbPopER minEn maxEn
END
```

where minEn and maxEn are both in Hartree, and have the defaults just specified. In order to get information on many more orbitals, simply specify a large negative value for minen and a large positive value to maxen.

Repeat

Control the repetition of output in Geometry iterations: optimization, computation of frequencies, transition state search.

```
EPRINT
  Repeat list
END
```

list contains one or more of the following items: NumInt, SCF.

NumInt Output from the numerical integration procedure, like parameters, numbers of points generated, test data is controlled by the *numint* subkey (see below). The *repeat* subkey controls whether the output is repeated for all geometries (if the flag is on) or only for the first (if the flag is off). Some concise info is produced (repeatedly) anyway if the print switch computation is on.

SCF Controls similarly the SCF output, like population analysis and orbital eigenvalues. If the flag is on, these items are printed at the last SCF cycle in every geometry, otherwise only at the last geometry.

By default both options are off.

SCF

Output during the SCF procedure.

```
EPRINT
  SCF list
END
```

list is a list of items, separated by blanks or commas. The following items are recognized: Eigval, Eigvec, Err, Fmat, Keeporb, MOPop, Occ, Pmat, Pop, Start.

Eigval Eigenvalues of the one-electron orbitals at the last SCF cycle. In a run with multiple SCF runs (Geometry Optimization,..) this printing occurs only for the last SCF procedure. See also the eigval subkey of EPRINT. (Use the *repeat* subkey of EPRINT to get output for the last SCF procedure at each SCF run, use DEBUG SCFEIGVAL to get output on all SCF cycles).

Eigvec MO eigenvector coefficients in the BAS representation. Only printed on the last SCF cycle.

Err SCF error data which are checked for convergence. By default this takes effect after cycle 25 of the SCF. If the key is set it takes effect at the first cycle. Optionally one may type ErrN,where n is an integer (written directly after Err without a blank in between), in which case the key takes effect at cycle n.

Fmat Fock matrix in the low representation.

Keeporb If the KeepOrbitals option is activated (see the key SCF), output is generated whenever this option actually results in a change of occupation numbers as regards the energy ordering.

Occ concise output of SCF occupation numbers on last SCF cycle if no eigenvalues are printed (see: Eigval).

moPop Mulliken populations in terms of the elementary basis functions (bas), per MO, for input-specified MOs (see the EPRINT subkey *orbpop*)

Pmat Density matrix

`Pop` General control of bas Mulliken populations. This supervises all printing (whether populations are printed or not) according to the EPRINT subkeys *atompop, fragpop, orbpop* (the latter only as regards the bas population analysis at the end of the SCF procedure).

`Start` Data pertaining to the *first* SCF cycle (of the *first* SCF procedure, in case of an optimization; use *repeat* to get this for *all* SCFs).

By default Eigval, Keeporb, Occ, and Pop are on, the others off.

SFO

Information pertaining to the use of Symmetrized Fragment Orbitals (for analysis purposes).

```
EPRINT
  SFO list
END
```

`list` A list of items, separated by blanks or commas. The following items are recognized: eig, eigcf, orbpop, grosspop, fragpop, ovl.

`Eig` The MO coefficients in terms of the SFOs.

`Eigcf` idem, but now also containing the coefficients pertaining to the CoreFunctions.

`OrbPop` population analysis of individual orbitals. The orbitals analyzed are set with the EPRINT subkey *orbpop*.

`GrossPop` Gross populations of the SFOs, split out in symmetry representations. GrossPop is automatically turned on when OrbPop is activated.

`FragPop` Population analysis on a per-FragmentType basis. This analysis does in fact not depend on the SFOs (ie, the result does not depend on how the SFOs are defined), but the computation of these populations takes place in the SFO-analysis module, which is why it is controlled by the SFO print option. FragPop output is given per orbital when OrbPop is activated, per symmetry representation when GrossPop is activated, and as a sum-over-all-orbitals-in-all-irreps otherwise (if FragPop is active).

`Ovl` Overlap matrix of the SFO basis, separately for each symmetry representation.

By default orbpop is on, the other options off.

In a Spin-Orbit calculation the SFO analysis is not yet implemented completely.

Remark: the options eig and eigcf replace the previous (now disabled) simple print options eigsfo and eigsfo.

Note that the simple print key SFO controls whether or not the EPRINT subkey *sfo* is effective at all.

TransitionField

Part of the bonding energy is computed and analyzed by the so-called Transition State procedure[2],[3]. This has nothing to do with physical transition states, but is related to the Fock operator defined by an average charge density, where the average is taken of the initial (sum-of-orthogonalized-fragments) and the final (SCF) charge density. There is also an analogous term where the average is taken of the sum-of-fragments and the sum-of-orthogonalized-fragments. Various terms, Fock operators and Density Matrices used in this approach may be printed. To avoid confusion with real Transition States (saddle points in the molecular Energy surface) the phrase TransitionField is used here.

```
EPRINT
  TF list
END
```

---

[2] T. Ziegler and A. Rauk, *On the calculation of Bonding Energies by the Hartree Fock Slater method. I. The Transition State Method*, Theoretica Chimica Acta 46, 1 (1977) (https://doi.org/10.1007/BF00551648)

[3] T. Ziegler and A. Rauk, *A theoretical study of the ethylene-metal bond in complexes between copper(1+), silver(1+), gold(1+), platinum(0) or platinum(2+) and ethylene, based on the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1558 (1979) (https://doi.org/10.1021/ic50196a034)

`List` A list of items, separated by blanks or commas. The following items are recognized: Energy, Fmat, DiagFmat, FragPmat, DiagFragPmat, F*dPmat, DiagF*dPmat, OrbE.

`Energy` Energy terms computed from the TransitionField.

`Fmat` TransitionField Fock matrices.

`DiagFmat` Idem, but only the diagonal elements.

`FragPmat` The molecular P-matrix constructed from the sum-of-fragments.

`DiagFragPmat` idem, but only the diagonal elements.

`F*dPmat` The TransitionField energy term can be expressed as a Fock operator times the difference between two P-matrices (initial and final density).

`DiagF*dPmat` only diagonal elements

`OrbE` Orbital energies in the TransitionField.

By default all options are off.

9.5.5 Other Eprint subkeys

We discuss now the remaining EPRINT sub keys that are not simple shortcuts for print switches.

```
EPRINT
  Eigval noccup {nvirtual}
END
```

This specifies the *number* of one-electron orbitals for which in the SCF procedure energies and occupation numbers are printed whenever such data is output: the highest noccup occupied orbitals and the lowest nvirtual empty orbitals. Default values are noccup=10, nvirtual=10. If only one integer is specified it is taken as the noccup value and nvirtual is assumed to retain its standard value (10). Printing can be turned off completely with the EPRINT sub key SCF, see above.

Mulliken Population Analysis

All population subkeys of EPRINT refer to *Mulliken* type populations.

```
EPRINT
  ATOMPOP level
END
```

Populations accumulated per atom.

level must be none, gross or matrix. none completely suppresses printing of the populations; gross yields the gross populations; matrix produces the complete matrix of net and overlap populations. Default value: matrix.

```
EPRINT
  BASPop level
END
```

Populations are printed per elementary (bas) basis function. The level options are none, short, gross, matrix. none, gross and matrix are as for atompop.

short yields a summary of BAS gross populations accumulated per angular momentum (*l*) value and per atom.

Default value: gross.

```
EPRINT
  FragPop level
END
```

Completely similar to the atompop case, but now the populations per *fragment*. Of course in the case of single-atom fragments this is the same as atompop and only one of them is printed. Default: matrix.

For all three population keys atompop, fragpop and baspop, specification of a higher level implies that the lower-level data, which are in general summaries of the more detailed higher level options, are also printed.

Printing of any populations at the end of the SCF procedure is controlled with the EPRINT sub key *SCF* (pop).

Population Analysis per MO

A very detailed population analysis tool is available: the populations *per orbital* (MO). The printed values are independent of the occupation numbers of the MOs, so they are not populations in a strict sense. The actual populations are obtained by multiplying the results with the orbital occupations.

The analysis is given in terms of the SFOs and provides a very useful characterization of the MOs at the end of the calculation, after any geometry optimization has finished. This feature is now also available in a Spin-Orbit coupled relativistic calculation, in the case there is one scalar relativistic fragment, which is the whole molecule.

The same analysis is optionally (see EPRINT subkey *SCF*, option mopop also provided in terms of the elementary basis functions (bas).

```
EPRINT
  OrbPop {noccup {nvirtual}} {tol=tol}
    subspecies orbitals
    subspecies orbitals
    ...
  subend
END
```

`noccup` Determines how many of the highest occupied orbitals are analyzed in each irrep. Default noccup=10.

`nvirtual` Determines in similar fashion how many of the lowest virtual orbitals are analyzed in each irrep. Default nvirtual=4.

`tol` Tolerance parameter. Output of SFO contributions smaller than this tolerance may be suppressed. Default: 1e-2.

`subspecies` One of the subspecies of the molecular symmetry group. Can not be used (yet) in a Spin-Orbit coupled calculation.

`orbitals` A list of integers denoting the valence orbitals (in energy ordering) in this subspecies that you want to analyze. This overrules the noccup,nvirtual specification for that symmetry representation. In an unrestricted calculation two sequences of integers must be supplied, separated by a double slash (//).

Any subset of the subspecies can be specified; it is not necessary to use all of them. No subspecies must occur more than once in the data block. This can not be used in a Spin-Orbit coupled equation (yet).

A total SFO gross populations analysis (from a summation over the occupied MOs) and an SFO population analysis per fragment type are preformed unless *all* MO SFO-populations are suppressed.

9.5.6 Reduction of output

One of the strong points of ADF is the analysis in terms of fragments and fragment orbitals (SFOs) that the program provides. This aspect causes a lot of output to be produced, in particular as regards information that pertains to the SFOs.

Furthermore, during the SCF and, if applicable, geometry optimizations, quite a bit of output is produced that has relevance merely to check progress of the computation and to understand the causes for failure when such might happen.

If you dislike the standard amount of output you may benefit from the following suggestions:

If you are not interested in info about progress of the computation:

```
NOPRINT Computation
```

If you'd like to suppress only the SCF-related part of the computational report:

```
NOPRINT SCF
```

If you don't want to see any SFO stuff:

```
NOPRINT SFO
```

To keep the SFO *definitions* (in an early part of output) but suppress the SFO-mo coefficients and the SFO overlap matrix:

```
EPRINT
 SFO noeig, noovl
END
```

Note: the SFO-overlap matrix is relevant only when you have the SFO-MO coefficients: the overlap info is needed then to interpret the bonding/anti-bonding nature of the various SFO components in an MO.

If you are not interested in the SFO *populations*:

```
EPRINT
 SFO noorbpop
END
```

9.6 Results on Output

The (standard) output file contains information of the main characteristics of the run, the SCF and geometry optimization results, bonding energy and population analyzes. Major parts of output can be regulated with print switches, see section *Controlling printed Output* (page 274).

By default the program produces quite a bit of output, for a large part related to (Mulliken-type) population analyzes of the molecule in total, as well as of individual orbitals, both in terms of the elementary basis functions and in terms of the SFOs, the symmetry-adapted Fragment Orbitals.

The fragment-oriented approach of ADF is very suitable for a thorough chemical analysis of molecular orbital properties and a conceptual representation of results. New users are advised to spend time and get familiar with the SFO-type analysis. It is an extremely more powerful tool to understand the electronic structure of the molecule than the classical atomic orbital populations.

The results for structure and reactivity, spectroscopic properties, transport properties that are printed are meant to be self-explanatory. See also the input options for each of these properties.

9.6.1 Electronic Configuration

The direct results from the SCF are the orbital energies and occupation numbers. This defines the electronic configuration: the occupation numbers and HOMO and LUMO energies for instance show whether or not the aufbau principle is satisfied in the final situation.

The energies of the Core Orbitals can be used to interpret for instance XPS (X-ray Photoelectron Spectroscopy) data: from Koopman's theorem these core orbital energies are an approximation to the core ionization energies. This neglects the effect of relaxation upon the ionization so that absolute energy values may not be very good; relative values, however, should be fair and can therefore be used to study (relative) chemical shifts.

9.6.2 Mulliken populations

Mulliken populations are based on the elementary atomic basis functions (bas). The individual BAS populations are printed together with summaries of the populations in all basis functions with the same angular moment quantum number on the same atom. A final summary is obtained by adding all functions on each atom, yielding the atom-atom populations. The atom-atom populations per l-value can be obtained if the key EXTENDEDPOPAN is included. The atomic gross charges are derived from the net and the overlap populations in the usual way. In addition, a population analysis may be given of individual MOs (by default this is suppressed). See the EPrint keys SCF (option mopop) and orbpop.

Mulliken-type populations are computed and printed at various levels of refinement (ranging from *per-basis function* to *per-fragment type*, data for the whole molecule as well as for individual MOs), and in two different representations, one based on the elementary basis functions (bas), the other on SFOs (Symmetrized Fragment Orbitals). This is potentially a very large amount of data. Precisely what is printed by default, and how this can be modified so as to suppress output or, alternatively, to get more information, is regulated by the print keys (print, eprint).

9.6.3 Hirshfeld charges, Voronoi deformation density

Mulliken populations can be summarized to yield atomic charges. Alternative methods exist to deduce atom charges from the self-consistent results of a molecular calculation. Several of those alternatives are provided by ADF: Hirshfeld analysis, Voronoi analysis, multipole derived charges, and charge model 5.

Of the methods applied in ADF to compute charges (Mulliken, Hirshfeld, Voronoi) we recommend the Hirshfeld analysis[2],[3] and the analysis based on Voronoi *deformation* density (VDD) charges[1],[4], see below. The fragments to which the Hirshfeld charges apply are enumerated in the early geometry part of the output file, where for each fragment the numbers of the atoms are given that belong to the fragment. The sum of the Hirshfeld charges may not add up to the analytical net total charge of the molecule. Any deviation from this is caused by numerical integration precision (small effect) and the neglect of long-distance terms that ADF uses to speed up the integral evaluations. This approximation does not affect very much the energy and molecular orbital properties, but it does show up in the sum-of-charges somewhat more. It does not indicate an error (unless the deviation is really large, say in the order of 1‰ of the total number of electrons).

The Hirshfeld analysis produces a charge value per fragment, computed as the integral of the SCF charge density over space, in each point weighted by the relative fraction of the (initial) density of that fragment in the total initial (sum-of-fragments) density:

$$Q^{\text{frag(i)}} = \int \rho^{\text{SCF}} \rho^{\text{initial frag(i)}} / \sum_j \rho^{\text{initial frag(j)}} \qquad (5.1.1)$$

---

[2] F.L. Hirshfeld, *Bonded-atom fragments for describing molecular charge densities*, Theoretica Chimica Acta 44, 129 (1977) (https://doi.org/10.1007/BF00549096)

[3] K.B. Wiberg and P.R. Rablen, *Comparison of atomic charges derived via different procedures*, Journal of Computational Chemistry 14, 1504 (1993) (https://doi.org/10.1002/jcc.540141213)

[1] G. te Velde, *Numerical integration and other methodological aspects of bandstructure calculations*, in *Chemistry*. 1990, Vrije Universiteit: Amsterdam.

[4] F.M. Bickelhaupt, N.J.R. van Eikema Hommes, C. Fonseca Guerra and E.J. Baerends, *The Carbon-Lithium Electron Pair Bond in* $(CH3Li)_n$ (n = 1, 2, 4), Organometallics 15, 2923 (1996) (https://doi.org/10.1021/om950966x)

The VDD method is based on the *deformation* density and a rigorous partitioning of space into non-overlapping atomic areas, the so-called Voronoi cells??[5]. The Voronoi cell of an atom *A* is the region in space closer to nucleus *A* than to any other nucleus (cf. Wigner-Seitz cells in crystals). The VDD charge of an atom *A* monitors the *flow* of charge into, or out of the atomic Voronoi cell as a result of 'turning on' the chemical interactions between the atoms. The VDD method summarizes the three-dimensional deformation density on a per-atom basis. It is conceptually simple and affords a transparent interpretation based on the plausible notion of charge redistribution due to chemical bonding, i.e. the gain or loss of charge in well-defined geometrical compartments of space. For the use of VDD in analyzes involving molecular fragments, see Ref.[6].

In the same fashion as for the Hirshfeld analysis, a summation over all atoms is given which should yield zero (for a neutral molecule). The deviation from zero is caused by numerical integration and by neglect-of-long-distance-terms; the same remarks apply as for the Hirshfeld analysis above.

The partitioning of space, using mid-way separation planes, is inappropriate to produce useful absolute numbers when neighboring atoms have very different sizes, for instance, Hydrogen and a heavy metal. However, *changes* in the density analyzed in this way do give a reasonable general insight in the effect of bonding on the location of charge densities, in particular because the Voronoi data per atom are split up in contributions within the atomic sphere and the rest of its Voronoi cell.

For symmetric molecules one can do a (partial) symmetry partitioning of the Voronoi deformation density if one includes

```
PRINT VDDANALYSIS
```

This analysis is especially useful if all fragments are calculated in the same symmetry as the full molecule, which is not always possible or useful, for example, symmetry NOSYM does not provide any extra information.

Hirshfeld and Voronoi charge analyzes are printed at the end of the SCF (of the last geometry, in case of an Optimization).

The Hirshfeld analysis in ADF produces charges *per fragment*, so that *atomic* charges are obtained only if single-atom fragments are used. This limitation does not apply to Voronoi charges (data per atom). Mulliken charges are given both per atom *and* per fragment.

In the printout of charges per fragment (as for the Hirshfeld analysis), you have to be aware of the *ordering* of fragments. A complete list of fragments is printed in the early GEOMETRY section of standard output, where you also find which atom(s) correspond(s) to which fragment. Note that even when you use single-atom fragments only, the order of fragments is usually quite different from the order of atoms in your input file. Typically (but not necessarily exactly in each case), when you use single-atom fragments: consider the first non-dummy atom in your ATOMS block. This defines the first atom *type*. Then browse the ATOMS list until you find an atom of a different type. This defines the second atom type, and so on. The single-atom fragment list will often be such that you first get *all* atoms of the first atom type, then all atoms of the second type, and so on. Check the printed list-of-fragments always, to avoid mistakes in assigning Hirshfeld charges to atoms (fragments).

---

[5] C. Fonseca Guerra, J.-W. Handgraaf, E. J. Baerends and F. M. Bickelhaupt, *Voronoi Deformation Density (VDD) charges. Assessment of the Mulliken, Bader, Hirshfeld, Weinhold and VDD methods for Charge Analysis*, Journal of Computational Chemistry 25, 189 (2004) (https://doi.org/10.1002/jcc.10351)

[6] C. Fonseca Guerra, F.M. Bickelhaupt, J.G. Snijders and E.J. Baerends, *The Nature of the Hydrogen Bond in DNA Base Pairs: The Role of Charge Transfer and Resonance Assistance*, Chemistry - A European Journal 5, 3581 (1999) (https://doi.org/10.1002/(SICI)1521-3765(19991203)5:12%3C3581::AID-CHEM3581%3E3.0.CO;2-Y)

9.6.4 Multipole derived charges

The multipole derived charges (MDC) analysis[7] uses the atomic multipoles (obtained from the fitted density) up to some level X, and reconstructs these multipoles exactly (up to level X) by distributing charges over all atoms. This is achieved by using Lagrange multipliers and a weight function to keep the multipoles local. Since the atomic multipoles are reconstructed up to level X, the molecular multipoles are represented also up to level X. The recommended level is to reconstruct up to quadrupole: MDC-q charges. The SCF should have converged for a meaningful MDC analysis.

9.6.5 Charge model 5

The charge model 5 (CM5)[8][9] uses the Hirshfeld analysis in combination with a parametrization to yield atomic charges that can accurately reproduce dipole moments obtained from experimental results. For input, use the keyword *CM5* (page 258).

9.6.6 Bond order analysis

The bond order analysis (see *Bond orders* (page 258)). produces the output in which the bond order values are printed for each pair of atoms for which the Nalewajski-Mrozek bond order value is larger than the threshold that can be specified with the keyword BONDORDER. For convenience the printed bond orders are accompanied by the corresponding interatomic distance. In the Nalewajski-Mrozek approach[11][12][16][17][18][19] the bond order indices b_{AB} are calculated based on the one- and two-center valence indices

$$b_{AB} = V_{AB} + W_A^{AB} V_A + W_B^{AB} V_B$$

with the weighting factors for one-center indices given by

$$W_X^{XY} = \frac{V_{XY}^{\mathrm{COV}}}{\sum_Z V_{XZ}^{\mathrm{COV}}}$$

Unlike other definitions of covalent bond orders, the Nalewajski-Mrozek valence indices comprise both, covalent and ionic contributions. There exist three alternative sets of the Nalewajski-Mrozek valence indices[?][?][?][?][?][?][10]. The bond order indices calculated from each set of the valence indices differ slightly due to arbitrariness in the way of splitting the one-center terms between bonds. More detailed description of alternative valence indices and their physical meaning is summarized in[?]; see also original papers[?][?][?][?][?].

---

[7] M. Swart, P.Th. van Duijnen and J.G. Snijders, *A charge analysis derived from an atomic multipole expansion*, Journal of Computational Chemistry 22, 79 (2001) (https://doi.org/10.1002/1096-987X(20010115)22:1%3C79::AID-JCC8%3E3.0.CO;2-B)

[8] A.V. Marenich, S.V. Jerome, C.J. Cramer, D.G. Truhlar, *Charge Model 5: An Extension of Hirshfeld Population Analysis for the Accurate Description of Molecular Interactions in Gaseous and Condensed Phases*, Journal of Chemical Theory and Computation 8, 527 (2012) (https://doi.org/10.1021/ct200866d)

[9] C.A. Peeples and G. Schreckenbach, *Implementation of the SM12 Solvation Model into ADF and Comparison with COSMO*, Journal of Chemical Theory and Computation 12, 4033 (2016) (https://doi.org/10.1021/acs.jctc.6b00410)

[11] A. Michalak, R.L. De Kock and T. Ziegler, *Bond Multiplicity in Transition-Metal Complexes: Applications of Two-Electron Valence Indices*, Journal of Physical Chemistry A 112, 7256 (2008) (https://doi.org/10.1021/jp800139g)

[12] M.S. Gopinathan and K. Jug, *Valency. I. A quantum chemical definition and properties*, Theoretica Chimica Acta 1983 63, 497 (1983) (https://doi.org/10.1007/BF00552652)

[16] R.F. Nalewajski and J. Mrozek, *Modified valence indices from the two-particle density matrix*, International Journal of Quantum Chemistry 51, 187 (1994) (https://doi.org/10.1002/qua.560510403)

[17] R.F. Nalewajski, J. Mrozek and A. Michalak, *Two-electron valence indices from the Kohn-Sham orbitals*, International Journal of Quantum Chemistry 61, 589 (1997) (https://doi.org/10.1002/(SICI)1097-461X(1997)61:3%3C589::AID-QUA28%3E3.0.CO;2-2)

[18] R.F. Nalewajski, J. Mrozek and A. Michalak, *Exploring Bonding Patterns of Molecular Systems Using Quantum Mechanical Bond Multiplicities*, Polish Journal of Chemistry 72, 1779 (1998) (http://ichf.edu.pl/pjch/pj-1998/pj07s98.htm#1779)

[19] R.F. Nalewajski, J. Mrozek and G. Mazur, *Quantum chemical valence indices from the one-determinantal difference approach*, Canadian Journal of Chemistry 74, 1121 (1996) (https://doi.org/10.1139/v96-126)

[10] I. Mayer, *Charge, bond order and valence in the ab initio SCF theory*, Chemical Physics Letters 97, 270 (1983) (https://doi.org/10.1016/0009-2614(83)80005-0)

By default the bond order indices based on the valence indices obtained from partitioning of $Tr(P\Delta P)$ are printed in the ADF output. Note that in this version the covalent two-center part (also printed in the output) is equal to the Gopinathan-Jug[?] bond order. The default values are:

$$
\begin{aligned}
V_A &= V_A^{\text{ion}} + V_A^{\text{COV}} \\
V_A^{\text{ion}} &= \sum_{a \in A} \left(P_{aa}^{\alpha} \Delta P_{aa}^{\alpha} + P_{aa}^{\beta} \Delta P_{aa}^{\beta} \right) \\
V_A^{\text{COV}} &= 2 \sum_{a \in A} \sum_{a' \in A, a < a`} \left(P_{aa'}^{\alpha} \Delta P_{a'a}^{\alpha} + P_{aa'}^{\beta} \Delta P_{a'a}^{\beta} \right) \\
V_{AB}^{\text{COV}} &= 2 \sum_{a \in A} \sum_{b \in B} \left(P_{ab}^{\alpha} \Delta P_{ba}^{\alpha} + P_{ab}^{\beta} \Delta P_{ba}^{\beta} \right)
\end{aligned}
$$

To produce the values from all alternative versions of Nalewajski-Mrozek valence indices, accompanied by the Gopinathan-Jug[?] and Mayer[?] bond orders, see the keyword BONDORDER.

The Mayer[?] bond orders can also be calculated (see *BondOrders%PrintAll* (page 259)). The two implementations of calculating the Mayer bond orders differ slightly if one uses frozen cores. They should agree exactly in all electron calculations.

9.6.7 Dipole moment, Quadrupole moment, Electrostatic potential

Dipole moment. Note that in a ion the value of the dipole moment depends on the choice of the origin, as follows from elementary electrostatic theory.

Quadrupole moment. Note that the value of the quadrupole moment often depends on the choice of the origin, as follows from elementary electrostatic theory.

Electrostatic potential at the nuclei: the Coulomb potential of the molecule at the nuclear positions, where the contribution from the nucleus itself is omitted.

9.6.8 Fragments and Basis Functions

`SFOs` SFOs: the Symmetry combinations of Fragment Orbitals. The SFOs are the basic conceptual entities for the analysis of MOs and other results. Note: The FO *coefficients* that expand the SFOs are normalized in the sense that they add up (squared) to unity. The resulting SFO *function* is not necessarily a normalized function. The FOs are normalized, so it depends on the *overlap* between the FOs what the self-overlap and hence the norm of the SFO is. Also printed are, for each subspecies in each irrep separately, the indices of the elementary basis functions from which the FOs and hence the SFOs are built up. (The overlap matrix of SFOs is printed much later, in the SFO Populations section after everything (SCF, Geometry) has cycled to convergence).

`Elementary basis functions (BAS)` First the lists of function *sets*, defined by radial behavior and the angular quantum number, are printed for all atom types on which the functions are centered. Thereafter follows the complete BAS list where the function sets have been expanded over all atoms (the *sets* are printed only for the atom *types*) and also over all Cartesian harmonics (6, not 5 *d*-functions, et cetera). In this printout the numbering can be found to which the SFO survey above refers.

9.6.9 MO analysis

`MOs expanded in SFOs` This gives a useful characterization of the character of the self-consistent molecular orbitals. Additional information is supplied by the SFO population analysis, see below. The definition of the SFOs in terms of the Fragment MOs has been given in a earlier part of output (section build). The SFO occupation numbers that applied in the fragments are printed. This allows a determination of the orbital interactions represented in a MO. Be aware that the bonding/antibonding nature of a SFO combination in a mo is determined by the relative signs of the coefficients *and* by the overlap of the SFOs. This overlap *may be negative!* Note also that SFOs are generally *not* normalized functions. The SFO overlap matrix is printed later, in the SFO-populations part below.

`SFO population analysis` For each irrep:

- Overlap matrix of the SFOs. Diagonal elements are not equal to 1.0 if the SFO is a linear combination of two or more Fragment Orbitals. The Fragment Orbitals themselves are normalized so the diagonal elements of the SFO overlap matrix give information about the overlap of the Fragment Orbitals that were combined to build the SFO.
- Populations on a per-fragment basis for a selected set of MOs (see EPrint, subkey *OrbPop*). This part is by default *not* printed, see EPRINT subkey *SFO*.
- SFO contributions per MO: populations for each of the selected MOs. In these data the MO occupation numbers are not included, so that also useful information about the virtual MOs is obtained. The printout is in matrix form, with the MOs as columns. In each printed matrix a row (corresponding to a particular SFO) is omitted if all populations of that SFO are very small in all of the MOs that are represented in that matrix. See eprint, subkey *orbpop*. Note that this method to define SFO populations (for orbitals) is very similar to the classical Mulliken type analysis, in particular regarding the aspect that *gross* populations are obtained as the diagonal (*net*) populations plus half of the related off-diagonal (overlap) populations. Occasionally this may result in negative (!) values for the population of certain SFOs, or in percentages higher than 100%. If you have such results and wonder if they can be right, work out one of the offending cases by hand, using the printed SFO overlap matrix and the printed expansion of the MOs in SFOs to compute 'by hand' the population matrix of the pertaining MO. To avoid doing large calculations it is usually sufficient to take only the few largest MO expansion coefficients; this should at least qualitatively give the correct outcomes.
- Total SFO gross populations in a symmetry representation: from a summation over all MOs (not only those analyzed in the previous section of output) in the symmetry representation under consideration. In the gross populations the MO occupation numbers have been included.
- (Per spin): A full list of all MOs (combining all symmetry representations), ordered by energy, with their most significant SFO populations. Since there might be several significant SFO populations for a particular MO, and an SFO may actually be a linear combination of several (symmetry-related) Fragment Orbitals, this table could get quite extensive. In order to confine each SFO population specification to one line of output, the SFOs are indicated by the characteristics of the first term (Fragment Orbital) of its expansion in Fragment Orbitals. So, if you see the SFO given as the '2 P:x on the first Carbon fragment', it may actually refer to the symmetry combination of, for instance, 2P:x and 2P:y orbitals on the first, second and third Carbon fragments. A full definition of all SFOs in terms of the constituting Fragment Orbitals is given in an early part of the output.

9.6.10 Bond energy analysis

The bond energy and its decomposition in conceptually useful terms: Pauli (exchange) repulsion, total steric repulsion, orbital interactions (partitioned into the contributions from the distinct irreducible representations), and corrections for some approximations (fitting and Transition State analysis procedure). For a discussion of bonding energy decompositions and applications see e.g. refs.[13][14][15][20][21][22][23][24][25][26].

The program prints the bonding energy (not in a Create or Frequencies run) and its decomposition in terms that are useful for chemical interpretation. The *total* energy is not computed. The bonding energy is defined relative to the fragments. When *basic atoms* are employed as fragments one should realize that these do not represent the atomic ground state since they are computed as spin-restricted and spherically symmetric objects, with possibly fractional occupation numbers. The correct multiplet state is not computed. To obtain the bonding energy with respect to isolated atoms you should therefore add atomic correction terms to account for spin polarization and the multiplet state. See also the SLATERDETERMINANTS key and the discussion on multiplet states.

The spin polarization energy can be computed by running the single atom unrestricted, using as fragment the corresponding (restricted) basic atom. The true multiplet state is not necessarily obtained in this way.

For the comparison of computed bonding energies with experimental data one should furthermore be aware of any aspects that are not represented in the computational formalism, such as zero-point motions and environment (solvent) effects.

In a Geometry Optimization or Transition State search, the program may print a bonding energy evaluation at each geometry (depending on print switches). A test-energy value is written in the log file. This is *not* the bonding energy, although the difference is usually small. The test-energy printed in the log file is the energy expression from which the energy gradients are computed. The true bonding energy contains in addition a few (small) correction terms that are mostly related to the fit incompleteness. These correction terms are usually very small.

If Electric Fields are used in the computation (homogeneous and/or point charges), the printed Bonding Energy is the energy of the molecule in the field minus the energy of the fragments in the same field. The energy terms due to the field are also printed separately so that one can subtract them from the total bonding energy to obtain the energy-change without field-terms.

---

[13] T. Ziegler and A. Rauk, *On the calculation of Bonding Energies by the Hartree Fock Slater method. I. The Transition State Method*, Theoretica Chimica Acta 46, 1 (1977) (https://doi.org/10.1007/BF00551648)

[14] T. Ziegler and A. Rauk, *A theoretical study of the ethylene-metal bond in complexes between copper(1+), silver(1+), gold(1+), platinum(0) or platinum(2+) and ethylene, based on the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1558 (1979) (https://doi.org/10.1021/ic50196a034)

[15] F.M. Bickelhaupt, N.M. Nibbering, E.M. van Wezenbeek and E.J. Baerends, *The Central Bond in the Three CN* Dimers NC_CN, CN-CN, and CN-NC: Electron Pair Bonding and Pauli Repulsion Effects*, Journal of Physical Chemistry 96, 4864 (1992) (https://doi.org/10.1021/j100191a027)

[20] K. Kitaura and K. Morokuma, *A new energy decomposition scheme for molecular interactions within the Hartree-Fock approximation*, International Journal of Quantum Chemistry 10, 325 (1976) (https://doi.org/10.1002/qua.560100211)

[21] T. Ziegler and A. Rauk, *Carbon monoxide, carbon monosulfide, molecular nitrogen, phosphorus trifluoride, and methyl isocyanide as sigma donors and pi acceptors. A theoretical study by the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1755 (1979) (https://doi.org/10.1021/ic50197a006)

[22] H. Fujimoto, J. Osamura and T. Minato, *Orbital interaction and chemical bonds. Exchange repulsion and rehybridization in chemical reactions*, Journal of the American Chemical Society 100, 2954 (1978) (https://doi.org/10.1021/ja00478a004)

[23] S. Wolfe, D.J. Mitchell and M.-H. Whangbo, *On the role of steric effects in the perturbational molecular orbital method of conformational analysis*, Journal of the American Chemical Society 100, 1936 (1978) (https://doi.org/10.1021/ja00474a055)

[24] A.J. Stone and R.W. Erskine, *Intermolecular self-consistent-field perturbation theory for organic reactions. I. Theory and implementation; nucleophilic attack on carbonyl compounds*, Journal of the American Chemical Society 102, 7185 (1980) (https://doi.org/10.1021/ja00544a003)

[25] F. Bernardi, A. Bottoni, A. Mangini and G. Tonachini, *Quantitative orbital analysis of ab initio SCF=MO computations : Part II. Conformational preferences in H2N—OH and H2N—SH*, Journal of Molecular Structure: THEOCHEM 86, 163 (1981) (https://doi.org/10.1016/0166-1280(81)85082-8)

[26] P.J. van den Hoek and E.J. Baerends, *Chemical bonding at metal-semiconductor interfaces*, Applied Surface Science 41/42, 236 (1989) (https://doi.org/10.1016/0169-4332(89)90063-9)

9.7 Densf: Volume Maps

densf is an auxiliary program to generate values of molecular orbitals, charge densities and potentials in a user-specified grid, to be used typically for plotting or graphical display. The TAPE41 result file can be used directly by the AMSview program to visualize these properties.

densf requires an ascii input file where the user specifies the grid and the items that he/she wishes to see calculated on the grid, plus the standard result file adf.rkf (TAPE21) from an *adf* calculation. * densf* writes a summary of the items that have been requested to standard output, together with some general information.

densf produces a (binary) KF file TAPE41, see OUTPUTFILE keyword below. TAPE41 is a KF file and all KF utilities can be used to inspect and process its data.

densf can also read and write cube files. See the CUBINPUT and CUBOUTPUT input options for details.

Examples of using *densf* are contained in the set of sample runs; see the *ADF Examples* (page 401).

9.7.1 Input

The input for *densf* is keyword oriented. The keywords may be specified in any order.

Below follows a list of the allowed keywords with their description.

```
$AMSBIN/densf << eor
ADFFILE {file}
OUTPUTFILE {file}
VTKFILE {file}
CUBINPUT {file}
CUBOUTPUT {file}
GRID ...
UNITS ...
Density ...
KinDens ...
Laplacian ...
DenGrad ...
DenHess ...
Potential ...
Orbitals ...
NOCV ...
NCI ...
SEDD
DualDescriptor
POLTDDFT ...
eor
```

Input/Output files

```
ADFFILE {file}
```

ADFFILE keyword specifies path to the adf.rkf file from which *densf* reads the input data. Absence of the keyword is treated as if **ADFFILE TAPE21** has been specified.

```
OUTPUTFILE {file}
```

OUTPUTFILE keyword specifies path to the (possibly existing) TAPE41 file. If the file exists, *densf* will read grid specifications from it ignoring GRID keyword in the input. Computed quantities are saved in the file overwriting existing data with the same name, if any.

```
VTKFILE {file}
```

VTKFILE keyword specifies path to a file in the format readable by VTK directly. This option exists primarily for better integration with ADF-GUI and the user should not specify it.

```
CUBINPUT {file}
```

If the CUBINPUT keyword is present then the grid as specified in the **file** is used to calculate all requested quantities. Any volume data found in the cube file is also saved in the output file. NOTE: CUBINPUT option cannot be used with a pre-existing TAPE41 file because they both specify the grid, which may lead to a conflict.

```
CUBOUTPUT {file}
```

Presence of the CUBOUTPUT keyword tells densf to save all computed quantities as cube files using **file** as filename prefix. The prefix can also contain a complete path including directories. For example, specifying the following in the densf input

```
CUBOUTPUT /home/myhome/H2O
Density SCF
```

will result in a file /home/myhome/H2O%SCF%Density.cub being created containing volume data for the total SCF density. One file per requested quantity is created.

The OUTPUTFILE, CUBOUTPUT and VTKFILE options are mutually exclusive. Absence of any of these options is treated as if **OUTPUTFILE TAPE41** has been specified.

Grid

The grid can be specified using the Grid block key.

In its simplest form the block can be written as follows:

```
GRID {save} {coarse|medium|fine}
END
EXTEND grid_extent
```

If the word *save* is specified, the program will store all grid points on TAPE41 (in addition to the specification of the grid that is always stored). The default is NOT to store all the grid points.

Either coarse, medium or fine may be specified. This instructs the program to generate the grid automatically within a box enclosing all atoms of the molecule. The distance between grid points is 0.5, 0.2 or 0.1 bohr for the coarse, medium and fine grid, respectively. Obviously, the size of the result file TAPE41 depends strongly on this specification. By default a coarse grid is generated. The size of the grid box is determined by the atomic coordinates and the grid extent. The latter is by default equal to 4.0 Bohr and can be changed using the EXTEND keyword, in which case it is affected by the unit_of_length value specified in the UNITS block (see below).

For more detailed grid specification the following Grid block can be used:

```
Grid {save}
 x0 y0 z0
 n1 [n2 [n3]]
 v1x v1y v1z length1
 [v2x v2y v2z length2]
```

(continues on next page)

(continued from previous page)

```
 [v3x v3y v3z length3]
END
```

The rows of the Grid block are specified as follows:

- Row 1: three coordinates for the 'origin' (one of the corners) of the grid.
- Row 2: up to three integers specifying the numbers of points in each direction. The number of integers defines the dimensionality of the grid.
- Rows 3 to 5: up to three rows, one per grid dimension. Each row contains components of the direction vector and the length of the grid in this dimension. The absolute magnitude of the vector components is irrelevant as it will be scaled to fit the given number of points into the grid length in this dimension. The unit of length for the grid size is by default Angstrom. The default can be overridden by using the input key UNITS, see below.

Notes:

- The second row ('three integers…') specifies the number of grid *points* in the different directions. The corresponding number of steps or intervals is one less!
- If the TAPE41 result file is to be used by AMSview, the grid must be a 3-dimensional orthogonal one, with the same step size for all three dimensions.
- If the output TAPE41 file already exists and it contains valid grid data or if CUBINPUT is specified then the GRID input is ignored.
- The unit of length used in the input file has no relation to how the data are stored on the result file and how the program processes the data internally. Internal processing and storage on file is in bohr (atomic units).

Inline Grid

DENSF can read grid as list of points. When specifying inline grid the GRID keyword should look as follows:

```
Grid Inline
  x1 y1 z1
  x2 y2 z2
  ...
  xN yN zN
End
```

Here, x#, y#, and z# are coordinates of points at which requested properties will be calculated. This feature may be used, for example, by external programs to calculate various properties at a number of points exactly and avoid interpolation with its inaccuracy. This feature should be used only when the output file has a TAPE41 format.

Units

The unit of length can be set with the Units block:

```
UNITS
 Length unit_of_length
END
```

The unit-of-length will apply to the grid specification in the input file. Default is angstrom.

Density

Generates the charge density in the grid. It is a simple keyword (not block-type).

```
density {fit} {frag} {ortho} {scf} {trans}
```

Occurrence of the word fit specifies that all densities specified in this record will be computed from the fit functions (an approximation to the exact density), rather than from the occupied molecular orbitals.

frag, ortho, scf, and trans causes each of the corresponding densities to be computed. frag stands for the sum-of-fragments (i.e. the initial) density, scf for the final result of the *adf* calculation, ortho for the orthogonalized fragments (orthogonalization to account for the Pauli repulsion, see the ADF User's Guide), and trans for excitation transition density.

Transition density is a product of initial and final states of an excitation. In the simplest case when initial and final states consist of one molecular orbital each, in this case the corresponding transition density is a product of the two MOs. To obtain transition densities one needs to perform an excitations calculation with ADF, see EXCITATIONS keyword in ADF User's Guide. Transition densities for all excitations found in the input adf.rkf (TAPE21) file will be calculated. The transition densities are always fit-densities.

If both the exact and the fit-densities are required the density keyword must be repeated, once with and once without the fit option specified.

The default (when the DENSITY key does not occur in the input file) is to calculate the final SCF density and the sum-of-fragments density.

The frozen core density is calculated with:

```
density core
```

Kinetic Energy Density and Electron Localization Function (ELF)

```
KinDens {frag} {orth} {scf}
```

Generates the Kinetic energy density and electron localization function on the grid.

Occurrence of any of the words requests calculation of the two quantities (KinDens and ELF) based on the corresponding density: sum-of-fragments, orthogonalized fragments, or SCF, respectively. If none of the options is present, scf is assumed.

Laplacian of the Density

The Laplacian of the exact SCF density is calculated with:

```
Laplacian
```

The Laplacian of the fitted SCF density is calculated with:

```
Laplacian fit
```

The LAPLACIAN key can occur multiple times. The LAPLACIAN feature is also supported by AMSview.

Gradient of the Density

The gradient of the exact SCF density is calculated with:

```
DenGrad
```

The gradient of the fitted SCF density is calculated with:

```
DenGrad fit
```

The gradient of the frozen core density is calculated with:

```
DenGrad core
```

The DENGRAD key can occur multiple times. This feature should be used only when the output file has a TAPE41 format.

Hessian of the Density

The Hessian of the exact SCF density is calculated with:

```
DenHess
```

The Hessian of the fitted SCF density is calculated with:

```
DenHess fit
```

The Hessian of the frozen core density is calculated with:

```
DenHess core
```

The DENHESS key can occur multiple times. This feature should be used only when the output file has a TAPE41 format.

Potential

Generates the coulomb and/or exchange-correlation potential in the grid.

```
potential {coul / XC} {frag} {ortho} {scf}
```

frag, ortho, and scf are as for the density: at least one must be specified.

coul and XC specify that the Coulomb potential, respectively the exchange-correlation potential must be computed. Precisely one of these options must be specified in the record. If both potential types are required, another input record with the potential key must be used.

In the present release the xc option is not yet operational.

The default (when the POTENTIAL key does not occur in the input) is to calculate the SCF Coulomb potential.

Orbitals

A block type key in which the required molecular orbitals are specified. The key can be repeated in input any number of times; all occurrences are read and applied.

```
Orbitals {GRAD} type
 (data)
END
```

The argument of the orbitals key (type) must be scf (for the scf orbitals) or loc (for the localized molecular orbitals, see the ADF User's Guide) or generic (see separate section).

The SCF type can have an optional GRAD specifier which requests calculation of the MO gradients.

In many data records in the ORBITALS block, as noted in the description of these data records, you may specify a HOMOLUMO range.

A HOMOLUMO range is the following:

```
{HOMO{{-}n}} {LUMO{{+}n}}
```

HOMO: the highest occupied orbital HOMO-n, with n an integer: the highest (n+1) occupied orbitals LUMO: the lowest virtual orbital LUMO+n, with n an integer: the lowest (n+1) virtual orbitals.

The HOMO part, or the LUMO part, or both must be specified. The integer n with sign is always optional, and the sign is always optional (and has no meaning, it is intended to enhance readability).

Thus, as an example,

```
HOMO-1 LUMO+1
```

means a range of 4 orbitals: the two highest occupied ones, and the two lowest virtuals.

Each data record in the orbitals block must have either of the following formats:

the word alpha or beta. This specifies that subsequent records refer to spin-alpha or spin-beta orbitals respectively. In a restricted calculation this has no meaning and beta must not be specified. alpha and/or beta may occur any number of times in the orbitals block. All records until the first occurrence of alpha or beta are assumed to refer to spin-alpha orbitals.

label n1, n2, n3, … label is one of the subspecies of the point group symmetry used in the adf calculation and n1 etc. are indices of the molecular orbitals (in that subspecies) that are to be computed. This format is meaningless and must not be used for the loc orbitals type, because localized orbitals do not (necessarily) belong anymore to a particular symmetry

label HOMOLUMO label is one of the subspecies of the point group symmetry used in the calculation, the orbitals follow from the HOMOLUMO range.

label occ or label virt occ specifies all orbitals (in that symmetry representation) up to and including the highest occupied one. virt specifies all orbitals above the highest occupied one. In this context partially occupied orbitals are considered occupied. Note carefully that if in a particular symmetry representation an empty orbital is computed below the highest occupied one in that same representation (excited state), that particular empty one is included in the list of occ. Again, this format is meaningless and must therefore not be used for the loc type of orbitals.

all occ or all virt or all HOMOLUMO Specifies for each symmetry representation:

- all orbitals up to and including the highest occupied one (in that symmetry), or
- all orbitals above the highest occupied one, or
- all orbitals defined by the HOMOLUMO range.

This form is not to be used for the LOC type of orbitals. However, using this for LOC will not result in an error but will be interpreted as identical to the following format.

all This format must be used only for the LOC type of orbitals and simply means: all computed localized orbitals (irrespective of occupation numbers).

n1, n2, … a simple list of integer indices. This format must be used only for the loc type of orbitals since no reference is made to any symmetry representation. The indices refer of course to the list of localized orbitals as computed by ADF.

The default value used when the ORBITALS key is not present is:

```
Orbitals SCF
 All HOMO-1 LUMO+1
End
```

NOCV

In ADF2009.01 it is possible to use DENSF to calculate $\epsilon^* \phi^2$ values of Natural Orbitals for Chemical Valence (NOCVs). Additional information on NOCVs is available in Ref.[1].

The relevant part of the DENSF input is as follows:

For spin-unrestricted:

```
NOCV
  Alpha
   N1:math:`\alpha`
   N2:math:`\alpha`
   ...
  Beta
   N1:math:`\beta`
   N2:math:`\beta`
   ...
END
```

For spin-restricted:

```
NOCV
   N1
   N2
   ...
END
```

N1, N2, etc. specify sequential numbers of the orbitals for which $\epsilon^* \phi^2$ is to be calculated.

Alpha and *Beta* specify that the numbers that follow refer to spin α and β, respectively. Both *Alpha* and *Beta* are optional, *Alpha* being assumed if omitted. The NOCV input block must be closed with “END”.

Alternatively, one can specify to calculate all (alpha- or beta-) NOCV’s:

For spin-unrestricted:

```
NOCV
  Alpha
   ALL
```

(continues on next page)

---

[1] A. Michalak, M. Mitoraj, and T. Ziegler, *Bond Orbitals from Chemical Valence Theory*, Journal of Physical Chemistry A 112, 1933 (2008) (https://doi.org/10.1021/jp075460u)

(continued from previous page)

```
  Beta
   ALL
END
```

For both spin-restricted and spin-unrestricted:

```
NOCV
   ALL
END
```

The last and probably the most convenient form of the NOCV input blocks lets one to specify an NOCV eigenvalue threshold as a criterion for selecting orbitals:

For spin-unrestricted:

```
NOCV
  Alpha
   THRESH threshold
  Beta
   THRESH threshold
END
```

For both spin-restricted and spin-unrestricted:

```
NOCV
   THRESH threshold
END
```

When this form of the input is used, only those NOCVs will be included whose absolute eigenvalue is equal to or larger than the given *threshold*.

NOCV profile

It is possible to compute an integral profile of the NOCV $\epsilon^*\phi^2$ function along a given line. To calculate a single point of the profile the $\epsilon^*\phi^2$ is integrated in the plane orthogonal to the profile axis and passing through the point. In addition to the input above used for selecting NOCVs, the following input is required to calculate the profile:

```
NOCV
   PROFILE
END
LINE
   x1 y1 z1
   x2 y2 z2
   delta
END
```

The LINE input block is used to define the axis along which the profile will be generated. The (*x1,y1,z1*) point defines the origin of the profile axis and (*x2,y2,z2*) defines the second point of the line and the positive direction. The distance between the point does not matter. The *delta* value is the distance between two point in the profile. All values are given in the units specified in the UNITS input block. The positions of the left- and rightmost profile points depend on coordinates of the atoms and the grid extent value (specified by the EXTEND keyword, see above).

Generic orbitals

There is also a possibility to calculate any orbital as long as it is present in the t21 file in the BAS representation. The input syntax is as follows:

```
Orbitals GenBas
 section1%variable1
 section2%variable2
End
```

In the example above, each line contains the section and variable name of the orbital in the input t21 file. The length of the variable should be equal to the number of atomic functions (naos) and it is supposed to contain expansion coefficients of the orbital on the basis of atomic (primitive) functions.

The calculation results are stored in the output file in sections and variables with exactly the same names as specified in the input. The section and variable names may contain spaces although the leading and training spaces are discarded.

NCI

The areas of non-covalent interactions (NCI), see Refs.[4][5], can be recognized by the a low value of the electron density coupled with a low value of RDG (reduced density gradient s = 1/2 (3 π^2)$^{-1/3}$ $|\nabla\rho|$ $\rho^{-4/3}$) and a negative (or a small positive) value of the second eigenvalue of the Hessian of the electron density (λ_2). The regions of significant hydrogen bonding are recognized by strictly negative λ_2 while in the regions of VdW interactions it may be slightly positive. The relevant DENSF input keyword is:

```
NCI {BOTH|FIT} {RHOVDW=*RhoVdW*} {RDG=*Rdg*}
```

All arguments are optional

By default, the exact density is used to calculate the NCI properties. If FIT is specified then the fitted density is used to calculate the fields and their names are prepended with “Fit”. If BOTH is specified then the NCI properties are calculated using both exact and the fitted density. Again names of the fields calculated from the fitted density start with “Fit”

The remaining arguments set relevant thresholds (all in atomic units): *RhoVdW*: density threshold for detection of weak interaction regions (default 0.02); *Rdg*: threshold on the reduced density gradient value s (default 0.5). A point is considered for NCI only if s value is smaller than Rdg.

DENSF creates three variables per density type (exact or fitted) when NCI is present in the input: *SCF%RDG* (or *SCF%FitRDG*): the reduced density gradient value s; *SCF%DenSigned* (or *SCF%FitDenSigned*): the sign(λ_2) ρ value for regions where $s < Rdg$; *SCF%NCI* (or *SCF%FitNCI*): the NCI flag value, see below;

If the point is considered for NCI (that is if $s < Rdg$), the sign(λ_2) ρ value (or ρ) is tested against *RhoVdW*. If $\rho <$ *RhoVdW* then the NCI value is set to 1 to flag a VdW interaction region. If sign(λ_2) $\rho <$ -*RhoVdW* then the NCI value is set to -1 to flag a hydrogen bonding region. In all other cases the NCI value is zero.

AMSview visualization of NCI: use ‘Isosurface:with phase’ isosurface of property NCI (at 0.5). Alternative visualization of NCI: use colored isosurface with isosurface of property RDGforNCI (at 0.5) colored with NCI (range -0.5 to 0.5). The value of RDGforNCI is the same as RDG (reduced density gradient) except in regions where the density is larger than 0.05, in order to get rid of surfaces around atoms, that one would see if one is using an isosurface of RDG. One may need a medium or fine grid for best visualization.

---

[4] E.R. Johnson, S. Keinan, P. Mori-Sánchez, J. Contreras-García A.J. Cohen, and W. Yang, *Revealing Non-Covalent Interactions*, Journal of the AmericanChemical Society 132, 6498 (2010) (https://doi.org/10.1021/ja100936w)

[5] J. Contreras-García E.R. Johnson, S. Keinan, R. Chaudret, J-P. Piquemal, D.N. Beratan, and W. Yang, *NCIPLOT: A Program for Plotting Noncovalent Interaction Regions*, Journal of Chemical Theory and Computation 7, 625 (2011) (https://doi.org/10.1021/ct100641a)

SEDD

The single exponential decay detector (SEDD), see Ref.[2], extracts information about bonding and localization in atoms, molecules, or molecular assemblies. The practical evaluation of SEDD does not require any explicit information about the orbitals. The only quantity needed is the electron density (calculated or experimental) and its derivatives up to the second order. For the exact equation to be used, and pictures, see Ref.[?].

Specifying SEDD in the input also invokes calculation of the Density Overlap Regions Indicator (DORI)[3]. The DORI is best visualized as a 0.9 isosurface colored with the sign(λ_2) ρ value (see the DenSigned variable from the NCI keyword above). AMSview visualization of DORI: use colored isosurface with isosurface of property DORI (at 0.9) colored with DenSigned (range -0.02 to 0.02). One may need a medium or fine grid for best visualization.

DualDescriptor

Calculate the dual descriptor (DD) in the frontier molecular orbitals approximations. The DD is essentially a difference between (an average of) the square of the lowest unoccupied and the highest occupied molecular orbitals: $DD = \frac{1}{N_{LUMO}}(\sum_i^{N_{LUMO}} \psi_i^2) - \frac{1}{N_{HOMO}}(\sum_j^{N_{HOMO}} \psi_j^2)$, where i and j are indices of orbitals within 10^{-3} Hartree of the LUMO and HOMO, respectively. Please note that this method will produce incorrect results for molecules with fractional occupation numbers.

PolTDDFT

ADF can calculate the complex dynamical polarizability with *PolTDDFT* (page 196), which can be used to calculate the photoabsorption and CD spectra. With densf one can calculate for a given frequency the three components of the induced (first-order perturbed) density (relevant for polarizability), the three components of the transition (first-order perturbed) density (relevant for absorption), and the three components of the rotator strength density (relevant for CD), see Ref.[6].

```
POLTDDFT n
```

Here n is the n'th frequency in the list of requested frequencies.

9.7.2 Result: TAPE41

Follows a description of the contents of TAPE41. We start with a brief discussion of the sections. At the end you can find an uncommented list of all variables and sections. Note that some data are only generated when certain keywords are provided.

Sections on TAPE41

Grid

This is a general section. It contains the grid data and some more general info.

The grid characteristics are stored as:

- The 'origin' of the grid.
- The numbers of points in three independent directions.

---

[2] P. de Silva, J. Korchowiec, T.A. Wesolowski, *Revealing the Bonding Pattern from the Molecular Electron Density Using Single Exponential Decay Detector: An Orbital-Free Alternative to the Electron Localization Function*, ChemPhysChem 13, 3462 (2012) (https://doi.org/10.1002/cphc.201200500)

[3] P. de Silva and C. Corminboeuf, *Simultaneous Visualization of Covalent and Noncovalent Interactions Using Regions of Density Overlap*, Journal of Chemical Theory and Computation 10, 3745 (2014) (https://doi.org/10.1021/ct500490b)

[6] L. Chang, O. Baseggio, L. Sementa, D. Cheng, G. Fronzoni, D. Toffoli, E. Aprà, M. Stener, and A. Fortunelli, *Individual Component Map of Rotatory Strength and Rotatory Strength Density Plots As Analysis Tools of Circular Dichroism Spectra of Complex Systems*, Journal of Chemical Theory and Computation 14, 3703 (2018) (https://doi.org/10.1021/acs.jctc.8b00250)

- Three vectors, called 'x-vector', 'y-vector' and 'z-vector'. They are the *steps* in the three independent directions that define the grid.

If the save option was used in input (key grid) also all grid coordinates are stored: for each point three coordinates (xyz), also if only a 2-dimensional or 1-dimensional grid has been generated (a 2D grid does not necessarily lie in the xy-plane).

Note that the grid values are now stored in a simpler manner than in previous (prior to 2004) versions of densf, because the 'x values', 'y values', and 'z values' now each have their own, separate sections.

The remaining (general) data in this section comprises:

- The number of subspecies ('symmetries') for which data such as Molecular Orbitals may be present.
- The names of the subspecies.
- A logical with the name 'unrestricted', which flags whether the data pertain to an unrestricted calculation.
- The total number of grid points.

SumFrag

Contains grid data of the Sum-of-fragments (charge density, coulomb potential, kinetic energy density, ELF, etc.).

Ortho

Contains similar data for the orthogonalized-fragments.

SCF

Contains the (spin) density, potential, etc. of the final (scf) solution.

Core

Contains grid data of the frozen core (charge density, gradients, Hessian).

TransDens_L1_L2

Contains grid data for electron transition densities. L1 is either SS or ST, and L2 is a symmetry label for all transitions in the section. Here SS and ST stand for Singlet-Singlet and Singlet-Triplet, respectively. Variables in each section are Fitdensity_N and Coulpot_N for the density and Coulomb potential for excitation N within this spin and symmetry.

SCF_label

'Label' is one of the symmetry subspecies.

Each such section contains the total number of orbitals in that subspecies (as used in the *adf* calculation), with their occupation numbers and energy eigenvalues.

In addition it contains the grid-values of the (user-specified subset of) MOs in that subspecies. The variable name corresponding to an orbital is simply its index in the energy-ordered list of all orbitals (in that subspecies): '1', '2', etc.

In case of spin-orbit coupling spinors are calculated. The label is one of the double group symmetry subspecies, and the square root of the density of the spinor is calculated with an approximate phase factor. The full description of the spinor can be found in the section SO_label.

SO_label

'Label' is one of the double group symmetry subspecies.

The grid-values of the (user-specified subset of) spinors in that subspecies. A spinor is a two-component complex wave function, which can be described with four real functions: real part alpha, real part beta, imaginary part alpha, and imaginary part beta. The variable name corresponding to a spinor is its index in the energy-ordered list of all orbitals (in that subspecies), combined with the real alpha part, the real beta part, the imaginary alpha part, or the imaginary beta part: '1_alpha_R', '1_beta_R', '1_alpha_I', '1_beta_I', '2_alpha_R', etc.

LocOrb

Values of the localized orbitals.

NOCV

Values related to the NOCVs.

Profile

Values related to the NOCV profile. The variables whose name start with “Dif” contain the in-plane integral (in e/Bohr) for the given point on the profile axis. A variable without “Dif” in the name contains the integral (in electrons) of the corresponding “Dif” variable on the (-infinity,x) interval.

Geometry

Some general geometric information: the number of atoms (not counting any dummy atoms that may have been used in the *adf* calculation), their Cartesian coordinates (in bohr) and nuclear charges.

Note: the *order* of the atoms here is not necessarily identical to the input list of atoms: they are grouped by atom type.

- In an unrestricted calculation the section SCF_label is replaced by SCF_label_A and SCF_label_B for the spin-alpha and spin-beta data, respectively, and similarly for LocOrb: LocOrb_A and LocOrb_B.
- One or more subspecies may not have been used in the *adf* calculation. This happens when the basis set used in that calculation does not contain the necessary functions to span symmetry-adapted combinations of basis functions for that subspecies. In such a case the corresponding section on TAPE41 will not be created by *densf*.
- If you want to verify the contents of TAPE41, use the *pkf* utility to obtain a survey or *dmpkf* to get a complete ASCII printout.

The information is presented in three columns. In the left-most column, section and variable names are printed, variable names being indented. In the middle column, variable’s type and size is given. If the type is omitted, double precision floating point is assumed. The right-most column contains comments, if any.

Note that the name of a section of variable may consist of more than one word and that blanks in such names are significant. Furthermore, they are case-sensitive. Each line below contains the name of only one section or variable.

```
      NAME           length          Comment
Grid
  Start_point           (3)
  nr of points x        (one integer)
  nr of points y        (idem)
  nr of points z        (idem)
  total nr of points    (idem)
  x-vector              (3)
  y-vector              (3)
  z-vector              (3)
  nr of symmetries      (one integer)
  labels                (nr of symmetries160 characters)
  unrestricted          (one logical)
SumFrag
  CoulPot               (total nr of points)
  XCPot_A               (idem)   spin-restricted: XCPot
  XCPOt_B               (idem)
  Density_A             (idem)   spin-restricted: Density
  Density_B             (idem)
  Fitdensity_A          (idem)   spin-restricted: Fitdensity
  Fitdensity_B          (idem)
  Kinetic Energy Density_A (idem)   spin-restricted:
                                    Kinetic Energy Density
  Kinetic Energy Density_B (idem)
  ELF_A                 (idem)   spin-restricted: ELF
```

(continues on next page)

(continued from previous page)

```
  ELF_B                (idem)
Ortho
  Same variables as in SumFrag
SCF
  Same variables as in SumFrag and Ortho, and:
  DensityLap_A         (idem)   spin-restricted: DensityLap
  DensityLap_B         (idem)
  DensityGradX_A       (idem)   spin-restricted: DensityGradX
  DensityGradX_B       (idem)
  DensityGradY_A       (idem)   spin-restricted: DensityGradY
  DensityGradY_B       (idem)
  DensityGradZ_A       (idem)   spin-restricted: DensityGradZ
  DensityGradZ_B       (idem)
  DensityHessXX_A      (idem)   spin-restricted: DensityHessXX
  DensityHessXX_B      (idem)
  DensityHessXY_A      (idem)   spin-restricted: DensityHessXY
  DensityHessXY_B      (idem)
  DensityHessXZ_A      (idem)   spin-restricted: DensityHessXZ
  DensityHessXZ_B      (idem)
  DensityHessYY_A      (idem)   spin-restricted: DensityHessYY
  DensityHessYY_B      (idem)
  DensityHessYZ_A      (idem)   spin-restricted: DensityHessYZ
  DensityHessYZ_B      (idem)
  DensityHessZZ_A      (idem)   spin-restricted: DensityHessZZ
  DensityHessZZ_B      (idem)
  DenSigned            (idem)   sign( :math:`\lambda`\ :sub:`2` ) :math:`\rho`
  FitDenSigned         (idem)
  RDG                  (idem)   reduced density gradient
  FitRDG               (idem)
  NCI                  (idem)   +/-1 flag for non-covalent interaction regions
  FitNCI               (idem)
  SEDD                 (idem)   single exponential decay detector
  DORI                 (idem)   density overlap regions indicator
Core
  Density              (total nr. of points)
  DensityGradX         (idem)
  DensityGradY         (idem)
  DensityGradZ         (idem)
  DensityHessXX        (idem)
  DensityHessXY        (idem)
  DensityHessXZ        (idem)
  DensityHessYY        (idem)
  DensityHessYZ        (idem)
  DensityHessZZ        (idem)
TransDens_L1_L2                   L1: SS or ST; L2 is excitation's symmetry
  Fitdensity_1         (total nr. of points)
  Fitdensity_2         (idem)
  Fitdensity_3         (idem)
  Coulpot_1            (idem)
  Coulpot_2            (idem)
  Coulpot_3            (idem)
SCF_label_A
     (label is a symmetry subspecies.
      Spin-restricted: SCF_label)
  nr of orbitals       (one integer)
  Occupations          (nr of orbitals)
  Eigenvalues          (idem)
```

(continues on next page)

(continued from previous page)

```
  1                     (total nr of points)
  2                     (idem)
  3                     (idem)
     (as many as there are Molecular Orbitals in that
      symmetry representation for the indicated spin)
SCF_label_B
     (only if spin-unrestricted same variable as
      in SCF_label_A)
SO_label
     (label is a double group symmetry subspecies.)
  1_alpha_R             (total nr of points) real alpha part of the 1st spinor
  1_beta_R              (idem) real beta part of the 1st spinor
  1_alpha_I             (idem) imaginary alpha part of the 1st spinor
  1_beta_I              (idem) imaginary beta part of the 1st spinor
  2_alpha_R             (idem) real alpha part of the 2nd spinor
  2_beta_R              (idem) real beta part of the 2nd spinor
  2_alpha_I             (idem) imaginary alpha part of the 2nd spinor
  2_beta_I              (idem) imaginary beta part of the 2nd spinor
     (as many as there are spinors in that
      double group symmetry representation)
LocOrb_A                           if unrestricted, otherwise
                                   LocOrb
  nr of orbitals        (one integer)
  1                     (total nr. of points)
  2                     (idem)
  (etc)
NOCV
  Dens_A number(occupation number)      (total nr. of points)
  Dens_B number(occupation number)      (idem)
  (etc)
Profile
  DifDens_A number*(NOCV eigenvalue)    (number of profile points)
  DifDens_B number*(NOCV eigenvalue)    (idem)
  DifSumDens_A number                   (idem)
  DifSumDens_B number                   (idem)
  DifRestSumDensities_A                 (idem)
  DifRestSumDensities_B                 (idem)
  DifSumBelow_A                         (idem)
  DifSumBelow_B                         (idem)
  Dens_A number*(NOCV eigenvalue)       (idem)
  Dens_B number*(NOCV eigenvalue)       (idem)
  SumDens_A number                      (idem)
  SumDens_B number                      (idem)
  RestSumDensities_A                    (idem)
  RestSumDensities_B                    (idem)
  SumBelow_A                            (idem)
  SumBelow_B                            (idem)
POLTDDFT
  InducedDensX_number          (total nr of points)  Induced (first-order perturbed)␣
↪density in the X-direction
  InducedDensY_number          (idem)                Induced (first-order perturbed)␣
↪density in the Y-direction
  InducedDensZ_number          (idem)                Induced (first-order perturbed)␣
↪density in the Z-direction
  TransDensX_number            (idem)                Transition (first-order␣
↪perturbed) density in the X-direction
  TransDensY_number            (idem)                Transition (first-order␣
↪perturbed) density in the Y-direction
```

(continues on next page)

(continued from previous page)

```
 TransDensZ_number            (idem)                 Transition (first-order␣
↪perturbed) density in the Z-direction
 RotatorStrengthDensX_number  (idem)                 Rotator strength density in the X-
↪direction
 RotatorStrengthDensY_number  (idem)                 Rotator strength density in the Y-
↪direction
 RotatorStrengthDensZ_number  (idem)                 Rotator strength density in the Z-
↪direction
Geometry
 nnuc                  (one integer)
     (nr of nuclei, omitting dummy atoms)
 xyznuc                (nnuc times 3)
     (the atoms are not in the same order as in the adf input
      file. Rather they are grouped by atomtype.)
 qtch                  (nnuc)       Atomic charges
x values
       x values        (total nr. of points)
y values
       y values        (idem)
z values
       z values        (idem)
```

9.8 Dos: Density of States

The auxiliary program *dos* computes various types of densities-of-states (DOS) for a user-specified energy interval.

dos can not be used in case of spin-orbit coupling

dos requires an ASCII input file where the user specifies the items to be calculated and computational details, plus the standard result file adf.rkf (TAPE21) from an *adf* calculation. The latter file must be present as a local file with name TAPE21 in the directory where *dos* is executed, or one can use the key adffile to point to the adf.rkf file,

dos produces as result one or more ASCII files with the density-of-states values. Error messages and computational info (if any) are written to standard output.

9.8.1 Introduction

The program *dos* gives information on the number and character of one-electron levels (molecular orbitals) as a function of the (orbital) energy. The total density of states N*(*E) is a well known concept in electronic structure theory of infinite systems (crystals). N*(*E)*dE* denotes the number of one-electron levels (orbitals) in the infinitesimal energy interval *dE*. The total density of states (TDOS) at energy *E* is usually written as

$$N(E) = \sum_i \delta(E - \epsilon_i) \qquad (4.5.1)$$

where the ϵ_i denote the one-electron energies. So the integral of $N(E)$ over an energy interval E_1 to E_2 gives the number of one-electron states in that interval. Usually the δ-functions are broadened to make a graphical representation possible.

When the δ-functions are multiplied by a weight factor that describes some property of the one-electron state ϕ_i at energy ϵ_i various types of densities-of-states are obtained that provide a graphical representation of the state character (orbital character) as a function of one-electron energy.

In calculations on finite molecules the total density of states as a function of (orbital) energy may also be useful, but the main use of various types of densities-of-states is to provide a pictorial representation of Mulliken populations. The weight factors employed are related to the orbital character determined by means of a Mulliken population analysis *per orbital* (see below). The program *dos,* therefore, provides the same information as can be generated by the ADF program (a population analysis per orbital) but *dos* enables an easy graphical representation and is particularly useful when there are many one-electron levels, for instance in calculations on clusters. You can obtain a simple view of the character of the orbitals in a certain energy range. You can also find out in which orbitals (at which energies) certain basis functions or fragment orbitals give a large contribution, and whether such contributions are bonding, nonbonding or antibonding with respect to particular bonds. Such information is provided by *dos* in the form of (weighted) density of states values over a user-specified energy range, which can for instance be plotted by *gnuplot*.

The following options are available for computations by *dos*:

- TDOS: Total Density of States
- GPDOS: Gross Population Density of States
- OPDOS: Overlap Population Density of States
- PDOS: Projected Density of States

The total density of states (TDOS) has large values at energies where there are many states per energy interval.

The GPDOS (Gross Population based Density Of States) of a function χ_μ (or a sum of such functions) has large values at energies where this function (these functions) occur(s) in the molecular orbitals.

The PDOS of a function χ_μ provides similar information, but with the projection of χ_μ onto the orbital ϕ_i as weight factor for the importance of χ_μ in the orbital ϕ_i .

The OPDOS (Overlap Population based Density Of States) between χ_μ and χ_ν has large positive values at energies where the interaction between them is bonding, and negative values where the interaction is anti-bonding. An example of the use of these plots is provided in[1].

We review below the Mulliken population analysis, and then describe the forms of density of states analysis performed by DOS. Finally an input description of DOS is given.

9.8.2 Mulliken population analysis

The orbitals ϕ_i with energies ϵ_i are expanded in basis functions χ_μ , which leads to the definition of density matrices P_i describing orbital densities, from which the total density matrix can be constructed:

$$\phi_i(r) = \sum_\mu \chi_\mu(r) C_{\mu i}$$

$$\begin{aligned}
\rho_i(r) &= \int |\phi_i(r)|^2 = \sum_{\mu\nu} P_{i,\mu\nu}\chi_\mu(r)\chi_\nu(r) \\
P_{i,\mu\nu} &= C_{\mu i} C_{\nu i} \\
\rho(r) &= \sum_i n_i \rho_i(r) = \sum_{\mu\nu} P_{\mu\nu}\chi_\mu(r)\chi_\nu(r) \\
P_{\mu\nu} &= \sum_i n_i C_{\mu i} C_{\nu i} \qquad (4.5.2)
\end{aligned}$$

Here μ and μ run over the basis functions, which may be either primitive functions, or combinations of primitive functions, for instance the SCF orbitals of atoms or larger fragments.

---

[1] P.J. van den Hoek, E.J. Baerends, and R.A. van Santen, *Ethylene epoxidation on silver(110): the role of subsurface oxygen*, Journal of Physical Chemistry 93, 6469 (1989) (https://doi.org/10.1021/j100354a038)

The Mulliken population analysis provides a partitioning of either the total charge density or an orbital density. The total density is written as

$$\begin{aligned}\rho(r) &= \sum_{\mu\nu} P_{\mu\nu}\chi_\mu(r)\chi_\nu(r)\\ &= \sum_{A\leq B}\sum_{\mu\in A}\sum_{\nu\in B} P_{\mu\nu}\chi_\mu(r)\chi_\nu(r)\\ &= \sum_{A\leq B}\rho_{AB} \qquad (4.5.3a)\end{aligned}$$

$$\rho_{AB} = \sum_{\mu\in A}\sum_{\nu\in B} P_{\mu\nu}\chi_\mu(r)\chi_\nu(r) \qquad (4.5.3b)$$

The total number of electrons, $N = \int\rho(r)dr$, is now partitioned over the atoms by assigning an overlap population $P_{\mu\nu}S_{\mu\nu} + P_{\nu\mu}S_{\nu\mu}$ for one half to the atom *A* of χ_μ and one half to atom *B* of χ_ν,

$$N = \int\rho(r)dr = \sum_{\mu\nu} P_{\mu\nu}S_{\mu\nu} = \sum_\mu GP_\mu \qquad (4.5.4a)$$

$$GP_\mu = \sum_\nu P_{\mu\nu}S_{\mu\nu} \qquad (4.5.4b)$$

GP_μ is the gross population of χ_μ . It contains the net population $P_{\mu\mu}$ and half of each total overlap population $P_{\mu\nu}S_{\mu\nu}+P_{\nu\mu}S_{\nu\mu}$ between χ_μ and χ_ν. Summing the gross population over the functions $\mu\in A$ yields the total number of electrons assigned to atom A, or the gross population of atom A, GP_A , and hence the gross charge Q_A of atom A,

$$GP_A = \sum_{\mu\in A} GP_\mu \qquad (4.5.5a)$$

$$Q_A = Z_A - GP_A \qquad (4.5.5b)$$

The overlap population $OP_{\mu\nu}$ between two functions and the overlap population Q_{AB} between two atoms are defined in an analogous manner,

$$OP_{\mu\nu} = P_{\mu\nu}S_{\mu\nu} + P_{\nu\mu}S_{\nu\mu} \qquad (4.5.6a)$$

$$Q_{AB} = \sum_{\mu\in A}\sum_{\nu\in B} OP_{\mu\nu} \qquad (4.5.6b)$$

These quantities can be evaluated for a single orbital density, $N = 1 = \int|\phi_i(r)|^2dr$. The gross population $GP_{i,\mu}$ of a function in a specific orbital density $|\phi_i(r)|^2$ is then associated with the fraction of the orbital density belonging to that function (or the percentage χ_μ character of orbital ϕ_i , and the overlap population $OP_{i,\mu\nu}$ gives an indication of the strength of bonding or anti-bonding between χ_μ and χ_ν in orbital ϕ_i ,

$$GP_{i\mu} = \sum_\nu P_{i,\mu\nu}S_{\mu\nu} = \sum_\nu C_{\mu i}C_{\nu i}S_{\mu\nu} \qquad (4.5.7a)$$

$$OP_{i,\mu\nu} = P_{i,\mu\nu}S_{\mu\nu} + P_{i,\nu\mu}S_{\nu\mu} = 2C_{\mu i}C_{\nu i}S_{\mu\nu} \qquad (4.5.7b)$$

9.8.3 Density of states analyses based on Mulliken population analysis

Total density of states

The total density of states TDOS at energy *E* is written as

$$TDOS : N(E) = \sum_i \delta(E-\epsilon_i)$$

so the integral of $N(E)$ over an energy interval E_1 to E_2 gives the number of one-electron states in that interval. In practice the delta functions are approximated by Lorentzians,

$$TDOS : N(E) = \sum_i L(E - \epsilon_i) = \sum_i \left(\frac{\sigma}{\pi} \frac{1}{(E - \epsilon_i)^2 + \sigma^2} \right)$$

A plot of $N(E)$ versus E reveals energetic regions where many levels are located. The width parameter s determines of course the appearance of the plot. A typical value is 0.25 eV (used as default in *dos*).

Partial (gross population and projected) density of states

In order to find out if a given function χ_μ contributes strongly to one-electron levels at certain energies, one may weigh a one-electron level with the percentage χ_μ character. We usually determine the χ_μ character by the gross populations, obtaining the GPDOS form of the partial density of states,

$$GPDOS : N_\mu(E) = \sum_i GP_{i,\mu} L(E - \epsilon_i)$$

If the weight factor is determined by projection of ϕ_i against χ_μ , we obtain the projected density of states PDOS,

$$PDOS : N_\mu(E) = \sum_i |\langle \chi_\mu | \phi_i \rangle|^2 L(E - \epsilon_i)$$

One should not use the PDOS for d-type or f-type primitive basis functions ('BAS'). A d-type function consists of 6 Cartesian functions, while there can of course be only 5 true d-type functions among them: one (linear combination) of them is in fact an s-type function (x2+y2+z2). Similarly, there are 10 f-type Cartesian functions, 3 of which are in fact p-functions. The PDOS is calculated for the 6 d-type and 10 f-type Cartesian functions, which leads to undesired results. An PDOS for SFOs does not suffer from this problem.

Overlap population density of states (OPDOS)

If the delta function representing orbital ϕ_i is weighed with the overlap population between χ_μ and χ_ν in ϕ_i , the overlap population density of states OPDOS is obtained,

$$OPDOS : N_{\mu\nu}(E) = \sum_i OP_{i,\mu\nu} L(E - \epsilon_i)$$

If an orbital ϕ_i at energy ϵ_i is strongly bonding between χ_μ and χ_ν the overlap population is strongly positive and OPDOS(e) will be large and positive around $E = \epsilon_i$. Similarly, OPDOS(E) will be negative around energy ϵ_i when there is antibonding between χ_μ and χ_ν in ϕ_i .

The OPDOS(E) has been used under the name coop (crystal orbital overlap population) in Extended-Hückel solid state calculations by Hoffmann and coworkers [2].

[2] R. Hoffmann, *A chemist's view of bonding in extended structures* (VCH Publishers, New York, 1988).

9.8.4 Generalizations of OPDOS, GPDOS, PDOS

As observed above, the basis functions in the above expressions may be primitive basis functions ('Slater type orbitals'), but of course the formulas are equally applicable for other types of MO expansions. In *dos* the user may select either the expansion in primitive basis functions ('BAS') or the expansion in SFOs (Symmetrized Fragment Orbitals) for the DOS analyses.

It is also possible in DOS to treat a *set* of basis functions simultaneously. For instance, the GPDOS for a set of basis functions μ_1 , μ_2 , … is simply defined as the summation of the corresponding single-function GPDOS(E) values

$$N_{\mu-set}(E) = \sum_{\mu \in \mu-set} \sum_i GP_{i,\mu} L(E - \epsilon_i)$$

In a similar fashion the OPDOS can be defined for *two sets* of basis functions μ_1 , μ_2 , … and ν_1 , ν_2 , … as

$$N_{\mu-set,\nu-set}(E) = \sum_{\nu\in\nu-set}\sum_{\mu\in\mu-set}\sum_i OP_{i,\mu\nu}L(E-\epsilon_i)$$

and finally for the PDOS we get in similar fashion

$$N_{\mu-set}(E) = \sum_{\mu\in\mu-set}\sum_i |\langle\chi_\mu|\phi_i\rangle|^2 L(E-\epsilon_i)$$

9.8.5 Input

The (ASCII) input for *dos* is keyword oriented. Reading input by *dos* terminates whenever it finds a line END INPUT or the end-of-file, whichever comes first.

Follows a list of keywords with their meaning. Generally keys may occur more than once and *the order in which they appear is relevant in some cases*. For instance the key energyrange (which defines for what energy values to compute densities-of-states, see below) applies to all items that come after it in input until the next occurrence of energyrange.

```
$AMSBIN/dos << eor
ADFFILE adffile
ENERGYRANGE {Npoint=nr} {E-start=e1} {E-end=e2 / E-step=de}
LORENTZIAN width=width
FILE file
TDOS { title }
OPDOS ...
GPDOS ...
PDOS ...
eor
```

ADFFILE adffile

```
ADFFILE adffile
```

Specifies a (relative) path to an adf.rkf file. If not specified dos requires a file named TAPE21 in the current directory.

Energy scan values

```
ENERGYRANGE {Npoint=nr} {E-start=e1} {E-end=e2 / E-step=de}
```

This specifies for which energy values the densities-of-states are computed that are specified *after* it in the input file and *until* the next occurrence of ENERGYRANGE.

ENERGYRANGE specifies the lower bound, upper bound and number of equidistant energy values (including endpoints). All items are optional with defaults applying for those omitted.

The E end and E-step values determine one another and must therefore not be specified both (or be consistent).

The initial defaults are:

```
nr=301
e1=-20
de=0.1
```

All energy data are in eV.

When values have been changed with the key ENERGYVALUE, the so-modified values are the defaults for the next occurrence of ENERGYVALUE.

Peak widening

The peaks in the DOS curves corresponding to the energies of the molecular orbitals are widened by a Lorentzian curve, the width of which can be adjusted.

```
LORENTZIAN width=width
```

Initial default width is 0.25 (eV).

As for ENERGYRANGE, the key LORENTZIAN may occur more than once and each occurrence sets the width for all items after it.

Result files

The computed densities-of-states are stored on one or more ASCII files, which have to be specified in input.

```
FILE file
```

The key FILE may occur any number of times in input. Each time it occurs the specified file is opened by *dos*. The file must not yet exist and the new file will accumulate (ASCII) the densities-of-states data of all DOS items subsequently specified, until the next occurrence of FILE. The first occurrence of the key FILE must be given before any DOS specification (by the keys TDOS, OPDOS, GPDOS, PDOS, see below).

The format of the result file is such that it can be fed directly into *gnuplot*.

Densities of States

Total density of states.

```
TDOS { title }
```

`TDOS` instructs the program to compute the total density of states.

`title` (optional) will appear as title to the section of corresponding Density-of-States data in the result file.

The other types of densities-of-states require block-type keyword input.

```
OPDOS { title }
Ftype numbers
Ftype numbers
...
*SUBEND*
Ftype numbers
Ftype numbers
...
END
```

`Ftype` Specifies the type of basis functions to use in the MO expansions. If the primitive basis functions are to be used Ftype must be bas. For the SFO representation Ftype must be one of the irreducible representations of the point group symmetry. All Ftype values in the data block must be consistent: either all are bas or all are irrep labels. The scope of this consistency requirement is the data block of the current key: in a next OPDOS data block, for instance, a different choice may be made.

`numbers` Must be a sequence of integers referring to the basis functions to be selected, i.e. the ‘μ-set’ and ‘ν-set’ in (4.5.13) etc. If bas-type basis functions are selected the numbers refer to the overall list of all basis functions as printed in the output file of the adf run. If SFOs are selected the numbers refer to the SFO list of the pertaining symmetry representation without the core functions, see the adf output file.

`SUBEND` Must be typed as such and separates the ‘μ-set’ and the ‘ν-set’: all records before subend specify together the ‘μ-set’ and all records below subend comprise the ‘ν-set’. Each of these two sections may consist of any number of records.

The input for GPDOS and PDOS are similar, but simpler because only one set of functions ('μ-set') has to be specified, so there is no subend in the data blocks for these keys.

```
GPDOS { title }
Ftype numbers
Ftype numbers
...
END
```

```
PDOS { title }
Ftype numbers
Ftype numbers
...
END
```

The keys GPDOS, OPDOS, PDOS and (TDOS) may occur any number of times in input and in any order. Each time the DOS key occurs the current energyrange and lorentzian settings apply and the results are written to the current file.

9.9 VCD Analysis: VCDtools

The auxiliary program *VCDtools* provides insight regarding the origin of the VCD intensity of a given normal mode. This is done by analyzing and visualizing the various contributions to the total Electric (E) and Magnetic (M) Dipole Transition Moments (DTM). *e.g.*, the contributions from atoms, electrons, nuclei and molecular fragments. The *VCDtools* program is accessible through the AMS-GUI. Documentation can be found in:

- VCDtools section of the AMS manual.

9.10 PyFrag: Activation Strain Model Analysis

The standalone script PyFrag allows one to analyse various quantities along a trajectory with ADF. Usually such a trajectory results from either an internal reaction coordinate (IRC) or a linear transit (LT) calculation and is analyzed in the context of extended activation strain model (ASM).

See also:

Example: Activation Strain Model Analysis using PyFrag (page 1024)

9.10.1 Requirements

The trajectory can thereby either be a file containing concatenated molecular geometries in the xyz-format (see example below) or a T21 binary output file from a previous LT or IRC calculation with ADF. ASM is based on the concept molecular fragments, which have to be defined by specifying the corresponding atomic indices for each fragment. Furthermore, frequently used ADF input options can be imported from a separate input file (see *example* (page 1024)).

9.10.2 Running PyFrag

Running PyFrag is done by starting the corresponding Python script as follows:

```
$AMSBIN/amspython $AMSHOME/scripting/standalone/pyfrag/PyFrag.py [options]
```

whereas the details of possible `[options]` are explained in the following

9.10.3 Specifying the Trajectory

The trajectory is specified by one of the following:

```
--xyzpath <PATH>/trajectory.xyz
--irct21 <PATH>/IRCtrajectory.T21
--lt <PATH>/LTtrajectory.T21
```

9.10.4 Molecular Fragments

The system can be split into an arbitrary number of disjoint subsystems, fragments. Each fragment is defined by its own argument `--fragment` followed by the indices of atoms in that fragment:

```
--fragment 1 3 4 5 6 --fragment 2
```

9.10.5 ADF Options

The settings for the ADF calculations are retrieved from either a pyfrag-specific settings file `adfsettings` (see *example* (page 1024)).

```
--adfinputfile adfsettings
```

or by individual command line arguments for each option, e.g.:

```
--adfinput basis.type=TZ2P
```

9.10.6 Analysis Options

The following options control the quantities reported by PyFrag in the final results table:

Offsets for the strain energies in kcal/mol of the fragments (one argument for each fragment, corresponding to the order of the fragment definition above):

```
--strain <StrainEnergy Fragment1> --strain <StrainEnergy Fragment2>
```

Bond length between selected pairs of atoms and equilibrium offset value in Angstrom

```
--bondlength <AtomIndex1> <AtomIndex2> <offset value>
```

Angle length between selected triples of atoms and equilibrium offset value in degrees

```
--angle <AtomIndex1> <AtomIndex2> <AtomIndex3> <offset value>
```

Hirshfeld charges of fragments

```
--hirshfeld frag<FragmentNumber>
```

Atomic charges (Voronoi deformation density charges, VDD)

```
--VDD  <AtomIndex1> [<AtomIndex2> [...]]
```

Orbital interaction energies for orbitals belonging to a point group symmetry irrep (specified by its symbol) of the whole system

```
--irrepOI <IrrepSymbol>
```

Orbital energies within fragments can be either specified for a certain orbital in a given irrep specifically for HOMO and LUMO levels in each fragment

```
--orbitalenergy <IrrepSymbol> <FragmentNumber> <OrbitalIndexWithinIrrep>
--orbitalenergy frag<FragmentNumber> HOMO
--orbitalenergy frag<FragmentNumber> LUMO
```

Orbital population numbers within fragments are again be either specified for a certain orbital in an irrep or for the HOMO or LUMO in each fragment

```
--population <IrrepSymbol> frag<FragmentNumber> <OrbitalIndexWithinIrrep>
--population frag<FragmentNumber> HOMO
--population frag<FragmentNumber> LUMO
```

Orbital overlaps between pairs of orbitals, e.g.

```
--overlap frag<FragmentNumber> HOMO  frag<FragmentNumber> LUMO
```

or

```
--overlap <IrrepSymbol> frag<FragmentNumber> <OrbitalIndexWithinIrrep> <IrrepSymbol>␣
↪frag<FragmentNumber> <OrbitalIndexWithinIrrep>
```

CHAPTER

TEN

ACCURACY AND EFFICIENCY

See also:

Examples: *accuracy* (page 1064)

10.1 Precision and Self-Consistency

The precision of a calculation is determined by

- The function sets (basis sets, levels of frozen core approximation, and fit sets for the computation of the Coulomb potential)
- Numerical integration settings in real space
- The accuracy of the density fitting procedure
- Convergence criteria (for the SCF procedure and the geometry optimization)
- A few more items that are rather technical and usually irrelevant (these are not discussed here).

The fragments you attach determine, through the fragment files, the function sets. Since each fragment traces back to one or more Create runs, the employed data base files in the Create runs determine the finally employed function sets.

In this part we examine numerical integration, density fitting and the SCF procedure.

Numerical Quality

With the key NUMERICALQUALITY one can set the density fitting quality (ZlmFit) and the numerical integration quality (BeckeGrid) simultaneously

```
NUMERICALQUALITY [basic|normal|good|verygood|excellent]
```

Note that the so called Excellent quality is only meant for debugging purposes. In case of real numerical issues the results with Excellent should not be trusted more than with VeryGood.

10.1.1 SCF

The SCF procedure is regulated with keys that set the maximum number of iterations, the convergence criterion, and various items that control the iterative update method. Molecules may display wildly different SCF-iteration behavior, ranging from easy and rapid convergence to troublesome oscillations. We expect that the default settings take care of most cases, but one should realize that this is a difficult and tricky subject. The user has a few (main) options to adapt the procedure to the situation at hand: simple damping or one of the SCF acceleration schemes.

See also:

In case of SCF convergence problems, see *SCF Convergence Guidelines for ADF* (page 356)

At each cycle the electron density is computed as a sum of occupied orbitals squared; the new density defines the potential from which the orbitals are re-computed. The cycle is repeated until convergence is reached. To speed-up convergence and to avoid non-convergent oscillatory behavior the values at the next iteration are constructed as a mixture of the computed new data and those used at the cycles before. This may involve only the previous cycle and is then called *damping*. Alternatively some sort of DIIS (direct inversion in iterative space) or LIST (LInear-expansion Shooting Technique) procedure can be invoked, which is a generalization of damping to include more previous iterations.

Starting from ADF2016, the SCF module has been rewritten from scratch. The old SCF is still available and it can be switched on using the `OldSCF` keyword. The description below refers to the new SCF code only. User documentation for the OldSCF module is available in the ADF2014 User's Guide.

There are various SCF acceleration techniques implemented in ADF. By default the mixed ADIIS+SDIIS method by Hu and Wang[1] is used, unless the number of DIIS expansion vectors is set to zero (see the `DIIS N` key below) or `NoADIIS` is specified. Here, SDIIS stands for the original Pulay DIIS scheme. Other available methods mostly belong to the `LIST` family developed in the group of Y.A. Wang. The ARH and Energy-DIIS methods are available with the `OldSCF` method only.

Main options

Subkeys in the data block of the master key SCF control the aspects mentioned above. Each subkey is optional. Curly brackets are not part of the key. Omission means the application of default values. Omission of the SCF block altogether implies defaults for all subkeys.

```
SCF
 Iterations Niter
 Converge SCFcnv { sconv2 }
 AccelerationMethod { ADIIS | LISTi | LISTb | fDIIS | LISTf | MESA | SDIIS }
 MESA [NoFDIIS] [NoLISTb] [NoLISTf] [NoLISTi] [NoSDIIS] [NoADIIS]
 ADIIS
   {THRESH1 a1}
   {THRESH2 a2}
 End
 NoADIIS
 DIIS
   {N n}
   {OK ok}
   {cyc cyc}
 End
 Mixing mix
 Mixing1 mix1
 OldSCF
End
```

---

[1] X. Hu and W. Yang, *Accelerating self-consistent field convergence with the augmented Roothaan-Hall energy function*, Journal of Chemical Physics 132, 054109 (2010) (https://doi.org/10.1063/1.3304922)

Iterations Niter

Niter The maximum number of SCF cycles allowed. In case of Geometry Optimizations it applies separately to each of the SCF procedures that are executed. Default is 300. The program executes at least one cycle, even if you specify a non-positive number.

Converge SCFcnv { sconv2 }

SCFcnv The criterion to stop the SCF updates. The tested error is the commutator of the Fock matrix and the P-matrix (=density matrix in the representation of the basis functions) from which the F-matrix was obtained. This commutator is zero when absolute self-consistency is reached. Convergence is considered reached when the maximum element falls below SCFcnv and the norm of the matrix below 10*SCFcnv. The default is 1e-6 (in Create mode: 1e-8).

sconv2 A second criterion which plays a role when the SCF procedure has difficulty converging. When in any SCF procedure the currently applicable criterion does not seem to be achievable, the program stops the SCF. When the secondary criterion (sconv2) has been met, only a warning is issued and the program continues normally. If the secondary criterion was not met either, the program terminates any further geometry optimizations, frequency steps, etc. You can prevent the program from terminating in such a case with the key ALLOW. The default for sconv2 is 1e-3.

AccelerationMethod Specify which SCF acceleration method must be invoked. By default the ADIIS+SDIIS method (denoted as ADIIS) is used and normally it results in optimal performance. In some cases it may be useful to use one of the methods from the LIST family or fDIIS. The LIST family methods have been implemented following publications by the group of Y.A. Wang[2][3]. The number of vectors to store specified by `DIIS N` also applies to the LIST methods. This number is a very important parameter and it is worthwhile increasing or decreasing it in case of SCF convergence problems. A word of caution: do not just blindly increase the number for every system. Testing showed that a large number breaks convergence for some, mainly small, systems.

MESA The MESA method has also been developed in the group of Y.A. Wang[7]. It combines several other acceleration methods (currently ADIIS, fDIIS, LISTb, LISTf, LISTi, and SDIIS). In order to improve the method's performance, some of the components can be disabled by specifying the corresponding "No" argument. For example, to remove SDIIS from MESA specify `MESA NoSDIIS`. Specifying this key invokes the MESA method regardless of the AccelerationMethod's value.

ADIIS This key has effect only if ADIIS acceleration method is used. The THRESH1 and THRESH2 arguments correspond to values of the maximum element of the [F,P] commutator matrix, ErrMax. If ErrMax $\geq a1$ then only A-DIIS coefficients are used to determine the next Fock matrix. If ErrMax $\leq a2$ then only SDIIS coefficients are used. For ErrMax between $a2$ and $a1$ the total DIIS coefficients are calculated from SDIIS and A-DIIS values weighted proportionally according to the ErrMax value. Thus, the weight of A-DIIS coefficients decreases as the ErrMax value becomes smaller. The default values for a1 and a2 are 0.01 and 0.0001, respectively. In difficult SCF cases, especially when the Pulay DIIS appears to be unstable, it may be worth trying to decrease the thresholds and let A-DIIS approach the final solution on its own.

NoADIIS If OldSCF is not set, specifying NoADIIS has the same effect as setting AccelerationMethod to SDIIS. it disables A-DIIS and switches SCF to a damping+SDIIS scheme. In this scheme, SCF starts with simple damping (a.k.a. mixing) and continues until the maximum element value of the [F,P] commutator drops below the `DIIS OK` parameter but for no more than `DIIS Cyc` iterations. After this, the pure SDIIS scheme becomes effective till the end of the SCF procedure. If OldSCF is set then specifying NoADIIS disables the automatic switching to ADIIS+SDIIS in case of a difficult SCF convergence.

DIIS The DIIS sub-block specifications can be given to control the SCF acceleration procedure. All these specifications are optional. Normally, ADIIS+SDIIS is activated starting from the 2nd iteration regardless of the `OK` and `Cyc` settings below. The two settings become effective only when `NoADIIS` is specified.

---

[2] Y.A. Wang, C.Y. Yam, Y.K. Chen, G. Chen, *Communication: Linear-expansion shooting techniques for accelerating self-consistent field convergence*, Journal of Chemical Physics 134, 241103 (2011) (https://doi.org/10.1063/1.3609242)

[3] M.A. Garcia Chavez, *Numerical methods in quantum chemistry to accelerate SCF convergence and calculate partial atomic charges*, PhD Thesis, University of British Columbia, (2018), (http://dx.doi.org/10.14288/1.0372885)

N The number of expansion vectors used for accelerating the SCF. The number of previous cycles taken into the linear combination is then n-1 (the new computed potential is also involved in the linear combination). By default n=10. An input value smaller than 2 disables the DIIS. Note that this number applies not only to Pulay DIIS scheme but also to other DIIS-like methods, such as A-DIIS and LIST methods. It should be noted that methods from the LIST family are quite sensitive to the number of expansion vectors used. For this reason, these methods have a built-in limit on the number depending on the iteration number and the degree of convergence. The number of vectors in LIST increases when approaching convergence but it is always limited by `DIIS N`, which is a hard limit. It may be useful to set `DIIS N` to a value larger than 10 when using a LIST method with a difficult to converge system. A value between 12 and 20 can sometimes get the job done.

OK The SDIIS starting criterion, when A-DIIS is disabled, by default 0.5 a.u..

Cyc When A-DIIS is disabled, the SDIIS will start at this iteration (by default 5) irrespective of the `DIIS OK` value above.

Mixing mix

mix When none of the SCF acceleration methods is active, the next Fock matrix is determined as $F = mixF_n + (1 - mix)F_{n-1}$. The default value is 0.2.

Mixing1 mix1

mix1 A different mixing parameter value for the first SCF cycle. By default, equal to `Mixing`.

OldSCF Enforce the old SCF procedure. It will also be used automatically when one of the following is true: `Occupations Steep` is specified, level-shifting is enabled (see the `Lshift` key below), the augmented Roothaan-Hall method is enabled (see the `ARH` key below), the Energy-DIIS method is enabled (see the `EDIIS` key below), the `RESTOCC` key is specified.

Lshift vshift Level shifting is currently not implemented in the new SCF, therefore specifying `Lshift` enables `OldSCF` automatically.

VShift The level shifting parameter. The diagonal elements of the Fock matrix, in the representation of the orbitals of the previous iteration, are raised by vshift hartree energy units for the virtual orbitals. This may help to solve convergence problems when during the SCF iterations charge is sloshing back and forth between different orbitals that are close in energy and all located around the Fermi level. Level shifting is not supported in the case of Spin-Orbit coupling. **At the moment properties that use virtuals, like excitation energies, response properties, NMR calculations, will give incorrect results if level shifting is applied.**

Lshift_err Shift_err

Shift_err Specifies that level shifting will be turned off by the program as soon as the SCF error drops below a threshold; default value: 0.

Lshift_cyc Shift_cyc

Shift_cyc Specifies that level shifting is not turned on before the given SCF cycle number (for the start-up geometry); default value: 1.

Note: *Electron smearing*, may be used to overcome convergence difficulties. The idea is to distribute electron occupations fractionally over the states around the Fermi level using the Fermi distribution function. This aspect is controlled by the Smear or the ElectronTemperature option of the Occupations key. See the Occupations key for more details.

Note: A-DIIS is not compatible with enforced non-aufbau electronic configurations and it is disabled in such a case automatically. A non-aufbau electronic configuration may be enforced using a block form of the Occupations key, but it may also result from the KeepOrbitals (a.k.a. orbital tracking) feature.

Energy-DIIS

```
SCF
 EDIIS
 ...
End
```

Energy-DIIS is implemented following the paper by Kudin, Scuseria, and Cances[4]. The method is invoked by specifying an EDIIS keyword in the SCF block. Please note that similar to ARH and unlike the standard SCF procedure in ADF this method requires energy evaluation at each SCF cycle, which makes it significantly slower compared to energy-free SCF. You might need a higher integration accuracy to get an accurate total energy. The same restrictions apply as for the key *TOTALENERGY* (page 252). The EDIIS method will start at the 2nd SCF cycle, and the size of the DIIS space will be the same as for the normal DIIS. This subkey EDIIS can be used in addition to the other subkeys of the block key SCF. This method has been superseded by A-DIIS and the LIST family methods.

Augmented Roothaan-Hall (ARH)

The Augmented Roothaan-Hall method has been developed by T. Helgaker and coworkers and is extensively discussed in Ref.[5]. The basic idea of the method is that the density matrix is optimized directly to minimize total energy. At each step, the new density matrix is parametrized in terms of matrix exponent:

P_{new} = exp(-X) P_{old} exp(X),

here, X is an anti-symmetric step matrix subject to the following conditions:

X = argmin{E(P(X))} - X minimizes the energy

|X| < h - length of X is smaller than or equal to some trust radius

The optimal X is found using a Conjugate Gradient method, possibly with pre-conditioning. The trust radius is updated based on how well the energy change is predicted.

ARH Input

The ARH procedure is invoked by including the ARH block in the SCF input block. This subkey ARH can be used in addition to the other subkeys of the block key SCF.

```
SCF
  ARH
    {CONV conv}
    {ITER iter}
    {NSAVED nsaved}
    {START start}
    {FINAL} ...
  End
  ...
End
SYMMETRY NOSYM
```

All parameters in the ARH keyword are optional. The following arguments determine the main parameters of the ARH procedure.

CONV conv ARH convergence criterion. When the RMS gradient and its maximum components are both lower than the criterion, the ARH procedure will be considered converged. The default value is 10^{-4} .

---

[4] K.N. Kudin, G.E. Scuseria and E. Cances, *A black-box self-consistent field convergence algorithm: One step closer*, Journal of Chemical Physics 116, 8255 (2002) (https://doi.org/10.1063/1.1470195)

[5] S. Høst, J. Olsen, B. Jansík, L. Thøgersen, P. Jørgensen and T. Helgaker, *The augmented Roothaan-Hall method for optimizing Hartree-Fock and Kohn-Sham density matrices*, Journal of Chemical Physics 129, 124106 (2008) (https://doi.org/10.1063/1.2974099)

`ITER iter` Maximum number of ARH iteration to perform. Please note that in difficult cases a huge number of iterations may be required for complete SCF convergence. The default value is 500.

`FINAL` Determines whether SCF is continued after ARH has completed. If this option is set, one Fock matrix diagonalization will be performed to get orbitals and the SCF procedure will be halted. By default this option is OFF.

`START start` Sets the SCF cycle number on which the ARH method is invoked. The default value is 2. Using a larger value may provide a better starting guess for the ARH minimization.

`NSAVED nsaved` Sets the number of saved density and Fock matrices used for augmentation of the electronic Hessian. The default value is 8. A larger nsaved value should be used in difficult cases when the number of orbitals very close to the Fermi level is large.

The default minimization method is Untransformed Pre-conditioned Conjugate Gradient. The following two parameters may be used to change this.

`NOPRECOND` Disables pre-conditioning during the CG minimization. This option should not be used if atoms heavier than the second-row elements are present.

`TRANSPCG` Specifying this option will enable the use of the Transformed Pre-conditioned CG method, which may result in better SCF convergence in some cases.

At each SCF step, the procedures begins by performing usual CG minimization keeping track of the total step length. If at some micro-iteration the step length exceeds the trust radius, the procedure switches to trust-radius optimization in the reduced space, which, in turn, is halted as soon as the level-shift parameter *mu* has converged. The final step is then calculated as a Newton step in the reduced space of all the trial vectors generated during CG minimization. The following options may be used to modify this behavior.

`NOSWITCHING` Setting this option turns OFF the switching from the normal CG to a trust-radius minimization in reduced space. Using this option helps to reduce the total number of SCF cycles is some cases.

`SHIFTED` Setting this option will turn ON the trust-radius optimization from the first micro- iteration.

`CGITER=cgiter` Sets the maximum number of micro-iterations.

The next two options determine the trust radius.

`TRUSTR=trustr` Initial value for the trust radius. Default: 0.5

`MAXTRUSTR=maxtrustr` The maximum trust radius value. This is set to 0.5 by default and should never be changed.

ARH Notes and Recommendations

Restriction: The method currently works for symmetry NOSYM calculations only. The NOSYM requirement comes from the fact that during direct optimization of the density matrix it may have a symmetry lower than that of the molecule.

The method requires the total energy to be calculated at each step, which makes it much more expensive compared to the standard SCF procedure that does not need the energy. Therefore, the method should only be used when the standard SCF procedure fails. Another complication caused by the use of the total energy is that somewhat higher integration accuracy may be required to get stable SCF convergence, and that the method may not be applicable in all cases. It is also recommended to use the *ADDDIFFUSEFIT* (page 328) keyword or a higher ZlmFit quality setting to increase accuracy of the total energy and, thus, improve convergence. Please refer to the *TOTALENERGY* (page 252) keyword for more information.

10.1.2 Numerical Integration

Many integrals in ADF are calculated via numerical integration. The integration grid used for computing these integrals is therefore an important technical aspect of an ADF calculation, affecting both the accuracy of the results and the computation time.

Becke Grid

The default numerical integration grid is a refined version of the fuzzy cells integration scheme developed by Becke[1]. The implementation in ADF is described in Ref.[2].

The quality of the Becke integration grid can be changed within the BECKEGRID block key:

```
BECKEGRID
   Quality [basic|normal|good|verygood|excellent]
End
```

`Quality` **Default: Normal**. For a description of the various "qualities" and the associated numerical accuracy, see Ref.[?]. The integration grid quality defined in the BECKEGRID block key overrules the *NumericalQuality* (page 315).

Advanced options:

```
BECKEGRID
  {QualityPerRegion
     Region myregion
     Quality {Basic|Normal|Good|VeryGood|Excellent}
  End}
  {qpnear qpnear}
  {RadialGridBoost boost}
End
```

`QualityPerRegion` One can overwrite the integration grid quality for atoms in a particular region. *Example: Multiresolution* (page 1079) illustrates how to use this option.

`qpnear` **Default: 4.0 Angstrom**. Only relevant if you have specified point charges in the input file. ADF generates grids only about those point charges that are close to any real atoms. The criterion, input with the qpnear subkey, is the closest distance between the point charge at hand and any real atom. Any input value is interpreted in the unit-of-length specified with the Units key.

`RadialGridBoost` **Default: 1.0** (or 3 if a numerically sensitive functional is used). The number of radial integration points will be boosted by this factor. Some XC functionals require very accurate radial integration grids, so ADF will automatically *boost* the radial grid for the following numerically sensitive functionals: `LibXC M05`, `LibXC M05-2X`, `LibXC M06-2X`, `LibXC M06-HF`, `LibXC M06-L`, `LibXC M08-HX`, `LibXC M08-SO`, `LibXC M11-L`, `LibXC MS0`, `LibXC MS1`, `LibXC MS2`, `LibXC MS2H`, `LibXC MVS`, `LibXC MVSH`, `LibXC N12`, `LibXC N12-SX`, `LibXC SOGGA11`, `LibXC SOGGA11-X`, `LibXC TH1`, `LibXC TH2`, `LibXC WB97`, `LibXC WB97X`, `MetaGGA M06L`, `MetaHybrid M06-2X`, `MetaHybrid M06-HF`, `MetaGGA MVS`

Notes:

- The space-partition function used in ADF differs from the one described in Ref.[?]. The unnormalized partition function used in the program is defined as (Ω_I is an element-dependent parameter: 0.1 Bohr for H, 0.3 Bohr for He-Xe and 0.6 Bohr for Cs-Og):

---

[1] A.D. Becke, *A multicenter numerical integration scheme for polyatomic molecules*, Journal of Chemical Physics 88, 2547 (1988) (https://doi.org/10.1063/1.454033)

[2] M. Franchini, P.H.T. Philipsen, L. Visscher, *The Becke Fuzzy Cells Integration Scheme in the Amsterdam Density Functional Program Suite*, Journal of Computational Chemistry 34, 1818 (2013) (https://doi.org/10.1002/jcc.23323).

$$\mathcal{P}_{i,U} = \begin{cases} 1 & \text{if } r_{i,U} < \Omega_I \\ 0 & \text{if } \exists j : r_{j,U} < \Omega_J \\ \eta_i \frac{e^{-2(r_{i,U}-\Omega_I)/a_0}}{(r_{i,U}-\Omega_I)^2} & \text{elsewhere} \end{cases}$$

- If either the (block) key `INTEGRATION` or the key `NOBECKEGRID` are used in the input, the Voronoi grid is used:

```
NOBECKEGRID
```

- A Becke grid of normal quality is roughly equivalent (in both absolute accuracy and computation time) to INTEGRATION 4 (Voronoi scheme), and a Becke grid of good quality is roughly equivalent to INTEGRATION 6 (Voronoi scheme).
- The Becke grid is not very well suited to calculate Voronoi deformation density (VDD) charges. For accurate calculation of VDD charges the Voronoi integration scheme is recommended.

Voronoi grid (deprecated)

In ADF2012 and previous versions the default integration scheme was the cellular Voronoi quadrature scheme, implemented by te Velde and Baerends[3],[4]. Starting from ADF2013 this method is no longer the default scheme for integration. The so called Becke grid is the default, see the key BECKEGRID. If the key INTEGRATION is used, the Voronoi scheme will be used. Thanks to a smoother behavior of the relative integration error as a function of the nuclear coordinates, the Becke grid is better suited for geometry optimization and TS search compared to the Voronoi scheme.

The Voronoi integration method requires only one input parameter which determines the precision of numerical integrals and derives from that the number of integration points.

```
INTEGRATION
  accint value
End
```

`accint` A positive real number: the numerical integration scheme generates points and weights such that a large number of representative test integrals are evaluated with an accuracy of accint significant digits. The default accint is 4.0 (10.0 in Create runs).

The *number* of integration points varies strongly with accint, and this determines to a large extent the computational effort. Decreasing accint from 4.0 to 3.0 for instance roughly halves the number of points (this depends somewhat on the molecule).

The defaults should yield good precision for the very large majority of applications. Lower values (3.0 or even 2.0) can be used if precision is not crucial and the purpose is to get an impression. We recommend that you experiment for yourself to get a feel for how results may vary in quality and computing time.

The default in Create mode is very large: 10.0. This is computationally no problem thanks to the simplicity of the single atom case, in particular due to the high symmetry. There is no reason to override the default integration settings when creating basic atoms.

We've now only explained the normal, simple application of the Integration key, which we hope and expect is adequate for all your computations. Next additional details will be discussed. The distribution of points over space is internally regulated by quite a few parameters. Each of these parameters can be controlled in input. By default they depend on one another, and all of them depend on the main parameter accint. Advanced users may wish to experiment and override the default relations between the parameters.

---

[3] P.M. Boerrigter, G. te Velde and E.J. Baerends, *Three-dimensional Numerical Integration for Electronic Structure Calculations*, International Journal of Quantum Chemistry 33, 87 (1988) (https://doi.org/10.1002/qua.560330204)

[4] G. te Velde and E.J. Baerends, *Numerical Integration for Polyatomic Systems*, Journal of Computational Physics 99, 84 (1992) (https://doi.org/10.1016/0021-9991(92)90277-6)

You may also have rather non-standard applications where the default relations are less adequate. A thorough understanding of the underlying method is required to make a sensible choice for all parameters[?],[5].

More options can be defined in the `Integration` block. Consult the literature for detailed information about the various schemes.

```
INTEGRATION
  data
  data
  ...
end
```

The block form is used to override default relations between various parameters that are applied in the generation of the integration grid in the polyhedron method[?]. All these parameters are accessible with subkeys in the data block of Integration. Most of the subkeys are simple keys with one single value as argument; a few subkeys are block-type (sub) keys themselves and hence require the usual format of a data block closed by subend.

`accint` The main precision parameter Its value defines the number of significant digits by which an internal set of standard integrals must be evaluated. The number and distribution of integration points is tuned accordingly. For normal applications this should yield a nearly optimal (given the underlying method) generation of points and weights. The default depends on the run type.

`accsph` The polyhedron method of generating integration points partitions space in atomic polyhedrons, partitioned in pyramids with their tops at the atom in the center of the polyhedron. A core like atomic sphere is constructed around the atom; this truncates the tops of the pyramids. accsph specifies the test precision for the generation of points within the spheres. By default accsph=accint.

`accpyr` Similarly this subkey sets the test level for the parts of the pyramids outside the atomic sphere. Default: accpyr=accint.

`accpyu, accpyv, accpyw` The truncated pyramids are mathematically transformed into unit cubes. A product Gauss integration formula is applied to the cubes, with three (test precision) parameters for the three dimensions. Accpyw controls the direction that is essentially the radial integration from the surface of the atomic sphere to the base of the pyramid. The other two control the orthogonal directions (angular). By default all three equal accpyr.

`accout` The region of space further away from the atoms, outside the polyhedrons, has its own precision parameter. By default accout=accint.

`nouter` This outer region is treated by a product formula: outwards times parallel. The latter involves two dimensions: the surface of the molecule say. The outward integration is performed with Gauss-Legendre quadrature, in a few separate steps. The lengths of the steps are not equal, they increase by constant factors. The total length is fixed. The number of steps is controlled with this subkey; default: 2.

`outrad` The parameter that defines the number of Gauss-Legendre integration points for each outward step. The precise relation between the actual number of points and this subkey, and the default relation between outrad and accout can be found in the implementation.

`outpar` Similarly the integration in the directions parallel to the surface of the atomic system is controlled by a parameter. See the implementation for details.

`dishul` Sets the distance between the outermost nuclei of the molecule and the boundary planes that define the boundary between the polyhedrons and the outer region. By default dishul=2.3*R, where Ris the radius of the largest atomic sphere in the molecule.

`frange` The outward range of the *outer region*: integration is not performed to infinity but to a distance frange from the outermost atoms, where all functions can be assumed to be essentially zero. By default frange is derived both from accint, the general precision parameter, and from the present chemical elements: heavier atoms have longer-range functions than hydrogen say. The precise relations can be found in the implementation.

---

[5] G. te Velde, *Numerical integration and other methodological aspects of bandstructure calculations*, in *Chemistry*. 1990, Vrije Universiteit: Amsterdam.

linrot This parameter is significant only for symmetries with an axis of infinite rotational symmetry: Cand D It is the highest rotational quantum number around this axis that occurs among the integrands. This depends on the employed basis functions and fit functions. By default the program finds this out for itself.

qpnear If you specify point charges in the input file, there are two considerations implied for the numerical integration grid. First, since the point charges create a Coulomb singularity. The integrands (of for instance the basis function products against the Coulomb potential) can only be evaluated with high precision if the grid around the point charges has spherical symmetry and uses local spherical coordinates, exactly as is done for the atomic nuclei. Second, the point charges do not carry fit or basis functions, hence they play only a role in the more diffuse tails of the actual functions involved in integrals. Therefore, a relative low precision of the integral part close to the point charge may have little effect on the total integration accuracy. Since additional 'spherical centers' with their own surrounding grids increase the total number of points significantly, typically a few thousands *per Coulomb center*, this may result in high computational effort. Therefore, the program generates spherical grids only about those point charges that are close to the other atoms. The criterion, input with the qpnear subkey, is the closest distance between the point charge at hand and any real atom. Default 4.0 Angstrom. Any input value is interpreted in the unit-of-length specified with the Units key.

Next come the subkeys that require a list of data. The subkey must be placed on one line, the data on the next. This somewhat peculiar structure suggests that the subkeys are block keys; however their data blocks have no end code (subend) as for normal block type subkeys.

The list of data for such a subkey contains one value for each atom type. The data must be in the order in which the atom types were defined under atoms, implicitly or explicitly: remember that atoms belonging to different fragment types automatically have different atom types, even if their atom type *names* have been specified as identical under atoms.

rspher gives the radii of the atomic spheres, one value for each atom type. By default, the radii are derived from the chemical element (heavier atoms get larger spheres) and from the environment: the sphere must not be too large for the atomic cell (polyhedron).

linteg The maximum angular momentum quantum number of integrands centered on an atom of that type (one value for each atom type). This depends on the basis functions and on the fit functions. By default the program checks the function sets and sets the linteg values accordingly. This subkey is applied for the generation of grid points in the atomic spheres.

Items that relate to geometric lengths (dishul, frange, rspher) must be given in bohr (=atomic units), irrespective of the unit of length defined with units.

Atomic radial grid

For each atom the charge densities and the coulomb potentials of frozen core and valence electrons are computed in a radial grid and stored on adf.rkf. The values in the points of the molecular numerical integration grid are then evaluated by interpolation from the table of radial values.

The radial grid consists of a sequence of r-values, defined by a smallest value, a constant multiplication factor to obtain each successive r-value, and the total number of points. Equivalently it can be characterized by the smallest r-value, the largest r-value, and the number of points; from these data the program computes then the constant multiplication factor. The characteristics are set with

```
RADIALCOREGRID {nrad=points} {rmin=rmin} {rmax=rmax}
```

points The number of radial grid points; default: 5000.

rmin The shortest distance used in the radial grid; default 1e-6 Angstrom

rmax The largest distance in the radial grid; default: 100 Angstrom.

rmin and rmax, when specified, are interpreted as specified in units of length defined by units.

The keyword name radialcoregrid has historical reasons: in earlier releases the radial grid was used only for the frozen core density and potential.

10.1.3 Density fitting

Zlm Fit: density fitting with radial spline functions and real spherical harmonics

Note: In ADF2013 and previous versions, a different density-fitting scheme (pair-fit) was used. Include the key `STOFIT` if you want to use the old fitting scheme.

The basic ideas behind the so-called Zlm Fit can be described as follows. The total electron-density is split into atomic densities (in a similar way as the volume is partitioned for the Becke grid). These atomic densities are then approximated by a combination of radial spline functions and real spherical harmonics (Zlm). The implementation in ADF is described in Ref.[2]. The algorithm that is used in ADF is related to the procedures proposed by Becke[3] and Delley[1]).

The Zlm Fit scheme (which is the default fitting scheme in ADF2016) offers certain advantages compared to the old pair-fit method, especially the possibility to calculate the Coulomb potential to very high precision.

```
ZLMFIT
 Quality {basic|normal|good|verygood|excellent}
 {QualityPerRegion
    Region myregion
    Quality {Basic|Normal|Good|VeryGood|Excellent}
 End}
End
```

`Quality` The default quality of the Zlm Fit is normal. It can be changed with the subkey Quality.

`QualityPerRegion` One can overwrite the Zlm Fit quality for atoms in a particular region. *Example: Multiresolution* (page 1079) illustrates how to use this option.

The Zlm Fit method can be used for most features of the ADF program. For the calculation of Hartree-Fock exchange integrals, ADF uses a different fitting method, see *the section on Hartree-Fock exchange* (page 326).

Pair fit: symmetric density fit

The non-default density fitting procedure in ADF, called pair fit method, is carried out separately for each pair of atoms. To use it one needs to include the keyword STOFIT.

```
STOFIT
```

The implemented approach has several advantages in efficiency but it has a drawback in that it necessitates the use of all available fit functions rather than only the symmetric combinations although the final result of course needs only a symmetric fit because the total density is a symmetric (A1) function. For atoms far apart the density fitting is performed with only symmetric functions. Given the implemented algorithm this entails an approximation which can be tuned:

```
A1FIT atomicseparation
```

`atomicseparation` is the threshold distance between atoms, *in Angstrom*. The symmetric fit approximation is applied only for atoms farther apart. Default is 10.0 Angstrom

Pair fit: fit integrals

---

[2] M. Franchini, P.H.T. Philipsen, E. van Lenthe, L. Visscher, *Accurate Coulomb Potentials for Periodic and Molecular Systems through Density Fitting*, Journal of Chemical Theory and Computation 10, 1994 (2014) (https://doi.org/10.1021/ct500172n)

[3] A.D. Becke, R.M. Dickson, *Numerical solution of Poisson's equation in polyatomic molecules*, Journal of Chemical Physics 89, 2993 (1988) (https://doi.org/10.1063/1.455005)

[1] B. Delley, *An all-electron numerical method for solving the local density functional for polyatomic molecules*, Journal of Chemical Physics 92, 508 (1992) (https://doi.org/10.1063/1.458452)

```
STOFIT
```

For the computation of the Coulomb potential with the pair fit method the program uses a large number of so-called *fit integrals*: the overlap integrals of a fit function with a *product* of two basis functions, where at least two of the involved three functions are centered on the same atom. In fact these are ordinary overlap integrals of STOs because the fit and basis functions are all STOs and a product of STOs on a center is itself also an STO. To use this STO fitting method, which was previously the default, use the key STOFIT in the input of adf (also include it in the create mode of an atom, if that is explicitly used). For the bond energy a first order fit correction term is included, which makes the bond energy accurate to first order in the fit.

Obviously, when the two involved atoms are far enough apart, such overlap integrals become negligibly small. All fit integrals are ignored (and not computed) that are smaller - according to a rough but reasonable estimate - than a preset threshold.

The value of this threshold can be set via input, using the subkey CUTOFF_FIT of the *LINEARSCALING* (page 339) block key word.

True density in XC potential

For the computation of the exchange-correlation potential (XC-potential) the program uses as default the fitted density. This is an approximation. For the XC potential the true density can be used if one includes the keyword EXACTDENSITY:

```
EXACTDENSITY
```

Using the EXACTDENSITY keyword makes the calculation more time-consuming but more accurate in the following cases:

- calculations that require accurate description of virtual orbitals, such as most of the TDDFT;
- when studying systems where weak interaction, such Van der Waals forces and hydrogen bonds, are important. For example, EXACTDENSITY should be switched on when performing geometry optimization of DNA pairs.

Precision of density fitting on standard Output

In the output file of the ADF calculation one can find at the end of the SCF concise information about the density-fit precision: the error integral for the SCF density. The error integral is the integral of the difference between the exact density and the fit density, squared. Such values have very little to do with numerical integration, rather they show whether or not the employed set of fit functions are adequate to describe the SCF density. Error integral values that significantly exceed 1e-4 times the number of atoms are suspicious and may indicate some deficiency in the fit set for the actual calculation. On the last geometry (in an optimization) the fit-error integrals are also printed for the initial (sum-of-fragments) density and the orthogonalized fragments. In the bond energy analysis a 1st-order fit-correction term is given.

10.1.4 Hartree-Fock RI scheme

The implementation in ADF of the calculation of exact exchange (**Hartree Fock exchange**), which is needed for the hybrid functionals, is based on work by Watson et al., Ref.[1]. This procedure is generally referred to as **resolution of the identity** (abbreviated as **RI**). It employs an auxiliary set of functions (**fit functions**) to efficiently approximate the Hartree Fock exchange matrix.

Technical aspects of the RI scheme can be tweaked in the `RIHartreeFock` key block:

---

[1] M.A. Watson, N.C. Handy and A.J. Cohen, *Density functional calculations, using Slater basis sets, with exact exchange*, Journal of Chemical Physics 119, 6475 (2003) (https://doi.org/10.1063/1.1604371)

```
RIHartreeFock
  UseMe               {True|False}
  Quality             {Auto|Basic|Normal|Good|Verygood}
  FitSetQuality       {VeryBasic|Basic|Normal|Good|VeryGood}
  DependencyThreshold rVal
  QualityPerRegion
     Region myregion
     Quality {VeryBasic|Basic|Normal|Good|VeryGood}
  End
End
```

`UseMe` **Default: True**. Set to False if you want to use the old RI scheme.

`Quality` **Default: Auto**. Set the quality of the **FitSet** and of other RI specific technical procedures, including numerical integration and linear scaling parameters. If `Auto`, the quality defined in *NumericalQuality* (page 315) will be used.

`FitSetQuality` The quality of auxiliary fit set employed in the RI scheme (If both `RIHartreeFock%Quality` and `RIHartreeFock%FitSetQuality` are specified, the value specified in `RIHartreeFock%FitSetQuality` will be used). `Normal` quality is generally sufficient for basis sets up to and including TZ2P. Numerical issues have been found, especially if the molecule has symmetry NOSYM and a basis set larger than TZ2P is used. For larger basis sets (or for benchmarking purposes) a `Good` or `VeryGood` fit set is recommended. For larger basis sets than TZ2P it is recommended to include the DEPENDENCY key with a larger than default criterion for the basis set dependence, namely bas=5e-4. **Note that the FitSetQuality heavily influences the computational cost of the calculation.**

`DependencyThreshold` **Default: 1.0E-3**. To improve numerical stability, almost linearly-dependent combination of basis functions are removed from the Hartree-Fock exchange matrix. If you obtain unphysically large bond energy in an Hybrid calculation, you might try setting the DependencyThreshold to a larger value (*e.g.* 3.0E-3). Starting from AMS2022 ADF DependencyThreshold=5e-3 in case of (any variant of) GW, if it is not specified in the input.

`QualityPerRegion` One can overwrite the fit set quality (`FitSetQuality`) for atoms in a particular region. *Example: Multiresolution* (page 1079) illustrates how to use this option.

In ADF2019.3 the fit set for quality GOOD has been improved by adding more diffuse l=0 fit functions to the fit set.

Note:

- In ADF2019.3 the fit set for `FitSetQuality Good` has been improved.
- In ADF2017 and previous versions, a different implementation of the Hartree-Fock RI scheme was used by default. See *Old Hartree-Fock RI scheme* (page 327).

Old Hartree-Fock RI scheme

To use the old Hartree-Fock RI scheme one must specify:

```
RIHartreeFock
  UseMe False
End
```

In ADF2017 and previous versions, a different implementation of the Hartree-Fock RI scheme was used by default.

The old Hartree-Fock RI uses a smaller fit set than the default scheme, and does not include *H* and *I* fit functions; the new scheme should give to more accurate results, especially for f-block elements.

Memory usage

Calculation of the Hartree-Fock exchange may require a lot of memory. Shared memory is used to buffer the necessary data used by all processes on a multi-processor node. By default, ADF will use 30% of the total physical memory of the computer for this buffer, which may be more than is desirable or possible. This amount, in megabytes, can be set using a HFMAXMEMORY input keyword.

```
HFMaxMemory mbytes
```

The amount of memory used is related how many atoms can be done in a single pass. Thus, another way to limit the amount of memory used by ADF is to limit the number of atoms done per pass. The latter can be done using the HFATOMSPERPASS keyword. The safest, but also the slowest, is to set HFATOMSPERPASS to 1.

```
HFAtomsPerPass AtomsPerPass
```

If both HFMAXMEMORY and HFATOMSPERPASS are present, the value specified in HFATOMSPERPASS takes precedence. To debug memory usage in the Hartree-Fock routine, one can use a PRINT HARTREEFOCK keyword.

Numerical issues

Numerical problems have been found with the present implementation of Hartree-Fock or (meta-)hybrids during the SCF, especially if the molecule has symmetry NOSYM and a basis set TZP or larger is used. Workaround is to use always the DEPENDENCY key with rather strict criteria for the basis set dependence, namely bas=4e-3. In ADF2010 these numerical problems have been reduced. The DEPENDENCY key is automatically switched on in the case of a Hartree-Fock or a (meta-)hybrid potential. The result of the DEPENDENCY key is that linear dependence of the basis set is reduced by removing linear combinations that correspond with eigenvalues in the virtual SFOs overlap matrix, which are smaller than, in this case, 4e-3. Note that this is a rather large value, such that it will have an effect on the bonding energy. For DZ and DZP basis sets this value will normally not result in reduction of the virtual space. However, for TZP, TZ2P, QZ4P and larger this will often result in reduction of the basis set, which will have an effect on the accuracy of the bonding energy. In these cases one could try a smaller value than 4e-3, but be aware that numerical problems may occur. If the molecule has symmetry the numerical problems are reduced.

The origin of this problem is that for an accurate description of Hartree-Fock exchange one needs more (diffuse) fit functions in the fit procedure which is used in ADF, which uses only fit functions on the two centers of the two STOs. One can get more diffuse fit functions if one adds in the Create run of an atom the key:

```
AddDiffuseFit
```

If the BASIS key is used one can also add this key in the molecular calculation (the scripts in ADF will then automatically add this in the Create runs of the atoms). If one adds this key preliminary results indicate that one can lower the value for the dependency key to bas=1e-4. Such a low value for the dependency key normally means that the basis set is not reduced for basis sets of TZP or TZ2P quality.

A different way to add fit functions that is useful for standard basis sets DZ, DZP, TZP, and TZ2P is to add the subkey ‘Fittype QZ4P’ in the BASIS key, thus:

```
BASIS
  Type ...
  Core ...
  FitType QZ4P
End
```

In ADF2013 use is made of distance cut-offs for the calculation of the HF exchange integrals. This can reduce computation time and memory usage, especially for large molecules, however, this can also reduce the precision, which can lead to numerical problems. It is possible to set the distance cut-off threshold (starting from ADF2013.01b) for the calculation of the HF exchange integrals, A value of 99 for hffit virtually excludes the possibility that something will be neglected:

```
LINEARSCALING
  HF_FIT hffit
END
```

SCF problems

However, for larger molecules and in case the molecule contains heavy elements (Z>36) one still should use rather strict criteria for the basis set dependence, such as bas=4e-3.

In case of SCF problems that might be related to numerical issues one can try one or more of the following, that were discussed before

```
LINEARSCALING
  HF_FIT 99
END
Basis
 FitType QZ4P
End
AddDiffuseFit
Dependency bas=5e-3
```

10.1.5 MBPT scheme

Note: This page describes technical aspects of the MBPT (Many-Body Perturbation Theory) module which is used in double-hybrid and MP2, RPA, GW and GW-BSE calculations. In order to use double-hybrids, MP2 or RPA in your calculation you should request it in the *XC input block* (page 72). In order to perform a GW calculation, you should request it in the *GW input block* (page 233).

ADF implements RPA, GW, and SOS-MP2 (spin-opposite-scaled) using a newly designed algorithm which in all cases scales quadratically with system size[1][3]. Full MP2 is at the moment only implemented using the canonical RI-algorithm which scales to the fifth power with system size. Thus, we strongly discourage using full MP2 or double-hybrids employing full MP2 for system larger than 1000-1500 basis functions. At the moment ADF features a large number of double-hybrids using SOS-MP2 only (For a list of implemented functionals see *XC input block* (page 72)) which are significantly faster than conventional double-hybrids while offering the same level of accuracy[2].

GW, MP2, RPA and double-hybrid functionals can be used with scalar relativistic effects within the `ZORA`, `X2C`, or `RA-X2C` formalism. GW should not be used in combination with solvent models, like COSMO, or other environments. In ADF2022 and later MP2 and double-hybrid functionals, GW and RPA can be used with spin-orbit coupling. Note that in case of spin-orbit coupling *approximate SS and OS contributions* (page 87) are calculated. In ADF2022 in case of ZORA by default the so called scaled ZORA orbital energies are used in the MBPT expressions.

The Formalism used in the double-hybrid calculation can be changed using the Formalism key. By default, ADF selects the most appropriate algorithm for your system and functional.

The calculation of the independent-particle polarizability or Kohn-Sham density response function in imaginary time is the key step in SOS-MP2, RPA and GW. The equations are solved in the atomic orbital basis exploiting sparsity via advanced density fitting techniques (so-called pair-atomic resolution of the identity or pair-atomic density fitting)[7]. In case of a SOS-MP2 or RPA calculation, the polarizability is than contracted with the Coulomb potential. For SOS-MP2, the correlation energy is then immediately evaluated in imaginary time while in a RPA calculation the product of

---

[1] A. Förster, M. Franchini, E. van Lenthe, L. Visscher, *A Quadratic Pair Atomic Resolution of the Identity Based SOS-AO-MP2 Algorithm Using Slater Type Orbitals*, Journal of Chemical Theory and Computation 16 875-891 (2020) (https://doi.org/10.1021/acs.jctc.9b00854)

[3] A. Förster, L. Visscher, *Low-order scaling |G0W0| by pair atomic density fitting*, Journal of Chemical Theory and Computation 16 (12), 7381 (2020) (https://doi.org/10.1021/acs.jctc.0c00693)

[2] A. Förster, L. Visscher, *Double hybrid DFT calculations with Slater type orbitals*, Journal of Computational Chemistry 41 1660-1684 (2020) (https://doi.org/10.1002/jcc.26209)

Coulomb potential and polarizability is Fourier transformed to the imaginary frequency axis where the correlation energy is evaluated using a matrix logarithm. In a GW calculation, the polarizability is Fourier transformed to the imaginary frequency axis as well where the so-called screened interaction is calculated. The QP states are then evaluated along the real-frequency axis using analytical continuation techniques.

Recommended numerical settings

For all calculations using the MBPT scheme (which includes GW, GW-BSE, RPA, MP2, and double hybrids), we recommend to consider the following points:

Dependency

In ADF2022 a similar method was implemented as was used in the RIHARTREEFOCK scheme to improve stability of the results with the subkey DEPENDENCY of the key MBPT:

```
MBPT
   Dependency Yes/No
End
```

MBPT

Dependency

Type Bool

Default value Yes

Description If true, to improve numerical stability, almost linearly-dependent combination of basis functions are removed from the Green's function that are used in the MBPT equations. Disabeling this key is stringly discouraged. Its value can however be changed. The key to adjust this value is RiHartreeFock%DependencyThreshold

In addition one may remove linear dependencies in the basis set, by using the `Dependency` key. For example

```
DEPENDENCY
   BAS 5e-04
END
```

Augmented basis sets: A value of 5e-3 is recommended
Non-augmented basis sets: Set the dependency key to values between 5e-4 and 5e-3
quasiparticle selfconsistent GW: A value of 5e-3 is recommended
GW-BSE: A value of 5e-3 is recommended

Starting from AMS2022 ADF will use a `Dependency bas=5e-3` and `RIHartreeFock DependencyThreshold=5e-3` in case of (any variant of) GW and GW-BSE. One can override these values in the input.

Numerical Quality

Augmented basis sets: Numerical quality should be `VeryGood`
Non-augmented basis sets, TZ: Numerical quality should range from `Normal` to `VeryGood`
Non-augmented basis sets, QZ: Numerical quality should range from `Good` to `VeryGood`

There is also the option to choose a different numerical quality for the MBPT and the preceding DFT calculation:

```
MBPT
   NumericalQuality Good
END
```

sets the numerical quality to Good only for the MBPT calculation. More info can be found in the *MBPT input block* (page 331) section.

The sizes of the imaginary time and imaginary frequency grids can be controlled with the `nTime` and `nFrequency` keys. For example:

```
MBPT
   nTime 32
   nFrequency 32
END
```

In case of SOS-MP2, `nFrequency` is ignored. The numerical quality automatically sets the number of grid points for imaginary time and frequency integration in case of a GW or RPA calculation:

Numerical quality	Number of points
VERYBASIC	8
BASIC	12
NORMAL	16
GOOD	20
VERYGOOD	24
EXCELLENT	32

For a MP2 or double hybrid calculation, see default is always 9 points, independently of the numerical quality. Note, that the requirements for this parameter are in general lower than for a GW or RPA calculation. Starting from AMS2022, the number of points is automatically adjusted in SOS-MP2 as well.

Note the following: The number of points actually used in a calculation can differ. At runtime, the MBPT algorithm decides what is the maximum number of integration points which is reasonable to use. So the actual number of points which has been used will be equal or smaller. In case the number of points is set by hand, some info is printed in the output. In case of a GW calculation, the numbers of points can also be found in adf.log file in the `GW` section under the names `nTime` and `nFrequency`.

Changing the defaults can be necessary in case your system contains 4th row elements or heavier and/or your basis set is very large and/or your system has a very small HOMO-LUMO gap. For a GW calculation, 24 points should be sufficient for 5th row elements. 32 points might be required for 6th row elements. The maximum number of points which can be used is 42. For a MP2 calculation, 16 points will usually be sufficient if your systems contains 4th row elements, and 20 points will usually suffice in case of 5th row elements. Note, that also a very small HOMO-LUMO.

```
MBPT
   Formalism [Auto | RI | LT | All]
   FrequencyGridType [LeastSquare | GaussLegendre]
   nTime integer
   nFrequency integer
   nLambda integer
   FitSetQuality [Auto | VeryBasic | Basic | Normal | Good | VeryGood]
   IntegrationQuality [VeryBasic | Basic | Normal | Good | VeryGood]
   ThresholdQuality [VeryBasic | Basic | Normal | Good | VeryGood | Excellent]
   Dependency Yes/No
   UseScaledZORA Yes/No
```

(continues on next page)

(continued from previous page)

```
   SigmaFunctionalParametrization [W1 | W2 | S1 | S2 | S1re]
End
```

MBPT

Type Block

Description Technical aspects of the MP2 algorithm.

Formalism

Type Multiple Choice

Default value Auto

Options [Auto, RI, LT, All]

Description Specifies the formalism for the calculation of the MP2 correlation energy.

'LT' means Laplace Transformed MP2 (also referred to as AO-PARI-MP2),

'RI' means that a conventional RI-MP2 is carried out.

If 'Auto', LT will be used in case of DOD double hybrids and SOS MP2, and RI will be used in all other cases.

'All' means that both RI and LT formalisms are used in the calculation.

For a RPA or GW calculation, the formalism is always LT, irrespective of the formalism specified with this key.

FrequencyGridType

Type Multiple Choice

Default value LeastSquare

Options [LeastSquare, GaussLegendre]

Description Use Gauss-legendre grid for imaginary frequency integration in RPA and GW calculations instead of the usually used Least-Square optimized ones. Has the advantage that it can be systematically converged and an arbitrary number of grid points can be used. Typically more grid points will be needed to get the same level of accuracy. However, the convergence of the results with the size of the grid can be more systematic. These grids can only be used when Formalism is set to RI.

nTime

Type Integer

GUI name Number of time points

Description Number of imaginary time points (only relevant in case the Laplace Transformed (LT) formalism is used).

In the many-body-perturbation theory module in ADF, the polarizability (or Kohn-Sham density response function) is evaluated in imaginary time to exploit sparsity in the AO basis. For MP2, this is often referred to as a Laplace transform. For MP2, 9 points are the default. This is a safe choice, guaranteeing accuracies higher than 1 Kj/mol for most systems (For many simple organic systems, 6 points are sufficient for good accuracy).

Only for systems with a very small HOMO-LUMO gap or low-lying core states (heavy elements starting from the 4th row of the periodic table) more points might be necessary.

In principle, the same considerations apply for RPA and GW as well, however, the accuracy requirements are somewhat higher and 12 point are the default for RPA. In a GW calculation, the number of points is adjusted according to the numerical quality. Using less than 9 points is strongly discouraged except for the simplest molecules.

In ADF2019, it can happen that the algorithm determining the imaginary time grid does not converge. In this case, the usual reason is that the number of points is too small and more points need to be specified. Starting from AMS2020, this does not happen any more. In case the imaginary time grid does not converge, the number of points is automatically adjusted until it does.

The computation time of AO-PARI-MP2, RPA, and GW scales linearly with the number of imaginary time points.

`nFrequency`

Type Integer

Default value 12

GUI name Number of frequency points

Description Number of imaginary frequency points. This key is only relevant for RPA and GW and will be ignored if used in an AO-PARI-MP2 calculation. 12 Points is the default for a RPA calculation. It is technically possible to use a different number of imaginary frequency points than for imaginary time. The maximum number of points which can be requested for imaginary frequency integration is 42. Important note: The computation time and memory requirements roughyl scale linearly with the number of imaginary frequency points. However, memory can be an issue for RPA and GW when the number of imaginary frequency points is high. In case a job crashes, it is advised to increase the number of nodes since the necessary memory distributes over all nodes.

`nLambda`

Type Integer

Default value 1

GUI name Number of lambda points

Description Size of coupling constant integration grid for SOSEX variants in RPA. Default is 4 points

`FitSetQuality`

Type Multiple Choice

Default value Auto

Options [Auto, VeryBasic, Basic, Normal, Good, VeryGood]

Description Specifies the fit set to be used in the MBPT calculation.

‘Normal’ quality is generally sufficient for basis sets up to and including TZ2P.

For larger basis sets (or for benchmarking purposes) a ‘VeryGood’ fit set is recommended. Note that the FitSetQuality heavily influences the computational cost of the calculation.

If not specified or ‘Auto’, the RIHartreeFock%FitSetQuality is used.

`IntegrationQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood]

Description Specifies the integration quality to be used in the MBPT calculation. If not specified, the RIHartreeFock%IntegrationQuality is used.

`ThresholdQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description Controls the distances between atomic centers for which the product of two basis functions is not fitted any more. Especially for spatially extended, large systems, 'VERYBASIC' and 'BASIC' can lead to large computational savings, but the fit is also more approximate. If not specified, the RIHartreeFock%ThresholdQuality is used.

`Dependency`

Type Bool

Default value Yes

Description If true, to improve numerical stability, almost linearly-dependent combination of basis functions are removed from the Green's function that are used in the MBPT equations. Disabeling this key is stringly discouraged. Its value can however be changed. The key to adjust this value is RiHartreeFock%DependencyThreshold

`UseScaledZORA`

Type Bool

Default value Yes

Description If true, use the scaled ZORA orbital energies instead of the ZORA orbital energies in the MBPT equations.

`SigmaFunctionalParametrization`

Type Multiple Choice

Default value S1re

Options [W1, W2, S1, S2, S1re]

Description Only relevant if a sigma-functional calculation is performed. Possible choices for the parametrization of the sigma-functional. Not all options are supported for all functionals.

10.1.6 Dependency (basis set, fit set)

Conceivably the sizes of basis and/or fit sets may be so large that the function sets become almost linearly dependent. Numerical problems arise when this happens and results get seriously affected (a strong indication that something is wrong is if the core orbital energies are shifted significantly from their values in normal basis sets). Although for the fit set a few (incomplete) tests are carried out, the program will generally not check such aspects and carry on without noticing that results may be unreliable.

A new feature has been implemented to take care of this. For reasons of compatibility with previous versions and also because our experience with it is limited so far, we have chosen to make application of it not the default.

You have to activate it explicitly. Our experience so far suggests that real problems only arise in case of large basis sets with very diffuse functions (i.e.: not with the normal basis sets provided in the standard package). Starting form ADF2022 it will be automatically activated in case of (any variant of) GW.

Use of the block DEPENDENCY turns internal checks on and invokes countermeasures by the program when the situation is suspect. A few technical (threshold-type) parameters can be set as well, but this is not necessary, assuming that the defaults are adequate.

```
DEPENDENCY
   {bas tolbas}
   {eig BigEig}
   {fit tolfit}
End
```

`tolbas` A criterion applied to the overlap matrix of unoccupied normalized SFOs. Eigenvectors corresponding to smaller eigenvalues are eliminated from the valence space. Default value: 1e-4. Note: if you choose a very coarse value, you'll remove too many degrees of freedom in the basis set, while if you choose it too strict, the numerical problems may not be countered adequately. Starting from AMS2022 ADF will use a rather large value 5e-3 in case of (any variant of) GW, if it is not specified on input.

`BigEig` Merely a technical parameter. When the DEPENDENCY key is activated, any rejected basis functions (i.e.: linear combinations that correspond with small eigenvalues in the virtual SFOs overlap matrix) are normally processed until diagonalization of the Fock matrix takes place. At that point, all matrix elements corresponding to rejected functions are set to zero (off-diagonal) and BigEig (diagonal). Default: 1e8.

`tolfit` Similar to tolbas. The criterion is now applied to the overlap matrix of fit functions. The fit *coefficients*, which give the approximate expansion of the charge density in terms of the fit functions (for the evaluation of the coulomb potential) are set to zero for fit functions (i.e.: combinations of) corresponding to small-eigenvalue eigenvectors of the fit overlap matrix. Default 1e-10.

Notes:

- Application / adjustment of tolfit is not recommended: it will seriously increase the cpu usage while the dependency problems with the fit set are usually not so serious anyway.
- Application of the dependency/tolbas feature should not be done in an automatic way: one should test and compare results obtained with different values: some systems look much more sensitive than others. We have, so far, not been able to understand an unambiguous pattern in these experiences. Of course, when things become clearer in this respect, we will implement the corresponding intelligence into the program.
- When the dependency key is used, the numbers of functions that are effectively deleted is printed in the output file, in the SCF part (cycle 1) of the computation section.
- The adf.rkf (TAPE21) result file of a calculation that used the DEPENDENCY key contains information about the omitted functions and these will also be omitted from the fragment basis when the adf.rkf is used as a fragment file.

10.2 Basis Set Superposition Error (BSSE)

The *Ghost Atom* (page 48) feature enables the calculation of Basis Set Superposition Errors (BSSE). The idea is as follows. In a normal calculation of the bonding energy of a molecule c, composed of fragments a and b, one compares the total energies of c vs. those of isolated a and isolated b added together. In ADF this can be done in one stroke by running c from fragments a and b.

The BSSE is determined as the bonding energies of a pseudo-molecule d composed of (1) a plus a ghost b and (2) b plus a ghost a. The ghost atoms in the calculations are at their normal positions in the true molecule c, and they have their normal basis (and fit) functions. However, they do not have a nuclear charge and no electrons to contribute to the molecule. To set such a calculation up one needs first to make the appropriate ghost basis set files: for each involved atom, copy the basis set file that was used for its creation and modify it so as to remove the frozen core. Next, Create the ghosts with zero mass and zero nuclear charge. Apply these ghost fragments in the BSSE runs.

See also:

- *Example: Basis Set Superposition Error (BSSE): Cr(CO)5 +CO* (page 1064)

- Tutorial: Basis set superposition error (BSSE)

10.3 Control of Program Flow

10.3.1 Limited execution

```
STOPAFTER programpart
```

`programpart` Must be a predefined name associated with a (major) part of the program With this key you tell ADF to terminate the job after the named program part has been executed.

A survey of the recognized names with a brief explanation follows below. The program parts are listed in order of execution: by taking a name further down the list you execute a larger part of the program.

`init` initialization procedure, input reading and printing of the output header with the job identification.

`input` input-reading module.

`geomet` geometry section: organization of atoms in types of atoms and fragments, checks of the actual fragments against information on the attached fragment files.

`config` electronic configuration (if not determined only by the SCF procedure), printout of symmetry subspecies.

`mainsy` generation of symmetry information, representation matrices, etc.

`symfit` construction of symmetry adapted fit functions.

`cblock` generation of integration points and the distribution of them in the blocks that control the internally used segmented vectorization loops.

`engrad` Relevant only in an optimization calculation. Engrad calculates energy gradients. The geometry is not yet updated and no printing of convergence tests and new coordinates is carried out.

`geopt` This routine evaluates energy gradients and updates the geometry accordingly; it also prints the convergence tests and the computed new coordinates. Compare 'stopafter engrad'.

`forcematrix` in a Frequencies run, terminate the calculation when all displacements have been done and before any further processing of the computed Hessian, such as the determination of normal modes, takes place.

10.3.2 Skipping

With the following key you can restrict which parts of the program are actually executed:

```
SKIP argumentlist
```

`argumentlist` A sequence of names, separated by blanks or commas. skip may occur any number of times in input. The names in the argument list refer to various items that are associated with parts of the program. With this key you tell ADF to skip the named program part(s) and to continue execution thereafter. The program does not check any consequences and may even crash when variables have not been initialized or have attained incorrect values due to the skipping.

Use of this key should be contemplated only in debugging and testing sessions, in which you may skip the computation of certain data when before that data will be needed you'll halt the program to inspect something.

Recognized and operational arguments are for instance (possibly not complete due to frequent extensions in this respect): atpair, ets, fitint, orthon, qmpot

10.3.3 Ignore checks

ADF performs several checks during a calculation, and stops with an error message when intermediate results are suspicious, when input-specified instructions are incompatible, etc. These controlled aborts can in some cases be overruled. Of course, the checks have been inserted for good reasons and one should realize that ignoring them probably produces incorrect results and/or may lead to a program-crash.

```
ALLOW argumentlist
```

`argumentlist` A sequence of names, separated by blanks or commas. allow may occur any number of times in input, see the list below for the names that can be used.

`BadCoreInt` Numerical integration of the frozen core density should closely approximate the analytical value. If the deviation is large compared to the user-specified numerical integration precision the program aborts with an error message like 'BAD CORE INTEGRAL'. This control is overruled by using this ALLOW option.

`BadIntegrals` Only applicable when the direct-SCF option is turned off for the basis functions. (This happens automatically for ZORA full-potential calculations). In that case, a sequence of elementary overlap integrals are evaluated with the numerical integration grid and the outcomes tested against the analytical value. If the deviation is too large a warning is issued. Above a certain threshold the program will abort, unless you override the exit with this Allow option.

`BadSCF` If the SCF procedure hasn't converged, any geometry manipulations (optimization, linear transit …) will be aborted because the energy gradients are not reliably computed in a non-self-consistent field.

`CloseAtoms` Atom-atom distances should not be less than 0.2 Bohr. This is checked in the program section where the numerical integration grid is generated.

`RelGeo` Geometry manipulation (optimization, linear transit…) is not supported for all of the relativistic options. See Relativistic

`SmallBlocks` The list of numerical integration points is partitioned in blocks, so as to fit data arrays (for instance values of all basis functions in the points of a block) in available memory. The program computes the maximum block length from available memory and size parameters such as numbers of basis functions. A small block size implies a severe reduction in CPU efficiency. Therefore, the program aborts (by default, to override by this ALLOW option) if the block length turns out to be very small (less than 10).

`xc` Certain combinations of the Density Functional options or application of them with some other features are not allowed. See XC.

10.4 Technical Settings

10.4.1 Memory usage

The amount of memory used by the program during a calculation is determined by three quantities:

- The size of the program itself (executable statements, static arrays). This quantity depends on the program version and is currently around 20 MB.
- Buffer space used by ADF for more efficient I/O handling. This quantity is set at installation. See the Installation Manual.
- Dynamically allocated arrays. The program allocates memory dynamically during the run conform the requirements of the actual calculation.

Starting from ADF2010 in case of parallel calculations some of the data arrays that are used within ADF will be shared by processes on the same node, provided the operating system allows shared memory. This will reduce the total amount of memory used by all ADF processes on each node because only one copy of certain large arrays per node will be present.

Note that shared arrays is not the same as distributed arrays. To disable the use of shared memory one can specify the following keyword:

```
NoSharedArrays
```

10.4.2 Direct SCF: recalculation of data

The program's performance can be defined in terms of the amounts of time (CPU and I/O seconds) and disk space used in a calculation. Also important for the human user is the turn-around time. On multi-user machines CPU-cheap jobs may take a lot of real time to execute due to I/O scheduling. Therefore ADF recomputes some items, like basis functions and fit functions in the integration points, rather than store them on disk. This will increase the amount of CPU time but reduce disk access and it may also improve the turn-around.

The direct SCF method is default and can not be changed. The keys DISK or DIRECTSCF are therefore obsolete and can not be used.

10.4.3 Vector length

Numerical integration is applied in ADF to evaluate Fock matrix elements and many other quantities that are defined as integrals over basis functions, the charge density, the potential, etc. As a consequence a large part of the CPU time is spent in simple do-loops over the integration points. The total number of points depends on the required precision and on the number of atoms, the geometry and symmetry. All such numerical integration loops are segmented into loops over *blocks* of points, each block consisting of a certain number of points. This latter defines the most inner do-loop and hence determines vectorization aspects.

Depending on the computer, c.f. the compiler, vector operations may be executed more efficiently using longer vectors. Long vectors increase the demand on Central Memory however because the program may sometimes have to access large numbers of such vectors in combination (for instance all basis functions) so that they must be available in memory simultaneously. The optimum vector length depends therefore on the balance between vectorization efficiency and memory usage. The maximum vector length that you allow the program to use can be set via input.

```
VECTORLENGTH vectorlength
```

The default is set at the installation of ADF on your platform, see the Installation manual. For organizational reasons the true vector length actually used in the computation may be smaller than the value defined with this key, but will not exceed it (except in a Create run, but in that case performance and memory usage are no hot topics).

10.4.4 Tails and old gradients

The key TAILS is currently obsolescent because of the introduction of the LINEARSCALING block and may be removed in future versions.

Each block of points (see above) covers (more or less) a certain region in space and can hence be assigned a distance value with respect to a particular atom. These distances are used to control whether or not to evaluate functions centered on that atom in that particular block of points.

```
TAILS {bas=tailbas} {fit=tailfit}
```

`tailbas, tailfit` Accuracy levels, similar to the integration parameter: a higher value implies higher precision: in this case, basis functions and fit functions respectively are assumed zero in blocks of points that are at a sufficiently large distance from the atom at which the function is centered. Sufficiently large is defined by comparing the integral of the (radial part of the) function beyond that distance with the total integral. By default tailbas and tailfit both depend on the numerical integration parameter

Note: in contrast with some of the older versions, supplying only the keyword without parameters does not switch off the use of function cutoffs. To effectively switch off the distance effects in gradients evaluation one should specify large values for the BAS and FIT parameters. The value of 100 should be more than enough, thus, for example:

```
TAILS bas=100 fit=100
```

Improved performance in geometry optimizations and frequency runs is achieved by a new implementation of the calculation of the gradients that now uses linear scaling techniques.

The key TAILS is not used in geometry optimizations anymore. For controlling the use of distance effects in normal SCF calculations, and for calculations with the RESPONSE or EXCITATIONS keywords, please check the LINEARSCALING block.

10.4.5 Linearscaling

The LINEARSCALING block has a very similar function to the TAILS keyword described above. In addition to defining the precision of operations related to operations in the numerical integration grid, it also defines the precision for the calculation of the overlap matrix, the fit integrals, and the density fit procedure. Default values have been chosen which result in negligible differences in the results for our test calculations, so that these defaults can be considered safe. They have been chosen similarly to the defaults for the TAILS keyword.

However, it may be advisable to modify the settings for the linear scaling parameters in two cases. First, if a very accurate result is needed, and numerical noise is to be completely eliminated, strict values can be specified. Especially for small molecules, where timings are not so large anyway, this may be of interest. Second, for large molecules, in which the calculations are very time-consuming, one can experiment with less strict values for the LINEARSCALING block keyword. In such a case one should be aware of the reduced accuracy and preferably test the influence of the changes on the results.

In the simplest application of the LINEARSCALING block, only one parameter is provided. All the subkeys described below will then be given this value. A very large value implies a calculation where no distance cut-offs are used. A normal value (almost default situation) would be 8 for linscal, 6 gives a faster but somewhat sloppier result. Whether this is acceptable is strongly case-dependent. A value of 10 or 12 is already quite strict and, unless there are some sort of numerical problems, there should not be much influence on the results by choosing a stricter value than that. A value of 99 for linscal virtually excludes the possibility that something will be neglected.

```
LINEARSCALING linscal
End
```

More refined control is possible by using the full block key

```
LINEARSCALING
  CUTOFF_FIT epsfit
  OVERLAP_INT ovint
  PROGCONV progconv
  CUTOFF_COULOMB epsvc
  CUTOFF_MULTIPOLES epsmp
END
```

CUTOFF_FIT determines how many atom pairs are taken into account in the calculation of the fit integrals and the density fit procedure. If the value is too low, charge will not be conserved and the density fitting procedure will become unreliable. This parameter is relevant for the timings of the FITINT and RHOFIH routines of ADF.

OVERLAP_INT determines the overlap criterion for pairs of AO's in the calculation of the Fock-matrix in a block of points. Indirectly it determines what the cut-off radii for AO's should be. The value of ovint has a strong influence on the timing for the evaluation of the Fock matrix, which is very important for the overall timings. The default value for ovint is accint + 2 (typically 6). Again, a higher value implies a safer but slower calculation.

PROGCONV determines how the overall accuracy changes during the SCF procedure ('progressive convergence'). The idea is that one might get away with a lower accuracy during the initial SCF cycles, as long as the last cycle(s) is/are sufficiently accurate. The current default is that progconv has the value 0, which means that the accuracy in the beginning of the SCF is the same as in the rest of the SCF. This keyword is currently still in the testing phase, so we do not recommend changing its default value. The value of progconv determines how much lower the other parameters in the LINEARSCALING input block are at the beginning of the SCF than at the end.

CUTOFF_COULOMB determines the radii for the fit functions in the evaluation of the (short-range part of) the Coulomb potential. As the Coulomb potential may take a sizable amount of time, the value chosen for epsvc may influence the total ADF timing significantly as well. The default value for epsvc is accint + 4 (typically 8).

CUTOFF_MULTIPOLES determines the cut-offs in the multipole (long-range) part of the Coulomb potential. This term scales quadratically with system size, but has a small prefactor. In most cases, change in the epsmp value will not affect the CPU time significantly. The default value for epsmp is accint + 4 (typically 8).

10.4.6 All Points

ADF makes use of symmetry in the numerical integrations. Points are generated for the *irreducible wedge,* a symmetry unique sub region of space. Optionally the symmetry equivalent points are also used. This is achieved by setting the key

```
ALLPOINTS
```

The key has no argument. The CPU time increases roughly by a factor equal to the number of symmetry operators, and the results should be the same. This key is available only as a debugging feature, to check the correctness of certain symmetry related algorithms.

10.4.7 Full Fock

At every cycle in the SCF procedure the Fock operator is computed in all integration points. By default the *difference* with the values of the previous cycle are used to compute *changes* in the Fock matrix elements. This leads in general to better computational efficiency in two ways: 1) when all such difference values in a block of integration points are very small such a block is skipped in the calculation. 2) if the values are not negligible but still rather small, the contribution from such a block to matrix elements between basis functions with small overlaps are neglected.

With the key

```
FULLFOCK
```

this is turned off, so that the complete matrix elements are computed, no blocks are skipped and the neglect of matrix elements between functions with small overlaps (see also the key TAILS) is controlled solely by the function characteristics and precision requirements, not by the development of the SCF.

10.4.8 Save info

Several types of information, gathered during the run, are lost on exit. The SAVE key allows you to prevent the removal of such information.

```
SAVE info
```

info A sequence of names separated by blanks or commas. save may occur any number of times in the input file. save turns save-info options on. Possible `info`:

TAPE10 File with numerical integration data: points and weights, values of functions (depends on direct-SCF options) and core densities and potentials.

TAPE13 Check point file. This file is lost (by default) only upon normal program exit, i.e. a program-controlled termination (including a program-detected error condition leading to controlled exit). In all such cases all info on TAPE13 is also present on `adf.rkf`. TAPE13 exists when the program crashes into a core dump for instance, in which case it is uncertain what the contents of `adf.rkf` will be. The save feature allows you to specify that TAPE13 is kept *also* upon normal exit.

TAPE14 Scratch file with numerical integration data, mainly pertaining to individual fragments.

TAPE15 Scratch file data related to matrices like the density matrix, the Fock matrix, and the overlap matrix.

CHAPTER

ELEVEN

RESTARTS

11.1 ADF Restart files

When an ADF calculation terminates abnormally - not controlled by the program itself, for instance after a core dump due to some bug - there will usually be a file TAPE13, which serves as a checkpoint file. TAPE13 can be used to restart the calculation at a point not too far before the fatal condition occurred. It contains only data for the restart, but none of the special analysis data on adf.rkf that would be useful for analysis, to serve as fragment file, etc.

TAPE13 is upgraded during the calculation but discarded upon normal termination, namely when all relevant information has been saved on adf.rkf. At that point all info that would have been on TAPE13 is present on adf.rkf. If you wish to keep TAPE13 anyway - for instance because you plan a restart after normal termination and don't intend to keep the substantially bigger adf.rkf - you must use the SAVE key.

Upon normal (i.e. program-controlled) termination of a calculation, the adf.rkf result file can be used for restart purposes. When a crash occurs, however, chances are that adf.rkf has not correctly been closed and that its data structure is inconsistent: during the calculation large portions of adf.rkf are kept in memory rather than on file, and only at the point of final termination, all data is flushed to file.

General remarks

In all restart calculations a normal input file must be supplied (you can, for instance, simply take the original one), with a specification of the restart file added: the restart file does *not replace* the input file. From the program's point of view, it first reads the 'normal' input file and then inspects whether a restart file is present to replace some of the information read from input.

The concept of restarts in ADF is rather simple and primarily directed at increasing computational efficiency by providing cost-expensive data. The continuation run is to a large extent independent from the one that generated the restart file. The runtype, the choice of density-functional and other features in the Hamiltonian, precision of numerical integration, thresholds on convergence, et cetera are all determined solely from the input file for the new run: no such data is read from the restart file. Most input items should, therefore, be supplied in the restart run again, even if it is a direct continuation of a previous calculation: omission implies using the standard defaults, which are not necessarily the settings of the calculation that generated the restart file.

Even the key ATOMS with the list of atomic coordinates must be supplied again: the program needs the information herein to deduce what fragments are used, which coordinates are free or frozen respectively in an optimization, etc. The coordinate *values* may be supplied with the restart file and these will then overwrite those specified in the input file.

Obviously, the two runs cannot be completely unrelated. To let the restart data make sense the runs should correspond to the same molecule (i.e. its general definition in terms of fragment building blocks). The program does not check all aspects related to this and certain abuses will therefore survive the internal tests, but will surely lead to some error later on: it is the user's responsibility to ensure that the restart data match the calculation one has in mind.

Interdependencies between data read from the restart file (rather than from input or fragment files) and other items imply that some input keys and some options to specific keys may be inaccessible when restart data are provided. In most cases supplying such inaccessible input options will simply be ignored; in some cases a warning is issued or an error abort occurs.

A restart file supplies data from a previous run that might be useful in the current one. The applications are (combinations are possible):

- Get a better start in the (first) SCF procedure by providing the electronic charge density (in the form of fit coefficients) from the preceding run,
- Continue an optimization by supplying the latest geometry (coordinates) from a previous run via the restart file (rather than typing them in),

WARNING: The SCF and optimization procedures use *history* to improve convergence behavior. Most of such history information is not stored on a restart file. As a consequence, a restart may not continue exactly as the original run would have done if it hadn't terminated. In a SCF restart, for instance, the DIIS procedure has to rebuild the information. The same holds for geometry optimizations, although history plays usually not a very big role there.

11.2 AMS restart file

In ADF2020 AMS is handling tasks related to exploring the PES. For restarts related to, for example, a Linear Transit, Transition State, IRC or IR Frequencies run, one should look in the AMS manual.

11.3 The restart key

The name of the restart file must be provided via the `EngineRestart` key in the AMS driver part of the input (see below). A list of data items is read from the file (if present on the file and only as far as significant for the new run) and used unless their usage is explicitly suppressed by the user.

Simple usage:

```
$AMSBIN/ams << eor
  ...

  EngineRestart restartfile

  Engine ADF
    ...
  EndEngine
eor
```

`restartfile` The name of a file with restart data. The path (absolute or relative) to the file must be included if the file is not local to the directory where the calculation executes. In most cases it will be a adf.rkf file from an ADF calculation, but this is not necessary. It may be any file - constructed by the user for instance - provided it has the right structure. It must be a kf file and the data to be used must be stored in sections and under variable names, exactly how such data are generated by a normal ADF run on adf.rkf or on the checkpoint file TAPE13. Note: the filename must not be one of the standard filenames used internally by the program, such as adf.rkf, TAPE13 etc. Generally: don't use a name like TAPEnn where nn is a two-digit number.

One can also specify a list of restart options in the `ADF` part of the input:

```
 Engine ADF
   ...
   ...
   Restart
```

(continues on next page)

(continued from previous page)

```
      optionlist
      optionlist
      ...
   End
EndEngine
```

This is the list of possible options:

`noSCF` Do not use any fit coefficients from the restart file as a first approximation to the (fitted) SCF density for the new calculation. Instead, the sum-of-fragments density will be used, as in a non-restart run. Note, typically noSCF should be used in combination with noORB.

`noORB` Do not use orbitals from the restart file.

`SPINFLIP atomnumbers` See the separate *section about the spin-flip method* (page 63) for converging broken-symmetry systems.

Note: In ADF2020 AMS is handling tasks related to exploring the PES. For restarts related to, for example, a Linear Transit, IRC or IR Frequencies run, one should look in the AMS manual.

11.4 Structure of the restart file

All data that may be retrieved from the restart file must be stored in a specific location on the restart file. If you're simply using an adf.rkf (TAPE21) result file or a TAPE13 checkpoint file you don't need to bother about this: ADF has put all data in the right place; the following discussion is primarily for those who want to manipulate the restart file or even construct one themselves.

Since the restart file must be a kf file, the location of the data is of the form Section%Variable, specifying the section and the variable name. The section and variable names are case sensitive. See the utilities document for general information about kf files.

If the specified variable is not present in the specified section on the restart file - or if there is no such section at all - the data is not used, usually without an error message. In some cases a few global tests are carried out on the retrieved data; if they fail the tests the data are not used and a warning - in some cases an error abort - may be issued by the program.

KF files are binary files and so are the adf.rkf result file, the TAPE13 checkpoint file and generally any restart files. If you wish to edit and modify the contents, or just inspect them, the standard KF utilities can be used. Apply pkf to get a survey of the sections and variables on the file, dmpkf to get a complete ASCII version of the file and udmpkf to transform an ASCII version - presumably edited and modified - back into binary format. Please consult the Scripting Section for further information about the standard KF utilities.

AMS tasks

In ADF2020 AMS is handling tasks related to exploring the PES. For restarts related to, for example, a Linear Transit, Transition State, IRC or IR Frequencies run, one should look in the AMS manual.

CHAPTER

TWELVE

RECOMMENDATIONS AND TROUBLESHOOTING

12.1 Recommendations

12.1.1 Precision

The quality of the calculation, *given* the selected model Hamiltonian - density-functional, relativistic features, spin-restricted/unrestricted… - is determined to a large extent by several technical precision parameters.

The most significant ones are:

`Basis set` Obviously, the quality of the basis set may have a large impact on the results. As a general rule, minimum and almost-minimum basis sets (types SZ and DZ) may be used for pilot calculations, but polarization functions should be included (DZP, TZP) for more reliable results.

`SCF convergence` The self-consistent-field (SCF) and geometry optimization procedures terminate when convergence criteria are satisfied. If these are set sloppy the results may carry large *error bars*. The default SCF convergence tolerance is tight enough to trust the results from that aspect. However, when the SCF procedure encounters severe problems an earlier abort may occur, namely if a secondary (less stringent) criterion has been satisfied (see the key SCF). Although this still implies a reasonable convergence, one should be aware that for instance the energy may be off by a few milli-Hartree (order of magnitude, may depend quite a bit on the molecule). It is recommended that in such cases you try to overcome the SCF problems in a secondary calculation, by whatever methods and tricks you can come up with, rather than simply accept the first outcomes. Note: in a geometry optimization the SCF convergence criteria are relaxed as long as the geometry optimization has not yet converged. This should generally not affect the final results: the SCF density and hence the energy gradients may be somewhat inaccurate at the intermediate geometries, but since these are not a goal in themselves the only concern is whether this might inhibit convergence to the correct final geometry. Our experiences so far indicate that the implemented procedure is reliable in this aspect.

`Geometry convergence` This is a far more troublesome issue. Three different types of convergence criteria are monitored: energy, gradients and coordinates. The energy does not play a critical role. Usually the energy has converged well in advance of the other items. The coordinates are usually what one is interested in. However, the program-estimated uncertainty in the coordinates depends on the Hessian, which is not computed exactly but estimated from the gradients that are computed in the various trial geometries. Although this estimated Hessian is usually good enough to guide the optimization to the minimum - or transition state, as the case may be - it is by far not accurate enough to give a reasonable estimate of force constants, frequencies, and as a consequence, neither of the uncertainties in the coordinates. An aspect adding to the discrepancy between the Hessian-derived coordinate-errors and the true deviations of the coordinates from the minimum-energy geometry is that the true energy surface is not purely quadratic and using the Hessian neglects all higher order terms. The gradients provide a better criterion for convergence of the minimizer and therefore it is recommended to tighten the criterion on the gradients, rather than anything else, when stricter convergence than the default is required. The default convergence criteria, in particular for the gradients, are usually more than adequate to get a fair estimate of the minimum energy. Tighter convergence should only be demanded to get more reliable coordinate values (and in particular, when the equilibrium geometry needs to be determined as a preliminary for a Frequencies run).

Numerical integration accuracy The key BeckeGrid key block (or alternatively the old INTEGRATION key block) determines the numerical precision of integrals that are evaluated in ADF by numerical integration, primarily the Fock matrix elements and most of the terms in the gradients. In addition the integration settings also determine several other computational parameters. The demands on numerical integration precision depend quite a bit on the type of application. The SCF convergence seems to suffer hardly from limited integration precision, but geometry convergence does, especially when tight convergence is required and also in transition state searches, which are generally more sensitive to the quality of the computed energy gradients. An extreme case is the computation of frequencies, since they depend on differences in gradients of almost-equal geometries. Frequency calculations on molecules with sloppy modes suggest that a BeckeGrid of "good" quality may be required. Note: a large integration value implies that a lot more points will be used in the numerical integrals, thereby increasing the computational effort (roughly linear in the number of points).

12.1.2 Electronic Configuration

Not specifying occupation numbers in input will *not* automatically result in the computational of the ground state. It may even lead to non-convergence in the SCF and/or in the determination of minimum-energy geometries or transition states. Therefore: whenever possible, specify occupation numbers explicitly in input (key OCCUPATIONS)!

Misunderstanding results of a calculation may easily result from a lack of awareness of how ADF treats the electronic configuration, which orbitals are occupied and which are empty. Unless you specify occupation numbers in input they will be determined from the aufbau principle but only during the first few SCF cycles. Thereafter the distribution of electrons over the different symmetry representations is frozen (see the key OCCUPATIONS, options AUFBAU and aufbau2). If at that point the potential has not yet sufficiently relaxed to self-consistency the *final* situation may be non-aufbau.

A related aspect is that the *ground state* does not necessarily *have* an aufbau occupation scheme. In principle, different competing electronic states have to be evaluated to determine which has the lowest total (strongest bonding) energy.

Check output always carefully as to which orbitals are occupied. In general, whenever possible, supply occupation numbers in input. Be aware that the automatic choice by the program may in a Geometry Optimization result in different configurations in successive geometries: the automatic assessment by the program will be carried out anew in each SCF procedure. If competing configurations with comparable energies have different equilibrium geometries, the geometry optimization has a high failure probability. The gradients computed from the SCF solution of a particular configuration drive the atoms in a certain direction, but in the next geometry, when the program re-determines the occupations and finds a different configuration, the resulting gradients may drive the atoms in another direction.

See the keys CHARGE and OCCUPATIONS for user-control of occupation numbers.

Spin-unrestricted versus spin-restricted, Spin states

If your molecule has unpaired electrons, you should run an unrestricted calculation, in principle. However, if this exhibits convergence problems (or if you simply want to save time: an unrestricted calculation takes a factor 2 more CPU time and data storage), you may consider to do it in two steps. First, run a spin-*restricted* calculation. Then perform a spin-unrestricted calculation using the restricted adf.rkf as a restart file. In the follow-up calculation you should specify the precise occupation numbers for the state you're interested in, *and* use the SCF input key to specify *only one* SCF cycle (iterations=1). This prohibits convergence (so you keep the converged *restricted* orbitals) and gives you a fairly adequate approximation to a converged unrestricted result. See also the H2 example run for a discussion in the Examples document.

An unrestricted calculation does not necessarily yield the multiplet configuration (triple, doublet …). This is a rather complicated matter, see the discussion on multiplet states, key SLATERDETERMINANTS.

12.1.3 What basis set should I use in ADF?

This question is hard to answer in general, but a few general suggestions can be made.

Note: This page discusses the basis set requirements for DFT calculations. For a discussion of this question for a GW calculations, see also the discussion in the *GW page input block* (page 233).

ZORA or non-relativistic calculation?

By default, *scalar relativistic effects with ZORA* (page 97) are included.

If you are doing a ZORA calculation, you will need the ZORA basis sets which can be found in $AMSHOME/atomicdata/ADF/ZORA. You may also use the all electron basis sets from the ET or AUG directory, but be aware that these were optimized to non-relativistic calculations. Currently the ZORA basis sets cover the entire periodic table and besides all electron basis sets offer a choice of frozen cores. At present the ZORA directory does not contain basis sets with very diffuse functions, which may be required in calculations for hyperpolarizabilities or high-lying excitation energies, but for the lighter elements (H-Kr) you can certainly use the all-electron basis sets from the ET or AUG directory. Warning: in a ZORA calculation use only the frozen core basis sets coming from the $AMSHOME/atomicdata/ADF/ZORA directory, or use all electron basis sets.

If you do not use ZORA, your basis sets should come from the directories SZ, DZ, DZP, TZP, TZ2P, or one of the ET or AUG basis sets. For many of the heavy elements only ZORA basis sets are available, but for such elements it would be inadvisable to do non-relativistic calculations anyway. For light elements the ZORA and normal basis sets should be identical except for the description of the frozen core. Usually the ZORA basis sets contain much steeper basis and fit functions to accurately describe the core region.

Large or small molecule?

For standard calculations (energies, geometries, etc.) we recommend the following hierarchy of basis sets: SZ < DZ < DZP < TZP < TZ2P < TZ2P+ < ET/ET-pVQZ < ZORA/QZ4P where the largest and most accurate basis is on the right. Not all basis sets are available for all elements. For small negatively charged atoms or molecules, like F^- or OH^- , basis sets with extra diffuse functions are needed, like they are available in the AUG or ET/QZ3P-nDIFFUSE directories. For example, the standard basis sets, or even the large ZORA/QZ4P basis set will often not be large enough for the accurate calculation of such anions.

In general it is advisable to use the best basis set that you can afford to use in terms of CPU time and memory. If you want to optimize the geometry or calculate the atomization energy of a diatomic molecule there is little reason not to use the very large ZORA/QZ4P basis, or (for light elements) a similarly large ET basis (we recommend the ET-pVQZ basis). If you are studying a molecule with 100 atoms or more, the use of such large basis sets does not only become prohibitive because of the required CPU time and memory, but it also is much less needed than for smaller systems. In medium-sized or large molecules even the moderately large basis sets will prove to be quite adequate because of the effect of basis set sharing. Each atom profits from the basis functions on its many neighbors. Additionally, if a large basis contains diffuse functions, linear dependency problems may occur. See also the input key DEPENDENCY. In many cases basis DZ or DZP will give acceptable accuracy for calculations on large systems. If you are inexperienced it may be prudent to test a few different basis sets to get a feel for the size of basis set effects. To get a rough idea for the size of various basis sets, we mention here the number of functions for all-electron basis sets from the directories ZORA/SZ up to ZORA/QZ4P. For carbon, the number of functions is 5 (basis ZORA/SZ), 10 (DZ), 15 (DZP), 19 (TZP), 26 (TZ2P), 43 (QZ4P). The same numbers for hydrogen are: 1 (SZ), 2 (DZ), 5 (DZP), 6 (TZP), 11(TZ2P, 21(QZ4P). These numbers arise because ADF uses 'pure' d and f functions. In other words, 5 instead of 6 d functions are used and 7 instead of 10 f functions. Note that especially the jump from TZ2P to QZ4P is quite steep.

In an overgeneralizing fashion we can state that the single zeta basis SZ is hardly ever sufficient to get more than a qualitative picture and should be used only when larger basis sets are not affordable. The double zeta basis DZ performs already much better and may give quite reasonable results, for example in geometry optimizations on large molecules. However, in more subtle situations, for example if hydrogen bonds are important, it is advisable to use at least one set of polarization functions. This is the double zeta polarized DZP basis set. Basis set TZP extends the valence space (but not the core space which remains double zeta) to triple zeta. In basis TZ2P an additional polarization function is added. For example, hydrogen gets a d polarization function in addition to its p polarization function and carbon gets an f polarization function on top of a d polarization function. Several tests have shown that often the second polarization function is of more use when it has an l value one higher than the first polarization function. This is reflected in the choice just described. If another polarization function is needed it is usually best to add another one of the lowest l-value (2p+1d for hydrogen, 2d+1f for carbon). This choice has been made in the ET basis ET-QZ3P. There, sometimes even three d polarization functions were added, for example 3 p functions for Be, and 3 d functions for S. The reason for this is that S can occur in hypervalent species such as SF_6 , which put special demands on the basis set. In the case of Be, the unoccupied p level is so close in energy to the occupied ones that it is sometimes called a valence level. Semantics aside, it is clear that a proper description of the p level of Be is very important and it is therefore not strange to add a third p function. In general, the unoccupied levels for the atoms on the left side of the periodic table are close to the occupied ones. This makes it necessary to add a few extra functions for the lowest unoccupied levels in order to get a description which corresponds to the general level of accuracy one expects from the hierarchical basis set names SZ-TZ2P. The basis set quality for a particular subdirectory is now rather uniform throughout the periodic system. At the same time we have attempted to increase the number of functions in a systematic fashion so that each element is described by at least as many functions of a particular l value as its predecessor.

The ZORA/QZ4P basis sets might be roughly described as core triple zeta, valence quadruple zeta, with 4 polarization functions (2 d and 2 f functions for C, 2 p and 2 d for H). The fit sets corresponding to these basis sets are also much larger than the fit sets found in basis sets SZ-TZ2P. If one has doubts about the adequacy of a fit set for a certain element, this can be tested by replacing its fit set by the large one from the QZ4P directory, see the subkey FitType of the key BASIS. In the ZORA/QZ4P basis sets, the choice for the exponents of the polarization functions was done in a systematic, but somewhat hand-waving manner. For this reason the exponents were always rounded to half integers. Also the geometric mean of the exponents usually does not coincide with the choices made in directories SZ-TZ2P and the ET basis sets. However, the fact that two polarization functions (with reasonable exponents) are present instead of a single one is far more important. A reasonable intermediate basis set, in size between TZ2P and QZ4P might be envisaged in which a single polarization function is added, as described above. This is roughly the choice for the polarization functions made in the ET directory ET-QZ3P.

Frozen core or all-electron?

In general we recommend the use of frozen core basis sets for LDA and GGA functionals if available. Especially for the heavier atoms the number of functions is much smaller than for their all-electron counterparts. Our tests indicate that the error made by invoking the frozen core approximation is usually clearly smaller than the difference with respect to slightly higher quality basis sets. For the ZORA/QZ4P basis sets, only all-electron basis sets are available as these are intended for near basis set limit calculations only in which the CPU time is not a major concern.

All electron basis sets are required in case of SAOP, meta-GGA and meta-hybrid functionals, functionals that use LibXC, post-KS calculations like GW, RPA, MP2 or double hybrids. Recommended to use all electron basis sets in case of Hartree-Fock or (range-separated) hybrids. The X2C and RA-X2C relativistic method requires the use of all electron basis sets.

Geometry optimizations involving atoms with a too large frozen core may give rise to numerical problems. In such cases it is recommendable to use a smaller frozen core. In previous occurrences we have removed such basis sets.

For accurate results on properties like nuclear magnetic dipole hyperfine interactions (ESR), nuclear quadrupole coupling constants, and chemical shifts (NMR), all electron basis sets are needed on the interesting atoms. For such properties tight functions might be necessary for high accuracy, especially in a ZORA calculation.

Diffuse functions needed?

For small negatively charged atoms or molecules, like F^- or OH^- , basis sets with extra diffuse functions are needed, like they are available in the AUG or ET/QZ3P-nDIFFUSE directories. For example, the standard basis sets, or even the large ZORA/QZ4P basis set will often not be large enough for the accurate calculation of such anions.

For accurate results on properties like polarizabilities, hyperpolarizabilities, and high-lying excitation energies, also diffuse functions are needed. This is especially true for calculations on small molecules. In larger molecules the nature of the relevant virtuals is much more 'molecular', much less Rydberg-like, so that the normal basis sets may be sufficient. Basically all properties calculated through the RESPONSE keyword may require diffuse functions. If you use the EXCITATIONS keyword, the necessity of diffuse functions depends on the type of excitation you are interested in. The lowest excitations do not require diffuse functions, but Rydberg excitations do.

In case of diffuse basis functions the risk of linear dependency in the basis increases. This can be checked, and corrected for with the DEPENDENCY keyword. It is recommended to use this keyword for all calculations involving diffuse functions. A good default setting is

DEPENDENCY bas=1d-4

However, it may be advisable to experiment a bit with the parameter, especially if many linear dependent combinations of AOs are removed. Using too many diffuse functions on a large molecule will lead to insurmountable numerical problems. In such a case it is not only useless, but even harmful, to add many diffuse functions.

In the previous release only some basis sets were provided which contained diffuse functions. These were gathered in the directory Vdiff. Now several ET basis sets have been developed for the elements up to Ar containing some or many diffuse functions. We recommend to use these instead of the Vdiff directory. Most of these basis sets are quite large and not very suitable for large molecules.

In ADF2005.01 augmented basis sets were added in the AUG directory, especially devised for use in in TDDFT calculations, such that one can do a reasonable accurate calculation of excitation energies, with a relatively small basis set, see D.P. Chong[1].

Normal or even-tempered basis?

For normal calculations (these form the vast majority) we recommend the use of the optimized basis sets in the directories SZ-TZ2P and, for ZORA calculations, ZORA/SZ-QZ4P. These should be sufficient in accuracy for even very demanding users and are available for the entire periodic system (in the case of the ZORA basis sets). They are also available with a frozen core variety, which saves much CPU time.

The ET basis sets on the other hand are available only in all-electron form at the moment. Furthermore, most are pretty large (larger, but also better than TZ2P). Additionally, relatively large basis set superposition errors were detected for molecules containing atoms in the row K-Kr. For this reason we only recommend ET basis sets for the elements H-Ar at the moment. There they have yielded quite nice, near basis set limit, results for the G2 test set. For these light elements the ET basis sets can be comparable in quality to the ZORA/QZ4P basis, even though it is smaller. The ET basis sets are considered to be especially useful when diffuse functions are required. In that case it is very easy to adapt the original ET basis and fit set. The utilities provided for this in ADF will be described below in an Appendix. The ET basis set utilities will also prove useful for users who want to experiment with making their own basis sets, or who have very special demands on the basis and fit. The provided utilities automate much of the work needed to make new atomicdata files.

---

[1] D.P. Chong, *Augmenting basis set for time-dependent density functional theory calculation of excitation energies: Slater-type orbitals for hydrogen to krypton*, Molecular Physics 103, 749 (2005) (https://doi.org/10.1080/00268970412331333618)

Recommendations for Double-hybrids and MP2

Double-hybrid and MP2 calculations are correlated-electrons calculations and consequently they converge slower to the complete basis set limit than independent-electron methods. Also for large molecules, calculations with basis sets smaller than TZ2P are usually not very precise. We thus recommend TZ2P as a minimum requirement. The special, pseudo-correlation-consistent basis sets Corr/TZ3P and Corr/QZ6P usually give better results. They are similar to TZ2P and QZ4P, respectively, but they contain more polarization functions. They can also be ued to extrapolate the **correlation energy** to the complete basis set limit according to

$$E_c^{CBS} = E_c^{QZ} - \frac{1}{N_{bas}^{QZ}} \frac{E_c^{QZ} - E_c^{TZ}}{\frac{1}{N_{bas}^{QZ}} - \frac{1}{N_{bas}^{TZ}}} \; .$$

where E_c^{QZ} and E_c^{TZ} are the values of the correlation energies calculated with Corr/TZ3P and Corr/QZ6P, respectively and N_{bas}^{QZ} and N_{bas}^{TZ} are the corresponding number of basis functions. In a double-hybrid calculation, one should only extrapolate the correlation contribution to the total energy and use the KS contribution from the calculation with the Corr/QZ6P basis set.

The dependency key should always be used also when non-augmented basis sets are used (augmented basis sets should be avoided):

```
DEPENDENCY
   Bas 5e-4
END
```

is usually sufficient.

What accuracy do the basis sets give?

Tests on many diatomics were performed to test the various basis sets. We now document the results of some of these tests, in order to give a feeling for the quality that can be obtained from the various basis sets. See also van Lenthe and Baerends[2].

Tests for non-relativistic calculations on 36 diatomics containing oxygen, namely the oxides of the first 36 elements (H-Kr). All-electron basis sets were used. The ZORA/QZ4P basis set was used to define the basis set limit result. The numbers in the table refer to bonding energies in eV. Differences were taken between the QZ4P results and the results in smaller basis sets. By construction, the errors in the QZ4P column are zero.

	QZ4P	DZ	DZP	TZP	TZ2P
Average error	0.0	1.33	0.39	0.18	0.06
Average absolute error	0.0	1.33	0.39	0.18	0.06
Maximum error	0.0	2.84	1.07	0.41	0.13
Worst case	all	SO	BeO	FO	O_2

A few comments are in order to explain this table.

The oxides were used as a small test set because their equilibrium bond lengths are known in many cases. Also, they have a large influence on the electronic structure of the molecule, so that they also test the adequacy of the polarization functions.

The errors in the small basis sets are systematic, because the isolated atoms are described reasonably well, but the molecular energy is not deep enough. For this reason the average errors and average absolute errors are (nearly) always equal.

---

[2] E. van Lenthe and E.J. Baerends, *Optimized Slater-type basis sets for the elements 1-118*, Journal of Computational Chemistry 24, 1142 (2003) (https://doi.org/10.1002/jcc.10255)

Test calculations on 100 diatomics containing oxygen, using all-electron ZORA basis sets. Many basis sets for (very) heavy elements are included here, which could not be included in the table above. The numbers have the same interpretation as above and are again in eV.

	QZ4P	DZ	DZ	TZP	TZP	TZ2P	TZ2P
	ae	fc	ae	fc	ae	fc	ae
Average error	0.00	0.95	1.07	0.20	0.20	0.05	0.05
Average absolute error	0.00	0.98	1.07	0.20	0.21	0.05	0.05
Maximum error	0.00	2.86	2.83	0.74	0.74	0.19	0.17
Worst case	all	SO	SO	OgO	OgO	ThO	OgO

Again we place a few comments on these frozen core and all-electron results.

The trends are very similar to those in the previous table for the lighter elements.

The frozen core results are very satisfactory, as they are very close to the results with the corresponding all-electron basis sets. The error introduced by the frozen core approximation is typically much smaller than the one introduced by basis set incompleteness.

The average errors are quite comparable to those from the previous table. The heavier elements do not seem to be much more difficult than the lighter ones.

For heavy elements no reliable ET basis set is yet available for comparison.

More results, all-electron, non-relativistic on roughly 140 different diatomics at experimental or 'reasonable' equilibrium geometries.

	QZ4P	DZ	TZP
Average error	0.00	0.89	0.11
Average absolute error	0.00	0.89	0.11
Maximum error	0.00	2.84	0.32
Worst case	all	SO	O_2

Only the non-relativistic basis sets DZ and TZP are fairly complete for heavier elements.

Also for these general diatomics (not just oxides) the average and maximum errors have decreased substantially, especially for basis TZP.

Same table, but now for frozen core basis sets. In all these tests the smallest frozen core files were employed (i.e. the largest basis).

	QZ4P	DZ	TZP
Average error	0.00	0.73	0.13
Average absolute error	0.00	0.75	0.16
Maximum error	0.00	2.87	1.80
Worst case	all	SO	ThO

The frozen core approximation has little influence on the accuracy for the new basis DZ, but a somewhat larger effect on the new basis TZP. This is especially due to certain worst cases, such as ThO.

ZORA, all electron, over 240 diatomics

	QZ4P	DZ	TZP	TZ2P
Average error	0.00	0.70	0.11	0.02
Average absolute error	0.00	0.70	0.11	0.03
Maximum error	0.00	2.83	0.44	-0.16
Worst case	all	SO	I_2	Cr_2

The average error goes down very nicely from 0.70 to 0.11 to 0.03 eV when going from DZ to TZP to TZ2P. The average error in basis TZ2P is clearly below 1kcal/mol (the famous chemical accuracy). Errors due to deficiencies in current xc functionals are still much larger than this. As a consequence, the ZORA/TZ2P basis will be more than adequate for all standard calculations.

It is to be expected that these conclusions will not dramatically change if larger test molecules are used. Also for geometry optimizations the improved basis sets SZ-TZ2P and ZORA/SZ-TZ2P should be more than sufficient for all standard cases. The ZORA/QZ4P can be considered a very safe (though expensive) option for basis set limit calculations.

12.1.4 Relativistic methods

The ZORA relativistic approach is often superior and in other cases at least similar to the older Pauli method. In particular for all-electron calculations generally, and for very heavy elements even within the frozen core approach, the Pauli method may exhibit significant shortcomings. This is mostly due to the variational instability of the Pauli formalism in the deep-core region near the nucleus. The bigger the basis set and the smaller the frozen core, the more likely this will show up, while generally speaking you might be tempted to use smaller cores and bigger basis sets to *improve* your results. The ZORA approach does not suffer from these problems and is, therefore, highly recommended over the Pauli formalism.

12.2 Troubleshooting

This chapter contains hints to help you solve some problems and comments on frequently asked questions.

12.2.1 License file corrupt

You may find that, after having installed the license file, the program still doesn't run and prints a message like 'your license file is corrupt'. To explain how this may come about, and how you overcome this, a few words on license files.

Each license file consists of pairs of lines. The first of each pair is text that states, in more or less readable format typical aspects such as an expiration date, the version number of the software and so on. The second line contains the same information in encrypted format: a (long) string of characters that seem to make little sense. The program reads the license file and checks, with its internal encrypting formulas, that the two lines match. If not, it stops and prints the 'corrupt' message. So, there are two common reasons why it may happen to you:

- You are using a license file for another version of the software than your executables correspond to. Newer (major) releases may contain a different encrypting formula, so that the match in old license files is not recognized anymore. So, please verify that your license file and executable belong to the same major release.
- More likely: the license file as it has been created has been modified in some way. Sometimes, people inspect it and 'clean it up' a little bit, for instance by removing 'redundant' spaces, or by making some other 'improvements'. Unfortunately, every such modification will destroy the encryption match and lead to the 'corrupt' error. Most of the times, however, the reason lies in the mailing system, by which the license file has been sent to you. If the encrypted line is rather long, the mailer may have cut it in two shorter lines. To verify (and correct) this: edit the license file and see if it consists of pairs of lines as described above. If not, re-unify the broken lines and try again.

- Finally, the problem may lie in your OS, which may have inserted additional hidden CR characters (Carriage-Return) into the license file. You can remove them with our fix_license utility (in $AMSHOME/Install), see the Installation manual.

12.2.2 Recover from Crash

A calculation may terminate in two ways: controlled or uncontrolled. Controlled termination includes cases where the program itself detects an error and decides that continuation of the calculation is impossible or pointless. In all such cases the standard exit routine is executed, resulting in an output section with some final information. This also ensures that the general result file adf.rkf is closed properly and all relevant information flushed to it.

Uncontrolled termination may occur, for instance when some bug causes the program to divide by zero, violate memory access restrictions, etc. Usually this leads to an immediate abort of the program by the Operating System and hence loss of control by the program. In such situations the information on adf.rkf may be incomplete because some of the data are kept in memory until the final termination of the program is carried out. It would be a terrible nuisance to see all time spent so far being lost. To remedy this ADF supports a check point file, named TAPE13, to help you recover at least some, if not most, of the results: not for analysis, but for continuation from a point not too long before the fatal condition occurred. TAPE13 can be used, just like adf.rkf, as a normal restart file in the ADF part of the input. See the restart key. For restarting, for example, a geometry optimization, one should use the file ams.rkf, which is produced by the AMS driver, which should be used in the AMS part of the input.

12.2.3 Memory Management

Problem: The program aborts with an error message “MEMORY ALLOCATION ERROR”. This message is issued both in the logfile and in the output file.

Cause: Memory allocation may fail due to:

- Insufficient virtual (i.e. total RAM + swap) memory
- On Unix: too low values for per-process memory limits
- Restrictions of the 32-bit architecture

Cure: Problem 1: add more physical RAM or increase the size of the swap space (page file). Problem 2: add one or more ulimit commands to your run script setting relevant limits to “unlimited”. Problem 3: Perform your calculations on a 64-bit system. ADF version for the most common 64-bit operating systems are available so use them!

All the three problems above can be avoided by reducing the size of the calculation. The most important parameter defining the amount of used memory is the size of the basis set or, more precisely, the total number of Cartesian Slater functions, naos. Current value can always be found in the out file of the calculation, just search for the “naos” string. The amount of memory used by a particular calculation depends on the naos value and of the type of the calculation and, for large naos, it scales as naos^2 . For example, a non-relativistic calculation during SCF can use up to 40 naos^2 bytes of memory. Using spin-orbit coupling may double this amount and using a hybrid or a meta-GGA XC functional will add extra on top of it. Also TDDFT calculations require additional memory.

What can be done to reduce memory usage? First of all, reducing the basis set size for non-critical parts of the molecule will reduce the memory requirement without reducing the quality of the results. Secondly, performing a calculation with a pure GGA instead of B3LYP will not only reduce the amount of memory used but also make the calculation faster. The latter especially applies to geometry optimizations because there B3LYP does not perform any better than some of the GGAs.

Note: If workspace problems occur for relatively small calculations, there might be a bug. Notify your ADF contact: send us the output file so that we can have a look and check things out.

12.2.4 SCF Convergence Guidelines for ADF

The self-consistent field (SCF) method is the standard algorithm for finding electronic structure configurations within Hartree-Fock and density functional theory. SCF is an iterative procedure and can, depending on the situation at hand, sometimes be difficult to converge. Indeed, convergence problems occur in many different types of classes of chemical systems. These problems are most frequently encountered when the electronic structure exhibits a very small HOMO-LUMO gap, in systems with d- and f-elements with localized open-shell configurations, and in transition state structures with dissociating bonds. Finally, many SCF convergence problems are rooted in non-physical calculation setups, such as high-energy geometries or an inappropriate description of the electronic structure.

We advise to check and test the following list of options and aspects when encountering SCF convergence issues, starting from the most common and trivial ones:

1. Ensure that the atomistic system under study is realistic. More specifically, check for the proper values of bond lengths, angles, and other internal degrees of freedom in the geometry. Unless specified otherwise, AMS expects atomic coordinates in Å. Also, when copying atomic structures into the graphical user interface, check that the imported structure is complete and that no atoms got lost during the process.
2. The initial electronic structure is typically initialized as linear combinations of atomic configurations. However, a moderately (but not fully) converged electronic structure from, say, an SCF iteration conducted previously, likely represents a better initial guess already. Indeed in the next step of a geometry optimization, this moderately converged electronic structure information is reused as initial guess. The SCF iterations of the following geometry steps then usually converge even if the first electronic structure calculation does not. In the case of single-point calculations, the electronic structure needs to be read in via a *manual restart* (page 343).
3. Assert that the correct spin multiplicity of the system is used. Open-shell configurations should be computed in a spin-unrestricted or, if necessary, a spin-orbit coupling formalism. It is needed to manually set the spin component (see the tutorial on spin coupling within an iron complex for more details). For non-converging open-shell systems, the evolution of the SCF errors during the iteration might also provide some insight into the problem. Strongly fluctuating errors may indicate an electronic configuration far away from any stationary point or an improper description of the electronic structure by the approximation used.

4. Change to a different SCF convergence acceleration method, like MESA, LISTi or EDIIS. To this end, examine at the *documentation section* (page 316) for further details about the available SCF acceleration methods and the parameters controlling them. In the graphical user interface these options are available under **Details → SCF** and **Details → SCF Convergence Details**. The performance of these methods was tested for a variety of chemical systems that are difficult to converge. The corresponding results are depicted in Fig. 1 below and show that significant changes in the convergence behavior can be achieved with different accelerators.

5. The *Augmented Roothaan-Hall (ARH) method* (page 319) is another alternative, though computationally more expensive, convergence acceleration method. ARH directly minimizes of the systems total energy as a function of the density matrix using a preconditioned conjugate-gradient method with a trust-radius approach. As shown in Fig. 2 below, ARH can in some situations be a viable alternative for difficult systems.
6. Some parameters of the DIIS algorithm may also be changed manually. Note, that sometimes it is necessary to try multiple values and combinations of values until the desired outcome is achieved:

- **Mixing** denotes the fraction of the computed Fock matrix that is added when constructing the next guess for this matrix. More specifically, the Fock matrix resulting from the current guess for the electron density is combined with the corresponding matrices from multiple previous SCF iteration steps by the DIIS algorithm in order to construct the next guess. The mixing parameter controls the proportion of the computed Fock matrix in this linear combination, whereas a higher than the default value of 0.2 corresponds to a more aggressive acceleration, while lower value will lead to a more stable iteration and should be used for problematic cases.
- **Mixing1** corresponds to the mixing parameter used in the very first SCF cycle. Its default of 0.2 should mainly be altered only in situation where one attempts to slowly converge the electronic structure starting from a restart file (see above).

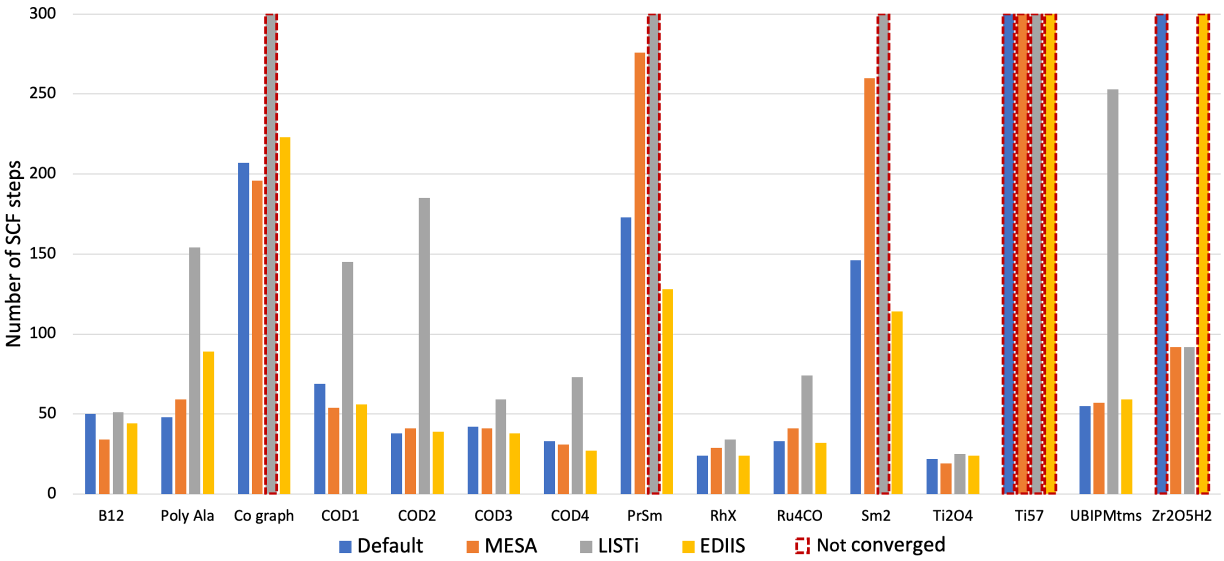


Fig. 12.1: Number of SCF iterations needed to converge a series of different types of chemical systems with the ADIIS+SDIIS (default), MESA, LISTi, and EDIIS accelerator methods. Non-converged SCF iterations are depicted by dashed borders.

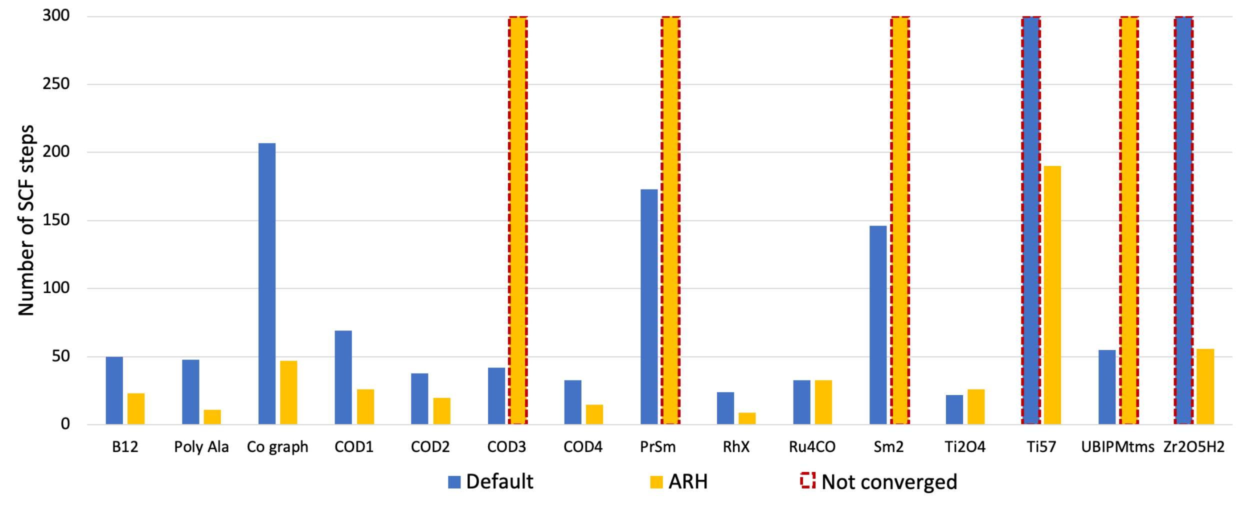


Fig. 12.2: Number of SCF iterations needed to converge a series of different types of chemical systems with the ADIIS+SDIIS (default) and ARH accelerators. Non-converged SCF iterations are depicted by dashed borders.

- **N** is the number of DIIS expansion vectors used for the SCF acceleration (default N=10) by default. An input value smaller than 2 disables the DIIS. A higher number of expansion vectors (e.g. up to N=25) makes the SCF iteration more stable, while a smaller number makes it more aggressive.
- **Cyc** is the number of initial SCF iteration steps after which SDIIS will start (default Cyc = 5). An initial equilibration will take place in the cycles before, through which a higher value of Cyc causes a more stable computation.
- As an example, the following parameter values can be used as a starting point for a slow but steady SCF iteration of a difficult system:

```
SCF
  DIIS
    N 25
    Cyc 30
  End
  Mixing 0.015
  Mixing1 0.09
End
```

Besides the methods discussed above, other algorithms and techniques can be used to converge a problematic SCF calculations. However, as opposed the aforementioned options, the techniques listed below slightly alter the end result, which needs to be carefully tested:

7. *Electron smearing* (page 59) simulates a finite electron temperature by using fractional occupation numbers to distribute electrons over multiple electronic levels. This is particularly helpful to overcome convergence issues in larger systems exhibiting many near-degenerate levels. As electron smearing alters the systems total energy, the value of this parameter should be kept as low a possible, e.g. by using multiple restarts with successively smaller smearing values.
8. The *level shifting technique* (page 318) artificially raises the energy of unoccupied (virtual) electronic levels and can be used to overcome SCF convergence problems as well. It will, however, give incorrect values for properties involving virtual levels, such as excitation energies, response properties, and NMR shifts. Likewise, the electronic structure of metallic systems with a vanishing HOMO-LUMO gap might be inadequately described by this technique.

12.2.5 Geometry Optimization troubleshooting

No convergence

First of all one should look how the energy changed during the latest ten or so iterations. If the energy is changing more or less in one direction (increasing or decreasing), possibly with occasional jumps, then there is probably nothing wrong with the optimization. This behavior is typical in the cases when the starting geometry was far away from the minimum and the optimization has a long way to go. Just increase the allowed number of iterations, restart from the latest geometry and see if the optimization converges.

If the **energy oscillates** around some value and the energy gradient hardly changes then you may need to look at the calculation setup. We give some recommendations below.

The success of geometry optimization depends on the accuracy of the calculated forces. The default accuracy settings are sufficient in most cases. There are, however, cases when one has to increase the accuracy in order to get geometry optimization converged. First of all, this may be necessary if you tighten the optimization convergence criteria. In some cases it may be necessary to increase the accuracy also for the default criteria. Here's what you can do to increase the accuracy of gradients:

- Increase the numerical quality to "good"
- Add an ExactDensity keyword or select "Exact" in the "Density used in XC-potential" list in the Details:Accuracy panel. This will make the calculation 2 or 3 times slower.

- Tighten the SCF convergence criteria, for example, to 1e-8.

Example input with some of these stricter settings using a TZ2P basis set. ExactDensity is not included here since it will make the calculation much slower.

```
NumericalQuality Good
Basis
  Type TZ2P
  Core None
End
SCF
 converge 1e-8
End
```

Small HOMO-LUMO gap: check the HOMO-LUMO gap at the last SCF cycle at a recent geometry. Is it comparable with the changes in the MO energies between geometries? If yes, then it is possible that the electronic structure changes between optimization steps, which may lead to non-convergence. This may indicate a fundamental problem with the calculation setup. First of all, check that you obtain a ground state in a single-point calculation. Is the spin-polarization value correct? Try calculating some of the high-spin states if possible and see if they have lower energy. If the MO repopulation that is taking place is between MOs of different symmetry, then you can try freezing the number of electrons per symmetry using an OCCUPATIONS block.

Are there **constraints** imposed on the optimization? If yes, then check that the constraints do not break the symmetry. ADF automatically preserves symmetry when the starting structure is symmetric and no *Symmetry NOSYM* has been specified. This symmetry preservation may conflict with constraints if the constraints are not specified according to symmetry. For example, if two inter-atomic distances are symmetry-equivalent and you freeze one of them, then you should also freeze the other explicitly.

Optimization in Cartesian coordinates usually needs more steps to converge compared to delocalized. If you haven't tried delocalized optimization for the system then you should do it.

Near 180-degree angles with optimization in delocalized coordinates. ADF normally does not have a problem with a near-180-degree valence angle if the initial value of the angle was larger than 175 degrees or if it is a terminal bond angle. If the initial angle was larger than 175 degrees then it gets a special treatment. For example, any torsion angle that contains the three atoms is removed or replaced with a torsion angle involving only the two end atoms of the near-180-degrees angle. If the angle defines a terminal bond then two angles in two coordinate planes are used to define the bond instead of a valence and a torsion angle. However, if the initial value of the angle was far from 180 degrees and has become close to it during optimization then optimization may become unstable, especially if this is an angle connecting large fragments. In this case, it may be useful to restart geometry optimization from the latest geometry. As a last resort, you may want to constrain the angle to a value close, but not equal, to 180 degrees.

Very short bonds

If the computed equilibrium geometry appears to exhibit unlikely values, typically significantly *too short* bond lengths, you may have run into a basis set problem, in particular (but not only) if the Pauli relativistic method is applied.

Problem: Optimized bond lengths are clearly too short. The energy may also look suspicious.

Possible cause 1: Basis set trouble: onset of Pauli variational collapse, if you have applied the Pauli relativistic option. Caused by small (or absent) frozen cores and/or relatively large basis sets, applied to heavy elements.

Possible cause 2: Basis set trouble also, but quite different from the previous potential cause: you have used relatively *large* frozen cores. When the atoms approach each other during the optimization and the frozen cores start to overlap, the energy computation and the computed energy gradients become more and more incorrect. This is a result of the inappropriateness of the frozen core approximation, which indeed assumes that frozen cores of neighboring atoms do not significantly overlap. Without going into a detailed explanation here, the net effect is that certain repulsive terms in the energy computation are missing and hence a spurious tendency to a 'core collapse' arises, yielding too short bond lengths.

Cure: Best is to abandon the Pauli method and use the ZORA approach instead for any relativistic calculation. If for whatever reason you insist on using the Pauli formalism, apply bigger frozen cores and, if that doesn't help, reduce the basis set (not by deleting polarization functions, but by reducing the flexibility of the occupied-atomic-orbitals space, in particular *s*- and *p*-functions). Note, however, that large frozen cores can be a cause for trouble by themselves, irrespective of any relativistic feature. If you have reason to believe that your frozen cores might be too *large*, given the resulting bond lengths in your calculation, you have to pick smaller cores (and hence be very wary of using the Pauli formalism for any relativity).

12.2.6 Frequencies

Imaginary Frequencies

Problem: totally unexpected significant imaginary frequencies are obtained (in a Frequencies run) where you are pretty convinced that all frequencies should be real.

Possible cause 1: problems with the electronic configuration. If there are competing configurations, the electronic *states* in the different displaced geometries may be different, resulting in energies and gradients belonging to different potential energy surfaces to be compared and combined into force constants (frequencies).

Check: orbital occupations and SCF convergence behavior: if the SCFs in the displaced geometries start with large errors and/or converge very slowly you are likely to have stumbled into different configurations, so that the results from the displaced geometries are incompatible.

Cure: This is a difficult situation that may require some experimenting and judicious manipulation of the various SCF options. The bottom line is that you should try anything you can to ensure that all involved geometries have the same electronic configuration. As long as you fail to achieve this, the results are meaningless.

Possible cause 2: flat potential energy surface (think about almost free rotation modes) coupled with relatively high noise level in gradients caused by numerical integration errors or not sufficiently converged geometry optimization.

Check: visualize the imaginary frequencies in AMSspectra and check that their respective normal modes correspond to movements that are expected to be have (nearly) flat energy profile.

Cure :

- restart geometry optimization with more strict convergence criteria. The default criterion on gradients 0.001 Hartree/Angstrom may be not strict enough for some systems. In such cases a value of 0.0001 is recommended, and for accuracy reasons use "good" numerical quality, and EXACTDENSITY (important for GGA's).

After the AMSification numerical frequencies or analytical frequencies can be computed immediately after a geometry optimization. Example input with strict settings using analytical frequencies, and a TZ2P basis set.

```
$ADFBIN/ams <<eor
   Task GeometryOptimization
   GeometryOptimization
      Convergence gradients=1e-4
   End
   Properties
      NormalModes True
   End
   NormalMode
      Hessian Analytical
   End

   Engine ADF
      NumericalQuality Good
      Basis
```

(continues on next page)

(continued from previous page)

```
         Type TZ2P
         Core None
       End
       ExactDensity
    EndEngine
eor
```

Geometry-displacement numbers in the logfile are not contiguous

Problem: successive displaced geometries in the logfile are numbered, but in your case these numbers make sudden jumps, like '0, 1, 2, 5, 6, 13…'

Cause: you're using Cartesian displacements in a system that has some symmetry in its equilibrium geometry. The program skips the displacements of symmetry-equivalent atomic coordinates to save time. The displacement counts in the logfile do not run over the actually performed displacements but over all atomic coordinates that could be displaced if no use were made of symmetry properties.

Cure: there is no error, don't worry.

12.2.7 Input ignored

Problem: the program doesn't get past input and aborts with a message eof while reading (….). Or the program seems to ignore some parts of input and as a consequence goes wrong somewhere. Or it seems that part of the input has not been read correctly or not at all.

Cause 1. You have used tab characters in your input file. These are not normally visible when you edit your file, but they will affect the program's scanning of the input. When you use tab characters in the input, it is very likely that the program will do something wrong somewhere. Tabs may be ignored by the program, so that items that you believed were separate (by a tab!) are in fact read as contiguous.

Check: the input file on tab characters.

Cause 2: misusage of one of the block-type keys or general keys.

A case that relatively often shows up is typing a title as first line of the input file, *without preceding it by the keyword title*. The program does not understand this as the title, but rather tries to interpret the first word as a keyword. This leads to an error if the first word is recognized as one of the pre-defined block-type keys (possibly abbreviated).

Check: the input file on usage of block-type keys and on proper usage of a title.

Cause 3: incorrect processing of expressions or unintended replacement of names by numerical values. Various kinds of mis-typing or incorrect usage of variables may cause this.

Check: how the program sees input, *after parsing*. This can be done by rerunning the job, with as first line in input : print parser.

This will cause the program to copy each input line *twice* to output, the second time after having parsed it. You may use StopAfter Input or StopAfter Init to let the program quit early so you can inspect what is going on with the input reading.

12.2.8 SFO Populations

In the section that prints the SFO populations of (selected) MOs you may occasionally find, for some SFOs in some MOs, *negative* SFO contributions. This may seem unphysical and hence suspicious, but it is ‘only’ a result of the Mulliken-type analysis method that underlies the computation of the SFO contributions. See the section below that discusses the output file. Likewise for larger-than-100% contributions: don’t worry too much, these numbers may be correct (mathematically, given the Mulliken population formulas).

12.2.9 Error Aborts

The program performs a large number of checks during the calculation and may stop when it detects and error. It is close to impossible to show here a complete list of all possible error messages. In a large number of cases, additional information is printed in the output file to provide a clue as to the cause of the error. It is always useful to carefully inspect the printed info and to try to understand the meaning of any error- or warning messages. If you can’t find your way out, try to get help from your ADF provider. If that fails, contact us directly at support@scm.com

12.2.10 Warnings

The program attempts to detect bugs, instabilities, convergence problems, et cetera and may issue warnings when something looks suspicious. This is not necessarily fatal to your results, but you should be cautious and try to understand what the messages are about. Most warnings are printed in the logfile. Usually there is corresponding and more extensive information in the standard output file.

CHAPTER

THIRTEEN

APPENDICES

13.1 Basis set file format

In `$AMSHOME/atomicdata/ADF` you can find standard basis sets (including fit sets and frozen core orbitals) for all chemical elements of the periodic table at different levels of accuracy. The database is partitioned in subdirectories. Some of these are special: for example, the subdirectory Dirac contains input files for the program *dirac* (computation of relativistic potentials and charge densities). Most subdirectories contain files for the create runs: for example, the subdirectories SZ through TZ2P. *The section about the STO basis sets* (page 38) describes all subdirectories in more details.

The names of the basis set files consist of two parts: the standard symbol for the chemical element and the level of frozen core approximation; e.g. Mn.2p is a basis set file for Manganese with a frozen core up to and including the 2p shell.

Many all-electron basis can be found in the data base, especially for the elements H-Kr. All electron basis sets for the heavier elements can be found in the ZORA subdirectory. Fit functions for the all-electron basis sets must include more, in particular more contracted functions than the standard fit sets that are provided in the frozen core basis set files. If you would combine a basis set with an inadequate fit set the results are unreliable and absolutely inadequate, in the same fashion as when you would have used a highly inadequate basis set.

13.1.1 Sections

The basis set file supplied to ADF in Create mode contains the following sections:

```
Title
Basis Functions
Core Expansion Functions
Core Description
Fit Functions
```

Each of these items is discussed below. The basis set file does *not* define the applied density functional, the electronic configuration, precision parameters (numerical integration, SCF convergence criterion…), etc etc. These items can be set in the normal input file if the default is not satisfactory.

Title The title is the first record of the file. It may contain any text. Only the first 60 characters are actually used. This title is (by default) printed in the output; it is also used to stamp an identification on the result file (adf.rkf). The file stamp will be printed whenever you use it as a fragment file in another calculation.

Basis functions A list of Slater type basis function characteristics. This part has the following format (example):

```
BASIS
  1s 5.4
  2s 1.24
```

(continues on next page)

(continued from previous page)

```
  ...
  (etc.)
  ...
end
```

The words basis and end signal the beginning and the end of this section in the data file. The records in-between list the basis functions; each record contains the main quantum number, the angular quantum number, and the exponential decay factor for a *set* of Slater type basis functions.

The order of specification of the basis functions is not free. First must come the Core Functions used for core-orthogonalization. The CFs must be in order: s-functions first, then p-functions, then d-functions, and finally f-functions (as far as applicable). In the valence basis set there must be exactly one core-orthogonalization function for each frozen core shell (1s, 2s, 2p, …).

Here as well as in all other function definitions below, the unit of length, implicit in the exponential decay factor, is bohr (atomic units), irrespective of the unit of length used in input for geometric items such as atomic positions (see units).

Core expansion functions This part has the form

```
CORE ns, np, nd, nf
  1s 7.68
  ...
  (etc.)
  ...
end
```

It looks very much like the *basis functions*: a list of Slater type function descriptions, closed by end. The header record however (core…) contains in addition four integers ns, np, nd, nf. They are the numbers respectively of s-, p-, d-, and f- frozen core shells in the atom. If you create for instance a Ruthenium atom with a frozen core up to the 4p shell, these numbers would be `4 3 1 0`: four frozen s-shells (1s,2s,3s,4s), three frozen p-shells (2p,3p,4p), one frozen d-shell (3d), and no frozen f-shells.

The core expansion sets defined in this section are used to describe the frozen core orbitals; they are not included in the valence basis set. In the list of core expansion sets all s-type functions must come first, then the p-type functions, then the d-functions, and then the f-functions (as far as applicable).

Core description Describes the frozen core shells as linear combinations of the core expansion functions. This section has the form

```
COREDESCRIPTION
  coefficients for the first frozen s-shell
  for the second s-shell
  for the n-th shell
  coefficients for the first frozen p-shell
  for the second p-shell
  for the d-shells
  for the f-shells
  pseudopotential parameters
end
```

For each of the angular momentum quantum numbers *l=s, p, d, f* all n *l* frozen shells are described by giving expansion coefficients. There are as many coefficients as there are function *sets* with the pertaining *l* value in the list of expansion functions. There are no separate coefficients for all *m* values: all *m* values are equivalent in a spherically symmetric model atom. See the Ca example below.

At the end of the (core) description section there is a record with pseudopotential parameters. The pseudopotential option, as an alternative to the frozen core approximation, is currently not supported, all values in this record must be zero, one for each frozen core shell. Equivalently you can put one zero, followed by a slash (/).

Fit functions is again a list of Slater type functions. These are used for an expansion of the density. The Coulomb potential due to the electronic charge distribution is computed from this expansion.

The format of this section is similar to the *basis functions*:

```
FIT
  1s 10.8
  ...
  (etc.)
  ...
end
```

The program cannot handle fit functions with *l*-value higher than 4, i.e. not higher than *g*-type functions. Bear this in mind if you construct alternative fit sets.

13.1.2 Example of a basis set file: Calcium

An example may serve to illustrate the format of a Create data file for DZ/Ca.2p (empty records inside and between the various sections are meaningless and ignored):

```
Calcium (DZ, 2p frozen)

BASIS
 1S  15.8
 2S   6.9
 2P   8.1

 3S   2.6
 3S   3.9
 3P   2.1
 3P   3.4
 4S   0.8
 4S   1.35
 4P   1.06

 3D   2.000
END

CORE    2  1  0  0
 1S  24.40
 1S  18.25
 2S   7.40
 2S   4.85
 3S   4.00
 3S   2.55
 4S   0.70
 4S   1.05
 4S   1.65
 2P  10.85
 2P   6.45
 3P   1.85
 3P   2.70
 3P   4.00
END
```

(continues on next page)

(continued from previous page)

```
DESCRIPTION
  0.2076143E+00  0.7975138E+00 -0.7426673E-04  0.1302616E-03 -0.6095738E-04
  0.1508446E-04  0.1549420E-06 -0.2503155E-07 -0.1843317E-05
  0.8487466E-01 -0.4505954E+00  0.1009184E+01  0.9627952E-01 -0.3093986E-01
  0.1678301E-01 -0.2381843E-02  0.6270439E-02 -0.8899688E-02
  0.3454503E+00  0.6922138E+00 -0.1610756E-02  0.5640782E-02 -0.5674517E-02

0/
END

FIT
 1S  31.80
 2S  29.37
 3S  25.15
 4S  21.06
 4S  13.99
 5S  11.64
 5S   8.05
 6S   6.69
 6S   4.76
 6S   3.39
 7S   2.82
 7S   2.06
 7S   1.50
 2P  24.10
 3P  14.78
 4P   9.29
 5P   5.98
 6P   3.94
 6P   2.24
 7P   1.50
 3D  16.20
 4D  10.47
 5D   6.91
 6D   4.65
 6D   2.70
 7D   1.85
 4F   7.00
 5F   4.00
 5G   3.50
END
```

13.1.3 Extending a basis set

Polarization functions are provided for most elements. We strongly recommend to use one of the default basis sets described *here* (page 38).

If you are considering making your own basis set by including one or more polarization functions, a good rule of thumb to choose the function characteristics is the following: take the next higher *l*-value that does not yet occur in the function set (however, do not go beyond *f*-functions: the program cannot (yet) handle *g*-type basis functions), select the minimum value for the main quantum number n that is compatible with the *l*-value (i.e.: 2p, 3d, 4f), and determine the exponential decay factor ζ, such that the function attains its maximum value at somewhere between 1/3 and 1/2 times the bond length. The functional maximum for a Slater-type function is at $R = (n-1)/\zeta$. The maximum for r^2 times the square of a Slater-type function is at $R = n/\zeta$.

13.2 Elements of the Periodic Table

A few characteristics are predefined in ADF for all elements of the periodic table, as shown below.

The electronic configuration defines the default occupation numbers in Create mode. Basis sets for the elements Bk-Ubn (Z=97-120) are only available in `$AMSHOME/atomicdata/ADF/ZORA/`.

	Nuclear Charge Z	mass number of default isotope used for mass	electronic configuration
H	1	1	$1s^1$
He	2	4	$1s^2$
Li	3	7	$2s^1$
Be	4	9	$2s^2$
B	5	11	$2s^2$ $2p^1$
C	6	12	$2s^2$ $2p^2$
N	7	14	$2s^2$ $2p^3$
O	8	16	$2s^2$ $2p^4$
F	9	19	$2s^2$ $2p^5$
Ne	10	20	$2s^2$ $2p^6$
Na	11	23	$3s^1$
Mg	12	24	$3s^2$
Al	13	27	$3s^2$ $3p^1$
Si	14	28	$3s^2$ $3p^2$
P	15	31	$3s^2$ $3p^3$
S	16	32	$3s^2$ $3p^4$
Cl	17	35	$3s^2$ $3p^5$
Ar	18	40	$3s^2$ $3p^6$
K	19	39	$4s^1$
Ca	20	40	$4s^2$
Sc	21	45	$3d^1$ $4s^2$
Ti	22	48	$3d^2$ $4s^2$
V	23	51	$3d^3$ $4s^2$
Cr	24	52	$3d^5$ $4s^1$
Mn	25	55	$3d^5$ $4s^2$
Fe	26	56	$3d^6$ $4s^2$
Co	27	59	$3d^7$ $4s^2$
Ni	28	58	$3d^9$ $4s^1$, $3d^8$ $4s^2$
Cu	29	63	$3d^{10}$ $4s^1$
Zn	30	64	$3d^{10}$ $4s^2$
Ga	31	69	$3d^{10}$ $4s^2$ $4p^1$
Ge	32	74	$3d^{10}$ $4s^2$ $4p^2$
As	33	75	$3d^{10}$ $4s^2$ $4p^3$
Se	34	80	$3d^{10}$ $4s^2$ $4p^4$
Br	35	79	$3d^{10}$ $4s^2$ $4p^5$
Kr	36	84	$3d^{10}$ $4s^2$ $4p^6$
Rb	37	85	$5s^1$
Sr	38	88	$5s^2$
Y	39	89	$4d^1$ $5s^2$
Zr	40	90	$4d^2$ $5s^2$
Nb	41	93	$4d^4$ $5s^1$

continues on next page

Table 13.1 – continued from previous page

	Nuclear Charge Z	mass number of default isotope used for mass	electronic configuration
Mo	42	98	$4d^5$ $5s^1$
Tc	43	[98]	$4d^5$ $5s^2$
Ru	44	102	$4d^7$ $5s^1$
Rh	45	103	$4d^8$ $5s^1$
Pd	46	106	$4d^{10}$
Ag	47	107	$4d^{10}$ $5s^1$
Cd	48	114	$4d^{10}$ $5s^2$
In	49	115	$4d^{10}$ $5s^2$ $5p^1$
Sn	50	120	$4d^{10}$ $5s^2$ $5p^2$
Sb	51	121	$4d^{10}$ $5s^2$ $5p^3$
Te	52	130	$4d^{10}$ $5s^2$ $5p^4$
I	53	127	$4d^{10}$ $5s^2$ $5p^5$
Xe	54	132	$4d^{10}$ $5s^2$ $5p^6$
Cs	55	133	$6s^1$
Ba	56	138	$6s^2$
La	57	139	$5d^1$ $6s^2$
Ce	58	140	$4f^1$ $5d^1$ $6s^2$
Pr	59	141	$4f^3$ $6s^2$
Nd	60	142	$4f^4$ $6s^2$
Pm	61	[145]	$4f^5$ $6s^2$
Sm	62	152	$4f^6$ $6s^2$
Eu	63	153	$4f^7$ $6s^2$
Gd	64	158	$4f^7$ $5d^1$ $6s^2$
Tb	65	159	$4f^9$ $6s^2$
Dy	66	164	$4f^{10}$ $6s^2$
Ho	67	165	$4f^{11}$ $6s^2$
Er	68	166	$4f^{12}$ $6s^2$
Tm	69	169	$4f^{13}$ $6s^2$
Yb	70	174	$4f^{14}$ $6s^2$
Lu	71	175	$4f^{14}$ $5d^1$ $6s^2$
Hf	72	180	$4f^{14}$ $5d^2$ $6s^2$
Ta	73	181	$4f^{14}$ $5d^3$ $6s^2$
W	74	184	$4f^{14}$ $5d^4$ $6s^2$
Re	75	187	$4f^{14}$ $5d^5$ $6s^2$
Os	76	192	$4f^{14}$ $5d^6$ $6s^2$
Ir	77	193	$4f^{14}$ $5d^7$ $6s^2$
Pt	78	195	$4f^{14}$ $5d^9$ $6s^1$
Au	79	197	$4f^{14}$ $5d^{10}$ $6s^1$
Hg	80	202	$4f^{14}$ $5d^{10}$ $6s^2$
Tl	81	205	$4f^{14}$ $5d^{10}$ $6s^2$ $6p^1$
Pb	82	208	$4f^{14}$ $5d^{10}$ $6s^2$ $6p^2$
Bi	83	209	$4f^{14}$ $5d^{10}$ $6s^2$ $6p^3$
Po	84	[209]	$4f^{14}$ $5d^{10}$ $6s^2$ $6p^4$
At	85	[210]	$4f^{14}$ $5d^{10}$ $6s^2$ $6p^5$
Rn	86	[222]	$4f^{14}$ $5d^{10}$ $6s^2$ $6p^6$
Fr	87	[223]	$7s^1$
Ra	88	[226]	$7s^2$

continues on next page

Table 13.1 – continued from previous page

	Nuclear Charge Z	mass number of default isotope used for mass	electronic configuration
Ac	89	[227]	$6d^1\ 7s^2$
Th	90	232	$6d^2\ 7s^2$
Pa	91	231	$5f^2\ 6d^1\ 7s^2$
U	92	238	$5f^3\ 6d^1\ 7s^2$
Np	93	[237]	$5f^4\ 6d^1\ 7s^2$
Pu	94	[244]	$5f^6\ 7s^2$
Am	95	[243]	$5f^7\ 7s^2$
Cm	96	[247]	$5f^7\ 6d^1\ 7s^2$
Bk	97	[247]	$5f^9\ 7s^2$
Cf	98	[251]	$5f^{10}\ 7s^2$
Es	99	[252]	$5f^{11}\ 7s^2$
Fm	100	[257]	$5f^{12}\ 7s^2$
Md	101	[258]	$5f^{13}\ 7s^2$
No	102	[259]	$5f^{14}\ 7s^2$
Lr	103	[266]	$5f^{14}\ 6d^1\ 7s^2$
Rf	104	[267]	$5f^{14}\ 6d^2\ 7s^2$
Db	105	[268]	$5f^{14}\ 6d^3\ 7s^2$
Sg	106	[269]	$5f^{14}\ 6d^4\ 7s^2$
Bh	107	[270]	$5f^{14}\ 6d^5\ 7s^2$
Hs	108	[269]	$5f^{14}\ 6d^6\ 7s^2$
Mt	109	[278]	$5f^{14}\ 6d^7\ 7s^2$
Ds	110	[281]	$5f^{14}\ 6d^8\ 7s^2$
Rg	111	[282]	$5f^{14}\ 6d^9\ 7s^2$
Cn	112	[285]	$5f^{14}\ 6d^{10}\ 7s^2$
Nh	113	[286]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^1$
Fl	114	[289]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^2$
Mc	115	[290]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^3$
Lv	116	[293]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^4$
Ts	117	[294]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^5$
Og	118	[294]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^6$
Uue	119	[315]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^6\ 8s^1$
Ubn	120	[299]	$5f^{14}\ 6d^{10}\ 7s^2\ 7p^6\ 8s^2$

Default (most abundant) isotope, used to set atomic mass (nr. of brackets gives mass directly). Default electronic configurations used in Create mode.

13.3 Multiplet States

Calculations with the ADF engine yield results for one-determinant electronic states, which are not always the 'true' states of the molecule. The evaluation of the correct multiplet energies is not trivial in this approach, see further below the section on multiplet energies. The point is to evaluate a specific multiplet state as a linear combination of selected one-determinant functions, each computed in the field of the so-called Average-of-Configuration (AOC). Typically, in an open shell system, the AOC is the spin-restricted system in which all orbitals in the open shell are degenerate and equally occupied. The AOC serves then as a fragment for the subsequent calculations, in which the different open shell orbitals are occupied differently by specifying the appropriate occupation numbers as explained below.

Important: in these follow-up calculations it is imperative that the results are obtained in the AOC field: no SCF convergence must be carried out, because we only want to assign the electrons differently, while keeping exactly the AOC orbitals. To achieve this, the follow-up calculations must use the keyword SCF, and the subkey iterations must be set to 0.

Since the ADF engine requires that the point-group symmetry matches not only to the nuclear frame but also to the electronic charge density and MO occupations, these calculations must run in a lower point group symmetry. Often you will also want to run the modified calculations spin-*un*restricted. For an example, see the set of sample runs that come with the package and the discussion in the *Examples* (page 401).

The calculation of the one-determinant states based on the AOC reference state is controlled with the block `SLATERDETERMINANTS`,

```
SLATERDETERMINANTS file
End
```

were the argument must be a file (including the path). The file must be an ASCII file containing data in the same format as you would supply in the data block, see below. All information on the file until the *eof* must be suitable for the data block, but no record 'end' on the file must be specified: only the *contents* of the data block.

Alternatively, you can use the SLATERDETERMINANTS block as follows

```
SLATERDETERMINANTS
 title1
    irrep occups
    irrep occups
    ...
 subend
 title2
    irrep occups
    ...
 subend
 title3
    ...
 subend
 ...
end
```

Each 'title' functions as a subkey, but is otherwise an arbitrary string to label the resulting one-determinant calculation. Each such subkey block contains the occupation numbers for a single one-determinant calculation. It is necessary that the calculation uses the reference AOC run as its only fragment file. The occupations in the subkey blocks must be rearrangements of the AOC open-shell electrons. In the Slaterdeterminants calculation you must explicitly specify the point group symmetry in which you want to run; this must be a lower symmetry than the AOC one, otherwise you couldn't rearrange the open shell electrons. See the sections below on multiplet energies. An sample run is included in *Examples* (page 401).

Each 'irrep occups' record specifies the occupations for the indicated irrep in the usual way (see for instance the occupations key). The irrep labels must correspond to the (lower) point group symmetry used in the slaterdeterminants calculation. Note that in an unrestricted calculations, occupations numbers must be given for both spins, using the double slash (//) to separate the occupations for spin-α and spin-β.

In this setup, the program will for each of the subkey blocks under the slaterdeterminants key execute an SCF calculation with only one cycle, i.e. no convergence, where the start-up field is the fragment field, i.e. the AOC field. So all one-determinant states in this calculation are evaluated in the AOC field. The resulting energies for the distinctly computed one-determinant states can then be combined to the desired multiplet values, corresponding to how the multiplet states are combinations of the one-determinant states.

13.3.1 Multiplet energies

The energies of atomic and molecular multiplet states that correspond to a given electron configuration can be calculated approximately with the method suggested in ref.[1]. There it is indicated that it would not be justified to take an arbitrary configuration-state function (CSF), defined in general as a linear combination of determinants that has specific spin and space symmetry properties, and use the corresponding alpha and beta spin densities in a DFT energy expression. The same holds true for the densities corresponding to the average-of-configuration (see section 'DFT energy of a one-determinantal wave function', the 'procedure' notes).

Therefore, we restrict ourselves to just computing the DFT energies of single-determinant wavefunctions. Usually (but not always) this is sufficient information to obtain the multiplet energies. The procedure, which is explained in[?], requires knowledge of the CSFs belonging to a given configuration. This means that a multiplet state with specific L, ML and S, MS values has to be written as a linear combination of the determinant wavefunctions that belong to the given configuration. In Ref.[2] Claude Daul exploited symmetry in order to simplify the relation between the multiplet energies and single-determinant energies.

It is often advantageous to search for CSFs that consist of one determinant only, since the energy of this determinant should correspond directly to the multiplet energy. Sometimes there is redundancy in the information and there may even be some inconsistency: two determinants may exist that both are CSFs belonging to the same multiplet state but yield somewhat different energies. We will illustrate this for the Carbon atom example treated below.

The discrepancies are a measure of 'error bars' associated with the theoretical multiplet energies. As a matter of fact, there are certain symmetry relations between the energies of the determinants of a configuration, calculated as the expectation value of the determinant for the full many-electron Hamiltonian. An example is the equal energy for the determinants of a p^1 configuration, whether the electron is placed in the p_0 (= p_z) orbital or in the $p_{+1}(= (p_x + ip_y)/\sqrt{2})$ orbital. This equality is not obtained with present-day density functionals, leaving an ambiguity ('error bar') in the determination of the energy. A more complete treatment of the symmetry relations between determinant energies is given in[?].

Warning: However, there have occasionally been found inconsistent results. A comparison to the results obtained by the procedure outlined in[?] may show significant differences and the latter seem more accurate and consistent.

DFT energy of a one-determinantal wave function

The determinant corresponds to a well defined ρ^α and ρ^β. Suppose we are dealing with a d^2 configuration and we wish to know the energy of

$$D_1 = |d_2^\alpha(1) d_1^\alpha(2)|$$

where d_m has the Y_{2m} angular part. This determinant is a CSF of the ^3F multiplet

$$D_1 = |^3F; M_L = 3; M_S = 1\rangle$$

We can easily transform to the real spherical harmonics that are used in the ADF engine:

$$Z_{lm}^C = \frac{1}{\sqrt{2}}(Y_l^{-m} + Y_l^{-m*}) = \frac{1}{\sqrt{2}}(Y_l^{-m} + (-1^m)Y_l^m)$$

$$Z_{lm}^S = \frac{i}{\sqrt{2}}(Y_l^{-m} + Y_l^{-m*}) = \frac{i}{\sqrt{2}}(Y_l^{-m} + (-1^m)Y_l^m)$$

with back transformations:

$$Y_l^m = \frac{1}{\sqrt{2}}(-1)^m(Z_{lm}^C + iZ_{lm}^S)$$

$$Y_l^{-m} = \frac{1}{\sqrt{2}}(Z_{lm}^C - iZ_{lm}^S)$$

---

[1] T. Ziegler, A. Rauk and E.J. Baerends, *On the calculation of Multiplet Energies by the Hartree Fock Slater method*, Theoretica Chimica Acta 43, 261 (1977) (https://doi.org/10.1007/BF00551551)

[2] C. Daul, *DFT applied to excited states*, International Journal of Quantum Chemistry 52, 867 (1994) (https://doi.org/10.1002/qua.560520414)

Here the superscripts c and s stand for the cosine, respectively sine type of combinations of $exp(-im\phi)$ and $exp(im\phi)$. This yields explicitly:

$$\begin{aligned}
d_{z^2} &= d_0 \\
d_{xz} &= 1/\sqrt{2}(d_{-1} - d_1) \\
d_{yz} &= i/\sqrt{2}(d_{-1} + d_1) \\
d_{x^2-y^2} &= 1/\sqrt{2}(d_{-2} + d_2) \\
d_{xy} &= i/\sqrt{2}(d_{-2} - d_2) \\
d_0 &= d_{z^2} \\
d_1 &= -1/\sqrt{2}(d_{xz} + id_{yz}) \\
d_{-1} &= 1/\sqrt{2}(d_{xz} - id_{yz}) \\
d_2 &= 1/\sqrt{2}(d_{x^2-y^2} + id_{xy}) \\
d_{-2} &= 1/\sqrt{2}(d_{x^2-y^2} - id_{xy})
\end{aligned}$$

For D_1 we obtain:

$$\begin{aligned}
&\rho_\alpha = |d_2|^2 + |d_1|^2 = 1/2|d_{x^2-y^2}|^2 + 1/2|d_{xy}|^2 + 1/2|d_{xz}|^2 + 1/2|d_{yz}|^2 \\
&\rho_\beta = 0
\end{aligned}$$

The fractional occupations have to be used in order to generate the densities ρα and ρβ and the corresponding density matrices ρ_α and ρ_α. The density matrices can be used to calculate the energy of D1 (and ${}^3F$) with respect to the energy of the 'master fragment', which should be the restricted atom with d^2 occupation. Other determinants of this configuration can be treated similarly to obtain more multiplet energies of the d^2 configuration.

Below is an example of how you can obtain determinant energies 'by hand', i.e. by carrying out a specific sequence of ADF calculations. The ADF engine supports an automatic procedure to do this, using the input key SLATERDETERMINANTS, see the ADF User's Guide, the *Examples* (page 401), and below.

Procedure

1. Determine a set of orbitals belonging to the given configuration. These orbitals are generated in what we call the average-of-configuration (AOC) calculation. This is a spin-restricted SCF calculation where the electrons of the configuration are distributed equally over the subspecies of the open shell irreps in order to retain the A1 symmetry of the total density in the symmetry group of the molecule. For instance, in case of an atomic d^2 configuration, the AOC calculation can be done in symmetry atom with occupation 2 in the d irrep. In case of an t_{2g} ^{5} e_g ^{1} configuration of an octahedral complex, the AOC calculation requires an occupation of 5 electrons in the t_{2g} , and 1 electron in the e_g . The result file adf.rkf of the AOC calculation has to be saved, to be used as a fragment file in the subsequent calculations.
2. The AOC is used as a fragment in all subsequent calculations that are performed to obtain single determinant energies. This means that those single determinant energies are always with respect to the AOC energy. This is a case where there is only one "fragment", which is actually the complete system, but in a different electronic configuration and in possibly a different symmetry group.

Suppose that a single determinant corresponds to spin-up and spin-down densities ρ^α and ρ^β , i.e. to specific spin-unrestricted occupations of the AOC orbitals. These densities ρ^α and ρ^β correspond to a symmetry group that will in general be a subgroup of the symmetry group of the molecule. For instance, the occupation $(p_{+1}\ \alpha)^1$ in the case of an atomic p^1 configuration corresponds to

$$\rho^\alpha = 1/2p_x^2 + 1/2p_y^2$$

with $D_{\infty h}$ symmetry.

To obtain the energy of the determinant wave function we must now perform *one* cycle (iterations= 0 in the block key SCF) of a spin-unrestricted calculation, with AOC as (the only) fragment with alpha and beta occupation numbers (using the input key occupations) such that ρ^α and ρ^β result. Note that the appropriate (lower) symmetry point group must be specified in the input file.

Occasionally, the single determinant corresponds to a closed shell configuration in the appropriate lower symmetry, for instance determinant D_{10} = $|0^+ \ 0^- \ |$ of the p^2 configuration of Carbon, with density $r=p_z{}^2$. In that case the one-cycle calculation can of course be spin-restricted.

N.B.1. One cycle will regenerate the SCF orbitals of AOC, if the same field is used as the converged AOC field. This will actually be the case because the starting potential is taken from the fragment adf.rkf file. The key modifystartpotential must not be used (the density should be distributed equally over the spins).

N.B.2. After diagonalization in the one-cycle run, the AOC orbitals have been obtained again and are occupied as specified. The ('bonding') energy is calculated from the resulting charge density.

Remarks:

- If one does not perform just one cycle, but instead converges the unrestricted calculation, the energy will be lowered by 'polarization' of the orbitals. It is theoretically not so clear what the status of such converged energies is. Usually the energy lowering is in the order of 0.1 eV, but it may be quite a bit larger.
- It is not necessary to use AOC as fragment in the single-determinant runs. It is also perfectly allowed to run all calculations (ground state, AOC, determinants) from one set of fragments, for instance the standard atomic fragments. Since we must arrange that the one-cycle determinant calculations use the AOC field, so as to reproduce the AOC orbitals, we must then supply the result file adf.rkf of the AOC run as a restart file, using the key restart; see the adf User's Guide. Of course, in such an approach the computed energies are with respect to another reference, for instance the restricted atoms.

Results for first period atoms

In one of the next sections tables are given for the energy lowering in going from the converged spherically symmetric spin-restricted atom (the 'master' fragment) to specific one-determinant wavefunctions with the orbital occupations as specified. Note that the p_x and p_y populations are always equal; only their sum is given. In many cases the determinant corresponds to a specific state, which is then given in the last column. For each atom, the first calculation is for the spherically symmetric spin-*un*restricted atom. These tables are now obsolete, all information needed to obtain the atomic reference energies, i.e. the ground state multiplet energy with respect to the AOC, can be found in ref.[3].

Examples worked out for all first period atoms:

H: Configuration $(1s)^1$. Only one determinant: | 1s α (1) |

He: Configuration $(1s)^2$. Closed shell.

Li: Configuration $(2s)^1$. Only one determinant: | 2s α (1) |

Be: Configuration $(2s)^2$. Closed shell.

B: Configuration (2p)^1 . Ground multiplet ^{2} P.

$$D_1 = |p_1\alpha(1)| = |^2P; M_L = 1; M_S = 1/2\rangle$$
$$\rho^\alpha = 1/2(p_x - ip_y)(p_x + ip_y) = 1/2(p_x^2 + p_y^2)$$

The occupation numbers for D_1 are

$$p_x^\alpha = p_y^\alpha = 1/2$$
$$p_z^\alpha = p_x^\beta = p_y^\beta = p_z^\beta = 0$$

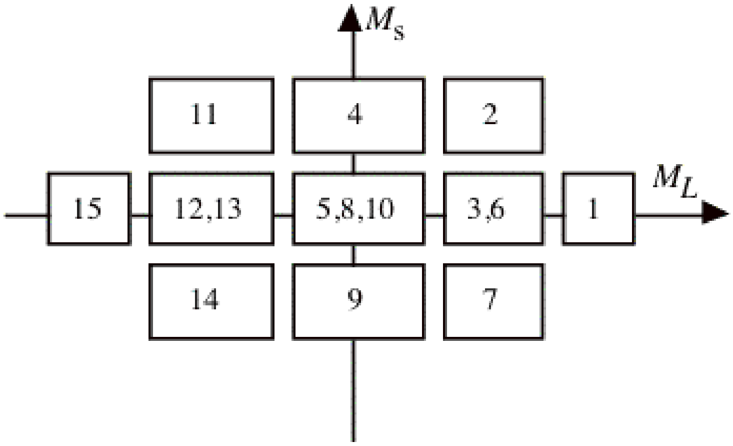
Another determinant belonging to ^{2} P is D_2 = $|..p_0\ \alpha\ (1)|$ with occupations $p_z^\alpha = 1$ and all other p-occupations zero. This determinant is 0.04 eV lower in energy than D_1 for LDA, but 0.15 eV for BP.

---

[3] E.J. Baerends, V. Branchadell and M. Sodupe, *Atomic reference-energies for density functional calculations*, Chemical Physics Letters 265, 481 (1997) (https://doi.org/10.1016/S0009-2614(96)01449-2)

C: Configuration p^2 . Multiplet states are ^{3} P, ^{1} S and ^{1} D. We use this atom as an example of the general procedure. First write down all determinants belonging to p^2 and group them according to M_S and M_L ($1^+ \equiv p_1\ \alpha$, …)

Determinant	M_S	M_L
$D_1 = \vert 1^+ 1^- \vert$	0	2
$D_2 = \vert 1^+ 0^+ \vert$	1	1
$D_3 = \vert 1^+ 0^- \vert$	0	1
$D_4 = \vert 1^+ -1^+ \vert$	1	0
$D_5 = \vert 1^+ -1^- \vert$	0	0
$D_6 = \vert 1^- 0^+ \vert$	0	1
$D_7 = \vert 1^- 0^- \vert$	-1	1
$D_8 = \vert 1^- -1^+ \vert$	0	0
$D_9 = \vert 1^- -1^- \vert$	-1	0
$D_{10} = \vert 0^+ 0^- \vert$	0	0
$D_{11} = \vert 0^+ -1^+ \vert$	1	-1
$D_{12} = \vert 0^+ -1^- \vert$	0	-1
$D_{13} = \vert 0^- -1^+ \vert$	0	-1
$D_{14} = \vert 0^- -1^- \vert$	-1	-1
$D_{15} = \vert -1^+ -1^- \vert$	0	-2



The presence of a determinant with M_L = 2, M_S = 0 and no M_L = 2, M_S 0 determinant indicates the presence of a ^{1} D multiplet, and E(^{1} D) = E(D_1). There is also a ^{3} P, the determinant with M_S = 1, M_L = 1. We should have E(^{3} P) = E(D_2) = E(D_4). The two determinants D_3 and D_6 in the M_S = 0, M_L = 1 box will mix, and the solutions of the 2 by 2 secular problem will be E(^{1} D) and E(^{3} P). Since the sum of the eigenvalues is equal to the sum of the initial diagonal elements, we have E(^{1} D) + E(^{3} P) = E(D_3) + E(D_6). We should also have E(D_3) + E(D_6) = E(D_1) + E(D_2). Such a relation provides a test on the consistency of the results. Finally we have the M_S = 0, M_L = 0 block. The sum of the energies of D_5 , D_8 and D_{10} should be the sum of the energies of ^{1} S, ^{3} P and ^{1} D. Since E(^{3} P) and E(^{1} D) are already known, E(^{1} S) can be calculated.

In the appendix we first locate for C the spherical unrestricted atom. Next we have E(D_4), yielding E(^{3} P) = -1.345 eV (LDA + Becke). Next E(D_2) = E(^{3} P) = -1.189 (always LDA + Becke). The difference is substantial: ~ 0.15 Next we have E(D_3) = - 0.812. Since E(D_6) = E(D_3), because ρ^α (D_6) = ρ^β (D_3) and ρ^β (D_6) = ρ^α (D_3), we should have 2E(D_3) = -1.624 = E(^{1} D) + E(^{3} P). Therefore E(^{1} D) = -1.624 - (-1.345) = - 0.279 or E(^{1} D) = -1.624 - (-1.189) = - 0.435. These numbers can be checked against E(D_1) which also should be E(^{1} D): E(D_1) =+ 0.044. The discrepancy is large! Finally, ^{1} S can be obtained: E(D_{10}) = + 0.319 (D_8) = E(D_1) = + 0.044 E(D_5) = E(D_1) = + 0.044. So 0.407 = E(^{1} S) + E(^{3} P) +E(^{1} D). Different results for E(^{1} S) are obtained depending on the E(^{3} P) and E(^{1} D) we choose: e.g. E(^{1} S) = 0.407 -(-1.345) - (- 0.279) = 2.031 or E(^{1} S) = 0.407 - (-1.189) -

(0.044) = 1.552. Comparing to experiment we might calculate the excitation energies w.r.t. E(^{3} P):

	calculated	experimental	HF
^{3} P → ^{1} D:	1.066 to 1.389	1.26	1.55
^{3} P → ^{1} S:	2.741 to 3.376	2.684	3.78

N: Configuration p^3 . Ground multiplet ^{4} S, corresponds to the spherical unrestricted atom, energy -2.943 eV. Other multiplets: ^{2} P, ^{2} D. According to the printed output for configuration p^3 we have $|^2$ D;M$_L$ =2;M$_S$ =1/2 = $|p_1{}^{\alpha}\ p_1{}^{\beta}\ p_0{}^{\alpha}|$ = D$_2$ ρ^α = 1/2 p$_x{}^2$ + 1/2 p$_y{}^2$ + p$_z{}^2$ ρ^β = 1/2 p$_x{}^2$ + 1/2 p$_y{}^2$ E(D$_2$) = - 0.745 according to the table in the Appendix (LDA + Becke). The energy of D$_1$, with ρ^α = p$_x{}^2$ + p$_y{}^2$, ρ^β = p$_z{}^2$, is E(|1A 2B 3A|) = -1.9702. The energy of D$_3$, with ρ^α = 1/2 p$_x{}^2$ + 1/2 p$_y{}^2$ + p$_z{}^2$, ρ^β = p$_z{}^2$ corresponding to |1A 2A 2B| or |2A 2B 3A|, is E(D$_3$) = - 0.158. Finally, D$_4$ has ρ^α = p$_x{}^2$ + p$_y{}^2$, ρ^β = 1/2 p$_x{}^2$ + 1/2 p$_y{}^2$, corresponding to |1A 1B 3A| and |1A 3A 3B|, and E(D$_4$) = - 0.109. The M$_L$ = 1, M$_S$ =1/2 determinants are |1A 1B 3A| and |1A 2A 2B|. Therefore E(^{2} D) + E(^{2} P) = E(D$_4$) + E(D$_3$), so E(^{2} P) = - 0.109 - 0.158 - (- 0.745) = + 0.478. We can use D$_1$ in the M$_L$ =0, M$_S$ =1/2 block, from which we find E(^{4} S) + E(^{2} D) + E(^{2} P) = 2E(D$_2$) =+ E(D$_1$). Hence E(^{2} P) = -1.490 - 1.9702 - (- 0.745) - (- 2.943) = + 0.2278.

O: Configuration p^4 . Multiplet states ^{3} P, ^{1} S, ^{1} D. D$_1$, with ρ^α = p$_x{}^2$ + p$_y{}^2$ + p$_z{}^2$, ρ^β = p$_z{}^2$ corresponds to |1A 2A 2B 3A| , the M$_L$ =1, M$_S$ =1 determinant of ^{3} P: E(^{3} P) = -1.836 D$_2$ with ρ^α = p$_x{}^2$ + p$_y{}^2$ + p$_z{}^2$, ρ^β = 1/2 p$_x{}^2$ + 1/2 p$_y{}^2$, corresponds to |1A 1B 2A 3A|, the determinant of ^{3} P: E(^{3} P) = -1.568 D$_3$, with ρ^α = 1/2 p$_x{}^2$ + 1/2 p$_y{}^2$ + p$_z{}^2$, ρ^β = p$_x{}^2$ + p$_y{}^2$, corresponds to |1A 1B 2A 3B|, and M$_L$ =1, M$_S$ =0 belonging to ^{3} P as well as ^{1} D.

F: Configuration p^5 . Ground multiplet ^{2} P. As in B, we have two determinants with different energies belonging to this state: D$_1$ = $|\ldots(p_0{}^{\alpha})^1\ (p_0{}^{\beta})^0|$ → E(D$_1$) = - 0.715. D$_2$ = $|(p_{-1})^2\ (p_0)^2\ (p_1{}^{\alpha})^1\ (p_1{}^{\beta})^0|$ → E(D$_2$) = - 0.467.

Ne: Configuration p^6 . Closed shell.

Ground and Excited State Multiplets

The computation of multiplets can be carried out with adf, using the input block `SLATERDETERMINANTS`.

The method described in? for the calculation of the energies of proper spin and spatial symmetry adapted Configuration State Functions is based on the calculation of the energies of single determinantal wavefunctions. The densities corresponding to those Slater determinants are inserted in the approximation used for the exchange-correlation energy.

The procedure as detailed above is somewhat involved. Moreover, one would like to have an easy procedure to calculate many determinants. This is particularly desirable if one wishes to calculate the energies of all Slater determinants of a given configuration, for instance if one wishes to calculate certain averages in view of the inconsistencies of the method.

We have implemented a semi-automatic procedure, using the block `SLATERDETERMINANTS`.

The general idea of this method is to first perform a restricted calculation in the symmetry that is appropriate for the molecule. This is called the average-of-configuration (AOC) calculation. This AOC calculation generates the orbitals which will be used in all the Slater determinants.

The AOC is the fragment that *must* be used in all subsequent calculations. The subsequent calculations are characterized by having the AOC as the (only) fragment, and by specifying the keyword `SLATERDETERMINANTS`. If an argument is given this must be a file name. The named file should contain the occupations for the determinants (see below). If no file name is specified, the occupations should be specified in the data block.

The required information, on file or in the data block, is the specification of the determinant or determinants that are to be calculated in the form of orbital occupation numbers for the AOC orbitals, using the irrep labels of the point group of the AOC calculation, see below for format. All specified determinants will be calculated, and the obtained energy will always be the energy difference with respect to the AOC. Default occupations for all subspecies of the AOC fragment are the occupations of the AOC itself. Therefore, only the open (modified) subspecies have to be specified.

One has to be careful with respect to the point group symmetry to use in the `SLATERDETERMINANTS` calculation. The density belonging to a specific determinant is usually lower than the AOC symmetry (which is the full symmetry group of the system). In that case this lower point group symmetry has to be specified in the `SLATERDETERMINANTS` calculation. Everything will always work fine if one just does not use any symmetry at all (nosym). However, if for reasons of computational efficiency one does want to use a subgroup of the system that corresponds to the determinant density or densities, this is perfectly possible. However, all the densities of the determinants specified *must* then have this (or a higher) symmetry. The program does not check on this, it is the user's responsibility to make sure that this condition is satisfied for all the determinants. The only check that is performed is that occupations of equivalent representations (subspecies of one irrep) in the lower point group of the `SLATERDETERMINANTS` run, that result from the specified occupations of the subspecies of the AOC symmetry, are equal.

Format of the input.

Important: in the SlaterDeterminants calculations it is imperative that the results are obtained in the AOC field: no SCF convergence must be carried out, because we only want to assign the electrons differently, while keeping the AOC orbitals exactly as they are. To achieve this, the calculations should use the keyword SCF, and the subkey iterations has to be set to 0 in the SCF data block.

Since adf requires that the point-group symmetry conforms not only to the nuclear frame but also to the electronic charge density and mo occupations, these calculations must run in a lower point group symmetry. Often you will also want to run the modified calculations spin-unrestricted.

For an example, see the set of sample runs that come with the package and the comments in the examples.

See: *SLATERDETERMINANTS block* (page 370). Each 'title' functions as a subkey, but is otherwise an arbitrary string to label the resulting one-determinant calculation. Each such subkey block contains the occupation numbers for a single one-determinant calculation. It is necessary that the calculation uses the reference AOC run as its only fragment file. The occupations in the subkey blocks must be re-arrangements of the AOC open-shell electrons. In the `SLATERDETERMINANTS` calculation you must explicitly specify the point group symmetry in which you want to run. The ρ^α and ρ^β densities of all determinants to be calculated must have this point group symmetry, or a higher symmetry.

Each 'irrep occups' record specifies the occupations for the indicated irrep in the usual way (see for instance the occupations key). The irrep labels must correspond to the AOC point group symmetry used in the AOC calculation, *not the point group symmetry used in the ``SLATERDETERMINANTS`` calculation!*. Note that in an unrestricted calculations, occupations numbers must be given for both spins, using the double slash (//) to separate the occupations for spin-alpha and spin-beta.

In this setup, the program will for each of the subkey blocks under the `SLATERDETERMINANTS` key execute an SCF calculation with only one cycle, i.e. no convergence, where the start-up field is the fragment field, i.e. the AOC field. So all one-determinant states in this calculation are evaluated in the AOC field. The resulting energies for the distinctly computed one-determinant states can then be combined to the desired multiplet values.

Example: Carbon p^2

```
SlaterDeterminants
C(p2) ALFA: s=1, px=py=2/3, pz=2/3; BETA: s=1, p=0 ! title
   S 1 // 1 ! irrep name and occupations
   P:x 0.666666666666666666 // 0 ! another irrep, et cetera
   P:y 0.666666666666666666 // 0
   P:z 0.666666666666666666 // 0
   D:z2 0 // 0
   D:x2-y2 0 // 0
   D:xy 0 // 0
   D:xz 0 // 0
   D:yz 0 // 0
SUBEND
C(p2) ALFA: S=1, px=py=1, pz=0; BETA: s=1 !next (Sl.Determinant) title
   S 1 // 1
   P:x 1 // 0
   P:y 1 // 0
   P:z 0 // 0
   D:z2 0 // 0
   D:x2-y2 0 // 0
```

(continues on next page)

(continued from previous page)

```
   D:xy 0 // 0
   D:xz 0 // 0
   D:yz 0 // 0
SUBEND
C(p2) ALFA: s=1, px=py=0.5, pz=1; BETA: s=1
   S 1 // 1
   P:x 0.5 // 0
   P:y 0.5 // 0
   P:z 1 // 0
   D:z2 0 // 0
   D:x2-y2 0 // 0
   D:xy 0 // 0
   D:xz 0 // 0
   D:yz 0 // 0
SUBEND
C(p2) ALFA: s=1, px=py=0.5, pz=0; BETA: s=1, px=py=0, pz=1
   S 1 // 1
   P:x 0.5 // 0
   P:y 0.5 // 0
   P:z 0 // 1
   D:z2 0 // 0
   D:x2-y2 0 // 0
   D:xy 0 // 0
   D:xz 0 // 0
   D:yz 0 // 0
SUBEND
C(p2) ALFA: s=1, px=py=0.5, pz=0; BETA: s=1, px=py=0.5, pz=0
   S 1 // 1
   P:x 0.5 // 0.5
   P:y 0.5 // 0.5
   P:z 0 // 0
   D:z2 0 // 0
   D:x2-y2 0 // 0
   D:xy 0 // 0
   D:xz 0 // 0
   D:yz 0 // 0
SUBEND
C(p2) ALFA: s= 1, px=py=0, pz=1; BETA: s=1, px=py=0, pz=1
   S 1 // 1
   P:x 0 // 0
   P:y 0 // 0
   P:z 1 // 1
   D:z2 0 // 0
   D:x2-y2 0 // 0
   D:xy 0 // 0
   D:xz 0 // 0
   D:yz 0 // 0
SUBEND
```

In the example the AOC calculation is the Carbon atom in spherical symmetry (symmetry name atom).

Several spin states can be generated from this AOC set of orbitals, but they all have a lower symmetry than the AOC. In the example the point group $D_{\infty h}$ (DLIN) could be used in the `SLATERDETERMINANTS` calculation. In $D_{\infty h}$ the *p* orbitals split into two sets, $p*_x$ * and $p*_y$ * occur in π_x and π_y respectively, so their occupations must be identical, and $p*_z$ * is a Σ_u orbital.

In the block `SLATERDETERMINANTS` (or in the file) we now specify the occupations for the subspecies of the atom irreps of a specific Slater determinant and the program will sort out the corresponding occupations in the d(lin) symmetry.

In all cases the orbitals used for the energy calculation(s) will be the self-consistent AOC orbitals.

In the given example, the first set of occupations does not correspond to a Slater determinant, but is the spin-polarized spherical case with the *p* electrons evenly distributed over all components.

LDA results, with and without GGA (Becke-Perdew)

Energy changes (eV) for atoms going from restricted to (one-cycle) unrestricted. Results between parentheses are for *converged* unrestricted calculations)

All calculations have been performed in $D_{\infty h}$ symmetry, since p_x and p_y always had equal occupations and therefore could occur as π_u -x and π_u -y partners of the Π_u irrep.

El.	Occupations						LDA	LDA+Becke	BP
	alpha-spin			beta-spin					
	s	px+py	pz	s	px+py	pz			
H	1	0	0	0	0	0	-0.868 (-0.898)	-0.758 (-0.837)	-0.889 (-0.948)
Li	1	0	0	0	0	0	-0.231 (-0.235)	-0.195 (-0.207)	-0.249 (-0.256)
Be	1	0	0	1	0	0	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
B	1	2/3	1/3	1	0	0	-0.247 (-0.255)	-0.231 (-0.242)	-0.276 (-0.281)
	1	0	1	1	0	0	-0.295 (-0.321)	-0.436 (-0.474)	-0.448 (-0.485)
	1	1	0	1	0	0	-0.266 (-0.279)	-0.296 (-0.316)	-0.333 (-0.348)
C	1	4/3	2/3	1	0	0	-1.163 (-1.203)	-1.109 (-1.158)	-1.252 (-1.285)
	1	2	0	1	0	0	-1.152 (-1.211)	-1.271 (-1.345)	-1.372 (-1.436)
	1	1	1	1	0	0	-1.152 (-1.197)	-1.134 (-1.189)	-1.267 (-1.307)
	1	1	0	1	0	1	-0.462 (-0.506)	-0.726 (-0.812)	-0.778 (-0.868)
	1	1	0	1	1	0	0.159 (0.150)	0.039 (0.044)	0.087 (0.086)
	1	0	1	1	0	1	0.730 (0.668)	0.322 (0.319)	0.480 (0.450)
N	1	2	1	1	0	0	-2.936 (-3.032)	-2.827 (-2.943)	-3.101 (-3.190)
	1	2	0	1	0	1	-1.362 (-1.454)	-1.811 (-1.972)	-1.943 (-2.108)
	1	1	1	1	1	0	-0.581 (-0.618)	-0.688 (-0.745)	-0.746 (-0.801)
	1	1	1	1	0	1	0.178 (0.088)	-0.104 (-0.158)	-0.069 (-0.140)
	1	2	0	1	1	0	0.197 (0.135)	-0.077 (-0.109)	-0.011 (-0.053)
O	1	2	1	1	2/3	1/3	-1.400 (-1.477)	-1.361 (-1.447)	-1.480 (-1.552)
	1	2	1	1	0	1	-1.442 (-1.583)	-1.698 (-1.836)	-1.816 (-1.957)
	1	2	1	1	1	0	-1.422 (-1.515)	-1.470 (-1.568)	-1.590 (-1.678)
	1	1	1	1	2	0	-0.564 (-0.623)	-0.866 (-0.960)	-0.913 (-1.013)
	1	1	1	1	1	1	0.358 (0.321)	0.255 (0.237)	0.292 (0.266)
	1	2	0	1	2	0	1.323 (1.220)	0.825 (0.789)	0.992 (0.932)
F	1	2	1	1	4/3	2/3	-0.374 (-0.398)	-0.366 (-0.391)	-0.394 (-0.416)
	1	2	1	1	2	0	-0.323 (-0.409)	-0.605 (-0.686)	-0.627 (-0.715)
	1	2	1	1	1	1	-0.349 (-0.389)	-0.401 (-0.441)	-0.427 (-0.467)

13.4 Dirac program: relativistic core potentials

In all relativistic calculations - scalar as well as spin-orbit - the relativistic atomic core densities and the relativistic atomic potentials *must* be made available to ADF on a file specified with the key COREPOTENTIALS. This is necessary only in the ‘create’ run of the atoms. In the molecular calculation this key is not required anymore. If supplied then the file must contain data for *all* atom types in the molecule, even for those atoms where relativistic aspects are expected to be negligible or that may not have a frozen core at all (such as Hydrogen). Excepted are any Ghost atoms (for instance for a BSSE calculation): these can not have any core potentials. This is tested by the program, internally, by looking at the nuclear charge and at the number of electrons belonging to an atom: if both numbers are zero, no (relativistic or other) core potential is allowed. Also the potential used in the ZORA kinetic energy operator in the MAPA (the Minimum of neutral Atomical potential Approximation) method should be present on this file (which will be the case if the program DIRAC is used to generate this file).

Relativistic potentials can and should be generated with the auxiliary program dirac, see the next section, and the examples (in particular *Example: Gold electrodes* (page 967)).

The recommended way to generate atomic fragments and relativistic potentials is by using the *BASIS* (page 42) keyword.

The auxiliary program DIRAC, which is installed in the Amsterdam Modeling Suite, serves to compute relativistic frozen core potentials (and densities), necessary to apply the (scalar) relativistic option in ADF. `$AMSHOME/atomicdata` has a subdirectory `Dirac`, which contains input files for DIRAC for all atoms of the periodic table of elements. The *names* of the input files indicate the frozen core level: *Ag.3d* for instance is the input file for a calculation on a Silver atom with a frozen core up to and including the 3d shell (i.e.: 1s, 2s, 2p, 3s, 3p, and 3d). The frozen core level used in the DIRAC calculation defines the core data computed and should therefore match the frozen core level in the ADF Create run for the atom that it will be used for.

A DIRAC run with the inputs provided in the `$AMSHOME/atomicdata/Dirac/` involves a fully relativistic calculation on the atom (spin-orbit coupling, double group symmetries). It generates a file TAPE12 with the corresponding core potential and density (a table of values for a sequence of radial distance values). Other files produced by DIRAC should be removed after the DIRAC run; they are needed for other applications of the program but play no role here.

If you run DIRAC while a file TAPE12 already exists the computed core data will be written at the end of it, after the existing data. The program will assume, however, that the existing data on the file are also core-data from DIRAC runs, and may abort otherwise.

Starting from ADF2006.01 it is not necessary anymore to make one big TAPE12 which contains data for all atoms involved in the molecular calculation. Instead only in the ADF Create run for each atom one needs a TAPE12 which contains data for the atom that is created. The corresponding core data is written to the adf.rkf of this atomic fragment. In the molecular ADF run one then should not include the CorePotentials key, such that ADF will read core data on the adf.rkf's of the atomic fragments. One can still use the CorePotentials key, but then one should proceed as in previous releases.

Implied options

The DIRAC calculations imply the *local* Density Functional in its simple X-alpha approximation without any gradient corrections. Not the *scalar* relativistic but the *fully* relativistic Hamiltonian is used, including spin-orbit coupling. In ADF you may use the *scalar* relativistic Hamiltonian and most users will employ a more sophisticated lda than X-alpha, such as the default vwn (Vosko, Wilk, Nusair) formulas, and may in addition routinely apply gradient corrections. The core potential may not exactly match the Fock operator applied in the molecular calculation. The effect is very small and one can neglect the discrepancy.

Input

The ASCII input files for DIRAC, as available in the directory `$AMSHOME/atomicdata/Dirac` (point nucleus) and `$AMSHOME/atomicdata/Dirac.gaussian` (finite nucleus), have a structure as described below. With this information you should be able to construct alternative input files, with other frozen cores for instance.

1. Title (60 characters at most). Plays no role
2. Ngrid, Nshell, rmin, rmax, Z, Xion, Anuc Ngrid=number of radial grid points, in which the core potentials are computed. Nshell=number of atomic orbital shells rmin, rmax=minimum and maximum radial grid values Z=nuclear charge Xion=net charge of the 'atom' Anuc=atomic weight
3. Pinit, Pfinal, eps, del, delrv Pinit, Pfinal= initial and final density iteration averaging factors. Each iteration cycle changes the actual averaging factor by taking the average of the previous and the final one, starting with the 'initial' one. eps=Exp(-sqrt(eps)) is set to zero, so eps determines the exponential underflow control. del=absolute convergence criterion for orbital eigenenergies. delrv=convergence criterion on the potential (multiplied by the radial distance *r*).

4. Idirc, Nmax, Ndebu, Nprin, Ipun, Ircor, Iwcor Idirc=zero for non-relativistic, otherwise one. Nmax=maximum number of iterations allowed to reach convergence. Ndebu=non-zero for additional output (for debugging purposes mainly) Nprin=print parameter. Use 2 or larger to get the orbitals printed. Ipun=punched output is produced if Ipun is non-zero. (out-of-date) Ircor=number of core orbitals from the fully relativistic run, to be kept frozen in the subsequent (if any) first-order perturbation calculation. Iwcor=number of core orbitals used to construct the core density and core potential, that are output on TAPE12. So, here you specify the relativistic core.
5. Xalph, Xlatt, Rnuc Xalph= Exchange parameter in the Xalpha formalism. Xlatt=Coulomb tail parameter Rnuc=size of nuclear radius, in bohr. If set to 1.0 or larger, it is recomputed as 0.0000208*Anuc**(1/3)
6. For each orbital shell: N, L, J, E, Z, D N, L, J = The usual orbital quantum numbers. J is used only for relativistic runs. E = Initial estimate of orbital energy, in atomic units. Z = Number of electrons on the shell D = Initial estimate of the error in the orbital energy
7. Icorp, Npcl, Demp, Peps Icorp = If 1 (one): Do a first order perturbation calculation after the fully relativistic run. This option plays no role in the current application for ADF. Npcl = Maximum number of cycles in the perturbation calculation. Demp = Damping factor in the perturbation iterations Peps = Convergence criterion in the perturbation iterations.

13.5 Symmetry

Note: Starting from the 2020 release, ADF does not automatically symmetrize and re-orient the molecule. If your molecule is symmetric and you want to make sure that the highest possible symmetry is used, you should either

- Symmetrize your molecule in AMSinput using the symmetrize button ★
- Use the **System%Symmetrize** option in the AMS driver

13.5.1 Schönfliess symbols and symmetry labels

A survey of all point groups that are recognized by ADF is given below. The table contains the Schönfliess symbols together with the names of the subspecies of the irreducible representations as they are used internally by ADF. The subspecies names depend on whether single-group or double-group symmetry is used. Double-group symmetry is used only in relativistic spin-orbit calculations.

Note that for some input of TDDFT (Response) calculations, other conventions apply for the subspecies, see below.

Table 13.2: Schönfliess symbols and the labels of the irreducible representations.

Point Group	Schönfliess Symbol in ADF	Irreducible representations in single-group symmetry	Irreducible representations in double-group symmetry
C_1	NOSYM	A	A1/2
R^3	ATOM	s p d f	s1/2 p1/2 p3/2 d3/2 d5/2 f5/2 f7/2
T_d	T(D)	A1 A2 E T1 T2	E1/2 U3/2 E5/2
O_h	O(H)	A1.g A2.g E.g T1.g T2.g	E1/2.g U3/2.g E5/2.g
		A1.u A2.u E.u T1.u T2.u	E1/2.u U3/2.u E5/2.u
$C_{\infty v}$	C(LIN)	Sigma Pi Delta Phi	J1/2 J3/2 J5/2 J7/2
$D_{\infty h}$	D(LIN)	Sigma.g Sigma.u Pi.g Pi.u	J1/2.g J1/2.u J3/2.g J3/2.u
		Delta.g Delta.u Phi.g Phi.u	J5/2.g J5/2.u J7/2.g J7/2.u
C_i	C(I)	A.g A.u	A1/2.g A1/2.u
C_s	C(S)	AA AAA	A1/2 A1/2*
C_n	C(N), n must be 2	A B	A1/2 A1/2*
C_{nh}	C(NH), n must be 2	A.g B.g A.u B.u	A1/2.g A1/2.g* A1/2.u A1/2.u*
C_{nv}	C(NV), n<9	A1 A2 B1 B2 E1 E2 E3 …	E1/2 E3/2 E5/2 …
		odd n: without B1 and B2	odd n also: An/2 An/2*
D_n	D(N), n<9	n=2: A B1 B2 B3	E1/2 E3/2 …
		other: A1 A2 B1 B2 E1 E2 E3 …	
		odd n: without B1 B2	odd n also: An/2 An/2*
D_{nh}	D(NH), n<9	n=2: A.g B1.g B2.g B3.g A.u B1.u B2.u B3.u	even n: E1/2.g E1/2.u
		other: A1.g A2.g B1.g B2.g E1.g	E3/2.g E3/2.u …
		E2.g E3.g … A1.u A2.u B1.u …	
		odd n: AA1 AA2 EE1 EE2 …	odd n: E1/2 E3/2 E5/2 …
		AAA1 AAA2 EEE1 EEE2 ….	
D_{nd}	D(ND), n<9	even n: A1 A2 B1 B2 E1 …	even n: E1/2 E3/2 …
		odd n: A1.g A2.g E1.g E2.g … E(n-1)/2.g	odd n: E1/2.g E1/2.u E3/2.g E3/2.u …
		A1.u A2.u E1.u E2.u … E(n-1)/2.u	An/2.g An/2.u An/2.g* An/2.u*

Most labels are easily associated with the notation usually encountered in literature. Exceptions are AA, AAA, EE1, EEE1, EE2, EEE2, etcetera. They stand for A', A", E1', E1", and so on. The AA, etc. notation is used in ADF to avoid using quotes in input files in case the subspecies names must be referred to.

The symmetry labeling of orbitals may depend on the choice of coordinate system. For instance, B1 and B2 representations in C_{nv} are interchanged when you rotate the system by 90 degrees around the z-axis so that x-axis becomes y-axis and vice-versa (apart from sign).

Labels of the symmetry subspecies are easily derived from those for the irreps. For one-dimensional representations they are identical, for more-dimensional representations a suffix is added, separated by a colon: For the two- and three-dimensional E and T representations the subspecies labels are obtained by adding simply a counting index (1, 2, 3) to the name, with a colon in between; for instance, the EE1 irrep in the Dnh pointgroup has EE1:1 and EE1:2 subspecies. The same holds for the two-dimensional representations of $C_{\infty v}$ and $D_{\infty h}$. For the R3 (atom) point group symmetry the subspecies are p:x, p:y, p:z, d:z2, d:x2-y2, etc.

All subspecies labels are listed in the Symmetry section, very early in the ADF output. To get this, perform a quick run of the molecule using the STOPAFTER key (for instance: stopafter config).

TDDFT irrep labels

The printed symmetry in the output in TDDFT calculations is actually the symmetry of transition density. For closed-shell systems, the symmetry of the excited state is the same as the symmetry of the transition density, while for open-shell systems, the symmetry of the excited states is the direct product between the symmetry of the transition density and the ground state symmetry. Note that the ground state symmetry of an open shell molecule is not necessarily A1.

The TDDFT module uses a different notation for some irrep names, for example, A’ is used instead of AA, A” (A two apostrophes) instead of AAA, A1’ instead of AA1, A2” instead of AAA2. In case of atoms, one should use a subgroup. In case of linear systems, if one does not specify the symmetry ADF will lower the symmetry automatically. For C(LIN) ADF will use internally C(7V), however, the names of the irreps in the TDDFT output will be S+, S-, Pi, De, Phi, instead of A1, A2, E1, E2, E3, respectively. For D(LIN) ADF will use internally D(7D), however, the names of the irreps in the TDDFT output will be S+.g, S-.g, Pi.g, De.g, Ph.g, S-.u, S+.u, Pi.u, De.u, Ph.u, instead of A1.g, A2.g, E1.g, E2.g, E3.g, A1.u, A2.u, E1.u, E2.u, E3.u, respectively.

Table 13.3: Schönfliess symbols and the labels of the irreducible representations for TDDFT.

Point	**Schönfliess**	**Irreducible representations**
Group	**Symbol in ADF**	
C_1	NOSYM	A
T_d	T(D)	A1 A2 E T1 T2
O_h	O(H)	A1.g A1.u A2.g A2.u E.g E.u T1.g T1.u T2.g T2.u
$C_{\infty v}$	C(LIN)	S+ S- Pi De Ph
$D_{\infty h}$	D(LIN)	S+.g S-.g Pi.g De.g Ph.g S-.u S+.u Pi.u De.u Ph.u
C_i	C(I)	A.g A.u
C_s	C(S)	A’ A”
C_n	C(N), n must be 2	A B
C_{nh}	C(NH), n must be 2	A.g B.g A.u B.u
C_{nv}	C(NV), n<9	n=2: A1 A2 B1 B2
		n=3: A1 A2 E
		n=4: A1 A2 B1 B2 E
		even n: A1 A2 B1 B2 E1 E2 …
		odd n: A1 A2 E1 E2 …
D_n	D(N), n<9	n=2: A B1 B2 B3
		n=3: A1 A2 E
		n=4: A1 A2 B1 B2 E
		even n: A1 A2 B1 B2 E1 E2 …
		odd n: A1 A2 E1 E2 …
D_{nh}	D(NH), n<9	n=2: A.g A.u B1.g B1.u B2.g B2.u B3.g B3.u
		n=4: A1.g A1.u A2.g A2.u B1.g B1.u B2.g B2.u B3.g B3.u E.g E.u
		even n: A1.g A1.u A2.g A2.u B1.g B1.u B2.g B2.u B3.g B3.u E1.g E1.u E2.g E2.u …
		n=3: A1’ A1” A2’ A2” E’ E”
		odd n: A1’ A1” A2’ A2” E1’ E1” E2’ E2” …
D_{nd}	D(ND), n<9	n=2: A1 A2 B1 B2 E
		even n: A1 A2 B1 B2 E1 E2 …
		n=3: A1.g A1.u A2.g A2.u E.g E.u
		odd n: A1.g A1.u A2.g A2.u E1.g E1.u E2.g E2.u …

13.5.2 Molecular orientation requirements

ADF requires that the molecule has a specific orientation in space, as follows:

- The origin is a fixed point of the symmetry group.
- The z-axis is the main rotation axis, xy is the σ_h -plane (axial groups, C(s)).
- The x-axis is a C_2 axis (D symmetries).
- The xz-plane is a σ_v -plane (C_{nv} symmetries).
- In T_d and O_h the z-axis is a fourfold axis (S_4 and C_4 , respectively) and the (111)-direction is a threefold axis.

If the user-specified symmetry equals the true symmetry of the nuclear frame (including electric field and point charges) the program will adapt the input coordinates to the above requirements, if necessary. If no symmetry has been specified at all ADF assumes you have specified the symmetry of the nuclear frame, accounting for any fields. If a subgroup has been specified for the molecular symmetry the input coordinates will be used as specified by the user. If a Z-matrix input is given this implies for the Cartesian coordinates: first atom in the origin, second atom on the positive x-axis, third atom in the xy-plane with positive y value.

Molecular orientation conventions

The molecular requirements for ADF may not fully determine the orientation of the molecule, like for D_{2h} molecules. See Mulliken[1] for standard conventions on the molecular orientation and notation. A few of these conventions are listed below:

- in planar C_{2v} molecules the x axis is perpendicular to the plane of the molecule.
- in planar D_{2h} molecules the x axis is perpendicular to the plane of the molecule and the z-axis passes through the greatest number of atoms. If the last condition is not decisive, the z-axis should pass through the greatest number of bonds.
- in planar D_{4h} and D_{6h} molecules the C_2-axis passes through the greatest number of atoms. If the last condition is not decisive, the C_2-axis should pass through the greatest number of bonds.

13.6 Binary result files

13.6.1 ams.rkf

The main ams.rkf file is written by the AMS driver. It contains high level information about the trajectory that the AMS driver took over the potential energy surface. Part of what used to be on TAPE21 is since the AMSification of ADF in ams.rkf.

This file contains information on geometry optimization, transition state search, linear transit, PES scan, NEB, or IRC, if that was requested in the calculation. For a geometry optimization or transition state search, for example, it contains the history of how the systems geometry changed during the optimization as well as the final optimized geometry. The format in which this information is written is independent from which engine was used for a calculation.

If a property, like vibrational modes, is tied to a special point on the potential energy surface, it is stored in the engine main output file adf.rkf.

- AMS manual
  - Section and variables on ams.rkf

---

[1] R.S Mulliken, *Report on Notation for the Spectra of Polyatomic Molecules*, Journal of Chemical Physics 23, 1997 (1955) (https://doi.org/10.1063/1.1740655)

13.6.2 adf.rkf (TAPE21)

adf.rkf is the general result file of an ADF calculation. In ADF<=2019 it was known as TAPE21. It is a KF file: Direct-Access, binary, and keyword driven. It contains information about the calculation. You can use it as a fragment file in a subsequent calculation on a bigger molecule, where the current one may be a part, or in an analysis program. Part of what used to be on TAPE21 is since the AMSification of ADF in ams.rkf.

A description of the various utilities that can be used to process adf.rkf can be found in other parts of this ADF manual.

- *Sections and variables on adf.rkf* (page 1220)

adf.rkf: Added notes

Section Multipole matrix elements

Information about a response calculation

`dipole elements` The matrix elements of the 3 dipole operator components between occupied and virtual orbitals: outer loop over the operators (in order: y, z, x), loop over virtual MOs, inner loop over occupied MOs

`quadrupole elements` Similar as for dipole. Order of operators: $\sqrt{3}$ *xy $\sqrt{3}$ *yz z2-(x2+y2)/2 $\sqrt{3}$ *xz $\sqrt{3}$ *(x2-y2)/2

`octupole elements` Similar as for dipole and quadrupole. Order of operators: $\sqrt{10}$ *y*(3*x2-y2)/4 $\sqrt{15}$ *xyz $\sqrt{6}$ *y*(4*z2-x2-y2)/4 z*(z2-3(x2-y2)/2) $\sqrt{6}$ *x*(4*z2-x2-y2)/4 $\sqrt{15}$ *z*(x2-y2)/2 $\sqrt{10}$ *x*(x2-3y2)/4

`hexadecapole elements` Similar as for dipole and quadrupole. Order of operators: $\sqrt{35}$ *xy*(x2-y2)/2 $\sqrt{70}$ *z*(3x2y-y3)/4 $\sqrt{5}$ *xy*(6z2-x2-y2)/2 $\sqrt{10}$ *(4yz3-3yz*(x2+y2))/4 (8z4-24*z2*(x2+y2)+3(x4+2x2y2+y4))/8 $\sqrt{10}$ *(4xz3-3xz*(x2+y2))/4 $\sqrt{5}$ *(x2-y2)*(6z2-x2-y2)/4 $\sqrt{70}$ *z*(x3-3xy2)/4 $\sqrt{35}$ (x4-6x2y2+y4)/8

Representation of functions and frozen cores

ADF uses the cartesian representation for the spherical harmonics part in functions:

$$f(x, y, z) = x^a y^b z^c r^d e^{-\alpha r}$$

The angular momentum quantum number *l* is then given by *l*=a+b+c, and the main quantum number *n* = *l* +d+1.

There are (*l* + 1)(*l* + 2)/2 different combinations of (a, b, c) for a given *l* - value, rather than (2 *l* +1). The excess is caused by the presence of spurious non-*l* Functions in the set; a Cartesian d-set for instance consists of six functions, five of which are true d-functions while one linear combination is in fact an s-type function (x2+y2+z2). Only the five true d-combinations are actually used as degrees of freedom in the basis set, but lists of primitive basis functions (bas) for instance run over all Cartesian functions including the improper ones.

A function *set* in ADF is characterized by the quantum numbers *l* and *n*, and by the exponential decay factor a. A set thus represents (*l* +1)(*l* +2)/2 Cartesian functions and (2 *l*+1) degrees of freedom.

The atomic frozen core orbitals are described as expansions in Slater-type functions; these are not the functions of the normal basis set but another set of functions, defined on the data files you use in Create mode.

Orthogonality of the valence space to the frozen core states is enforced as follows: for each frozen core shell (characterized by the quantum numbers *l* and *n*: all orbitals with *m* = - *l* … + *l* are identical apart from rotation in space) the set of valence basis functions is augmented with a so-called core orthogonalization function set. You may conceptually interpret the core orthogonalization functions as single zeta expansions of the true frozen core states. Each of the normal valence basis functions is now transformed into a linear combination of that valence function with all core orthogonalization functions, where the coefficients are uniquely defined by the requirement that the resulting function is orthogonal to all true core functions.

So the list of all Cartesian basis functions is much larger than the degree of freedom of the basis: it contains the spurious *non-l* combinations and it contains also the core orthogonalization functions.

Evaluation of the charge density and molecular orbitals

adf.rkf contains all the information you need to evaluate the charge density or a Molecular Orbital (MO) in any point in space. Most of the information is located in section Basis:

A list of function characteristics (kx,ky,kz,kr,alf), including the core orthogonalization functions. This list does *not* run over *all* bas functions used in the molecule: if a particular function is used on the atoms of a given atomtype, the function occurs only once in the list, but in the molecule it occurs as many times as there are atoms of that type.

With array nbptr you locate the subsections in the function list that correspond to the different types of atoms: for atom type i the functions nbptr(i)…nbptr(i+1)-1. The distinct atom types are listed in an early section of the standard output file.

Array nqptr gives the number of atoms for type i: nqptr(i+1)-nqptr(i). With this information you construct the complete list of all functions. Repeat the subsection of type i as many times as there are atoms of that type: the complete list can be considered to be constructed as a double loop, the outer being over the atom types, the inner over the atoms that belong to that type. The total 'overall' list of functions you obtain in this way contains naos functions. Note that in this way we have implicitly also defined a list of all atoms, where all atoms that belong to a particular atom type are contiguous. This list is the so-called 'internal' atom ordering, which may not be identical to the order in which atoms were specified in input, under atoms.

For a given symmetry representation (Sections S) the array npart gives the indices of the basis functions in the overall list that are used to describe orbitals in this representation. In case of an unrestricted run the array npart applies for either spin: the same basis functions are used; the expansion coefficients for the molecular orbitals are different of course.

In the symmetry-representation sections Eigen_bas gives the expansion coefficients that describe the MOs. The expansion refer to the functions indicated by npart, and the function characteristics are given by the arrays kx,ky,kz,kr,alf, *and* bnorm, i.e. the expansion is in* normalized*functions.

The value of an MO is now obtained as a summation of values of primitive basis functions. For the evaluation of any such basis function you have to take into account that its characteristics are defined in the local coordinate system of its atom.

To obtain the charge density you sum all MOs, squared and multiplied by the respective occupation numbers (array froc in the appropriate irrep section).

Note that the auxiliary program densf, which is provided with the ADF package, generates orbital and density values on a user-specified grid.

13.6.3 TAPE10

TAPE10 is a binary KF file. By default TAPE10 is a scratch file. TAPE10 contains data related to the numerical integration: points and weights, values of functions (depends on direct-SCF options) and core densities and potentials.

Use the keywords 'SAVE TAPE10' in the ADF calculation such that TAPE10 is saved. NMR requires a TAPE10 file, see also documentation on *NMR chemical shifts* (page 205) and *NMR spin-spin couplings* (page 213).

13.6.4 TAPE13

TAPE13 is a binary KF file. TAPE13 is the checkpoint file for restarts after a crash. It is a concise version of adf.rkf, containing only the data the program uses for restarting the calculation. See the restart keyword. Note that since the AMSification of ADF in most case this TAPE13 has lost it usefulness. Instead one should use the file ams.rkf, which is produced by the AMS driver, for restarting a geometry optimization, for example.

Like adf.rkf, TAPE13 is a binary, keyword driven KF file.

The calculation that produces TAPE13 determines which section are written on it. The actual values of the variables should be identical to the corresponding variables on adf.rkf. Also they should have the same names and be located in the same sections. In some cases, TAPE13 contains the complete corresponding section of adf.rkf.

13.6.5 TAPE15

TAPE15 is a binary KF file. By default TAPE15 is a scratch file. TAPE15 contains data related to matrices like the density matrix, the Fock matrix, and the overlap matrix.

Use the keywords 'SAVE TAPE15' in the ADF calculation such that TAPE15 is saved. Some external programs may need data that is stored on TAPE15. See, for example, the documentation on *adf2nbo to produce an input for NBO* (page 262) and *adf2damqt to produce an input for DAMQT* (page 273).

13.6.6 TAPE41

TAPE41 is a binary KF file. It is produced by the *densf* program to generate values of molecular orbitals, charge densities and potentials in a user-specified grid, to be used typically for plotting or graphical display, see the documentation on *densf* (page 291).

13.6.7 Other binary files

In the results directory that ams creates one may find other binary KF files. There may be binary adf.rkf-like output files for some points on the potential energy surface that was visited during the calculation. One may find t21.*.elementname files which are atomic adf.rkf-like files, that were created during an ADF calculation. One may find the binary t12.rel file that is not a KF file, which is an auxiliary binary file for relativistic calculations. Depending on the kind of ADF calculation more binary KF (scratch) files may be used.

13.7 Error messages

ERROR: STOP GEOMETRY ITERATIONS

The SCF procedure did not converge, which caused the geometry optimization to be aborted. See WARNING: SCF NOT COMPLETELY CONVERGED below.

ERROR: imo is not occupied PT1W

ERROR: imo is occupied PT1W

The electronic structure is not auf-bau, that is the LUMO is lower in energy than the HOMO. Disable the KeepOrbitals option by setting it to a large number and try different SCF version until you get the SCF properly converged. See the SCF Troubleshooting section for details.

ERROR: More cycles needed. Geometry NOT CONVERGED
ERROR: GEOMETRY DID NOT CONVERGE
ERROR: GEOMETRY NOT CONVERGED
ERROR: TRANSIT FAILED

The geometry optimization did not converge in the allowed number of steps. See the Geometry Optimization Troubleshooting section for recommendations on how to proceed.

ERROR: TAPE12 NOT SUPPLIED with TAPE10
ERROR: Relativistic option is used but no file with core potentials was found

Before running a relativistic calculation you need to run the dirac program to create a TAPE12 file. This can be done automatically by adding a Basis block to the input and executing ADF as $AMSBIN/adf instead of adf.exe.

ERROR: calculating FDE interactions for the same fragment

It's a program bug. Send your input file to SCM

ERROR: Abort because one processor has no integration points

Repeat the calculation with fewer processors in the MPI job.

ERROR: Davidson method not converged

The Davidson method did not converge that is used in the calculation of excitation energies. Increase the number of ITERATIONS in the EXCITATIONS block key, or use a larger TOLERANCE.

ERROR: EMPTY FIELD FOUND AT END OF …
ERROR: Cannot read … even as a string

Check the QMMM block in the input. Every simple key in the block (as opposed to a block key) must be followed by a value. See the QMMM Manual for the correct syntax.

ERROR: LenRecUnits: reclen units not determined

This error is most likely caused by compiler problems or by incorrect compilation flags. See https://www.scm.com/support/downloads/platform-specific-information/ for supported compiler versions and contact SCM for an advice on compilation flags. Do not forget to include the $AMSHOME/buildinfo file in your request.

13.8 Warnings

WARNING: (slightly) inaccurate ETA integral(s)
WARNING: SUSPICIOUS ETA INTEGRALS

Eta's are symmetry adapted STO's. This error is usually caused by numerical stability problems in the symmetric matrix eigensolver. On the x86(-64) platforms you can try using a different version of the Intel MKL library (newer is not always better).

WARNING: 1-character identifier(s) in DEFINE

The Define input block does not allow single-character identifiers.

WARNING: activating keyword SDFTENERGY for the calculation of subsystem DFT energy.

For more than two frozen subsystems the sDFT energy needs the corresponding logical switch, which is normally activated via the SDFTENERGY key in the FDE input block.

WARNING: AMAT not far from singularity

WARNING: Bmatrix not far from singular

Amat and Bmatrix are used in COSMO solvation model. This warning could be printed when the distance between two COSMO surface points is shorter than 0.01 Bohr. You may want to try different COSMO accuracy parameters.

WARNING: Atom (…) exceeded the max allowed bonds in DimQMHybridization

The number of bonds per atom is exceeded in DimQM. Check your input geometry for correctness (coordinate units, etc.). The bond limit per atom type is currently hardcoded in the program and cannot be increased with recompiling the program.

WARNING: inaccurate integration of Core Density

WARNING: BAD CORE INTEGRAL …

The core density must integrate to number of core electrons very accurately. If this is not the case then something is wrong either with the integration grid, with the input geometry or with the core orbitals. In any case, check your input geometry (Angstrom vs. Bohr vs. the Units input block, check bond lengths). Try using a smaller core and/or better grid.

WARNING: BAD FIT

WARNING: CANNOT NORMALIZE THE FIT

WARNING: LOSS OF CHARGE

These warnings are printed when the accuracy of the STO fit is not high enough. This might, for example, happen for negatively charged molecules for which the electron density cannot be adequately represented by the fit functions. This can be solved by using a larger fit type (e.g. FitType QZ4P in the Basis input block) or using the AddDiffuseFit keyword, or by using the ZlmFit instead of STOfit.

WARNING: The use of the SOLVATION option with non-basic fragments is not supported

WARNING: Bond energies are incorrect

Do not use the energy decomposition analysis if the fragments are calculated with COSMO solvation.

WARNING: C(lin) or D(lin) not supported in Freq runs

When performing numerical frequency calculation for a linear molecule you should not set the symmetry explicitly to C(lin) or D(lin). Use something like C(8v) or D(8h) instead.

WARNING: CD spectrum for triplex excit. NOT SUPPORTED

WARNING: CD spectrum for triplet excitations is NOT SUPPORTED

A CD spectrum calculation has been requested for triplet-only excitations, which is not supported by the program. The excitation rotary strengths will not be calculated.

WARNING: In ReadEpsilons: HOMO energy is positive

WARNING: Check if basis or fit sets are dependent

The HOMO energy is positive where it should usually be negative. The positive HOMO energy means the solution is not stable and the following TDDFT calculation may produce incorrect results.

WARNING: Combination of unrestricted and MODEL potential not extensively tested

Using the model XC functionals for spin-unrestricted calculations is not officially supported. Read the literature on the functional used to make sure you do not obtain nonsense results.

WARNING: Virtuals almost lin. dependent

WARNING: Consider using keyword DEPENDENCY

This warning is printed is when the smallest eigenvalue of the virtual SFO overlap matrix is smaller than 1e-5. You may want to add a Dependency keyword to your input.

WARNING: convMCD very small. Achieving convergence is unlikely

This warning is printed when the MCD convergence criterion is smaller than 1e-4. You may want to set the CONVMCD parameter in the MCD input block to a larger value if the CG solver does not converge.

WARNING: Core orb. en. with MODEL xc pot not implemented

Calculation of the core orbital energies in combination with model XC potentials (e.g. LB94) is not implemented.

WARNING: Could not find vecdimension excitations on SUBTAPE

WARNING: Trying to find vecdimension in ResponsePmats section …

When performing a Subsystem DFT excitations calculation the excitations must be calculated for each sub-system (fragment).

WARNING: Density fitting may not be accurate enough!

WARNING: H12 and Electronic Coupling possibly INACCURATE

WARNING: Diagonal energies and overlap(s) should be accurate

These messages are printed together and already contain a hint. To avoid the problem you may want to increase the quality of the fit using one or more of the following options: FitType QZ4P, AddDiffuseFit, NumericalQuality {Good|VeryGood|Excellent}.

WARNING: diagonalization error in dvdrvrhf. Results will be incomplete.

WARNING: diagonalization error in dvdrvrhfso. Results will be incomplete.

The LAPACK subroutine dsygv (dzhegv in case of dvdrvrhfso) returned a non-zero status for the (A-B)(A+B) matrix. Analysis of the obtained solution will not be done.

WARNING: dihedral jumps 180 degrees

This warning is printed by the old optimizer branch. Check the optimization results and switch to the new branch if you encounter any problems.

WARNING: DIM dipoles not converged upon SCF convergence

Use more tight SCF convergence criteria.

WARNING: No default parameters for classical Hirshfeld partitioning

WARNING: Dispersion added with factor 1.0

You may want to set the steepness factor ATT0 and the scaling factor BTT0 for the SCBR empirical dispersion.

WARNING: doing final FDE cycle - FDE NOT converged

The FDE freeze-and-thaw procedure has not converged. You may want to increase the number of the FDE freeze-and-thaw cycles.

WARNING: DPERTURB calculation, no CI-type couplings calculated

Since the CI-coupling does not make much sense in combination with DPERTURB, the former is disabled.

WARNING: ERROR OPENING ONE OF THE NBO DATA FILES. SEE OUTPUT

Check the output to see which NBO file is missing and look for clues in the NBO output preceding the calculation.

WARNING: Eucledian distance between NEB images is too short. NEB may have problems

The total Eucledian distance between two NEB images should be large compared to the Cartesian optimization step size. Otherwise the NEB optimization may result in a zig-zag path and may eventually fail. One way to increase the distance is by reducing the number of images.

WARNING: Extra DFT term related to g_xc suspiciously large

This warning is printed when the abs(beta(diff)) value is larger than 1e-3 and is larger than abs(beta(total)) in a hyperpolarizability calculation. This may be an indication of a numerical instability.

WARNING: FDE in combination with ZlmFit not thoroughly tested. Consider using STOFIT

[[[Mirko???]]]

WARNING: FILE POSSIBLY TOO BIG IN AORUN

This warning is printed when the AORUN input keyword is specified in certain types of TDDFT calculations (e.g. polarizability). In such a case the writing of the orbital values on the grid is disabled because otherwise it would result in a huge TAPE10 file. Note: the AORUN keyword serves no other purpose but to disable writing of the orbitals to TAPE10, so it should be used with care.

WARNING: Geometry optimization did not converge - frequencies are not calculated

This warning is printed when ADF is instructed to compute the vibrational spectrum of a molecule after geometry optimization but the optimization did not converge. In this case the frequencies are not computed.

WARNING: GRADIENT ILL-DEFINED FOR BOND ANGLE 180 OR 0

This warning is generated by the old geometry optimization branch when a valence angle gets close to 0 or 180 degrees. Such angles cannot be treated properly by the optimizer so one should avoid them, for example by adding dummy atoms.

WARNING: Hessian is incomplete

This warning just informs the user that only a fraction of the Hessian matrix has been done, as specified by the NUC keyword of the AnalyticalFreq input block. The missing matrix elements are calculated from an empirical MM forcefield and the combined Hessian matrix is analyzed.

WARNING: inaccurate elimination of XX motions, recover

This warning is generated by the old geometry optimization branch. Add PRINT FREEMV to your input to get more information.

WARNING: Inconsistent relativistic options used

This warning is printed when inconsistent relativistic options are used for the fragments and the final calculation and the RELOPTINCONS keyword is used to prevent ADF from stopping the calculation. If you add RELOPTINCONS then you probably know what you're doing.

WARNING: Infrared Intensities are wrong

WARNING: Infrared Intensities wrong

These warnings are issued when the computed dipole moment derivative is larger than 0.1 a.u. for a normal modes for which it must be zero by symmetry.

WARNING: Inter-node communication is slow! You may want to run on fewer nodes (or even one).

WARNING: Intra-node communication is VERY slow! Check CPU affinity, hyperthreading and over-subscription

The program tests the latency of the MPI communication at the beginning of each parallel calculation. The overall (i.e. the whole MPI "world" including all processes from all nodes) and the intra-node (i.e. including only the processes within one node) latency are measured separately. The first warning is printed if the overall latency is higher than 1 ms. This usually occurs when running on multiple nodes with a slow interconnect (e.g. 1 Gbps Ethernet), or when running on too many nodes. In this case it is recommended to run on fewer nodes. It is **not** recommended to run multi-node jobs at all if the cluster does not have a fast interconnect. The second warning is printed when the intra-node latency is higher than 1 ms, which should **never** happen. If the intra-node communication is slow the user is advised to stop running further MPI calculations until the issue is resolved. Possible reasons for this are: incorrect CPU affinity settings enforced by the batch scheduler (e.g. more than one process is bound to the same core), host is oversubscribed (i.e. other jobs are running on the same machines and ADF has to compete with them for resources). When two or more MPI jobs have to compete for the same resources both jobs will a low performance because the MPI synchronization calls also consume CPU cycles.

WARNING: INVALID SPRING PARAMETER. DEFAULTS ARE ENFORCED.

The valid values for the first NEBSPRING parameter are 1 through 6. If a value outside this range is specified it is set to 1 and this warning is printed.

WARNING: IRC START REQUEST IMPOSSIBLE: ADJUSTED

This warning is printed when the IRC eigevector index specified by the IRC HESS parameter is larger than the number of free coordinates.

WARNING: KeepOrbitals on

This warning is printed when the MO populations become non-aufbau due to the KeepOrbitals feature. The KeepOrbitals usually kicks in after N/2 SCF iterations, where N is the max number of SCF iterations allowed. See the Occupations keyword for details.

WARNING: LB94 XC FUNCTIONAL USED FOR ENERGY GRADIENTS

The LB94 XC functional is supposed to be used to calculate spectroscopic properties such as optical excitation spectra and is not intended for the bonding energy (and gradients thereof) calculations.

WARNING: LMO beta analysis has nonzero B-field frequency. Check output.

This warning is printed when the magnetic perturbation field in the dynamic beta tensor (optical rotation) calculation has a finite frequency. Use the DEBUG keyword of the LOCORB input block to get more information.

WARNING: localized MOs: unknown symmetry key …

This warning is printed when there is an unrecognized keyword is found in the LOCORB input block. Check that your input conforms the documentation of the LOCORB input block.

WARNING: Low value for "numvec" detected in AllocDavidson

WARNING: Using numvec = [number of excitations] + 5 instead

WARNING: Specify larger value for VECTORS in EXCITATION input block

These warnings are printed together and are self-explanatory.

WARNING: GIAOS ARE NOT IMPLEMENTED WITH LIFETIME. SWITCHING OFF GIAOS

WARNING: MAGOPTROT AND QUADRUPOLE OPTIONS ARE MUTUALLY EXCLUSIVE. SWITCHING OFF QUADRUPOLE

WARNING: MAGPERT AND QUADRUPOLE OPTIONS ARE MUTUALLY EXCLUSIVE. SWITCHING OFF QUADRUPOLE

WARNING: MAGPERT AND VROA OPTIONS ARE MUTUALLY EXCLUSIVE. SWITCHING OFF MAGPERT

WARNING: OPTICALROT AND VROA OPTIONS ARE MUTUALLY EXCLUSIVE. SWITCHING OFF OPTICALROT

WARNING: OPTROT AND GIAO AND NBO OPTIONS WITHOUT MAGPERT ARE NOT IMPLEMENTED. SWITCHING OFF NBO

WARNING: OPTROT AND QUADPERT OPTIONS ARE MUTUALLY EXCLUSIVE. SWITCHING OFF QUADPERT

WARNING: VROA OPTION MUST BE USED WITH QUADRUPOLE. SWITCHING ON QUADRUPOLE

WARNING: VELOCITYORD DOES NOT WORK WITH GIAO. SWITCHING OFF GIAO

WARNING: ZLMFIT AND FITAODERIV ARE NOT IMPLEMENTED. SWITCHING OFF FITAODERIV

These warnings are results of the AOResponse internal checks.

WARNING: max no. of iterations reached

This warning is printed when the CPKS procedure did not converge in the given number of steps. This may indicate a numerical stability issue or some other problem. In any case, you should check if the SCF has converged, if the electronic configuration is correct. Increasing the numerical accuracy might also help. Finally, you can try increasing the number of CPKS cycles for the feature in question (TD-DFT, analytical frequencies, etc.).

WARNING: maxBlockSize does not divide (maxBasisSize - minRestartSize). Suboptimal performance expected!

This warning is printed by the Davidson method using the PRIMME library. The fact that (maxBasisSize - minRestartSize) is not divisible by maxBlockSize does not affect the results so there is no reason to worry. [[[Robert is there any way to influence this?]]]

WARNING: MIXED OCCUPATIONS in locmo (localized MO)

This warning is printed when occupation numbers of the localized MOs after 2x2 rotations do not match the original ones, which essentially means that the occupied and virtual orbitals have been mixed together.

WARNING: mo levels nearly degenerate

This warning is usually issued by the CPKS driver routine when the energy difference between an occupied and a virtual orbital is less than 1e-3 a.u.. This usually means that the HOMO-LUMO gap is very small which may lead to problems in the CPKS procedure.

WARNING: MODTRC refers to non-existing free variable

WARNING: … selecting the first mode instead

This warning is printed by the old optimization branch in response to an invalid setting for the transition state search vector. In this case the TS procedure will use the eigenvector corresponding to the lowest eigenvalue of the Hessian matrix.

WARNING: More iterations needed. MODERATELY CONVERGED

WARNING: More iterations needed. NOT CONVERGED

WARNING: SCF did not converge - frequencies are not calculated

WARNING: SCF NOT COMPLETELY CONVERGED

One of these warnings is issued when the SCF procedure did not (completely) converge. See the SCF Troubleshooting section for recommendation on how to improve the SCF convergence.

WARNING: More nodes than energies on the contour: this is inefficient

This warning is printed by the NEGF code when the number of energies on the contour is smaller than the number of MPI tasks. This means that you can probably perform the calculation in the same time using fewer processors.

WARNING: multiple small orbital energy differences found

This warning is printed when the undocumented ASYMPCOR keyword is specified and more than one occupied-virtual orbital pair is found for which the energy difference is less than 1e-3 a.u..

WARNING: Negative root found. AllocDavidson

WARNING: Negative root found. AllocDavidsonso

These warnings are issued when the Davidson procedure finds a negative eigenvalue. This usually indicates a triplet instability.

WARNING: negative sqrt in the CI-type coupling parameters

WARNING: the results might be not trustworthy!!!!!

This warning is printed by the subsystem TDDFT module when checking requirements for the CI coupling calculation. The latter will not be done if any element of the CI_omega matrix is imaginary.

WARNING: no convergence in svdcmp

This is a generic warning from a singular value decomposition routine. When this occurs the SVD procedure will be aborted, which may lead to incorrect results in the module that called the routine.

WARNING: No DIM points found for projection matrix.

This warning is printed when projecting rigid rotations/translations from the DIM/QM energy gradients in the case there are no DIM points found within the cutoff radius specified by the PROJECTIONMATRIXPOINTS CUTOFF keyword.

WARNING: no optimization done in DVDSON: using guess vector results

This warning is printed to remind the user that he/she has specified SFGUESS or TRUSTSFGUESS in the input.

WARNING: root must be bracketed in rtsafe

The rtsafe function finds a root of the given callback-function in the given interval. The requirement is that the function has the opposite signs at the ends of the interval. This warning is printed when the requirement is not fulfilled.

WARNING: rtsafe exceeding maximum iterations

This warning is printed when the rtsafe function fails to find a root within the allowed number of iterations, currently 100.

WARNING: Non Aufbau energies used

This warning is printed by the Response code (excitations, polarizabilities, etc.) when the HOMO lies higher than LUMO in energy or when these are degenerate.

WARNING: Non-standard use of LAP functional

WARNING: Non-standard use of LYP functional

The LAP and LYP functionals already contain a term for correlation at the LDA level. This warning is printed when an LDA functional is specified in addition to LAP or LYP.

WARNING: NONRELATIVISTIC RUN WITH RELATIVISTIC COREPOTENTIALS

WARNING: RELATIVISTIC RUN WITHOUT RELATIVISTIC COREPOTENTIALS

This warning is printed when the CorePotentials keyword is used (or not used) inconsistently with the Relativistic option. Before running a relativistic calculation you need to run the dirac program to create a TAPE12 file, which then needs to be specified in the CorePotentials keyword. This can be done automatically by adding a Basis block to the input and executing ADF as $AMSBIN/adf instead of adf.exe.

WARNING: not all AIM bond paths converged

This warning is printed by the AIM bond path search module when some of the paths failed to converge to an attractor (a nucleus).

WARNING: not all scratch files were closed

This warning probably indicates a programmer error. Send the output to support@scm.com.

WARNING: not enough initial estimates

WARNING: Too many initial estimates.?

One of these warnings is issued when the number of initial vectors in the Davidson procedure is either too high or too low.

WARNING: Number of states in energy window changed

This warning is printed when the number of molecular orbitals in the energy window given with the ELTHRES/EUTHRES sub-keys of the VIBRON input block changes during calculation.

WARNING: occupations may be NOT AUFBAU

This warning is printed during a single-point calculation or the last iteration of a geometry optimization if the HOMO has a higher energy than the LUMO. For fractionally occupied orbitals, the HOMO is the highest MO that is populated by at least 1e-6 electrons and the LUMO is the lowest MO that misses at least 1e-6 electron to be fully occupied.

WARNING: omega and d(omega) incompatible in Verdet run

This warning is printed during calculation of the Verdet's constant when the value of the frequency ω is lower than the half-width of the interval used for numerical differentiation $\Delta\omega$.

WARNING: Out of parameterized Coulomb radii limits

The Coulomb radius parameters used to determine the COSMO sphere radii are currently available for elements up to 105. For heavier elements this warning is printed and the calculation stops.

WARNING: partially occupied orbitals

This warning is printed at the end of the SCF process if it results in partially occupied orbitals. The user should examine the resulting electronic structure carefully to see if this is the result they expected. One way to avoid fractional occupations is to repeat the calculation as spin-unrestricted with a non-zero total spin.

WARNING: Patchkovskii routines for PBE used. Check output

This warning is printed when USESPCODE is specified in the input together with the PBE functional. The bonding energy and other fragment analysis data may be incorrect.

WARNING: potential problem with dependency in AOResponse. Check output
WARNING: potential problem with dependency in LOCMO. Check output

When using the Dependency keyword, the linearly-dependent orbitals are not removed completely but are effectively excluded from variation by setting the corresponding diagonal matrix element to a large positive value. One of these warnings is issued when the any of the molecular orbitals included in the variation have the energy close to that large value, which probably means there is some bug in the SCF bookkeeping.

WARNING: Pressure must be positive, adjusted

This warning is printed when the pressure specified in the Thermo keyword is smaller than 1e-4 atmosphere.

WARNING: PRIMITIVE FIT COEFFICIENTS A BIT LARGE
WARNING: PRIMITIVE FIT COEFFICIENTS LARGE
WARNING: PRIMITIVE FIT COEFFICIENTS TOO LARGE

One or more of these warnings are printed when the number of electrons due to a single STO fit-function exceeds certain limits (20, 200, and 2000 times the largest number of valence electrons on a single atom in the molecule, respectively). This may indicate a linear dependence (over-fitting) problem in the fit or the basis set.

WARNING: problem with NEB optimisation. Re-initializing Hessian

This warning is printed during an NEB calculation if the Eucledian angle between three images becomes less than 30 degrees. This may indicate that there are too many NEB images for such a short reaction path.

WARNING: ReadExSym: found large energy differences

This warning is printed in a Vibron calculation when the excitation energy difference between two geometries is larger than 0.2 eV. You should check the results carefully.

WARNING: Restricted Open-Shell calculation, Unrestricted setup may be needed

This warning is printed when a molecule with an odd number of electrons (a radical) is calculated spin-restricted. This often indicates an error in the input (either wrong geometry or a missing Charge keyword). This warning may safely be ignored if the result of this calculation is to be used as a fragment.

WARNING: SchmidtOrth keeps finding noise

This warning is printed by the Schmidt orthogonalization procedure when it fails to orthogonalize the input vector to the reference set within 4 attempts. Two normalized vectors are considered orthogonal when their dot product is smaller than 1e-8 in the absolute value.

WARNING: sDFT energies should be used together with the FULLGRID option! Results might not be reliable!

Enable the FULLGRID option.

WARNING: SLOW SCF

WARNING: … apply ElectronSmearing

In case of really problematic SCF a finite electron temperature (smearing of electrons around the Fermi level) is applied automatically if the user a) has not specified smearing via input and b) has not defined the occupation numbers explicitly. Furthermore, smearing is not automatically turned on in single-point calculations and in the final geometry. That is, it is only allowed during geometry optimization in the intermediate geometries. To disable automatic smearing specify Occupations Smearq=0 in the input.

WARNING: small delta epsilon found

This warning is printed when the energy difference in an occupied-virtual MO pair is smaller than 0.001 a.u., which means the HOMO and LUMO are nearly degenerate.

WARNING: SOLVATION block: DISC is not compatible with C-Mat POT

This warning is printed when the DISC option in the Solvation input block is set for the C-Matrix POT (which is the default), where it does not make sense. The user should check their input and either remove the DISC option or set C-Matrix to EXACT or FIT.

WARNING: SOMCD indicated but closed-shell calculation

The SOMCD feature (used to calculate the C terms of the magnetic circular dichroism spectra) is available only in spin-unrestricted calculation. If SOMCD is specified in a spin-restricted calculation then it is ignored and this warning is printed.

WARNING: SSB-D XC FUNCTIONAL USED FOR ENERGY GRADIENTS

The use of "GGA SSB-D" in a geometry optimization may lead to numerical issues for some systems. Check carefully that the energy goes down smoothly and continuously. Also compare with a calculation performed with the "METAGGA SSB-D" specified in the XC input block.

WARNING: Strange column number EXCITATIONS

The second (undocumented) argument of the irrep sub-keys of the Davidson and Exact sub-blocks can optionally specify that the 2nd or 3rd projection of a multi-dimensional irrep should be used to calculate intensities. This warning is printed when the number specified is larger that the dimensionality of the irrep.

WARNING: Suspicious delocalized coordinates. Change GEigenvalueThreshold or optimize in Cartesian coords

This warning is printed when the number of generated delocalized coordinates is not equal to 3N-6 for a system of 5 atoms and more. This may indicate some problems in generation of internal coordinates for the input geometry, for example because of isolated molecules (which should normally not be a problem). This warning may also be caused by the presence of a block constraint, in which case it may be safely ignored. Otherwise, one should check the number of the generated delocalized coordinates in the output file. If it is larger than 3N-6 then it is advised to look at the "G-matrix eigenvalues" table in the output and set the GEigenvalueThreshold parameter in the Geometry block to an average of the eigenvalues number 3N-6 and 3N-5.

WARNING: Symmetry is not yet supported in VCD

The vibrational circular dichroism (VCD) intensities are not calculated if the symmetry used in the calculation is other than C:sub:*1* (i.e. NOSYM)

WARNING: TDA in FDEc also requires TDA excitations as input

Use TDA consistently in coupled frozen density embedding (FDEc).

WARNING: Temperature must be positive, adjusted

The temperature value specified in the Thermo keyword must be larger than 0.1 K.

WARNING: The full TDDFT kernel icw symmetry may give incorrect results.
WARNING: Results can be checked with symmetry NOSYM

As this warning already suggest, you can check the results with SYMMETRY NOSYM.

WARNING: thin plane error. HULL33
WARNING: thin planes inconsist 1. HULL33

These warnings can appear when the Te Velde integration grid is used. Use the Becke grid in this case.

WARNING: Unknown GGA functional for UFF dispersion.'

The UFF dispersion parameters depend on the XC functional used. Currently the parameters are defined for the PBE, B3LYP, PW91 and B3PW91 functionals. For any other functional the PBE parameters are used and this warning is printed.

WARNING: Unknown GGA functional for MBD dispersion.'

The MBD dispersion parameters depend on the XC functional used. Currently the parameters are defined for the PBE, PBE0, HSE03 and HSE06 functionals. For any other functional no short-range damping is applied and this warning is printed.

WARNING: Unknown GGA used. Dispersion added with factor 1.0

The Grimme dispersion corrections are defined per XC functional. For any unknown functional the dispersion is, by default, added with a factor 1.0, which can be changed from input.

WARNING: Unknown global optimization type requested. Using Quasi-Newton

WARNING: Unknown NEB optimization type specified. Using default

WARNING: Unknown optimization type requested. Using Quasi-Newton

One of these warnings is printed when the NEBOPT sub-key of the Geometry block contains an invalid argument.

WARNING: Unknown XC functional for DFT-D3

This warning is printed when an unknown XC functional is specified together with "Dispersion Grimme3" keyword in the XC input block. In such a case the PBE parameters are used for the dispersion correction.

WARNING: Untested MBD dispersion parameters are used. Check results.

WARNING: Untested dispersion parameters are used. Check results.

This warning is printed when the system under study contains atoms with an element index greater than the maximum defined for the given correction. Currently, the limits are set as follows: 54 for Grimme, 86 for MBD, 94 for Grimme3, 102 for sCBR, and 103 for the UFF dispersion type.

WARNING: Unrecognized key for Irrep

This warning is printed when an unrecognized irreducible representation name is given in the Davidson or Exact sub-blocks of the Excitations block.

WARNING: use of multiple fragments not yet allowed !!!

WARNING: Use results from parallel runs with caution!

These self-explanatory informative warnings are printed by the (relatively new) SubsystemResponse module.

WARNING: USING >ADIABS< scheme in vibron.

WARNING: DO YOU REALLY KNOW WHAT YOU ARE DOING?

The ADIABS scheme in the Vibron module can only be used for totally symmetric modes. Add ONLYSYM to the Vibron block key.

WARNING: using LDA-only response. Please check your output

This warning is printed by the AOResponse code in cases when the use of the GGA is disabled automatically, usually in open-shell calculations.

WARNING: VERY HIGH VALUE FOR FRANGE IN RANG2
WARNING: VERY HIGH VALUE FOR QPNEAR IN RANGQP
This warning is printed by the grid generation routine when a basis or an STO fit function has a very long range (>10,000 a.u.). This may occur for an extremely diffuse function when a very high integration accuracy is requested.

WARNING: VERY SMALL A(0) IN LOCORB1. Check output!
This warning is printed by the routines performing the Pipek-Mezey (PM) and Boys localization on first-order perturbed molecular orbitals when the absolute value of the A(0) denominator becomes smaller than 1e-10.

WARNING: VROA OPTION MUST BE USED WITH QUADRUPOLE
The AOResponse Quadrupole options is switched on and this warning is printed when the vibrational Raman optical activity (VROA) is to be calculated.

WARNING: WRONG COORDINATES
This warning is printed by QUILD in the same situations where ADF exits with the same "WRONG COORDINATES" message: the input geometry does not satisfy the point group specified by the Symmetry keyword. This usually happens when the molecule's orientation does not match the symmetry operations. Note that when the point group is set from input, the program does not re-orient the input geometry to match the symmetry.

WARNING: wrong nr of functions deleted. ORTNRM
Internal ADF error message. Related to STO fit. Send your input file to SCM. Alternative is to use ZlmFit.

WARNING: zero number of filtered transitions found, try larger demax
This warning is printed by the approximate TD-DFT module (TD-DFTB, sTDDFT, sTDA, KSSPECTRUM) when there are no transitions found for the given energy range. This should not happen if there is at least one occupied and one virtual orbital present because in ADF the range is by default very large (~1e308 a.u.)

CHAPTER

FOURTEEN

EXAMPLES

14.1 Introduction

The AMS package contains a series of sample runs. They consists of UNIX scripts (to run the calculations) and the resulting output files.

The examples serve:

- To demonstrate how to do calculations. The number of options available in AMS is substantial and the sample runs do not cover all of them. They should be sufficient, however, to get a feeling for how to explore the possibilities.
- To work out special applications that do not fit well in the User's Guide.

Note: Most of the provided samples have been devised to be short and simple, at the expense of physical or chemical relevance and precision or general quality of results. They serve primarily to illustrate the use of input, necessary files, and type of results. The descriptions have been kept brief. Extensive information about using keywords in input and their implications is given in the User's Manual.

Where references are made to the operating system (OS) and to the file system on your computer, the terminology of a UNIX type OS is used.

All sample files are stored in subdirectories under `$AMSHOME/examples/`, where `$AMSHOME` is the main directory of the AMS package. There are many subdirectories in `$AMSHOME/examples/`: the examples presented in this section are located in `$AMSHOME/examples/adf/`. Each sample run has its own directory. For instance, `$AMSHOME/examples/adf/HCN/` contains an ADF calculation on the HCN molecule.

Each sample subdirectory contains:

- A file TestName.run: the UNIX script to execute the calculation(s). A sample may involve several calculations, for instance a molecular AMS calculation (using ADF engine) followed by NMR calculation (using the NMR program) to compute chemical shifts.
- A file TestName_orig.out.gz: the resulting output(s) against which you can compare the outcome of your own calculation. Note: the files are compressed using gzip (https://en.wikipedia.org/wiki/Gzip).
- Zero or more files with a .ams extension. These files, if present, are intended for AMSinput and demonstrate the same functionality as the two files above. However, there are also differences between the .ams and the TestName.run files so the results obtained with the .ams files **cannot be compared directly** with TestName_orig.out. Also, the TestName.run file usually contains more than one calculation, for which more than one .ams file is required. That's why in some directories you may find more than one .ams file.

Technical notes:

- Running the examples on Windows: You can run an example calculation by double-clicking on the appropriate .run file. After the calculation has finished, you can compare the TestName.out file with the reference TestName_orig.out file. See remarks about comparing output files below.

- The UNIX scripts make use of the *rm* (remove) command. Some UNIX users may have aliased the *rm* command. They should accordingly adapt these commands in the sample scripts so as to make sure that the scripts will remove the files. New users may get stuck initially because of files that are lingering around after an earlier attempt to run one of the examples. In a subsequent run, when the program tries to open a similar (temporary or result) file again, an error may occur if such a file already exists. Always make sure that no files are left in the run-directory except those that are required specifically.
- It is a good idea to run each example in a separate directory that contains no other important files.
- The run-scripts use the environment variables `AMSBIN` and `AMSRESOURCES`. They stand respectively for the directory that contains the program executables and the main directory of the basis set files. To use the scripts as they are you must have defined the variables `AMSBIN` and `AMSRESOURCES` in your environment. If a parallel version has been installed, it is preferable to have also the environment variable `NSCM`. This defines the default number of parallel processes that the program will try to use. Consult the Installation Manual for details.
- As you will note the sample run scripts refer to the programs by names like 'ams', 'nmr', and so on. When you inspect your `$AMSBIN` directory, however, you may find that the program executables have names 'ams.exe', 'nmr.exe'. There are also files in `$AMSBIN` with names 'ams', 'nmr', but these are in fact scripts to execute the binaries. We strongly recommend that you use these scripts in your calculations, in particular when running parallel jobs: the scripts take care of some aspects that you have to do otherwise yourself in each calculation.
- You need a license file to run any calculations successfully. If you have troubles with your license file, consult the Installation manual. If that doesn't help contact us at support@scm.com

When you compare your own results with the sample outputs, you should check in particular (as far as applicable):

- Occupation numbers and energies of the one-electron orbitals;
- The optimized geometry;
- Vibrational frequencies;
- The bonding energy and the various terms in which it has been decomposed;
- The dipole moment;
- The logfile. At the end of a calculation the logfile is automatically appended (by the program itself) to the standard output.

General remarks about comparisons:

- For technical reasons, the discussion of which is beyond the scope of this document, differences between results obtained on different machines (or with different numbers of parallel processes) may be much larger than you would expect. They may significantly exceed the machine precision. What you should check is that they fall well (by at least an order of magnitude) within the *numerical integration* precision used in the calculation.
- For similar reasons the orientation of the molecule used by the program may be different on different machines, even when the same input is supplied. In such cases the different orientations should be related and only differ in some trivial way, such as by a simple rotation of all coordinates by 90 degrees around the z-axis. When in doubt, contact an SCM representative.

14.2 Model Hamiltonians

14.2.1 Special exchange-correlation functionals

Example: r2SCAN-3c

`Download r2SCAN-3c.run`

```
#!/bin/sh

# r2SCAN-3c example

# To use r2SCAN-3c in ADF one needs the r2SCAN functional icw a predefined basis set (mTZ2P),
# D4-dispersion and a geometrical counterpoise (gCP) correction.
# r2SCAN-3c also needs relativistic effects to be included,
# which are included by default in ADF, and (at least) a numericalquality GOOD.

# The usage of metagga r2scan-3c in the XC block automatically includes the correct
# D4 dispersion and gCP correction. It is in fact the same as using
# XC
#   gcp stor2scan3c
#   libxc r2scan
#   Dispersion Grimme4 s6=1.0 s8=0.0 s9=1.53 a1=0.42 a2=5.65 g_a=2.0 g_c=1.0
# end
# Note that LibXC is used to calculate the r2SCAN part of r2SCAN-3c.

$AMSBIN/ams << eor
System
  GeometryFile $AMSHOME/examples/adf/r2SCAN-3c/struc.xyz
end

Task SinglePoint

Engine ADF
  basis
    core None
    type mTZ2P
  end
  xc
    metagga r2scan-3c
  end
  numericalquality good
EndEngine
eor
```

Example: Asymptotically correct XC potentials: CO

Download CO_model.run

```
#!/bin/sh


# For property calculations, XC potentials with asymptotically correct (-1/r)
# behavior outside the molecule, the results tend to be superior to regular LDA
# or GGA calculations. This is especially true for small molecules and for
# properties that depend heavily on the proper description of the outer region
# of the molecule. In the example, all-electron basis sets are used. This is
# mandatory for the SAOP potential.

# In the first example, excitation energies are calculated with the GRACLB
# potential. This potential requires one number as argument: the experimental
# ionization potential in atomic units. This number can be either based on an
# experimental value, or on previous GGA total energy calculations.


AMS_JOBNAME=GRACLB $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     O  0              0   0
     C  1.128205364    0   0
  end
end

Task SinglePoint

Engine ADF
  title CO excitations grac potential
  beckegrid
    quality good
  end
  excitations
    lowest 10
    onlysing
  end
  basis
    core None
    type TZ2P
  end
  xc
    model GRACLB 0.515
  end
EndEngine

eor


# The same calculation with the SAOP XC potential would differ in the XC block
# only:


AMS_JOBNAME=SAOP $AMSBIN/ams <<eor
System
```

(continues on next page)

(continued from previous page)

```
  symmetrize
  atoms
     O  0              0    0
     C  1.128205364    0    0
  end
end

Task SinglePoint

Engine ADF
  title CO excitations SAOP
  beckegrid
    quality good
  end
  excitations
    lowest 10
    onlysing
  end
  basis
    core None
    type TZ2P
  end
  xc
    model SAOP
  end
EndEngine

eor


# SAOP depends on the orbitals which makes it more expensive to evaluate than
# GRAC for large molecules, but is much easier to use, since it does not require
# an ionization potential parameter as input.
```

Example: Meta-GGA energy functionals: OH

Download OH_MetaGGA.run

```
#! /bin/sh


AMS_JOBNAME=TPSSh $AMSBIN/ams <<eor
System
  atoms
     O     0.000000    0.000000    0.108786
     H     0.000000    0.000000   -0.870284
  end
end

Task SinglePoint

Engine ADF
  title OH radical, C*v symm. MP2(full)/6-31G* Geom. MP2=-75.5232063
  dependency
    bas 1.0e-4
```

(continues on next page)

(continued from previous page)

```
  end
  irrepoccupations
     A1 3 // 3
     A2 0 // 0
     B1 1 // 1
     B2 1 // 0
  end
  metagga
  basis
    type ET/ET-QZ3P
  end
  noprint BAS FIT FUNCTIONS SFO
  numericalquality VeryGood
  spinpolarization 1
  symmetry C(2V)
  unrestricted
  xc
    metahybrid TPSSh
  end
EndEngine
eor


AMS_JOBNAME=TPSS $AMSBIN/ams <<eor
System
  atoms
     O     0.000000    0.000000    0.108786
     H     0.000000    0.000000   -0.870284
  end
end

Task SinglePoint

Engine ADF
  title OH radical, C*v symm. MP2(full)/6-31G* Geom. MP2=-75.5232063
  dependency
    bas 1.0e-4
  end
  hartreefock
  irrepoccupations
     A1 3 // 3
     A2 0 // 0
     B1 1 // 1
     B2 1 // 0
  end
  metagga
  basis
    type ET/ET-QZ3P
  end
  noprint BAS FIT FUNCTIONS SFO
  numericalquality VeryGood
  spinpolarization 1
  symmetry C(2V)
  unrestricted
  xc
    metagga SCF TPSS
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/H
mv TAPE12 t12.rel

AMS_JOBNAME=H_HF $AMSBIN/ams <<eor
System
  Atoms
    H 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create H file=$AMSRESOURCES/ADF/ET/ET-QZ3P/H
  hartreefock
  xc
    gga PBE
  end
  corepotentials t12.rel
     H 1
  end
EndEngine
eor

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/O
mv TAPE12 t12.rel


AMS_JOBNAME=O_HF $AMSBIN/ams <<eor
System
  Atoms
    O 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create O file=$AMSRESOURCES/ADF/ET/ET-QZ3P/O
  hartreefock
  xc
    gga PBE
  end
  corepotentials t12.rel
     O 1
  end
EndEngine
eor


AMS_JOBNAME=H_HF_unr $AMSBIN/ams <<eor
System
  atoms
     H      0.0     0.0     0.0
  End
End

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  title H unrestricted correction
  fragments
     H H_HF.results/adf.rkf
  end
  hartreefock
  metagga
  numericalquality VeryGood
  spinpolarization 1
  unrestricted
  xc
    gga PBE
  end
EndEngine
eor


AMS_JOBNAME=O_HF_unr $AMSBIN/ams <<eor
System
  atoms
     O      0.0     0.0     0.0
  end
end

Task SinglePoint

Engine ADF
  title O unrestricted correction
  fragments
     O O_HF.results/adf.rkf
  end
  hartreefock
  irrepoccupations
     sigma  3 // 3
     pi     2 // 0
  end
  metagga
  numericalquality VeryGood
  spinpolarization 2
  symmetry C(lin)
  unrestricted
  xc
    gga PBE
  end
EndEngine
eor


AMS_JOBNAME=OH_HF $AMSBIN/ams <<eor
System
  atoms
     O      0.000000     0.000000     0.108786
     H      0.000000     0.000000    -0.870284
  end
end
Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  title OH radical, C*v symm. MP2(full)/6-31G* Geom. MP2=-75.5232063
  dependency
    bas 1.0e-4
  end
  energyfrag
     O  O_HF_unr.results/adf.rkf
     H  H_HF_unr.results/adf.rkf
  end
  fragments
     O  O_HF.results/adf.rkf
     H  H_HF.results/adf.rkf
  end
  hartreefock
  irrepoccupations
     A1 3 // 3
     A2 0 // 0
     B1 1 // 1
     B2 1 // 0
  end
  metagga
  noprint BAS FIT FUNCTIONS SFO
  numericalquality VeryGood
  spinpolarization 1
  symmetry C(2V)
  unrestricted
  xc
    gga PBE
  end
EndEngine
eor

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/H
mv TAPE12 t12.rel


AMS_JOBNAME=H_OLYP $AMSBIN/ams <<eor
System
  atoms
     H      0.0     0.0     0.0
  End
End
Task SinglePoint
Engine ADF
  create H file=$AMSRESOURCES/ADF/ET/ET-QZ3P/H
  hartreefock
  xc
    gga OLYP
  end
  corepotentials t12.rel
     H 1
  end
EndEngine
eor

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/O
```

(continues on next page)

(continued from previous page)

```
mv TAPE12 t12.rel

AMS_JOBNAME=O_OLYP $AMSBIN/ams <<eor
System
  atoms
     O     0.0    0.0    0.0
  End
End
Task SinglePoint
Engine ADF
  create O file=$AMSRESOURCES/ADF/ET/ET-QZ3P/O
  hartreefock
  xc
    gga OLYP
  end
  corepotentials t12.rel
     O 1
  end
EndEngine
eor


AMS_JOBNAME=H_OLYP_unr $AMSBIN/ams <<eor
System
  atoms
     H     0.0    0.0    0.0
  end
end

Task SinglePoint

Engine ADF
  title H unrestricted correction
  fragments
     H H_OLYP.results/adf.rkf
  end
  hartreefock
  metagga
  numericalquality VeryGood
  spinpolarization 1
  unrestricted
  xc
    gga OLYP
  end
EndEngine
eor

AMS_JOBNAME=O_OLYP_unr $AMSBIN/ams <<eor
System
  atoms
     O     0.0    0.0    0.0
  end
end

Task SinglePoint

Engine ADF
```

(continues on next page)

(continued from previous page)

```
  title O unrestricted correction
  fragments
     O O_OLYP.results/adf.rkf
  end
  hartreefock
  irrepoccupations
     sigma  3 // 3
     pi     2 // 0
  end
  metagga
  numericalquality VeryGood
  spinpolarization 2
  symmetry C(lin)
  unrestricted
  xc
    gga OLYP
  end
EndEngine
eor


AMS_JOBNAME=OH_OLYP $AMSBIN/ams <<eor
System
  atoms
     O     0.000000    0.000000    0.108786
     H     0.000000    0.000000   -0.870284
  end
end

Task SinglePoint

Engine ADF
  title OH radical, C*v symm. MP2(full)/6-31G* Geom. MP2=-75.5232063
  dependency
    bas 1.0e-4
  end
  energyfrag
     O  O_OLYP_unr.results/adf.rkf
     H  H_OLYP_unr.results/adf.rkf
  end
  fragments
     O  O_OLYP.results/adf.rkf
     H  H_OLYP.results/adf.rkf
  end
  hartreefock
  irrepoccupations
     A1 3 // 3
     A2 0 // 0
     B1 1 // 1
     B2 1 // 0
  end
  metagga
  noprint BAS FIT FUNCTIONS SFO
  numericalquality VeryGood
  spinpolarization 1
  symmetry C(2V)
  unrestricted
```

(continues on next page)

(continued from previous page)

```
  xc
    gga OLYP
  end
EndEngine
eor
```

Example: Hartree-Fock: HI

Download HI_EFG.run

```
#! /bin/sh


# Example shows a Hartree-Fock calculation with a non-relativistic, scalar␣
↪relativistic ZORA,
# and a spin-orbit coupled ZORA Hamiltonian. In this case ADF also calculates
# the electric field gradient (EFG) at the H and I nuclei (keyword QTENS).

# First the non-relativistic calculation.
# Note that in this case the all-electron basis sets are obtained from the
↪$AMSRESOURCES/ADF/ZORA directory

# == non-relativistic ==


AMS_JOBNAME=NR $AMSBIN/ams <<eor
System
  atoms
     H 0 0 0
     I 0 0 1.609
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type ZORA/TZ2P
    CreateOutput Yes
  end
  qtens
  xc
    hartreefock
  end
EndEngine
eor


# == scalar relativistic ZORA ==

# Next the scalar relativistic ZORA calculation. Note that in this case the all-
↪electron basis sets are
```

(continues on next page)

(continued from previous page)

```
# also obtained from the $AMSRESOURCES/ADF/ZORA directory, but this is default place␣
↪where the key BASIS will
# search for basis sets in case of ZORA. ADF will also calculate the EFG including␣
↪the small component density,
# also called SR ZORA-4.

AMS_JOBNAME=SR $AMSBIN/ams <<eor
System
  atoms
     H 0 0 0
     I 0 0 1.609
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  qtens
  relativity
    level scalar
    formalism ZORA
  end
  xc
    hartreefock
  end
EndEngine
eor



# == Relativistic Spinorbit ZORA ==

# Next the spin-orbit coupled relativistic ZORA calculation. Note that in this case␣
↪the all-electron basis sets are
# also obtained from the $AMSRESOURCES/ADF/ZORA directory, but again this is default␣
↪place where the key BASIS will search
# for basis sets in case of ZORA. ADF will calculate the EFG including the small␣
↪component density, also called ZORA-4.

AMS_JOBNAME=SO $AMSBIN/ams <<eor
System
  atoms
     H 0 0 0
     I 0 0 1.609
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  qtens
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    hartreefock
  end
EndEngine
eor


# == Relativistic Spinorbit ZORA with no symmetry ==

# Should give the same results as with symmetry.

AMS_JOBNAME=SO_nosym $AMSBIN/ams <<eor
System
  atoms
     H 0 0 0
     I 0 0 1.609
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  qtens
  relativity
    level spin-orbit
    formalism ZORA
  end
  symmetry nosym
  xc
    hartreefock
  end
EndEngine
eor
```

Example: B3LYP: H2PO

Download H2PO_B3LYP.run

```
#! /bin/sh


# Example shows an unrestricted B3LYP calculation. In this case ADF also
# calculates the hyperfine interactions at H, P, and O nuclei (keyword ESR).

# The 'RIHartreeFock%DependencyThreshold' key is set to 1E-4.
# One should check that the results remain reliable if one uses a smaller value
# for the DependencyThreshold key.


AMS_JOBNAME=TZ2P $AMSBIN/ams <<eor
System
  atoms
     O   1.492  0.000  0.000
     P   0.000  0.000  0.000
     H  -0.600 -0.650  1.100
     H  -0.600 -0.650 -1.100
  end
end

Task SinglePoint

Engine ADF
  title hfs H2PO B3LYP TZ2P
  esr
  end
  basis
    core None
    type TZ2P
  end
  numericalquality good
  rihartreefock
    dependencythreshold 1E-4
    quality Normal
  end
  spinpolarization 1
  unrestricted
  xc
    hybrid B3LYP
  end
EndEngine
eor



# For the hyperfine interactions it is important to use all-electron basis sets
# on the interesting nuclei. One can get more accurate results if one uses a
# larger basis set, like the QZ4P basis set, which is present in the
# $AMSRESOURCES/ADF/ZORA directory.


AMS_JOBNAME=QZ4P $AMSBIN/ams <<eor
System
```

(continues on next page)

(continued from previous page)

```
  atoms
     O   1.492  0.000  0.000
     P   0.000  0.000  0.000
     H  -0.600 -0.650  1.100
     H  -0.600 -0.650 -1.100
  end
end

Task SinglePoint

Engine ADF
  title hfs H2PO B3LYP QZ4P
  esr
  end
  basis
    core None
    type ZORA/QZ4P
  end
  numericalquality good
  rihartreefock
    dependencythreshold 1E-4
    quality Normal
  end
  spinpolarization 1
  unrestricted
  xc
    hybrid B3LYP
  end
EndEngine
eor

# The QZ4P results for the isotropic value of the A-tensor are approximately:
# -24.61 MHz for 17O, 957.33 MHz for 31P, and 110.83 MHz for 1H.

# You may want to compare the results with previous B3LYP results by N. R.
# Brinkmann and I. Carmichael, J. Phys. Chem. A (2004), 108, 9390-9399, which
# give for the Isotropic Fermi Contact Couplings (MHz) for the 2 A' State of H2
# PO using B3LYP, with an aug-cc-pCVQZ basis set: -24.24 MHz for 17O, 963.33
# MHz for 31P, and 111.51 MHz for 1H.
```

Example: Long-range corrected GGA functional LCY-BP: H2O

Download LCY-BP_H2O.run

```
#! /bin/sh


# Example for the range separated functional LCY-BP for a geometry optimization
# of water.

$AMSBIN/ams <<eor
System
  atoms
    O  -0.00000000   0.00000000   0.46983780
    H   0.00000000   0.63395481  -0.23491890
```

(continues on next page)

(continued from previous page)

```
    H   0.00000000  -0.63395481  -0.23491890
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    Gradients 0.0001
  End
End

Engine ADF
  title H2O geometry with LCY-BP functional
  basis
    type DZP
  end
  noprint BAS FIT FUNCTIONS SFO
  numericalquality Good
  xc
    gga Becke Perdew
    rangesep GAMMA=0.75
    xcfun
  end
EndEngine
eor
```

Example: Range-separated functional CAMY-B3LYP: H2O

Download CAMY-B3LYP_H2O.run

```
#! /bin/sh


# Example for the range separated functional CAMY-B3LYP for a geometry
# optimization of water. Note that CAMY-B3LYP is not the same as CAM-B3LYP.

$AMSBIN/ams <<eor
   Task GeometryOptimization
   System
      ATOMS
         O      0.000000    0.000000    0.000000
         H      0.000000    0.500000   -0.6
         H      0.000000   -0.500000   -0.6
      End
   End
   GeometryOptimization
      Convergence Gradients=1.0E-4
   End
   Engine ADF
      TITLE  H2O geometry with LC-BP functional
      NOPRINT BAS FIT FUNCTIONS SFO
      XC
         HYBRID CAMYB3LYP
         xcfun
```

(continues on next page)

(continued from previous page)

```
         RANGESEP GAMMA=0.34
      End
      BeckeGrid
         Quality good
      End
      RIHartreeFock
         UseMe False
      End
      basis
         Type DZP
      End
  EndEngine
eor
```

Example: Single point MP2

Download Water_MP2_AO.run

```
#! /bin/sh

# spin-restricted calculation on water, SOS-MP2
# AO-based algorithm, 6 imaginary time points

$AMSBIN/ams <<eor
System
  symmetrize
  atoms
     O   -0.702196054  -0.056060256   0.009942262
     H   -1.022193224   0.846775782  -0.011488714
     H    0.257521062   0.042121496   0.005218999
  End
End
Symmetry
  SymmetrizeTolerance 0.001
End

Task SinglePoint

Engine ADF
  Basis
    core None
    type TZP
  End

  XC
    MP2
    EmpiricalScaling SOS
  End

  MBPT nTime=6
EndEngine

eor
```

Example: Water Dimer SOS-AO-PARI-MP2

Download WaterDimer_SOS-MP2_AO.run

```
#! /bin/sh

# calculation on water, SOS-MP2
# AO-based algorithm, 8 imaginary time points

$AMSBIN/ams <<eor
System
  atoms
     O   -0.702196054  -0.056060256   0.009942262
     H   -1.022193224   0.846775782  -0.011488714
     H    0.257521062   0.042121496   0.005218999
     O    2.220871067   0.026716792   0.000620476
     H    2.597492682  -0.411663274   0.766744858
     H    2.593135384  -0.449496183  -0.744782026
  End
End

Task SinglePoint

Engine ADF
  Basis
    core None
    type TZ2P
  End

  XC
    empiricalscaling SOS
    MP2
  End

  MBPT
    formalism LT
    nTime 8
  End
EndEngine

eor
```

Example: Single point B2GPPLYP

Download Water_MeOH_B2GPPLYP.run

```
#! /bin/sh

# Dispersion corrected double hybrid calculation on
# a Water-Methanol complex using the double hybrid
# B2GPPLYP and standard settings
#
# First the fragments Water and MeOH are calculated.
# Next the bond energy of the Water-Methanol complex
# is calculated wrt to these fragments.
#
```

(continues on next page)

(continued from previous page)

```
# Note that for the calculation of the Pauli repulsion
# term the MP2 part of the double hybrid is not included.
#
# Next the BSSE is calculated for Water and MeOH.
# If one uses a TZ2P instead of a DZP basis set,
# the BSSE corrections will be smaller.

AMS_JOBNAME=Water $AMSBIN/ams <<eor
System
  atoms
     O     -0.525330     -0.050971     -0.314517
     H     -0.942007      0.747902      0.011253
     H      0.403697      0.059786     -0.073568
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type DZP
  end
  xc
    dispersion GRIMME3 BJDAMP
    doublehybrid B2GPPLYP
  end
EndEngine

eor

AMS_JOBNAME=MeOH $AMSBIN/ams <<eor
System
  atoms
     O      2.316633      0.045501      0.071858
     H      2.684616     -0.526577      0.749387
     C      2.781638     -0.426129     -1.190301
     H      2.350821      0.224965     -1.943415
     H      3.867602     -0.375336     -1.264613
     H      2.453296     -1.445999     -1.389381
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type DZP
  end
  xc
    dispersion GRIMME3 BJDAMP
    doublehybrid B2GPPLYP
  end
EndEngine

eor
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=Water_MeOH $AMSBIN/ams <<eor
System
  atoms
     O     -0.525330     -0.050971     -0.314517 adf.f=Water
     H     -0.942007      0.747902      0.011253 adf.f=Water
     H      0.403697      0.059786     -0.073568 adf.f=Water
     O      2.316633      0.045501      0.071858 adf.f=MeOH
     H      2.684616     -0.526577      0.749387 adf.f=MeOH
     C      2.781638     -0.426129     -1.190301 adf.f=MeOH
     H      2.350821      0.224965     -1.943415 adf.f=MeOH
     H      3.867602     -0.375336     -1.264613 adf.f=MeOH
     H      2.453296     -1.445999     -1.389381 adf.f=MeOH
  end
end

Task SinglePoint

Engine ADF
  fragments
     Water Water.results/adf.rkf
     MeOH MeOH.results/adf.rkf
  end
  xc
    dispersion GRIMME3 BJDAMP
    doublehybrid B2GPPLYP
  end
EndEngine

eor

# ==============
# BSSE for Water
# ==============

AMS_JOBNAME=Water_BSSE $AMSBIN/ams <<eor
System
  atoms
     O     -0.525330     -0.050971     -0.314517 adf.f=Water
     H     -0.942007      0.747902      0.011253 adf.f=Water
     H      0.403697      0.059786     -0.073568 adf.f=Water
     Gh.O   2.316633      0.045501      0.071858
     Gh.H   2.684616     -0.526577      0.749387
     Gh.C   2.781638     -0.426129     -1.190301
     Gh.H   2.350821      0.224965     -1.943415
     Gh.H   3.867602     -0.375336     -1.264613
     Gh.H   2.453296     -1.445999     -1.389381
  end
end

Task SinglePoint

Engine ADF
  fragments
     Water Water.results/adf.rkf
  end
  basis
```

(continues on next page)

(continued from previous page)

```
    core None
    type DZP
  end
  xc
    dispersion GRIMME3 BJDAMP
    doublehybrid B2GPPLYP
  end
EndEngine

eor

# =============
# BSSE for MeOH
# =============

AMS_JOBNAME=MeOH_BSSE $AMSBIN/ams <<eor
System
  atoms
     Gh.O  -0.525330     -0.050971     -0.314517
     Gh.H  -0.942007      0.747902      0.011253
     Gh.H   0.403697      0.059786     -0.073568
     O      2.316633      0.045501      0.071858 adf.f=MeOH
     H      2.684616     -0.526577      0.749387 adf.f=MeOH
     C      2.781638     -0.426129     -1.190301 adf.f=MeOH
     H      2.350821      0.224965     -1.943415 adf.f=MeOH
     H      3.867602     -0.375336     -1.264613 adf.f=MeOH
     H      2.453296     -1.445999     -1.389381 adf.f=MeOH
  end
end

Task SinglePoint

Engine ADF
  fragments
     MeOH MeOH.results/adf.rkf
  end
  basis
    core None
    type DZP
  end
  xc
    dispersion GRIMME3 BJDAMP
    doublehybrid B2GPPLYP
  end
EndEngine

eor
```

Example: Water-MeOH - DODSCAN

Download Water_MeOH_DODSCAN.run

```
#! /bin/sh

# Calculation on the bonding energy of the Water-methanol dimer using
# the meta-hybrid based, dispersion corrected double-hybrid functional DOD-SCAN.

# the AO-based algorithm is employed here by default, 8 numerical quadrature points
# are used for the numerical Laplace transform (as opposed to 9, which is the default).
# For the MP2 calculation only, the Normal fitset is used, while during the SCF
# default numerical settings are used together with the VeryGood fitset.


$AMSBIN/ams <<eor
System
  atoms
     O     -0.525330     -0.050971     -0.314517
     H     -0.942007      0.747902      0.011253
     H      0.403697      0.059786     -0.073568
     O      2.316633      0.045501      0.071858
     H      2.684616     -0.526577      0.749387
     C      2.781638     -0.426129     -1.190301
     H      2.350821      0.224965     -1.943415
     H      3.867602     -0.375336     -1.264613
     H      2.453296     -1.445999     -1.389381
  End
End

Task SinglePoint

Engine ADF
  basis
    core None
    type DZP
  End

  XC
    DoubleHybrid DODSCAN
  End

  MBPT
    nTime 8
    FitSetQuality Normal
  END
  RIHartreeFock
    FitSetQuality VeryGood
  End
EndEngine

eor
```

Example: unrestricted MP2: Li

Download MP2_Li.run

```
#! /bin/sh

# unrestricted MP2 calculation for Li

$AMSBIN/ams << eor
System
  Atoms
     Li      0.0000000000      0.0000000000      0.0000000000
  End
End

task SinglePoint

Engine adf
  SpinPolarization 1
  Unrestricted Yes
  basis
     Core None
     Type DZP
  End
  xc
    mp2
  End
EndEngine
eor
```

Example: RPA@PBE calculation: H2O

Download RPA_H2O.run

```
#! /bin/sh

# RPA@PBE calculation for Water.
# We use an all-electron basis set since core-correlation effects are important

# For H2O, PBE will be a suitable reference since it is only weakly correlated and␣
↪the
# KS HOMO-LUMO gap will be reasonably large.

# RECOMMENDED: Good numerical quality

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
End

System
  Atoms
    O    2.220871067   0.026716792   0.000620476
    H    2.597492682  -0.411663274   0.766744858
    H    2.593135384  -0.449496183  -0.744782026
  End
```

(continues on next page)

(continued from previous page)

```
  Symmetrize Yes
End

task SinglePoint

Engine adf

  Basis
    Core None
    Type TZ2P
  end

  NumericalQuality GOOD
  RIHartreeFock fitsetQuality=Normal

  Relativity
    Level None
  End

  XC
    GGA PBE
    RPA Direct
  end
EndEngine
eor
```

Example: RPASOX@PBE0 calculation: H2O

Download RPA-SOSSX_H2O.run

```
#! /bin/sh

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
End

System
  Atoms
    O    2.220871067   0.026716792   0.000620476
    H    2.597492682  -0.411663274   0.766744858
    H    2.593135384  -0.449496183  -0.744782026
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  symmetry nosym
  Basis
    Core None
    Type TZ2P
  end
  NumericalQuality GOOD
```

(continues on next page)

(continued from previous page)

```
  Relativity
    Level None
  End
  XC
    hybrid pbe0
    RPA SOSSX
  end
EndEngine
eor
```

Example: Grimme Molecular Mechanics dispersion-corrected functionals (DFT-D3-BJ)

Download MM_Dispersion.run

```
#! /bin/sh


# In this example a structure with 2 benzene molecules and a hydrogen molecule
# is optimized with the Grimme dispersion corrected PBE, in this case the (so
# far) latest variant the DFT-D family, which is DFT-D3-BJ. Needed is the subkey
# DISPERSION in the key XC, and the arguments Grime3 and BJDAMP. If one starts
# with atomic fragments the part of the bond energy that is due to the Grimme
# dispersion corrected functional is both inter-molecular as well as intra-
# molecular.


AMS_JOBNAME=Grimme_dispersion $AMSBIN/ams <<eor
System
  atoms
        C  0.0         1.39730802 -3.09
        H  0.0         2.48914064 -3.09
        C  1.21010424  0.69865401 -3.09
        H  2.15565903  1.24457032 -3.09
        C  1.21010424 -0.69865401 -3.09
        H  2.15565903 -1.24457032 -3.09
        C  0.0        -1.39730802 -3.09
        H  0.0        -2.48914064 -3.09
        C -1.21010424 -0.69865401 -3.09
        H -2.15565903 -1.24457032 -3.09
        C -1.21010424  0.69865401 -3.09
        H -2.15565903  1.24457032 -3.09
        C -1.21010424 -0.69865401  3.09
        H -2.15565903 -1.24457032  3.09
        C  0.0        -1.39730802  3.09
        H  0.0        -2.48914064  3.09
        C  1.21010424 -0.69865401  3.09
        H  2.15565903 -1.24457032  3.09
        C  1.21010424  0.69865401  3.09
        H  2.15565903  1.24457032  3.09
        C  0.0         1.39730802  3.09
        H  0.0         2.48914064  3.09
        C -1.21010424  0.69865401  3.09
        H -2.15565903  1.24457032  3.09
        H  0.0         0.0        -0.37
        H  0.0         0.0         0.37
```

(continues on next page)

(continued from previous page)

```
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    Gradients 1E-4
  End
End

Engine ADF
  title Geometry optimization with Grimme3-BJ dispersion correction for GGA
  basis
    type TZP
    core small
  end
  noprint sfo
  xc
    dispersion Grimme3 BJDAMP
    gga PBE
  end
EndEngine
eor


# The next calculations in this example demonstrate dispersion corrections when
# using non-atomic fragments. First three molecules (2 benzene molecules and a
# hydrogen molecule) are calculated . Needed again is the subkey DISPERSION in
# the key XC.


AMS_JOBNAME=benzene1 $AMSBIN/ams <<eor
System
  atoms
     C         0.000000    1.398973   -3.054539
     H         0.000000    2.490908   -3.049828
     C         1.211546    0.699486   -3.054539
     H         2.157190    1.245454   -3.049828
     C         1.211546   -0.699486   -3.054539
     H         2.157190   -1.245454   -3.049828
     C         0.000000   -1.398973   -3.054539
     H         0.000000   -2.490908   -3.049828
     C        -1.211546   -0.699486   -3.054539
     H        -2.157190   -1.245454   -3.049828
     C        -1.211546    0.699486   -3.054539
     H        -2.157190    1.245454   -3.049828
  end
end

Task SinglePoint

Engine ADF
  title Grimme3-BJ dispersion-corrected GGA
  basis
    core small
    type TZP
```

(continues on next page)

(continued from previous page)

```
  end
  noprint sfo
  scf
    converge 1.0E-06  1.0E-6
    iterations 60
  end
  xc
    dispersion Grimme3 BJDAMP
    gga PBE
  end
EndEngine
eor


AMS_JOBNAME=benzene2 $AMSBIN/ams <<eor
System
  atoms
     C        -1.211546   -0.699486    3.054539
     H        -2.157190   -1.245454    3.049828
     C         0.000000   -1.398973    3.054539
     H         0.000000   -2.490908    3.049828
     C         1.211546   -0.699486    3.054539
     H         2.157190   -1.245454    3.049828
     C         1.211546    0.699486    3.054539
     H         2.157190    1.245454    3.049828
     C         0.000000    1.398973    3.054539
     H         0.000000    2.490908    3.049828
     C        -1.211546    0.699486    3.054539
     H        -2.157190    1.245454    3.049828
  end
end

Task SinglePoint

Engine ADF
  title Grimme3-BJ dispersion-corrected GGA
  basis
    core small
    type TZP
  end
  noprint sfo
  xc
    dispersion Grimme3 BJDAMP
    gga PBE
  end
EndEngine
eor


AMS_JOBNAME=h2 $AMSBIN/ams <<eor
System
  atoms
     H         0.000000    0.000000   -0.377906
     H         0.000000    0.000000    0.377906
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title Grimme3-BJ dispersion-corrected GGA
  basis
    core small
    type TZP
  end
  noprint sfo
  xc
    dispersion Grimme3 BJDAMP
    gga PBE
  end
EndEngine
eor


AMS_JOBNAME=fragments $AMSBIN/ams <<eor
System
  atoms
     C         0.000000    1.398973   -3.054539  adf.f=b1
     H         0.000000    2.490908   -3.049828  adf.f=b1
     C         1.211546    0.699486   -3.054539  adf.f=b1
     H         2.157190    1.245454   -3.049828  adf.f=b1
     C         1.211546   -0.699486   -3.054539  adf.f=b1
     H         2.157190   -1.245454   -3.049828  adf.f=b1
     C         0.000000   -1.398973   -3.054539  adf.f=b1
     H         0.000000   -2.490908   -3.049828  adf.f=b1
     C        -1.211546   -0.699486   -3.054539  adf.f=b1
     H        -2.157190   -1.245454   -3.049828  adf.f=b1
     C        -1.211546    0.699486   -3.054539  adf.f=b1
     H        -2.157190    1.245454   -3.049828  adf.f=b1
     C        -1.211546   -0.699486    3.054539  adf.f=b2
     H        -2.157190   -1.245454    3.049828  adf.f=b2
     C         0.000000   -1.398973    3.054539  adf.f=b2
     H         0.000000   -2.490908    3.049828  adf.f=b2
     C         1.211546   -0.699486    3.054539  adf.f=b2
     H         2.157190   -1.245454    3.049828  adf.f=b2
     C         1.211546    0.699486    3.054539  adf.f=b2
     H         2.157190    1.245454    3.049828  adf.f=b2
     C         0.000000    1.398973    3.054539  adf.f=b2
     H         0.000000    2.490908    3.049828  adf.f=b2
     C        -1.211546    0.699486    3.054539  adf.f=b2
     H        -2.157190    1.245454    3.049828  adf.f=b2
     H         0.000000    0.000000   -0.377906  adf.f=h2
     H         0.000000    0.000000    0.377906  adf.f=h2
  end
end

Task SinglePoint

Engine ADF
  title Grimme3-BJ dispersion-corrected GGA
  fragments
     b1 benzene1.results/adf.rkf
     b2 benzene2.results/adf.rkf
     h2 h2.results/adf.rkf
```

(continues on next page)

(continued from previous page)

```
  end
  noprint sfo
  xc
    dispersion Grimme3 BJDAMP
    gga PBE
  end
EndEngine

eor
```

Example: Density-Dependent Dispersion Correction (dDsC): CH4-dimer

Download methane_dimer_dispersion.run

```
#! /bin/sh


# The density-dependent dispersion energy correction, dDsC, by S.N. Steinmann
# and C. Corminboeuf, is used to calculate interactions between non-overlapping
# densities, which standard density functional approximations cannot accurately
# describe. The example is for the methane dimer.

# The 'debug dispersion' is included such that in the output one can find more
# details on the exact parameters that are used.


$AMSBIN/ams <<eor
System
  atoms
     C  -0.000959    0.000775    1.853082
     H  -0.747186    0.712608    1.489389
     H   0.987865    0.294742    1.490258
     H  -0.241511   -0.998876    1.480724
     H  -0.002970   -0.005330    2.946903
     C   0.000962   -0.000776   -1.853082
     H   0.004264    0.004684   -2.946903
     H  -0.989749   -0.289559   -1.491241
     H   0.743184   -0.716138   -1.488107
     H   0.246099    0.997870   -1.481022
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    Gradients 1E-3
    Energy    0.1
    Step      0.1
  End
End

Engine ADF
  title methane-dimer
  debug dispersion
```

(continues on next page)

(continued from previous page)

```
  basis
    type TZP
  end
  numericalquality good
  symmetry nosym
  xc
    dispersion DDsC
    gga Becke LYP
  end
EndEngine
eor
```

Example: DFT-ulg Dispersion Correction: Benzene dimer T-shaped

Download Benzene_dimer_tshaped.run

```
#! /bin/sh


# The universal correction of density functional theory to include London
# dispersion (DFT-ulg) is used, which is related to UFF. This is yet another MM
# dispersion correction. The example is for the T-shaped benzene dimer.

$AMSBIN/ams <<eor
 Task SinglePoint
 System
   Atoms
      C  0.000000  0.000000  1.059035
      C  0.000000 -1.206008  1.757674
      C  0.000000 -1.207177  3.151591
      C  0.000000  0.000000  3.848575
      C  0.000000  1.207177  3.151591
      C  0.000000  1.206008  1.757674
      H  0.000000  0.000000 -0.021580
      H  0.000000 -2.141639  1.214422
      H  0.000000 -2.143566  3.692995
      H  0.000000  0.000000  4.930150
      H  0.000000  2.143566  3.692995
      H  0.000000  2.141639  1.214422
      C -1.394063  0.000000 -2.454152
      C -0.697047  1.207238 -2.454628
      C  0.697047  1.207238 -2.454628
      C  1.394063  0.000000 -2.454152
      C  0.697047 -1.207238 -2.454628
      C -0.697047 -1.207238 -2.454628
      H -2.475399  0.000000 -2.450322
      H -1.238232  2.143565 -2.453676
      H  1.238232  2.143565 -2.453676
      H  2.475399  0.000000 -2.450322
      H  1.238232 -2.143565 -2.453676
      H -1.238232 -2.143565 -2.453676
   end
 end
 Engine ADF
   Title Calculate the optimum structure of a dibenzene molecule and dispersion
```

(continues on next page)

(continued from previous page)

```
   basis
      type TZP
      CreateOutput Yes
   End
   XC
      GGA PBE
      dispersion uff
   End
 EndEngine
eor

$AMSBIN/densf <<eor
 adffile ams.results/adf.rkf
 Grid Medium
 End
 NCI BOTH
eor
```

14.2.2 Relativistic effects: ZORA, X2C, spin-orbit coupling

Example: ZORA Relativistic Effects: Au2

Download `Au2_ZORA.run`

```
#! /bin/sh


# A relativistic geometry optimization with the ZORA formalism. Both a scalar
# relativistic geometry optimization, as well as a spin-orbit coupled
# relativistic geometry optimization are performed. Spin-orbit effects on the
# geometry are often not so large for closed shell molecules, but takes
# typically a factor 4 times as much as a scalar relativistic calculation.

AMS_JOBNAME=Au2_GO_scalar $AMSBIN/ams <<eor
System
  atoms
     Au   0.0   0.0 -1.250
     Au   0.0   0.0  1.250
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1E-4
End

Engine ADF
  basis
    type TZ2P
    core Small
    CreateOutput Yes
  end
  relativity
```

(continues on next page)

(continued from previous page)

```
    level scalar
    formalism ZORA
  end
  xc
    gga Becke Perdew
  end
EndEngine

eor


AMS_JOBNAME=Au2_GO_spinorbit $AMSBIN/ams <<eor
System
  atoms
     Au  0.0  0.0 -1.250
     Au  0.0  0.0  1.250
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1E-4
End

Engine ADF
  basis
    type TZ2P
    core Small
    CreateOutput Yes
  end
  relativity
    level spin-orbit
    formalism ZORA
  end
EndEngine

eor
```

Example: Spin-Orbit coupling: Bi and Bi2

Download `SO_ZORA_Bi2.run`

```
#! /bin/sh


# Application of the Spin-Orbit relativistic option (using double-group
# symmetry) to Bismuth (atom and dimer).

# For comparison with the full double-group calculation, the 'standard'
# unrestricted calculation on Bismuth is carried out, using the scalar
# relativistic option.

# A net spin polarization of 3 electrons is applied (key charge).
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=Bi_SR $AMSBIN/ams <<eor
System
  atoms
     Bi   0.000000      0.000000      0.00000000
  end
end

Task SinglePoint

Engine ADF
  title Bi unrestricted
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 3
  unrestricted
  xc
    gga becke perdew
  end
EndEngine
eor


# The CHARGE key, in conjunction with the UNRESTRICTED key is used to specify
# that 3 electrons must be unpaired (second value of the CHARGE key), while the
# system is neutral (first value of the CHARGE key).

# Next we do a Spin-Orbit calculation on the Bismuth atom.

# Note that it is a 'unrestricted' run using the noncollinear approximation, and
# SYMMETRY NOSYM. The electronic charge density is spin-polarized.

AMS_JOBNAME=Bi_SO $AMSBIN/ams <<eor
System
  atoms
     Bi   0.000000      0.000000      0.00000000
  end
end

Task SinglePoint

Engine ADF
  title Bi spinorbit unrestricted noncollinear
  beckegrid
    quality good
  end
  basis
    core None
```

(continues on next page)

(continued from previous page)

```
    type TZ2P
    CreateOutput Yes
  end
  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization noncollinear
  end
  symmetry nosym
  unrestricted
  xc
    gga becke perdew
  end
EndEngine
eor


# Because an all electron basis set is used, the bond energy is huge, due to the
# very large higher order spin-orbit effect on the 2p orbitals.

# == Bi2 dimer ==

# Now we turn to the dimer Bi2: a series of Single Point calculations, all with
# the same inter atomic distance.

# First the scalar relativistic run.

AMS_JOBNAME=Bi2_SR $AMSBIN/ams <<eor
System
  atoms
     Bi       0.0              0.0              1.33
     Bi       0.0              0.0             -1.33
  end
end

Task SinglePoint

Engine ADF
  title Bi2, scalar relativistic
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
# The calculated scalar relativistic atomization energy will be close to 2.74
# eV. This is the bond energy of the dimer minus 2 times the bond energy of the
# unrestricted scalar relativistic atom.

# The result file tape21 is used as reference in subsequent calculations: run
# the spin-orbit case starting from the just completed dimer calculation as a
# fragment. The resulting 'bonding energy', ie the energy w.r.t. the scalar
# relativistic ZORA dimer, gives directly the effect of the full-relativistic
# ZORA versus the scalar relativistic ZORA option: the energy is lowered by huge
# amount, again mainly due to the large spin-orbit effect on the 2p orbitals.

AMS_JOBNAME=Bi2_SO_frag $AMSBIN/ams <<eor
System
  atoms
     Bi    0.0    0.0    1.33  adf.f=Bi2
     Bi    0.0    0.0   -1.33  adf.f=Bi2
  end
end

Task SinglePoint

Engine ADF
  title Bi2 from fragment Bi2,  with SpinOrbit coupling
  beckegrid
    quality good
  end
  fragments
     Bi2     Bi2_SR.results/adf.rkf
  end
  print SpinOrbit
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine
eor


# If one looks at the SFO analysis in the output of this calculation, one can
# observe that a first-order spin-orbit splitting of the scalar relativistic
# fragment orbitals is a good approximation to many of the calculated valence
# spinors.

# A final consistency check: run the spin-orbit dimer from single-atom
# fragments. The bonding energy should equal the sum of the bonding energies of
# the previous two runs: scalar relativistic dimer w.r.t. single atom fragments
# plus spin-orbit dimer w.r.t. the scalar relativistic dimer.

AMS_JOBNAME=Bi2_SO $AMSBIN/ams <<eor
System
  atoms
     Bi       0.0              0.0               1.33
     Bi       0.0              0.0              -1.33
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  title Bi2 from atomic fragments, SpinOrbit coupling
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  print SpinOrbit
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine
eor


# The calculated spin-orbit coupled relativistic atomization energy will be
# close to 2.18 eV. This is the bond energy of the dimer minus 2 times the bond
# energy of the unrestricted non-collinear spin-orbit coupled relativistic atom.
# Note that one has to subtract huge numbers.
```

Example: Spin-Orbit unrestricted non-collinear: Tl

Download Tl_noncollinear.run

```
#! /bin/sh


# Application of the Spin-Orbit relativistic option (using double-group
# symmetry, in this case NOSYM) to Tl using the collinear and non-collinear
# approximation for unrestricted Spin-Orbit calculations

# Note: For the collinear and the non-collinear approximation one should use
# symmetry NOSYM and use the key UNRESTRICTED.

# == Tl scalar restricted ==

AMS_JOBNAME=Scalar $AMSBIN/ams <<eor
System
  atoms
     Tl 0 0 0
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  title Tl scalar restricted
  beckegrid
    quality good
  end
  basis
    PerAtomType Symbol=Tl File=ZORA/TZ2P/Tl.4f
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga Becke Perdew
  end
EndEngine

eor

# == Tl spinorbit restricted ==

AMS_JOBNAME=SpinOrbit_Restricted $AMSBIN/ams <<eor
System
  atoms
     Tl 0 0 0
  end
end

Task SinglePoint

Engine ADF
  title Tl spinorbit restricted
  beckegrid
    quality good
  end
  eprint
    sfo eig
  end
  fragments
     Tl Scalar.results/adf.rkf
  end
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine

eor

# == Tl spinorbit collinear z-axis ==

AMS_JOBNAME=SpinOrbit_Collinear $AMSBIN/ams <<eor
```

(continues on next page)

(continued from previous page)

```
System
  atoms
     Tl 0 0 0
  end
end

Task SinglePoint

Engine ADF
  title Tl spinorbit collinear z-axis
  beckegrid
    quality good
  end
  eprint
    sfo eig
  end
  fragments
     Tl Scalar.results/adf.rkf
  end
  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization collinear
  end
  symmetry nosym
  unrestricted
  xc
    gga becke perdew
  end
EndEngine

eor

# == Tl spinorbit noncollinear ==

AMS_JOBNAME=SpinOrbit_NonCollinear $AMSBIN/ams <<eor
System
  atoms
     Tl 0 0 0
  end
end

Task SinglePoint

Engine ADF
  title Tl spinorbit noncollinear
  beckegrid
    quality good
  end
  eprint
    sfo eig
  end
  fragments
     Tl Scalar.results/adf.rkf
  end
  relativity
    level spin-orbit
```

(continues on next page)

(continued from previous page)

```
    formalism ZORA
    spinorbitmagnetization noncollinear
  end
  symmetry nosym
  unrestricted
  xc
    gga becke perdew
  end
EndEngine

eor

# == Tl spinorbit noncollinear ==

AMS_JOBNAME=SpinOrbit_SOUEXACT $AMSBIN/ams <<eor
System
  atoms
     Tl 0 0 0
  end
end

Task SinglePoint

Engine ADF
  title Tl spinorbit noncollinear
  beckegrid
    quality good
  end
  eprint
    sfo eig
  end
  fragments
     Tl Scalar.results/adf.rkf
  end
  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization noncollinear
    souexact
  end
  symmetry nosym
  unrestricted
  xc
    gga becke perdew
  end
EndEngine

eor
```

Example: Excitation energies including spin-orbit coupling: AuH

Download AuH_analyse_exciso.run

```
#! /bin/sh


# Calculation of the excitation energies of AuH including spin-orbit coupling.

# ADF can not handle ATOM and linear symmetries in excitation calculations.
# In case of linear symmetries ADF will automatically use a
# a subsymmetry, in this case symmetry C(7v).
# In the excitation part of the output the symmetry labels are those of C(LIN) in␣
↪this case.

# A relatively small TZ2P basis set is used, which is not sufficient for
# excitations to Rydberg-like orbitals, one needs more diffuse functions.

# The key STCONTRIB is used, which will give a composition of the spin-orbit
# coupled excitation in terms of singlet-singlet and singlet-triplet scalar
# relativistic excitations. In order to use the key STCONTRIB the scalar
# relativistic fragment should be the complete molecule.

# For precision reasons the Beckegrid quality is set to good. One might also
# increase the ZlmFit quality for better accuracy

# One needs to include the subkey SFO of the key EPRINT with arguments eig and
# ovl in order to get the SFO MO coefficients and SFO overlap matrix printed on
# standard output.


AMS_JOBNAME=scalar $AMSBIN/ams <<eor
System
  atoms
     Au 0.0000  0.0000  1.5238
     H  0.0000  0.0000  0.0000
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  eprint
    sfo eig ovl
  end
  excitations
    descriptors
    lowest 40
    nto
    sfoanalysis
    esestdm
  end
  print SOMATRIX
  sopert
  end
```

(continues on next page)

(continued from previous page)

```
  basis
    type TZ2P
    core None
    CreateOutput Yes
  end
EndEngine
eor



AMS_JOBNAME=spinorbit $AMSBIN/ams <<eor
System
  atoms
     Au 0.0000  0.0000  1.5238  adf.f=Frag
     H  0.0000  0.0000  0.0000  adf.f=Frag
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  eprint
    sfo eig ovl
  end
  excitations
    descriptors
    lowest 40
    nto
    sfoanalysis
    esestdm
  end
  fragments
     Frag scalar.results/adf.rkf
  end
  relativity
    level spin-orbit
  end
  stcontrib
EndEngine
eor
```

Example: ZORA, X2C and RA-X2C: HgI2 = Hg + I2

Download HgI2.run

```
#! /bin/sh


# The scalar relativistic ZORA, scalar relativistic X2C, and scalar relativistic
# RA-X2C are compared in this example for the bond dissociation energy of HgI2
# in Hg and I2. The division between scalar and spin-orbit relativistic effects
# is somewhat arbitrary and depends on the method of separation. Thus the scalar
```

(continues on next page)

(continued from previous page)

```
# relativistic results of X2C and RA-X2C will be different even in the basis set
# limit. However, the differences for bond energies are not so large. The
# results for scalar relativistic ZORA are often closer to scalar relativistic
# RA-X2C, than to scalar relativistic X2C.

for f in "ZORA" "X2C" "RA-X2C"
do

AMS_JOBNAME=HgI2_$f $AMSBIN/ams <<eor
System
  atoms
     Hg 0 0  0
     I  0 0  2.554
     I  0 0 -2.554
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type ZORA/DZ
  end
  numericalquality good
  relativity
    level scalar
    formalism $f
  end
  xc
    gga bp86
  end
EndEngine
eor

AMS_JOBNAME=I2_$f $AMSBIN/ams <<eor
System
  atoms
     I  0 0  1.333
     I  0 0 -1.333
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type ZORA/DZ
  end
  numericalquality good
  relativity
    level scalar
    formalism $f
  end
  xc
    gga bp86
```

(continues on next page)

(continued from previous page)

```
  end
EndEngine
eor

done
```

Example: spin-orbit coupled MP2: atomization energy I2

Download I2_SO_MP2.run

```
#!/bin/sh

# MP2 results are calculated for I2 and the open shell atom I, such
# that the atomization energy of I2 can be calculated.
# The relativistic RA-X2C method is used and spin-orbit coupling is included.

# Remark: the atomization energy will be almost the same if one would compare
# the results for the relativistic methods ZORA, X2C, and RA-X2C in case
# spin-orbit coupling is included.

AMS_JOBNAME=I2 $AMSBIN/ams <<eor
System
  atoms
     I   0 0  1.333
     I   0 0 -1.333
  end
end

Task SinglePoint

engine ADF
  Basis
    Type TZ2P
    Core None
  End
  relativity
    level spin-orbit
    formalism RA-X2C
  end
  xc
    mp2
  end
endengine
eor

AMS_JOBNAME=I_SO $AMSBIN/ams <<eor
System
  atoms
     I 0.0 0.0 0.0
  end
end

Task SinglePoint

engine ADF
```

(continues on next page)

(continued from previous page)

```
  Basis
    Type TZ2P
    Core None
  End
  relativity
    level spin-orbit
    formalism RA-X2C
    SpinOrbitMagnetization NonCollinear
  end
  unrestricted
  occupations integeraufbau
  symmetry nosym
  xc
    mp2
  end
endengine
eor
```

14.2.3 Solvents, other environments

Example: COSMO: HCl

Download Solv_HCl.run

```
#! /bin/sh


# Computing solvent effects, with the COSMO model, is illustrated in the HCl
# example.

# After a non-solvent (reference) calculation, which is omitted here, two
# solvent runs are presented, with somewhat different settings for a few input
# parameters. The block key Solvation controls all solvent-related input.

# All subkeys in the SOLVATION block are discussed in the User's Guide. Most of
# them are rather technical and should not severely affect the outcome.
# Physically relevant is the specification of the solute properties, by the
# SOLVENT subkey: the dielectric constant and the effective radius of the
# solvent molecule.

# Note that a non-electrostatic terms as a function of surface area is included
# in the COSMO calculation, by setting the values for CAV0 and CAV1 in the
# subkey SOLVENT of the key SOLVATION. In ADF2010 one should explicitly include
# such values for CAV0 and CAV1, otherwise this non-electrostatic term will be
# taken to be zero, since the defaults have changed in ADF2010.

# A rather strong impact on the computation times has the method of treating the
# 'C-matrix'. There are 3 options (see the User's Guide): EXACT is the most
# expensive, but presumably most accurate. POTENTIAL is the cheapest alternative
# and is usually quite adequate. EXACT uses the exact charge density for the
# Coulomb interaction between the molecular charge distribution and the point
# charges (on the Van der Waals type molecular surface) which model the effects
# of the solvent. The alternatives, notably 'POTENTIAL', use the fitted charge
# density instead. Assuming that the fit is a fairly accurate approximation to
```

(continues on next page)

(continued from previous page)

```
# the exact charge density, the difference in outcome should be marginal.


AMS_JOBNAME=Gas $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     H    0.000000     0.000000    0.000000
     Cl   1.304188     0.000000    0.000000
  end
end

Task SinglePoint

Engine ADF
  title HCl(0) reference run (gas phase)
  eprint
    scf NoEigvec
  end
  basis
    type DZP
    CreateOutput Yes
  end
  noprint Bas EigSFO EKin SFO, frag, functions
EndEngine
eor



AMS_JOBNAME=Exact $AMSBIN/ams <<eor
System
  atoms
     H    0.000000     0.000000    0.000000    adf.R=1.18
     Cl   1.304188     0.000000    0.000000    adf.R=1.75
  end
end

Task SinglePoint

Engine ADF
  title HCl(1) Solv-excl surface; Gauss-Seidel (old std options)
  eprint
    scf NoEigvec
  end
  noprint Bas EigSFO EKin SFO, frag, functions
  basis
    type DZP
  end
  solvation
    c-mat Exact
    charged Method=Gauss-Seidel
    disc SCale=0.01  LEGendre=10 TOLerance=1.0e-2
    div ND=4  min=0.5  Ofac=0.8
    scf Variational
    solv epsilon=78.8 radius=1.4 cav0=1.321 cav1=0.0067639
    surf delley
  end
```

(continues on next page)

(continued from previous page)

```
  symmetry NOSYM
EndEngine
eor


# In the second solvent run, another (technical) method is used for determining
# the charge distribution on the cavity surface (conjugate-gradient versus
# Gauss-Seidel in the previous calculation), and the POTENTIAL variety is used
# for the C-matrix handling. The results show that it makes little difference in
# outcome, but quite a bit in computation times.

AMS_JOBNAME=Potential $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     H    0.000000     0.000000    0.000000    adf.R=1.18
     Cl   1.304188     0.000000    0.000000    adf.R=1.75
  end
end

Task SinglePoint

Engine ADF
  title HCl(9) NoDisk and Cmatrix potential
  eprint
    scf NoEigvec
  end
  noprint Bas EigSFO EKin SFO, frag, functions
  basis
    type DZP
  end
  solvation
    c-mat POTENTIAL
    charged Method=conjugate-gradient
    div ND=4  min=0.5  Ofac=0.8
    scf Variational
    solv epsilon=78.8 radius=1.4 cav0=1.321 cav1=0.0067639
    surf delley
  end
EndEngine
eor
```

Example: solvation model SM12: Acetamide

Download Acetamide_SM12.run

```
#! /bin/sh


# In this example solvent effects are calculated with the solvation model SM12
# for Acetamide.

# After a non-solvent (reference gas phase) calculation, the same geometry is
# used in the solvated case. At the moment it is not possible to do a geometry
# optimization with the SM12 model in ADF. The default SM12 parameters are used,
```

(continues on next page)

(continued from previous page)

```
# which means a solvation in Water is simulated.


AMS_JOBNAME=Acetamide $AMSBIN/ams <<eor
System
  atoms
     O      -0.08557000       1.83373000       0.00836000
     C       0.09782000       0.61787000      -0.01303000
     C      -1.04418000      -0.38854000      -0.01026000
     N       1.35837000       0.07376000      -0.01671000
     H      -0.79705000      -1.31834000      -0.54107000
     H      -1.92168000       0.07841000      -0.46849000
     H      -1.29425000      -0.64220000       1.03061000
     H       1.51930000      -0.92568000      -0.01800000
     H       2.15296000       0.70225000       0.03378000
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type TZ2P
  end
  xc
    hybrid B3LYP
  end
EndEngine
eor


$AMSBIN/ams <<eor
System
  atoms
     O      -0.08557000       1.83373000       0.00836000 adf.f=Acetamide
     C       0.09782000       0.61787000      -0.01303000 adf.f=Acetamide
     C      -1.04418000      -0.38854000      -0.01026000 adf.f=Acetamide
     N       1.35837000       0.07376000      -0.01671000 adf.f=Acetamide
     H      -0.79705000      -1.31834000      -0.54107000 adf.f=Acetamide
     H      -1.92168000       0.07841000      -0.46849000 adf.f=Acetamide
     H      -1.29425000      -0.64220000       1.03061000 adf.f=Acetamide
     H       1.51930000      -0.92568000      -0.01800000 adf.f=Acetamide
     H       2.15296000       0.70225000       0.03378000 adf.f=Acetamide
  end
end

Task SinglePoint

Engine ADF
  fragments
     Acetamide Acetamide.results/adf.rkf
  end
  solvation SM12
  end
  xc
    hybrid B3LYP
```

(continues on next page)

(continued from previous page)

```
  end
EndEngine
eor
```

Example: Electric Field: PtCO

Download Field_PtCO.run

```
#! /bin/sh


$AMSBIN/ams <<eor
System
  Atoms
     Pt  0  0  0
     C   0  0  1.70
     O   0  0  2.85
  End

  ElectrostaticEmbedding
    ElectricField 0  0  0.01  [a.u.]
  End
end

Task SinglePoint

Engine ADF
  title Pt-CO in electric field
  beckegrid
    quality good
  end
  basis
    Core None
    Type DZ
    PerAtomType Symbol=Pt File=DZ/Pt.4d
  end
  symmetry C(LIN)
  scf
    converge 1.0E-10
  end
EndEngine
eor
```

Example: Electric Field, Point Charge: N2

Download Efield.PntQ_N2.run

```
#! /bin/sh

# For N2, three calculations are provided: 1) a normal N2 run as a reference
# 2) with a homogeneous electric field, 3) with a point charge.

AMS_JOBNAME=reference $AMSBIN/ams <<eor
```

(continues on next page)

(continued from previous page)

```
System
  atoms
     N   0 0 -0.55
     N   0 0  0.55
  end
end

Task SinglePoint

Engine ADF
  title N2 reference for comparison with E-Field runs
  basis
    type DZP
    core Small
    CreateOutput Yes
  end
EndEngine

eor


AMS_JOBNAME=efield $AMSBIN/ams <<eor
System
  atoms
     N   0 0 -0.55
     N   0 0  0.55
  end
    ElectrostaticEmbedding
        ElectricField 0.0 0.0 0.01 [a.u.]
    End
end

Task SinglePoint

Engine ADF
  title N2 in a homogeneous electric field
  Symmetry NoSym
  basis
    type DZP
    core Small
  end
EndEngine

eor


AMS_JOBNAME=point_charge $AMSBIN/ams <<eor
System
  atoms
     N 0 0 -0.55
     N 0 0  0.55
  end
    ElectrostaticEmbedding
        MultipolePotential
            # Format of each entry in the Coordinates block: x y z q (length units:␣
↪angstrom)
            Coordinates
```

(continues on next page)

(continued from previous page)

```
                0 0 3.0 1.0
            End
        End
    End
end

Task SinglePoint

Engine ADF
  title N2 polarized by a point charge on the axis
  Symmetry NoSym
  basis
    type DZP
    core Small
  end
EndEngine

eor
```

Example: 3D-RISM: Glycine

Download 3DRISM-Glycine.run

```
#! /bin/sh


# == Expert option ==

# Computing solvent effects with the 3D-RISM model is illustrated on the glycine
# example.

# All subkeys in the RISM block are discussed in the User's Guide. The things to
# pay attention to here are adf.SigU and EpsUadf. parameters for each atom in the␣
↪ATOMS
# block, the solvent parameters in the SOLVENT sub-block and the FFT box
# parameters in the SOLUTE sub-block. Both adf.SigU and EpsUadf. values as well as the
# solvent parameters may be obtained from force field parameter lists.
# Parameters for some common solvents are available in the ADF User's Guide.

# One should take into account the following when choosing FFT box parameters in
# the SOLUTE block:

# - the box should be at least twice as large as your model in each dimension,
# - the number of grid points in each dimension must be a power of 2, and
# - accuracy of the results and the memory usage depend on the number of grid-
#   points

# In this example only the gradients are calculated, no geometry optimization is␣
↪performed.


# == Remarks for different solvent than water ==

# The default values for the RISM1D subblock key are only applicable if the
# solvent is water. You must change the FLUIDPARAM subkey of the subblock key
```

(continues on next page)

(continued from previous page)

```
# RISM1D when modeling a different solvent, at least the dielectric constant and
# the density. In that case you may also have to change other subkeys of the
# subblock key RISM1D. For a different solvent than water, of course, you also
# have to change the description of the subblock SOLVENT. You may have to change
# the boxsize.


$AMSBIN/ams <<eor
System
  atoms
     C  0.0                  0.0                0.0                   adf.SigU=3.50  ␣
 ↪ adf.EpsU=0.066
     O  1.208031058311462    0.0                0.0                   adf.SigU=2.96  ␣
 ↪ adf.EpsU=0.200
     O -0.741720974445343    1.118348836898804  0.0                   adf.SigU=2.96  ␣
 ↪ adf.EpsU=0.200
     C -0.8529478907585144  -1.228639006614685  0.0                   adf.SigU=3.50  ␣
 ↪ adf.EpsU=0.066
     N -0.1451586186885834  -2.467742681503296  0.0                   adf.SigU=3.25  ␣
 ↪ adf.EpsU=0.170
     H -0.09300804138183594  1.869869947433472  0.0                   adf.SigU=1.00  ␣
 ↪ adf.EpsU=0.046
     H -1.528575420379639   -1.167157530784607  0.8756611347198486    adf.SigU=1.00  ␣
 ↪ adf.EpsU=0.046
     H -1.528575420379639   -1.167157530784607 -0.8756611347198486    adf.SigU=1.00  ␣
 ↪ adf.EpsU=0.046
     H  0.4767249822616577  -2.513782262802124  0.8179888725280762    adf.SigU=1.00  ␣
 ↪ adf.EpsU=0.046
     H  0.4767249822616577  -2.513782262802124 -0.8179888725280762    adf.SigU=1.00  ␣
 ↪ adf.EpsU=0.046
  end
end

Task SinglePoint

Properties
  Gradients Yes
End

Engine ADF
  title 3D-RISM test
  basis
    core small
    type DZP
  end
  nobeckegrid
  Relativity Level=None
  rism glycine 1N
    RISM1D
    SUBEND
    SOLVENT1 water
      UNITS      uWeight=g/mol  ULJsize=A  ULJenergy=kcal/mol Ucoord=A Udens=1/A3
      Parameters Weight=18.015   nAtoms=2
      1    -0.8476   3.166    0.1554       0.000000  0.00000  0.000000
      2     0.4238   1.000    0.0460      -0.816490  0.00000  0.577359
                                           0.816490  0.00000  0.577359
      DenSpe=0.03333
```

(continues on next page)

(continued from previous page)

```
    SUBEND
    SOLUTE  CO
      BOXSIZE 32.0 32.0 32.0
      BOXGRID   64   64   64
    SUBEND
  end
  symmetry C(s)
  xc
    lda
  end
EndEngine
eor


# The densf utility can be used to convert the 3D-RISM grid data stored in the adf.
↪rkf file to
# the TAPE41 format suitable for visualization by AMSview. The following fields are␣
↪created
# for each solvent site: Huv (solvent total correlation function), Uuv (solvent␣
↪potential in RT),
# Guv (solvent pair distribution function), Cuv (solvent direct correlation),
# and PMF (potentials of mean force in kcal/mol)


$AMSBIN/densf << eor
adffile ams.results/adf.rkf
RISM
eor
```

Example: 3D-RISM: Entropy

Download 3DRISM-Entropy.run

```
#! /bin/sh


# == Expert option ==

# Computing entropy contributions from the solvent with the 3D-RISM model is␣
↪illustrated
# on a water molecule with a constant point charge electrostatic potential

# All subkeys in the RISM block are discussed in the User's Guide.
# See also the 3DRISM-Glycine example

# density, dielectric constant as well as all DOEPS keys need to be taken from␣
↪experiment!

$AMSBIN/ams <<eor
System
  ATOMS
    O         0.000000    0.000000   -0.079297  adf.SigU=3.1660 adf.EpsU=0.1554 adf.
↪ChgU=-0.8476
    H         0.000000    0.768138    0.520093  adf.SigU=1.0000 adf.EpsU=0.0560 adf.
↪ChgU=+0.4238
```

(continues on next page)

(continued from previous page)

```
    H         0.000000   -0.768138    0.520093  adf.SigU=1.0000 adf.EpsU=0.0560 adf.
↪ChgU=+0.4238
  End
End


Task SinglePoint

Properties
  Gradients Yes
End

Engine ADF
  title 3D-RISM Entropy test

  ZLMfit
    Quality verygood
  End

  basis
    core small
    type DZP
  end

  Relativity Level=None

  XC
    GGA PBE
  End

  RISM Water in Water
    RISM1D theory=DRISM closure=HNC
     FLUIDPARAM temper=298.15 DielConst=78.46 UTotDens=G/CM3 0.9971
     DODEPS alpha=0.257e-3 kappa=0.457e-9 dedtp=-0.360 dedpt=0.037e-6
    SUBEND

    SOLVENT1 water (SPC/E)
      UNITS      uWeight=g/mol  ULJsize=A  ULJenergy=kcal/mol Ucoord=A Udens=MOL
      Parameters Weight=18.015   nAtoms=2
       1    -0.847600   3.16600   0.1554     0.00000000    0.00000000    0.06461505
       2    +0.423800   1.00000   0.0560     0.81649029    0.00000000   -0.51274392
                                          -0.81649029    0.00000000   -0.51274392
        DenSpe=1.00000
    SUBEND

    SOLUTE  Water
      BOXSIZE 32.0 32.0 32.0
      BOXGRID   64   64   64
      CHRGLVL=MM DERIV
    SUBEND
  end
EndEngine
eor
```

14.2.4 FDE: Frozen Density Embedding

Example: FDE: H2O in water

Download FDE_H2O_128.run

```
#! /bin/sh

# This example demonstrates how to use FDE in combination with a large
# environment, that is modeled as a superposition of the densities of isolated
# molecules. Here, the excitation energies of a water molecule surrounded by an
# environment of 127 water molecules. For details, see also: C.R. Jacob, J.
# Neugebauer, L. Jensen and L. Visscher, Comparison of frozen-density embedding
# and discrete reaction field solvent models for molecular properties, Physical
# Chemistry Chemical Physics 8, 2349 (2006)

# This calculation consists of two steps:
# - First a prototype water molecule is calculated.
# - Next the embedding calculation of water in water is performed.

# To reduce the amount of output the next lines are included in the adf
# calculations:
#   EPRINT
#     SFO NOEIG NOOVL NOORBPOP
#     SCF NOPOP
#   END
#   NOPRINT BAS FUNCTIONS


# create atomic fragment files


AMS_JOBNAME=O $AMSBIN/ams <<eor
System
  Atoms
    O 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create O file=$AMSRESOURCES/ADF/DZP/O
  Relativity Level=None
EndEngine
eor
mv O.results/adf.rkf t21.DZP.O

AMS_JOBNAME=H $AMSBIN/ams <<eor
System
  Atoms
    H 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create H file=$AMSRESOURCES/ADF/DZP/H
  Relativity Level=None
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor
mv H.results/adf.rkf t21.DZP.H


# First, a prototype water molecule is calculated. The density of this isolated
# water molecules will afterwards be used to model the environment.


AMS_JOBNAME=H2O $AMSBIN/ams <<eor
System
  atoms [bohr]
     O       -11.38048700000000    -11.81055300000000     -4.51522600000000
     H       -13.10476265095705    -11.83766918322447     -3.96954531282721
     H       -10.51089289290947    -12.85330720999229     -3.32020577897331
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fragments
     O  t21.DZP.O
     H  t21.DZP.H
  end
  noprint BAS FUNCTIONS
  numericalquality good
  scf
    converge 1.0e-6 1.0e-6
    diis
      cx 5.0
      cxx 10.0
      cyc 5
      n 10
      ok 0.5
    end
    iterations 50
    lshift 0.0
    mixing 0.2
  end
  symmetry NOSYM
  title Input generated by modco
  xc
    lda
  end
  Relativity Level=None
EndEngine
eor

mv H2O.results/adf.rkf t21.mol_1


# Afterwards, the FDE calculation is performed. In this FDE calculation, there
# is one nonfrozen water molecule and the previously prepared water molecule is
```

(continues on next page)

(continued from previous page)

```
# included as a frozen fragment that is duplicated 127 times. For this frozen
# fragment, the more efficient fitted density is used.



AMS_JOBNAME=FDE $AMSBIN/ams <<eor
System
  atoms [Bohr]
     O          0.00000000000000       0.00000000000000       0.00000000000000
     H         -1.43014300000000       0.00000000000000       1.10739300000000
     H          1.43014300000000       0.00000000000000       1.10739300000000
     O        -11.38048700000000     -11.81055300000000      -4.51522600000000    adf.
↪f=frag1|1
     H        -13.10476265095705     -11.83766918322447      -3.96954531282721    adf.
↪f=frag1|1
     H        -10.51089289290947     -12.85330720999229      -3.32020577897331    adf.
↪f=frag1|1
     O         -1.11635000000000       9.11918600000000      -3.23094800000000    adf.
↪f=frag1|2
     H         -2.82271357869859       9.71703285239153      -3.18063201242303    adf.
↪f=frag1|2
     H         -0.12378551814273      10.53819303003839      -2.70860866559857    adf.
↪f=frag1|2
     O        -16.96901200000000      -3.65835300000000     -10.78571300000000    adf.
↪f=frag1|3
     H        -15.70823656958949      -4.23996689198245      -9.62648920152293    adf.
↪f=frag1|3
     H        -16.47814038897754      -4.36080868684451     -12.37857352487904    adf.
↪f=frag1|3
     O        -10.76560800000000      -4.16357700000000       0.51803200000000    adf.
↪f=frag1|4
     H         -9.16115845895954      -3.98623150326079      -0.29800688710696    adf.
↪f=frag1|4
     H        -11.97483852095230      -3.79636551545084      -0.77600078745886    adf.
↪f=frag1|4
     O         -6.55423300000000      -9.20297500000000     -11.23125400000000    adf.
↪f=frag1|5
     H         -8.15301944239612      -9.65354271558258     -10.51536304076499    adf.
↪f=frag1|5
     H         -6.90715220963067      -7.74998343249339     -12.24902957370683    adf.
↪f=frag1|5
     O        -14.16713600000000      -6.55813200000000      -6.62957400000000    adf.
↪f=frag1|6
     H        -13.74689800192609      -8.31204468643143      -6.76675830514850    adf.
↪f=frag1|6
     H        -15.43294077464585      -6.49537593456625      -5.33906885435316    adf.
↪f=frag1|6
     O         -1.01233800000000      -6.53737500000000     -20.63228300000000    adf.
↪f=frag1|7
     H          0.51358895613388      -7.01674081354516     -19.78766204266023    adf.
↪f=frag1|7
     H         -2.25846754637793      -6.48721751306148     -19.32222614537351    adf.
↪f=frag1|7
     O         -7.97372300000000       4.33381600000000       8.85869700000000    adf.
↪f=frag1|8
     H         -7.39209897744333       2.71703281820261       9.42381190837837    adf.
↪f=frag1|8
```

(continues on next page)

(continued from previous page)

```
    H        -7.34133495205587      4.47772485325760      7.17021271491547   adf.
↪f=frag1|8
    O         1.68768000000000    -11.29316300000000     -2.87636700000000   adf.
↪f=frag1|9
    H         1.52000307205134    -11.65999321781241     -4.63958827351129   adf.
↪f=frag1|9
    H         0.92563970293820     -9.66478590395236     -2.67815953059656   adf.
↪f=frag1|9
    O         1.09184600000000     -1.69811700000000    -18.81717100000000   adf.
↪f=frag1|10
    H         2.07215988260319     -2.34326255514507    -20.19354345122146   adf.
↪f=frag1|10
    H         0.13398919688371     -3.11903886392137    -18.23829710737747   adf.
↪f=frag1|10
    O       -13.85258700000000     -1.51317200000000     -3.42847700000000   adf.
↪f=frag1|11
    H       -13.34561409429037     -1.90947670800034     -5.11890601995987   adf.
↪f=frag1|11
    H       -15.31887324674441     -2.53345289474331     -3.14451376182312   adf.
↪f=frag1|11
    O         5.28960400000000     -6.81266800000000    -13.10585800000000   adf.
↪f=frag1|12
    H         7.01709448187639     -6.62178372023976    -12.60488989024026   adf.
↪f=frag1|12
    H         4.34671789396407     -5.97791594280375    -11.80749041866683   adf.
↪f=frag1|12
    O       -11.47407200000000    -10.90708300000000     -9.44085000000000   adf.
↪f=frag1|13
    H       -11.21611173248622    -12.45441460630369    -10.34132244308043   adf.
↪f=frag1|13
    H       -11.17634020521638    -11.33593027595366     -7.70907363612158   adf.
↪f=frag1|13
    O        -1.64592100000000    -11.43523300000000      1.48098000000000   adf.
↪f=frag1|14
    H        -0.30557034252542    -10.47362907146570      0.73908565451392   adf.
↪f=frag1|14
    H        -1.72993170289473    -12.94068698496448      0.48190338473989   adf.
↪f=frag1|14
    O        -7.32468400000000     -5.01818100000000      4.38774900000000   adf.
↪f=frag1|15
    H        -8.81838077448469     -4.73028985072788      3.40916949260461   adf.
↪f=frag1|15
    H        -7.75923926090726     -6.38911120628163      5.48470646359115   adf.
↪f=frag1|15
    O         5.20879700000000     -6.83850600000000      3.69146100000000   adf.
↪f=frag1|16
    H         5.26579130261858     -6.68511818556767      1.89011343551333   adf.
↪f=frag1|16
    H         6.39162739624924     -5.60718291987245      4.28841938067655   adf.
↪f=frag1|16
    O         6.73749100000000     -0.26065600000000     -6.58850100000000   adf.
↪f=frag1|17
    H         6.56301678911955     -2.04681439624047     -6.81394756463717   adf.
↪f=frag1|17
    H         8.29925639010887      0.14828805200118     -7.40413401232653   adf.
↪f=frag1|17
    O         5.30624300000000     -5.54309500000000     -6.43599300000000   adf.
↪f=frag1|18
```

(continues on next page)

(continued from previous page)

```
    H         3.81279050995479    -6.09072594663625    -7.29699777727678    adf.
↪f=frag1|18
    H         6.65189674819999    -6.43527789528092    -7.25136362341575    adf.
↪f=frag1|18
    O        -5.72214300000000    -7.59376400000000    -5.60989200000000    adf.
↪f=frag1|19
    H        -5.63766272358672    -7.49959469188493    -7.41422690365166    adf.
↪f=frag1|19
    H        -3.99734274429106    -7.51425600722341    -5.07105423295232    adf.
↪f=frag1|19
    O        -4.96412100000000     4.32334900000000    -3.25899700000000    adf.
↪f=frag1|20
    H        -6.26920382212011     5.55951483151335    -3.05829302735551    adf.
↪f=frag1|20
    H        -3.43114054713299     5.22630232648312    -2.93303718111639    adf.
↪f=frag1|20
    O       -10.94232600000000     0.55437900000000   -13.18244900000000    adf.
↪f=frag1|21
    H        -9.99364526446256     2.08335246724114   -13.36649850073860    adf.
↪f=frag1|21
    H       -12.25798450895047     0.68645755108753   -14.41662950801929    adf.
↪f=frag1|21
    O        -9.29636700000000     5.39923600000000   -12.62463400000000    adf.
↪f=frag1|22
    H        -7.78716246923940     5.39969899801535   -11.62767327349474    adf.
↪f=frag1|22
    H        -8.71366631604628     5.57073538127912   -14.32835095341294    adf.
↪f=frag1|22
    O         2.97641300000000   -12.44506100000000     3.62605000000000    adf.
↪f=frag1|23
    H         3.66110125710447   -10.77090992621746     3.63307801290360    adf.
↪f=frag1|23
    H         1.22148596207982   -12.21500271544869     3.25334655288432    adf.
↪f=frag1|23
    O        -7.52033800000000    -2.70960600000000   -16.43185500000000    adf.
↪f=frag1|24
    H        -8.54144292463491    -2.01657808726233   -15.10947598274278    adf.
↪f=frag1|24
    H        -8.12619297191493    -1.91113986970532   -17.93750940057940    adf.
↪f=frag1|24
    O        -1.69701700000000    -2.04382200000000    -8.20675700000000    adf.
↪f=frag1|25
    H        -1.16935910005967    -0.43265303835196    -8.83710685355905    adf.
↪f=frag1|25
    H        -0.63663168422293    -2.33781858979711    -6.77122484449879    adf.
↪f=frag1|25
    O         5.30132200000000    -9.56047300000000     8.45223600000000    adf.
↪f=frag1|26
    H         5.27386189469879    -8.61762538446407     6.90889108173823    adf.
↪f=frag1|26
    H         4.55017557362690    -8.45544294990206     9.67137143822390    adf.
↪f=frag1|26
    O         5.99489300000000    -5.84398700000000    -1.27866800000000    adf.
↪f=frag1|27
    H         6.10499825242707    -5.94881824033588    -3.08103213209623    adf.
↪f=frag1|27
    H         7.02298278120587    -4.42064928461433    -0.84422038296580    adf.
↪f=frag1|27
```

(continues on next page)

(continued from previous page)

```
     O        -8.49933400000000      6.30730300000000      1.26206900000000     adf.
↪f=frag1|28
     H        -9.26414421853983      5.21634012956926      0.03875605345364     adf.
↪f=frag1|28
     H        -9.84131776823032      6.72028388514297      2.40230140460829     adf.
↪f=frag1|28
     O         1.42265800000000      7.15100500000000     -6.70245700000000     adf.
↪f=frag1|29
     H         0.49123529007929      7.74531228135872     -5.27036768330312     adf.
↪f=frag1|29
     H         2.87243479479048      6.33486591183760     -5.99276142000362     adf.
↪f=frag1|29
     O         1.16572300000000    -15.86327300000000     -8.03011700000000     adf.
↪f=frag1|30
     H        -0.43980652899404    -16.56646285956896     -8.47669081602252     adf.
↪f=frag1|30
     H         1.27023658519214    -14.33069688850776     -8.98504681689730     adf.
↪f=frag1|30
     O        -1.89194800000000    -10.63552100000000     -8.58469800000000     adf.
↪f=frag1|31
     H        -2.78816305171149    -10.71751973024788    -10.15367983850320     adf.
↪f=frag1|31
     H        -2.90779524204234    -11.59056143861734     -7.43250195607063     adf.
↪f=frag1|31
     O         1.30322100000000     -6.54666700000000     -9.29066800000000     adf.
↪f=frag1|32
     H         0.04144584195376     -7.80383112457906     -8.97588071525340     adf.
↪f=frag1|32
     H         0.43997569188912     -4.97291783823648     -9.06767765025571     adf.
↪f=frag1|32
     O        -4.10800400000000     -6.59387200000000    -16.43397700000000     adf.
↪f=frag1|33
     H        -4.90090013771901     -8.11414568563407    -17.00992218470793     adf.
↪f=frag1|33
     H        -5.44796070593591     -5.38061594011036    -16.36971704696261     adf.
↪f=frag1|33
     O        -5.19983400000000    -12.82687600000000     -4.96878500000000     adf.
↪f=frag1|34
     H        -6.70471369416354    -13.43327229448220     -5.76831602526536     adf.
↪f=frag1|34
     H        -5.42955949756766    -11.03545071340780     -4.87067679172327     adf.
↪f=frag1|34
     O        -4.89786700000000      5.50178800000000    -16.51771100000000     adf.
↪f=frag1|35
     H        -4.96675000682788      7.30587317882606    -16.40743398906903     adf.
↪f=frag1|35
     H        -3.22762417250174      5.07795982200470    -15.96792726530632     adf.
↪f=frag1|35
     O       -17.57898700000000     -4.99346700000000     -2.97681500000000     adf.
↪f=frag1|36
     H       -18.76387465651032     -3.84193458717524     -2.24085062321750     adf.
↪f=frag1|36
     H       -17.39866900954461     -6.30611916234463     -1.74554195748830     adf.
↪f=frag1|36
     O        -3.50776800000000    -17.32380200000000    -10.10442500000000     adf.
↪f=frag1|37
     H        -4.40193812130934    -18.63309464188228    -10.97496609167801     adf.
↪f=frag1|37
```

(continues on next page)

(continued from previous page)

```
    H       -4.53952158473348   -16.94953791159138    -8.66671344522861    adf.
↪f=frag1|37
    O      -10.81911600000000    -4.66640800000000   -11.14487100000000    adf.
↪f=frag1|38
    H      -11.62185710527301    -4.88296610749300    -9.53852778505075    adf.
↪f=frag1|38
    H      -10.60915967583231    -2.87735761828481   -11.30864113107593    adf.
↪f=frag1|38
    O       -6.39205500000000     0.01793500000000    -6.73248500000000    adf.
↪f=frag1|39
    H       -5.82125006722311     1.22625909145182    -5.51356587267281    adf.
↪f=frag1|39
    H       -4.87489255123650    -0.72801711445480    -7.37542911048570    adf.
↪f=frag1|39
    O      -11.30505500000000     0.46379700000000    -8.25065700000000    adf.
↪f=frag1|40
    H       -9.64226007733910     0.80674572776125    -7.62685540050746    adf.
↪f=frag1|40
    H      -11.19896085378185     0.79109534466454   -10.02638879504226    adf.
↪f=frag1|40
    O        9.00392400000000     1.26046800000000   -13.97805200000000    adf.
↪f=frag1|41
    H        8.87294935217592     1.02947962110145   -12.18888481086687    adf.
↪f=frag1|41
    H        7.29424845398649     1.45338312957695   -14.53608144577292    adf.
↪f=frag1|41
    O      -10.89041700000000     3.43422700000000    -2.58901500000000    adf.
↪f=frag1|42
    H      -11.04828427051772     1.96350979597573    -3.63002318964871    adf.
↪f=frag1|42
    H      -12.48348293054753     3.56267021936991    -1.74211071819919    adf.
↪f=frag1|42
    O       -5.77532000000000    11.85364600000000    -3.10184700000000    adf.
↪f=frag1|43
    H       -7.42098370181530    11.26896732951829    -2.63113365512790    adf.
↪f=frag1|43
    H       -5.59494422717391    13.43644886209393    -2.24522382582680    adf.
↪f=frag1|43
    O       -1.56797700000000     3.41013100000000    -9.46461500000000    adf.
↪f=frag1|44
    H       -0.62647981010343     4.69229425860768    -8.60364482634544    adf.
↪f=frag1|44
    H       -1.22975840621731     3.70013224540235   -11.21764410491830    adf.
↪f=frag1|44
    O       -5.85114500000000     5.94789900000000    -8.75297600000000    adf.
↪f=frag1|45
    H       -4.36435430864595     4.93739949116572    -8.95290949638818    adf.
↪f=frag1|45
    H       -5.23356351746774     7.64390869587085    -8.63572496098009    adf.
↪f=frag1|45
    O       -5.55413300000000    -7.92360000000000     0.25554800000000    adf.
↪f=frag1|46
    H       -4.38326585687764    -9.28844448621082     0.45022435153192    adf.
↪f=frag1|46
    H       -6.01769107687214    -7.47995835111906     1.94667176216252    adf.
↪f=frag1|46
    O      -10.80147900000000     4.92460700000000   -18.42825600000000    adf.
↪f=frag1|47
```

(continues on next page)

(continued from previous page)

```
     H        -9.46772203676955       4.11444445205056     -19.34278223208070      adf.
↪f=frag1|47
     H       -12.08285494971162       3.65684979597704     -18.27835293347214      adf.
↪f=frag1|47
     O        11.84765600000000       6.04652000000000      -5.60335700000000      adf.
↪f=frag1|48
     H        12.23822731048506       7.54282902057004      -4.66519785219186      adf.
↪f=frag1|48
     H        10.14515882225660       6.29374114330687      -6.16191644024222      adf.
↪f=frag1|48
     O         9.40177200000000      -2.16451100000000       0.06593400000000      adf.
↪f=frag1|49
     H         9.99192042697073      -0.60334687050214      -0.63128950443925      adf.
↪f=frag1|49
     H        10.90850263662718      -3.09972302317630       0.42194641911128      adf.
↪f=frag1|49
     O        -3.83766900000000      11.00009100000000      -8.55948700000000      adf.
↪f=frag1|50
     H        -3.74373964993300      10.83305937748681      -6.76090229682198      adf.
↪f=frag1|50
     H        -4.87440105609424      12.45821519867243      -8.82531410042130      adf.
↪f=frag1|50
     O        -8.92277200000000       7.51515500000000      -5.13005000000000      adf.
↪f=frag1|51
     H        -7.64240242456374       6.79630036125453      -6.18623699995818      adf.
↪f=frag1|51
     H        -9.98750814060132       6.11915279273326      -4.69517258001857      adf.
↪f=frag1|51
     O        -0.32879900000000       4.85643700000000     -14.28493900000000      adf.
↪f=frag1|52
     H        -0.20121901190276       6.61816864447670     -13.89551098390984      adf.
↪f=frag1|52
     H         1.10559652418602       4.51955612567933     -15.33404544784882      adf.
↪f=frag1|52
     O        -2.88119200000000     -21.16143900000000     -13.73773700000000      adf.
↪f=frag1|53
     H        -1.09646919425263     -20.88154858029820     -13.82748709656689      adf.
↪f=frag1|53
     H        -3.12787937393977     -22.76872743498338     -14.52980568926457      adf.
↪f=frag1|53
     O         4.08430100000000     -17.32350200000000       4.72137100000000      adf.
↪f=frag1|54
     H         4.02413151632988     -15.51826748869058       4.62578223161330      adf.
↪f=frag1|54
     H         2.36935484299032     -17.81891859379031       5.01313401917970      adf.
↪f=frag1|54
     O       -12.71856600000000       7.20558900000000       4.46079500000000      adf.
↪f=frag1|55
     H       -14.45999505747860       7.69416069947355       4.47955006523861      adf.
↪f=frag1|55
     H       -11.90975612966972       8.36888327939108       5.58515108034031      adf.
↪f=frag1|55
     O         3.64630900000000      -5.77729500000000      11.93616900000000      adf.
↪f=frag1|56
     H         4.69770672445712      -5.28069451760816      13.32165808553043      adf.
↪f=frag1|56
     H         1.98018929821852      -5.85215747794930      12.63620066423870      adf.
↪f=frag1|56
```

(continues on next page)

(continued from previous page)

```
    O        -11.58290200000000     -8.00977000000000    -14.71744100000000    adf.
↪f=frag1|57
    H        -11.15452934215944     -6.96177583707747    -13.30682612671804    adf.
↪f=frag1|57
    H        -13.31014912674778     -7.60719902330076    -15.07262764680259    adf.
↪f=frag1|57
    O        -22.69131900000000     -4.60206700000000    -11.33851100000000    adf.
↪f=frag1|58
    H        -22.22272486043639     -6.12263420792208    -12.19869942051499    adf.
↪f=frag1|58
    H        -21.13544211851963     -3.70468130859725    -11.12508740068061    adf.
↪f=frag1|58
    O          4.56872600000000     15.36672100000000     -5.27290200000000    adf.
↪f=frag1|59
    H          6.23209824653603     14.72493681821289     -5.57775451137379    adf.
↪f=frag1|59
    H          4.56888064677390     17.00621625499974     -6.03687769799284    adf.
↪f=frag1|59
    O        -19.42024500000000      0.10101200000000    -13.92810800000000    adf.
↪f=frag1|60
    H        -18.31405236374814     -1.07169409540655    -13.10790833801040    adf.
↪f=frag1|60
    H        -20.71321177209865      0.43336205711680    -12.70770480925147    adf.
↪f=frag1|60
    O          4.02837000000000     -8.18071600000000    -24.03669300000000    adf.
↪f=frag1|61
    H          5.65512609076238     -8.96585900909256    -24.13073712086871    adf.
↪f=frag1|61
    H          3.08016870759219     -9.23647651818391    -22.91513710771192    adf.
↪f=frag1|61
    O        -16.59841600000000    -12.07983100000000     -3.02725000000000    adf.
↪f=frag1|62
    H        -16.86792190502099    -10.47849662259938     -2.23055432463633    adf.
↪f=frag1|62
    H        -17.75928521418718    -12.09663441135308     -4.41422653012776    adf.
↪f=frag1|62
    O         -7.26966600000000     11.01317800000000    -13.83432500000000    adf.
↪f=frag1|63
    H         -8.50533384743054     11.78916627619676    -12.76540199703817    adf.
↪f=frag1|63
    H         -5.99921818801193     12.28193782381027    -14.05299015201462    adf.
↪f=frag1|63
    O         14.92607800000000      0.72989200000000     -0.88069200000000    adf.
↪f=frag1|64
    H         14.68468379195669      0.44423157042895      0.88898485624517    adf.
↪f=frag1|64
    H         13.43262120167169      1.60465761841764     -1.40602569068362    adf.
↪f=frag1|64
    O        -10.24993000000000    -15.51193700000000    -12.56703500000000    adf.
↪f=frag1|65
    H        -10.81569785539303    -16.80205979146874    -11.43252226892721    adf.
↪f=frag1|65
    H         -8.81118264459048    -16.23360868276287    -13.39214081855071    adf.
↪f=frag1|65
    O          6.15377400000000     -6.81034900000000    -18.37876400000000    adf.
↪f=frag1|66
    H          5.66882347580422     -7.60131711703521    -16.82608191819916    adf.
↪f=frag1|66
```

(continues on next page)

(continued from previous page)

```
    H         7.80239299356986     -6.14232206320171    -18.05104612139325     adf.
↪f=frag1|66
    O         3.76793000000000    -11.87319400000000    -12.00003600000000     adf.
↪f=frag1|67
    H         5.53017998300823    -12.15041382890993    -12.29878904864723     adf.
↪f=frag1|67
    H         3.52066686277755    -10.11552847274610    -12.34799636768586     adf.
↪f=frag1|67
    O         4.23865500000000      2.16969600000000    -16.37654900000000     adf.
↪f=frag1|68
    H         3.03592238765764      0.92692640056214    -16.90625382926868     adf.
↪f=frag1|68
    H         5.02860305993563      2.71732123822040    -17.90876707954152     adf.
↪f=frag1|68
    O       -14.68271000000000      0.72381400000000    -16.71344500000000     adf.
↪f=frag1|69
    H       -15.14615918038738     -0.55817879758533    -17.90228415832504     adf.
↪f=frag1|69
    H       -16.24034166467332      1.19193697588290    -15.92208663689043     adf.
↪f=frag1|69
    O        -8.56171900000000     -1.14046300000000    -21.11237400000000     adf.
↪f=frag1|70
    H        -7.24726382704255      0.10164744794682    -21.14397231666602     adf.
↪f=frag1|70
    H        -7.96722618972135     -2.45668030593841    -22.20129726831652     adf.
↪f=frag1|70
    O         3.09803900000000    -19.30437100000000     -4.72950400000000     adf.
↪f=frag1|71
    H         4.47592822929893    -18.53926720940844     -3.84199812304267     adf.
↪f=frag1|71
    H         2.31606384558652    -17.93639511851180     -5.61762585411266     adf.
↪f=frag1|71
    O       -11.48197100000000     -2.62773000000000      8.98577000000000     adf.
↪f=frag1|72
    H       -12.10181120734499     -1.96357292082485      7.42169752349214     adf.
↪f=frag1|72
    H       -12.53715834883977     -4.05824316030399      9.32015800703900     adf.
↪f=frag1|72
    O       -15.32683200000000      4.99378800000000      0.71520300000000     adf.
↪f=frag1|73
    H       -16.28231903578986      3.62492951609735      1.41155075487152     adf.
↪f=frag1|73
    H       -14.24707811603547      5.52532871717421      2.06546208560370     adf.
↪f=frag1|73
    O         4.69924600000000     -3.09331300000000    -22.21218100000000     adf.
↪f=frag1|74
    H         4.44004838777616     -3.02721933390031    -24.00105702850179     adf.
↪f=frag1|74
    H         4.90782404353437     -4.85303119199437    -21.84956458183150     adf.
↪f=frag1|74
    O         5.48125400000000    -14.87156900000000     -2.30008800000000     adf.
↪f=frag1|75
    H         6.04793062228805    -15.53700673074120     -0.71651626102270     adf.
↪f=frag1|75
    H         4.08123415431168    -13.81006617390298     -1.87025109596218     adf.
↪f=frag1|75
    O         7.71692700000000      7.49170500000000     -0.28146900000000     adf.
↪f=frag1|76
```

(continues on next page)

(continued from previous page)

```
    H         7.15009959892487      7.47463483734514      1.43610036604224    adf.
↪f=frag1|76
    H         6.70446411622425      6.23931957052779     -1.10490887468917    adf.
↪f=frag1|76
    O        -0.09971700000000     10.17858400000000    -12.50337600000000    adf.
↪f=frag1|77
    H         1.43320771620038     10.52645688573944    -11.60852657687974    adf.
↪f=frag1|77
    H        -1.38648439654237     10.22688417289634    -11.23313966659599    adf.
↪f=frag1|77
    O         0.17403100000000    -13.35840700000000    -17.21967600000000    adf.
↪f=frag1|78
    H        -1.53885665094696    -13.32910704018195    -16.63934279586859    adf.
↪f=frag1|78
    H         1.16036533522916    -13.51667655796879    -15.71179705645900    adf.
↪f=frag1|78
    O       -12.87620000000000     -6.35854400000000      5.05140600000000    adf.
↪f=frag1|79
    H       -11.96560105641672     -7.83983990822539      5.54960396913384    adf.
↪f=frag1|79
    H       -12.07232354985510     -5.82614364999607      3.52106744189462    adf.
↪f=frag1|79
    O        10.37496800000000      3.01364200000000     -1.65565900000000    adf.
↪f=frag1|80
    H        10.56060903951036      4.59152033506358     -0.79109750165770    adf.
↪f=frag1|80
    H        10.50573068374942      3.43968959450338     -3.40865305303111    adf.
↪f=frag1|80
    O       -10.54672300000000     11.48736100000000     -1.14189100000000    adf.
↪f=frag1|81
    H       -10.87795175377343      9.85135127695166     -1.83858958546889    adf.
↪f=frag1|81
    H       -11.17355773718848     12.64365945215504     -2.38352527948959    adf.
↪f=frag1|81
    O        16.77188300000000      5.46660900000000     -7.08505700000000    adf.
↪f=frag1|82
    H        15.05944999981585      5.87188178668200     -6.66677909180415    adf.
↪f=frag1|82
    H        17.62236540790539      5.39342341222859     -5.49040119833247    adf.
↪f=frag1|82
    O         3.75457600000000     11.31207900000000     -9.30406600000000    adf.
↪f=frag1|83
    H         3.86003631554596     12.75207030857270     -8.21460174023312    adf.
↪f=frag1|83
    H         3.06160535139532     10.00588129699877     -8.26231611492555    adf.
↪f=frag1|83
    O       -12.29225900000000      3.56799200000000     15.55877400000000    adf.
↪f=frag1|84
    H       -11.55132238527440      2.03988079459175     14.93626432369441    adf.
↪f=frag1|84
    H       -13.88471345817920      3.05255666747083     16.24435317034552    adf.
↪f=frag1|84
    O        -6.10108600000000    -22.91748700000000     -8.30037600000000    adf.
↪f=frag1|85
    H        -4.39703335260680    -23.51961699551631     -8.22775875933953    adf.
↪f=frag1|85
    H        -6.81851364400339    -23.76013175020010     -9.73105799918286    adf.
↪f=frag1|85
```

(continues on next page)

(continued from previous page)

```
     O      -16.47411100000000    -10.06716200000000    -10.99026800000000    adf.
↪f=frag1|86
     H      -16.57824909007337     -8.41061556718373    -11.70904362169910    adf.
↪f=frag1|86
     H      -14.81816261652035    -10.64777696598199    -11.42881650898706    adf.
↪f=frag1|86
     O      -19.79746500000000     -9.66763600000000     -6.29115500000000    adf.
↪f=frag1|87
     H      -19.30039040153150     -9.92412834081941     -8.01125899188181    adf.
↪f=frag1|87
     H      -20.69151106695596     -8.09537391433191     -6.30780271266956    adf.
↪f=frag1|87
     O       -5.17342500000000    -12.71280800000000    -15.39516900000000    adf.
↪f=frag1|88
     H       -6.29838943798551    -11.96053470711536    -16.59523946722675    adf.
↪f=frag1|88
     H       -5.88688186704289    -12.30005902553624    -13.78513137150979    adf.
↪f=frag1|88
     O       -0.98586800000000      2.69855900000000    -21.38983900000000    adf.
↪f=frag1|89
     H       -2.77204881462932      2.96779943896921    -21.48309619116062    adf.
↪f=frag1|89
     H       -0.80499110215822      1.20344697675802    -20.38807696838254    adf.
↪f=frag1|89
     O       -6.03694100000000      3.69212200000000    -20.94210000000000    adf.
↪f=frag1|90
     H       -6.14071267530565      5.16732160002454    -21.98356377744071    adf.
↪f=frag1|90
     H       -5.38953086150776      4.29598484602752    -19.36481963241442    adf.
↪f=frag1|90
     O        8.56921900000000     -3.79464900000000      6.33985500000000    adf.
↪f=frag1|91
     H        9.50635468663971     -2.25984004884893      6.53419338664101    adf.
↪f=frag1|91
     H        9.73243046417235     -5.09792887162821      6.80890110437856    adf.
↪f=frag1|91
     O        1.62911600000000     13.17414900000000     -1.60348400000000    adf.
↪f=frag1|92
     H        2.66594110342472     13.03446639023091     -0.12798012211089    adf.
↪f=frag1|92
     H        2.66565709768483     14.04258475743428     -2.80473562652369    adf.
↪f=frag1|92
     O       -1.79383000000000     17.03966900000000     -2.19674400000000    adf.
↪f=frag1|93
     H       -0.53200588641356     15.75727680306069     -2.00993024293071    adf.
↪f=frag1|93
     H       -1.42725302708876     17.80645974200545     -3.79338324012655    adf.
↪f=frag1|93
     O       -7.66877400000000    -18.37494100000000     -6.40129600000000    adf.
↪f=frag1|94
     H       -8.48426657117146    -18.61293045852318     -4.80443592340425    adf.
↪f=frag1|94
     H       -7.09966518295817    -20.02669116913405     -6.86974035671357    adf.
↪f=frag1|94
     O       10.40397100000000     -3.09718200000000    -22.16627000000000    adf.
↪f=frag1|95
     H        8.72972380889992     -2.49326115967796    -21.84408188422351    adf.
↪f=frag1|95
```

(continues on next page)

(continued from previous page)

```
    H      11.11812043738098     -1.87363862919898    -23.29078579938355     adf.
↪f=frag1|95
    O     -15.30392700000000      1.38087600000000      7.30094900000000     adf.
↪f=frag1|96
    H     -14.62180203945480      2.68484739436449      8.35261118963587     adf.
↪f=frag1|96
    H     -16.43220546537447      0.47060667386509      8.38261486369176     adf.
↪f=frag1|96
    O       8.58612500000000     -8.60056200000000     -9.05727500000000     adf.
↪f=frag1|97
    H       8.70489512259771     -9.81319946167069    -10.39407463523348     adf.
↪f=frag1|97
    H       8.62856020038021     -9.58970781613463     -7.54354008582316     adf.
↪f=frag1|97
    O      16.03912900000000      1.51573000000000    -10.24266300000000     adf.
↪f=frag1|98
    H      15.15629413827772      2.19658252062845    -11.66697745696204     adf.
↪f=frag1|98
    H      16.44244740848228      2.95789688018386     -9.22822101876796     adf.
↪f=frag1|98
    O     -15.02490500000000      4.61696500000000     -7.22941300000000     adf.
↪f=frag1|99
    H     -13.96280647503120      5.31092864978721     -8.51859506482347     adf.
↪f=frag1|99
    H     -14.28853929511416      3.00824288473906     -6.85340464006069     adf.
↪f=frag1|99
    O      -3.38051600000000    -15.91202400000000     -1.00061000000000     adf.
↪f=frag1|100
    H      -4.22410184376431    -15.04185487754888     -2.34329307606914     adf.
↪f=frag1|100
    H      -1.90382516434679    -16.61447063748981     -1.77362777442889     adf.
↪f=frag1|100
    O     -15.26048000000000      6.49459500000000    -20.09531600000000     adf.
↪f=frag1|101
    H     -13.52897204979343      6.24438887420643    -19.63607060457812     adf.
↪f=frag1|101
    H     -15.47768447407662      5.50940621247909    -21.59659174135803     adf.
↪f=frag1|101
    O     -18.03907100000000     -8.43103500000000      0.81066000000000     adf.
↪f=frag1|102
    H     -19.83364526460744     -8.64920488317246      0.75120975933977     adf.
↪f=frag1|102
    H     -17.60426054198021     -8.72510837516712      2.54157429534231     adf.
↪f=frag1|102
    O      -1.66133700000000    -24.85433900000000     -7.59562300000000     adf.
↪f=frag1|103
    H      -0.93912195987639    -26.25608107787563     -6.70953395077215     adf.
↪f=frag1|103
    H      -2.16378824805376    -25.53854670810930     -9.19280994513164     adf.
↪f=frag1|103
    O     -13.11411600000000      4.62406100000000     10.65633500000000     adf.
↪f=frag1|104
    H     -12.77074621481150      4.11089532103550     12.35644893641187     adf.
↪f=frag1|104
    H     -11.48603903104276      4.71705763077764      9.87382583582304     adf.
↪f=frag1|104
    O     -12.41558900000000     11.84804700000000     -6.99447300000000     adf.
↪f=frag1|105
```

(continues on next page)

(continued from previous page)

```
    H    -11.30168153595745     10.44978337527290     -6.71933990939709     adf.
↪f=frag1|105
    H    -12.47123642287374     12.03565896451106     -8.79261334416275     adf.
↪f=frag1|105
    O     -6.99029300000000     15.09813200000000     -9.14840700000000     adf.
↪f=frag1|106
    H     -8.45117315114006     15.70172502049625    -10.02768731357616     adf.
↪f=frag1|106
    H     -7.14889401470228     15.78429001622505     -7.48238284738053     adf.
↪f=frag1|106
    O    -12.34625400000000     10.56150500000000    -12.23873500000000     adf.
↪f=frag1|107
    H    -11.44958979728050      8.99509889893791    -12.35703315867646     adf.
↪f=frag1|107
    H    -12.49821882155289     11.12665028131793    -13.95020252415008     adf.
↪f=frag1|107
    O     11.40803400000000     -1.03651800000000     -9.30311700000000     adf.
↪f=frag1|108
    H     13.12420580954423     -0.46728539979694     -9.35167405119891     adf.
↪f=frag1|108
    H     11.31625950576437     -2.30813269496276    -10.58615133000387     adf.
↪f=frag1|108
    O     -5.11896700000000     -0.12253200000000     11.02394500000000     adf.
↪f=frag1|109
    H     -3.92681627947473      1.22947829217064     10.87403152613883     adf.
↪f=frag1|109
    H     -4.37657440580777     -1.48587544764282     10.09563930570322     adf.
↪f=frag1|109
    O     -9.99475100000000     -0.33650000000000     13.26488800000000     adf.
↪f=frag1|110
    H     -8.34486345702615      0.24573595608403     12.80608345859702     adf.
↪f=frag1|110
    H    -10.59609456116402     -1.19896273277279     11.79310420533769     adf.
↪f=frag1|110
    O     -2.30216800000000     10.69499300000000      2.99722800000000     adf.
↪f=frag1|111
    H     -1.93606116560115     10.01830345774807      1.36025326914961     adf.
↪f=frag1|111
    H     -4.04923519757226     11.15628108452666      2.91615338962990     adf.
↪f=frag1|111
    O     13.02295800000000     -5.27441800000000      1.23847500000000     adf.
↪f=frag1|112
    H     14.31033554297163     -4.63482922705865      0.14065238973499     adf.
↪f=frag1|112
    H     13.91152105495786     -5.78567177035616      2.72866923628817     adf.
↪f=frag1|112
    O      7.56659100000000      6.94211200000000     -9.52506900000000     adf.
↪f=frag1|113
    H      8.22168391057836      5.87923316520027    -10.83379780356249     adf.
↪f=frag1|113
    H      6.67225115956347      8.24497386381330    -10.40501831435555     adf.
↪f=frag1|113
    O     -2.17755000000000      7.03555800000000      6.40129900000000     adf.
↪f=frag1|114
    H     -2.96551413318437      5.65850851485069      5.53269295552737     adf.
↪f=frag1|114
    H     -2.21627359256819      8.40062348976667      5.21525504034563     adf.
↪f=frag1|114
```

(continues on next page)

(continued from previous page)

```
      O         9.00788300000000      12.60897400000000      -5.62029800000000     adf.
↪f=frag1|115
      H        10.26246081511818      11.86068291737544      -4.55365645678727     adf.
↪f=frag1|115
      H         7.97453521284851      11.22086747860001      -6.14656429973985     adf.
↪f=frag1|115
      O        -1.44785500000000      -3.72480700000000     -24.99109500000000     adf.
↪f=frag1|116
      H         0.23683814213660      -4.37145767071870     -25.11471193705296     adf.
↪f=frag1|116
      H        -1.92063360111663      -4.01890345598422     -23.27016756076567     adf.
↪f=frag1|116
      O        -7.64825200000000      12.61552300000000       3.06517700000000     adf.
↪f=frag1|117
      H        -7.81167293245439      14.38063407144895       3.42476805177055     adf.
↪f=frag1|117
      H        -8.83689943470846      12.31223844466806       1.73598952764555     adf.
↪f=frag1|117
      O         4.46933200000000      -0.50378800000000       2.21087700000000     adf.
↪f=frag1|118
      H         4.27357465941304      -2.08510082915893       3.06688986215400     adf.
↪f=frag1|118
      H         5.91907845739645      -0.73702803715971       1.15473407570514     adf.
↪f=frag1|118
      O        -1.24895400000000      -6.97750900000000      -2.93109400000000     adf.
↪f=frag1|119
      H        -2.37833578618925      -6.59579569242172      -1.57079221286929     adf.
↪f=frag1|119
      H        -0.21288275291882      -5.50308282339477      -3.08674554325879     adf.
↪f=frag1|119
      O        -5.92082900000000      -3.03107100000000      -2.51072800000000     adf.
↪f=frag1|120
      H        -5.78071219226800      -4.56089081895937      -3.46553050415439     adf.
↪f=frag1|120
      H        -6.21267678923423      -1.74791100910179      -3.75166671237991     adf.
↪f=frag1|120
      O         1.14317500000000      -2.76826300000000      -3.93086600000000     adf.
↪f=frag1|121
      H         2.94971664531870      -2.78102101161944      -3.84213191918490     adf.
↪f=frag1|121
      H         0.62850650734009      -1.73939210305163      -2.53510649267729     adf.
↪f=frag1|121
      O        -5.21546600000000      -0.71679400000000       1.97558900000000     adf.
↪f=frag1|122
      H        -5.88193806317475      -1.35755002215181       0.42095852001025     adf.
↪f=frag1|122
      H        -5.82594782496319      -1.86151827602300       3.23595476257813     adf.
↪f=frag1|122
      O        -5.42031600000000       3.66296900000000       4.47881900000000     adf.
↪f=frag1|123
      H        -6.47734991529745       4.75371897638942       3.49669222043088     adf.
↪f=frag1|123
      H        -5.28387721038046       2.15431400285173       3.49041213110813     adf.
↪f=frag1|123
      O         4.64978600000000       3.67542600000000      -3.78351700000000     adf.
↪f=frag1|124
      H         5.30464821970122       2.16136749204504      -4.52541424890080     adf.
↪f=frag1|124
```

(continues on next page)

(continued from previous page)

```
     H         3.40016650933066      3.10039840282737     -2.60903925511290    adf.
↪f=frag1|124
     O         1.38687600000000     -3.56100200000000      5.14710300000000    adf.
↪f=frag1|125
     H         2.35586583709350     -2.84555947523344      6.49648430622687    adf.
↪f=frag1|125
     H         2.25995224007779     -5.08706726956497      4.72228980076397    adf.
↪f=frag1|125
     O        -3.30635200000000     -3.51075400000000      7.52143300000000    adf.
↪f=frag1|126
     H        -1.82144215646564     -3.75776796826252      6.51860894948698    adf.
↪f=frag1|126
     H        -4.64977283415043     -4.18421132786886      6.51481526117547    adf.
↪f=frag1|126
     O         5.96480100000000      4.51370300000000      3.70332800000000    adf.
↪f=frag1|127
     H         5.24291272273548      3.06620845434369      2.89384293177905    adf.
↪f=frag1|127
     H         4.73614594944492      5.00201400735317      4.93765482424434    adf.
↪f=frag1|127
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  excitations
    lowest 5
    onlysing
  end
  fde
    pw91k
  end
  fragments
     O      t21.DZP.O
     H      t21.DZP.H
     frag1  t21.mol_1 type=fde &
     fdedenstype SCFfitted
     SubEnd
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  scf
    converge 1.0e-6 1.0e-6
    diis
      cx 5.0
      cxx 10.0
      cyc 5
      n 10
      ok 0.5
    end
    iterations 50
    lshift 0.0
```

(continues on next page)

(continued from previous page)

```
    mixing 0.2
  end
  symmetry NOSYM
  title Input generated by modco
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor
```

Example: FDE freeze-and-thaw: HeCO2

Download FDE_HeCO2_freezeandthaw.run

```
#!/bin/sh

# This example demonstrates how a freeze-and-thaw FDE calculation can be
# performed. As test system, a He-CO2 van der Waals complex is used. It will
# further be shown how different exchange-correlation potential can be used for
# different subsystems, and how different basis set expansions can be employed.
# For details, see C.R. Jacob, T.A. Wesolowski, L. Visscher, J. Chem. Phys. 123
# (2005), 174104. It should be stressed that the basis set and integration grid
# used in this example are too small to obtain good results.

# Summary:
# - PW91 everywhere
# - SAOP for He; PW91 for CO2
# - FDE(s) calculation with PW91 everywhere

# Important, this kind of FDE geometry optimization only works with the non-
# default STO pair fitting method. This means that one has to include the key
# STOFIT in the input for ADF. If one does not use the BASIS key, like in this
# example, one should also include the key STOFIT in all fragment calculations
# (also in the create runs).


# Create atom


AMS_JOBNAME=C $AMSBIN/ams <<eor
System
  Atoms
    C 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create C   $AMSRESOURCES/ADF/TZ2P/C
  stofit
  Relativity Level=None
EndEngine
eor
mv C.results/adf.rkf t21.C
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=O $AMSBIN/ams <<eor
System
  Atoms
    O 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create O   $AMSRESOURCES/ADF/TZ2P/O
  stofit
  Relativity Level=None
EndEngine
eor
mv O.results/adf.rkf t21.O

AMS_JOBNAME=He $AMSBIN/ams <<eor
System
  Atoms
    He 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create He    $AMSRESOURCES/ADF/TZ2P/He
  stofit
  Relativity Level=None
EndEngine
eor
mv He.results/adf.rkf t21.He


# == PW91 everywhere ==

# In the first part, the PW91 functional will be used for both the He and the
# CO2 subsystems. In this part, the FDE(m) basis set expansion is used, i.e.,
# basis functions of the frozen subsystem are not included in the calculation of
# the nonfrozen subsystem.

# First, the CO2 molecule is prepared. In this calculation, the C2v symmetry of
# the final complex is used, and the NOSYMFIT option has to be included because
# this molecule will be rotated as a frozen fragment.


############################
# Preparation of frozen CO2
############################

AMS_JOBNAME=CO2 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     C          0.000000  0.000000  0.000000
     O         -2.192000  0.000000  0.000000
     O          2.192000  0.000000  0.000000
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fragments
     C  t21.C
     O  t21.O
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  stofit
  symmetry C(2V)
  title TEST 1 -- Preparation of frozen CO2
  xc
    gga pw91
  end
  Relativity Level=None
EndEngine
eor

mv CO2.results/adf.rkf t21.co2.0


# Afterwards, the FDE calculation is performed. In this calculation, the He atom
# is the nonfrozen system, and the previously prepared CO2 molecule is used as
# frozen fragment. For this frozen fragment the RELAX option is specified, so
# that the density of this fragment is updated in freeze-and-thaw iteration (a
# maximum number of three iteration is specified).

########################
# Embedding calculation
########################


AMS_JOBNAME=FDE $AMSBIN/ams <<eor
System
  atoms [Bohr]
     He    0.000000  0.000000  6.019000 adf.f=He
     C     0.000000  0.000000  0.000000 adf.f=co2
     O    -2.192000  0.000000  0.000000 adf.f=co2
     O     2.192000  0.000000  0.000000 adf.f=co2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    fullgrid
    pw91k
```

(continues on next page)

(continued from previous page)

```
    relaxcycles 3
  end
  fragments
     He   t21.He
     co2  t21.co2.0  type=fde   &
     fdeoptions RELAX
     SubEnd
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  stofit
  title TEST 1 -- Embedding calculation: He with frozen CO2 density -- freeze-and-thaw
  xc
    gga pw91
  end
  Relativity Level=None
EndEngine
eor

# == SAOP for He; PW91 for CO2 ==

# In this second part, the above example is modified such that PW91 is employed
# for the CO2 subsystem, while the SAOP potential is used for He. This can be
# achieved by choosing SAOP in the XC key (this sets the functional that will be
# used for the nonfrozen subsystem). Additionally, for the frozen fragment the
# XC option is used to chose the PW91 functional for relaxing this fragment.
# Furthermore, the PW91 functional is chosen for the nonadditive exchange-
# correlation functional that is used in the embedding potential with the
# GGAPOTXFD and GGAPOTCFD options in the FDE key.


########################
# Embedding calculation
########################

AMS_JOBNAME=FDE1 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     He    0.000000   0.000000   6.019000 adf.f=He
     C     0.000000   0.000000   0.000000 adf.f=co2
     O    -2.192000   0.000000   0.000000 adf.f=co2
     O     2.192000   0.000000   0.000000 adf.f=co2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    fullgrid
    pw91k
    relaxcycles 3
```

(continues on next page)

(continued from previous page)

```
    xcnadd PW91
  end
  fragments
     He   t21.He
     co2  t21.co2.0  type=fde   &
     fdeoptions RELAX
     XC         GGA PW91
     SubEnd
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  stofit
  title TEST 2 -- Embedding calculation: He with frozen CO2 density -- freeze-and-thaw
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

rm t21.co2.0


# == FDE(s) calculation with PW91 everywhere ==

# In this third part, the PW91 functional is applied for both subsystems again,
# but in contrast to part 1, now the FDE(s) basis set expansion is used, i.e.,
# the basis functions of the frozen subsystem are included in the calculation of
# the nonfrozen subsystem. This can be achieved by employing the USEBASIS
# option. This option can be combined with the RELAX option.


############################
# Preparation of frozen CO2
############################


AMS_JOBNAME=CO2_PW91 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     C          0.000000  0.000000  0.000000
     O         -2.192000  0.000000  0.000000
     O          2.192000  0.000000  0.000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fragments
     C  t21.C
     O  t21.O
```

(continues on next page)

(continued from previous page)

```
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  stofit
  symmetry C(2V)
  title TEST 3 -- Preparation of frozen CO2
  xc
    gga pw91
  end
  Relativity Level=None
EndEngine
eor

mv CO2_PW91.results/adf.rkf t21.co2.0

########################
# Embedding calculation
########################

AMS_JOBNAME=FDE_PW91 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     He    0.000000  0.000000  6.019000 adf.f=He
     C     0.000000  0.000000  0.000000 adf.f=co2
     O    -2.192000  0.000000  0.000000 adf.f=co2
     O     2.192000  0.000000  0.000000 adf.f=co2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    fullgrid
    pw91k
    relaxcycles 3
  end
  fragments
     He   t21.He
     co2  t21.co2.0  type=fde   &
     fdeoptions RELAX USEBASIS
     SubEnd
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  stofit
  title TEST 3 -- Embedding calculation: He with frozen CO2 density -- freeze-and-thaw
  xc
    gga pw91
  end
  Relativity Level=None
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor
```

Example: FDE energy: NH3-H2O

Download FDE_Energy_NH3-H2O.run

```
#! /bin/sh

# This is example for a calculation of FDE interaction energies in ADF in case
# of closed shell fragments#

# It performs single point runs for H2O and NH3
# with LDA/DZ (all-electron) and uses these fragments in
# A) an FDE energy embedding calculation calculation in which the energy of
#    water in presence of a frozen ammonia is computed
#    This requires a supermolecular integration grid
# B) a fully variational FDE energy calculation (with freeze-and-thaw)
#
# NumericalQuality is good which should give total energies for the
# fragments accurate at least up to 10**(-4) atomic units


# NH3 LDA/DZ


AMS_JOBNAME=NH3 $AMSBIN/ams <<eor
System
  atoms
     N       -1.51248       -0.03714       -0.00081
     H       -1.71021        0.95994       -0.11003
     H       -1.96356       -0.53831       -0.76844
     H       -1.92899       -0.35123        0.87792
  end
end

Task SinglePoint

Engine ADF
  title H2O LDA/DZ single point
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZ
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  symmetrytolerance 1e-2
  Relativity
    Level None
```

(continues on next page)

(continued from previous page)

```
  End
EndEngine
eor

mv NH3.results/adf.rkf t21.ammonia


# H2O LDA/DZ


AMS_JOBNAME=H2O $AMSBIN/ams <<eor
System
  atoms
     O        1.45838        0.10183        0.00276
     H        0.48989       -0.04206        0.00012
     H        1.84938       -0.78409       -0.00279
  end
end

Task SinglePoint

Engine ADF
  title H2O LDA/DZ single point
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZ
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  symmetrytolerance 1e-2
  Relativity
    Level None
  End
EndEngine
eor
mv H2O.results/adf.rkf t21.water


# NH3-H2O: FDE energy of H2O in presence of frozen NH3


AMS_JOBNAME=FDE $AMSBIN/ams <<eor
System
  atoms
     O        1.45838        0.10183        0.00276    adf.f=frag1
     H        0.48989       -0.04206        0.00012    adf.f=frag1
     H        1.84938       -0.78409       -0.00279    adf.f=frag1
     N       -1.51248       -0.03714       -0.00081    adf.f=frag2
     H       -1.71021        0.95994       -0.11003    adf.f=frag2
     H       -1.96356       -0.53831       -0.76844    adf.f=frag2
     H       -1.92899       -0.35123        0.87792    adf.f=frag2
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  title NH3-H2O LDA/Thomas-Fermi/DZ FDE single point with interaction energy
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  fde
    energy
    fullgrid
    thomasfermi
  end
  fragments
     frag1  t21.water
     frag2  t21.ammonia type=FDE
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  symmetrytolerance 1e-2
  Relativity
    Level None
  End
EndEngine
eor


# NH3-H2O FDE interaction energy


AMS_JOBNAME=FDE1 $AMSBIN/ams <<eor
System
  atoms
     O        1.45838        0.10183        0.00276    adf.f=frag1
     H        0.48989       -0.04206        0.00012    adf.f=frag1
     H        1.84938       -0.78409       -0.00279    adf.f=frag1
     N       -1.51248       -0.03714       -0.00081    adf.f=frag2
     H       -1.71021        0.95994       -0.11003    adf.f=frag2
     H       -1.96356       -0.53831       -0.76844    adf.f=frag2
     H       -1.92899       -0.35123        0.87792    adf.f=frag2
  end
end

Task SinglePoint

Engine ADF
  title NH3-H2O LDA/Thomas-Fermi/DZ FDE single point with interaction energy
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  fde
```

(continues on next page)

(continued from previous page)

```
    energy
    relaxcycles 3
    thomasfermi
  end
  fragments
     frag1  t21.water
     frag2  t21.ammonia type=FDE  &
     fdeoptions RELAX
     SubEnd
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  symmetrytolerance 1e-2
  Relativity
    Level None
  End
EndEngine
eor
```

Example: FDE energy: unrestricted fragments: Ne-H2O

Download FDE_Energy_H2O-Ne_unrestricted.run

```
#! /bin/sh

# This is example for a calculation of FDE interaction energies in ADF for an
# open-shell frozen fragment.

# It performs single point runs for H2O and Ne, the latter unrestricted
# with LDA/DZ (all-electron) and uses these fragments in an FDE energy
# embedding calculation in which the energy of water in presence of a frozen
# (open-shell) neon atom is computed
# This is a bit of an artificial test but it serves its purpose
#
# No freeze-thaw is done, this is at present not possible with unrestricted
# (open shell) fragments, but has to be done manually, see the example in
# $AMSHOME/examples/Test/FDE_Spin.run
#
# NumericalQuality is good which should give total energies for the
# fragments accurate at least up to 10**(-4) atomic units
#
# This test has been checked to yield the same energy as a run with a closed-
# shell (restricted) Ne atom (just comment UNRESTRICTED in the input below)
#

# Ne LDA/DZ (unrestricted)

AMS_JOBNAME=Ne $AMSBIN/ams <<eor
System
  atoms
     Ne      -1.51248       -0.03714       -0.00081
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZ
  end
  noprint BAS FUNCTIONS
  nosymfit
  numericalquality Good
  scf
    converge 1.0e-06 1.0e-06
    iterations 100
  end
  spinpolarization 0
  symmetrytolerance 1e-2
  title Ne LDA/DZ single point, unrestricted
  unrestricted
  xc
    lda
  end
  Relativity Level=None
EndEngine
eor

mv Ne.results/adf.rkf t21.ne


# H2O LDA/DZ


AMS_JOBNAME=H2O $AMSBIN/ams <<eor
System
  atoms
     O        1.45838        0.10183        0.00276
     H        0.48989       -0.04206        0.00012
     H        1.84938       -0.78409       -0.00279
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZ
  end
  noprint BAS FUNCTIONS
```

(continues on next page)

(continued from previous page)

```
  nosymfit
  numericalquality Good
  spinpolarization 0
  symmetrytolerance 1e-2
  title H2O LDA/DZ single point
  xc
    lda
  end
  Relativity Level=None
EndEngine
eor
mv H2O.results/adf.rkf t21.water


# NH3-H2O: FDE energy of H2O in presence of frozen (unrestricted) Ne


AMS_JOBNAME=FDE $AMSBIN/ams <<eor
System
  atoms
     O        1.45838        0.10183        0.00276    adf.f=frag1
     H        0.48989       -0.04206        0.00012    adf.f=frag1
     H        1.84938       -0.78409       -0.00279    adf.f=frag1
     Ne      -1.51248       -0.03714       -0.00081    adf.f=frag2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  fde
    energy
    fullgrid
    thomasfermi
  end
  fragments
     frag1  t21.water
     frag2  t21.ne type=FDE
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  spinpolarization 0
  symmetrytolerance 1e-2
  title Ne-H2O LDA/Thomas-Fermi/DZ FDE single point with interaction energy
  xc
    lda
  end
  Relativity Level=None
EndEngine
eor
```

Example: FDE geometry optimization: H2O-Li(+)

Download GO_FDE_H2O-Li.run

```
#!/bin/sh

# This example demonstrates how to perform a geometry optimization of an active
# subsystem (a lithium atom) in the presence of a frozen FDE fragment (the water
# molecule).

# Notes:
#
# a) Only the geometry of the active fragment will be relaxed (in this case, the
#    Li atom). The geometry of the FDE fragment (the water molecule) will not
#    change during the optimization.
#
# b) When performing geometry optimization ICW FDE one must:
#    - Disable symmetry ('Symmetry NoSym')
#    - It is advised to use the FullGrid option ('FDE => FullGrid')



# Prepare the water molecule fragment:
# ====================================

AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
System
  atoms
     O        0.0000000    0.0000000   -1.53403200
     H       -0.7782160    0.0000000   -2.13596600
     H        0.7782160    0.0000000   -2.13596600
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type DZP
  end
  numericalquality Good
  symmetry NOSYM
  totalenergy
  xc
    lda
  end
  Relativity
    Level None
  End
EndEngine
eor

mv Iso2.results/adf.rkf t21.iso.rho2


# H2O-Li(+) Geometry optimization
# (Li optimization in the presence of a 'frozen' water molecule)
```

(continues on next page)

(continued from previous page)

```
# ===============================================================

AMS_JOBNAME=GO_FDE $AMSBIN/ams <<eor
System
  atoms
     LI         0.0000000    0.0000000    0.4529560
     O          0.0000000    0.0000000   -1.5340320  adf.f=rho2 region=frozen
     H         -0.7782160    0.0000000   -2.1359660  adf.f=rho2 region=frozen
     H          0.7782160    0.0000000   -2.1359660  adf.f=rho2 region=frozen
  end
  charge 1
end

Task GeometryOptimization

GeometryOptimization
   Convergence Step=1.0e-3
End

Properties
  Gradients Yes
End

Constraints
  Block frozen
End

Engine ADF
  fde
    fullgrid
    sdftenergy
    thomasfermi
  end
  fragments
     rho2 t21.iso.rho2 type=fde &
     SubEnd
  end
  basis
    core None
    type DZP
  end
  numericalquality Good
  symmetry NOSYM
  totalenergy
  xc
    lda
  end
  Relativity
    Level None
  End
EndEngine
eor

echo "Final Geometry"
echo ""
echo "Final Geometry after FDE optimization"
```

(continues on next page)

(continued from previous page)

```
$AMSBIN/amsreport GO_FDE.results/adf.rkf 'distances#labels'
```

Example: FDE NMR shielding: Acetonitrile in water

Download FDE_NMR_relax.run

```
#! /bin/sh


# This examples demonstrates both the calculation of NMR shieldings using FDE,
# and how the approximate environment density can be improved by partial
# relaxation of individual solvent molecules. The test system is a cluster of
# acetonitrile and 12 solvent water molecules, of which for two the densities
# are relaxed, while for the remaining 10 the frozen density of the isolated
# water is used. For details, see Refs. C. R. Jacob, J. Neugebauer, and L.
# Visscher, A flexible implementation of frozendensity embedding for use in
# multilevel simulation, submitted, 2007. R. E. Bulo, Ch. R. Jacob, and L.
# Visscher, NMR Solvent Shifts of Acetonitrile from Frozen-Density Embedding
# Calculation

# First, the isolated solvent water molecule is prepared. Again, because this
# will be rotated and translated afterwards, the option NOSYMFIT has to be
# included.


# create atomic fragment files

AMS_JOBNAME=O $AMSBIN/ams <<eor
System
  Atoms
    O 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create O file=$AMSRESOURCES/ADF/DZP/O
  Relativity Level=None
EndEngine
eor
mv O.results/adf.rkf t21.O.DZP

AMS_JOBNAME=H $AMSBIN/ams <<eor
System
  Atoms
    H 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create H file=$AMSRESOURCES/ADF/DZP/H
  Relativity Level=None
EndEngine
eor
mv H.results/adf.rkf t21.H.DZP
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=C $AMSBIN/ams <<eor
System
  Atoms
    C 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create C file=$AMSRESOURCES/ADF/DZP/C
  Relativity Level=None
EndEngine
eor
mv C.results/adf.rkf t21.C.DZP

AMS_JOBNAME=N $AMSBIN/ams <<eor
System
  Atoms
    N 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create N file=$AMSRESOURCES/ADF/DZP/N
  Relativity Level=None
EndEngine
eor
mv N.results/adf.rkf t21.N.DZP

#############################
# prepare H2O
#############################

AMS_JOBNAME=H2O $AMSBIN/ams <<eor
System
  atoms
     O       -1.46800        2.60500        1.37700
     H       -0.95200        3.29800        0.96500
     H       -1.16100        1.79900        0.96100
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fragments
     H        t21.H.DZP
     O        t21.O.DZP
  end
  noprint BAS FUNCTIONS
  numericalquality GOOD
  scf
    converge 1.0e-06 1.0e-06
    iterations 100
```

(continues on next page)

(continued from previous page)

```
  end
  symmetry NOSYM
  symmetrytolerance 1e-2
  title Input generated by PyADF and later modified
  xc
    lda
  end
  Relativity Level=None
EndEngine
eor

mv H2O.results/adf.rkf t21.h2o


# Afterwards, the FDE calculation is performed. In addition to the nonfrozen
# acetonitrile molecule, three different fragments are used for the solvent
# water molecules. The first two fragments frag1 and frag2 are relaxed (in up to
# two freeze-and-thaw cycles), while the third fragment is used for the
# remaining 10 solvent molecules. Since a calculation of the shielding is
# performed afterwards, the option has to be included.

######################################
# and the embedding calculation
######################################

AMS_JOBNAME=FDE $AMSBIN/ams <<eor
System
  atoms
     C        0.83000        0.66100       -0.44400
     N        0.00000        0.00000        0.00000
     C        1.87800        1.55900       -0.81900
     H        1.78500        2.40300       -0.13500
     H        1.76200        1.94900       -1.83000
     H        2.82900        1.12200       -0.51300
     O       -1.46800        2.60500        1.37700    adf.f=frag1|1
     H       -0.95200        3.29800        0.96500    adf.f=frag1|1
     H       -1.16100        1.79900        0.96100    adf.f=frag1|1
     O        2.40400       -2.51000       -0.36200    adf.f=frag2|1
     H        2.70000       -3.41900       -0.40900    adf.f=frag2|1
     H        1.77500       -2.50000        0.35900    adf.f=frag2|1
     O       -3.22800       -1.61500        1.18500    adf.f=frag3|1
     H       -3.33300       -2.55300        1.03000    adf.f=frag3|1
     H       -3.14200       -1.23600        0.31000    adf.f=frag3|1
     O        0.84000       -2.61200        2.89000    adf.f=frag3|2
     H        0.58800       -3.43700        3.30500    adf.f=frag3|2
     H        0.02500       -2.11500        2.82900    adf.f=frag3|2
     O        2.95400       -0.85100        2.99700    adf.f=frag3|3
     H        2.12000       -1.22400        2.71200    adf.f=frag3|3
     H        2.71800       -0.24100        3.69600    adf.f=frag3|3
     O        3.62200       -0.74000       -2.19300    adf.f=frag3|4
     H        3.05100       -1.25200       -1.62100    adf.f=frag3|4
     H        4.08100       -0.14200       -1.60200    adf.f=frag3|4
     O       -3.80000       -1.13100       -1.71100    adf.f=frag3|5
     H       -3.02600       -0.80900       -2.17400    adf.f=frag3|5
     H       -4.31600       -0.34500       -1.53300    adf.f=frag3|5
     O       -1.77100       -3.79600       -2.15500    adf.f=frag3|6
     H       -2.71500       -3.79000       -2.31700    adf.f=frag3|6
```

(continues on next page)

(continued from previous page)

```
     H       -1.65100       -3.19400       -1.42100    adf.f=frag3|6
     O        1.60000       -0.17800       -3.98800    adf.f=frag3|7
     H        2.40800       -0.18200       -3.47500    adf.f=frag3|7
     H        1.13900       -0.97100       -3.71300    adf.f=frag3|7
     O       -1.63900       -1.73400        3.28100    adf.f=frag3|8
     H       -1.97000       -1.69700        4.17900    adf.f=frag3|8
     H       -2.38200       -2.04200        2.76400    adf.f=frag3|8
     O        1.57900        2.85500        2.45800    adf.f=frag3|9
     H        0.92600        2.71500        3.14400    adf.f=frag3|9
     H        1.85200        3.76600        2.57000    adf.f=frag3|9
     O       -3.44400        2.36700        3.13700    adf.f=frag3|10
     H       -2.70200        2.29200        2.53700    adf.f=frag3|10
     H       -3.47300        3.29500        3.36800    adf.f=frag3|10
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
  end
  fragments
     H        t21.H.DZP
     C        t21.C.DZP
     N        t21.N.DZP
     frag1  t21.h2o   type=FDE  &
     fdeoptions RELAX
     RELAXCYCLES 2
     SubEnd
     frag2  t21.h2o   type=FDE  &
     fdeoptions RELAX
     RELAXCYCLES 2
     SubEnd
     frag3  t21.h2o   type=FDE  &
     FDEDENSTYPE SCFexact
     SubEnd
  end
  noprint BAS FUNCTIONS
  numericalquality GOOD
  save TAPE10
  scf
    converge 1.0e-07 1.0e-07
    iterations 100
  end
  symmetrytolerance 1e-2
  title Input generated by PyADF and later modified
  xc
    gga BP86
  end
  Relativity Level=None
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```


# Finally, the calculation of the NMR shielding of the nitrogen atom is
# performed using the NMR program.


$AMSBIN/nmr <<eor
 tape10file FDE.results/TAPE10
 adffile FDE.results/adf.rkf
 NMR
  out tens iso
  nuc 3
 END
eor
```

Example: FDE NMR spin-spin coupling: NH3-H2O

Download FDE_CPL_NH3-H2O.run

```
#! /bin/sh

# This example demonstrates calculation of NMR spin-spin couplings (J coupling)
# in case of frozen density embedding (FDE). It performs single point runs for
# H2O and NH3 with PBE/(ZORA/jcpl basis) (all-electron) and uses these fragments
# in:
# - an FDE embedding calculation calculation in which water in presence of a
#   frozen ammonia is computed (first the electron density / KS orbitals with
#   ADF then the NMR J couplings with CPL (scalar code branch))
# - a corresponding FDE freeze-thaw calculation followed by calculation of NMR J
#   couplings with CPL scalar, then spin-orbit code branch

# First NH3 with PBE and an all electron TZP basis set. No cpl calculation for
# NH3, thus regular basis is OK. This is not shown. Next H2O with PBE and an all
# electron ZORA/jcpl basis set:

# NH3 PBE/TZP (no cpl calculation for NH3, thus regular basis is OK)

AMS_JOBNAME=NH3 $AMSBIN/ams <<eor
System
  atoms
     N       -1.39559       -0.02156        0.00004
     H       -1.62981        0.96110       -0.10622
     H       -1.86277       -0.51254       -0.75597
     H       -1.83355       -0.33077        0.86231
  end
end

Task SinglePoint

Engine ADF
  title NH3 PBE/TZP single point
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
```

(continues on next page)

(continued from previous page)

```
  exactdensity
  basis
    core None
    type TZP
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  symmetry NOSYM
  xc
    gga PBE
  end
  Relativity Level=None
EndEngine
eor

mv NH3.results/adf.rkf t21.ammonia


# H2O PBE/(ZORA/jcpl)


AMS_JOBNAME=H2O $AMSBIN/ams <<eor
System
  atoms
     O        1.56850        0.10589        0.00001
     H        0.60674       -0.03396       -0.00063
     H        1.94052       -0.78000        0.00022
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type ZORA/jcpl
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  symmetry NOSYM
  title H2O PBE/(ZORA/jcpl) single point
  xc
    gga PBE
  end
  Relativity Level=None
EndEngine

eor
mv H2O.results/adf.rkf t21.water


# NH3-H2O: FDE calculation of H2O in presence of frozen NH3
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=FDE $AMSBIN/ams <<eor
System
  atoms
     O        1.56850        0.10589        0.00001    adf.f=frag1
     H        0.60674       -0.03396       -0.00063    adf.f=frag1
     H        1.94052       -0.78000        0.00022    adf.f=frag1
     N       -1.39559       -0.02156        0.00004    adf.f=frag2
     H       -1.62981        0.96110       -0.10622    adf.f=frag2
     H       -1.86277       -0.51254       -0.75597    adf.f=frag2
     H       -1.83355       -0.33077        0.86231    adf.f=frag2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  fde
    pw91k
  end
  fragments
     frag1  t21.water
     frag2  t21.ammonia type=FDE
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  save TAPE10
  symmetry NOSYM
  title NH3-H2O PBE/PW91k/(ZORA/jcpl) FDE single point calculation (no freeze-thaw)
  xc
    gga PBE
  end
  Relativity Level=None
EndEngine
eor


# NH3-H2O CPL calculation J(O-H) in water H-bond donor


"$AMSBIN/cpl" <<eor
 tape10file FDE.results/TAPE10
 adffile FDE.results/adf.rkf
 GGA
 NMRCoupling
  fc
  dso
  pso
  AtomPert 1
  AtomResp 2
  scf iterations=15 converge=1.0e-05
```

(continues on next page)

(continued from previous page)

```
 End
eor
rm TAPE21 TAPE10 logfile


# NH3-H2O FDE calculation (1 freeze-thaw cycle)


AMS_JOBNAME=FDE1 $AMSBIN/ams <<eor
System
  atoms
     O        1.56850        0.10589        0.00001    adf.f=frag1
     H        0.60674       -0.03396       -0.00063    adf.f=frag1
     H        1.94052       -0.78000        0.00022    adf.f=frag1
     N       -1.39559       -0.02156        0.00004    adf.f=frag2
     H       -1.62981        0.96110       -0.10622    adf.f=frag2
     H       -1.86277       -0.51254       -0.75597    adf.f=frag2
     H       -1.83355       -0.33077        0.86231    adf.f=frag2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  fde
    pw91k
    relaxcycles 1
  end
  fragments
     frag1  t21.water
     frag2  t21.ammonia type=FDE  &
     fdeoptions RELAX
     SubEnd
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  save TAPE10
  symmetry NOSYM
  title NH3-H2O PBE/PW91k/(ZORA/jcpl) FDE single point calculation (1 freeze-thaw␣
↪cycle)
  xc
    gga PBE
  end
  Relativity Level=None
EndEngine
eor


# NH3-H2O CPL calculation J(O-H) in water H-bond donor
# tests scalar code branch (no sd term)

```

(continues on next page)

(continued from previous page)

```
"$AMSBIN/cpl" <<eor
 tape10file FDE1.results/TAPE10
 adffile FDE1.results/adf.rkf
 GGA
 NMRCoupling
  fc
  dso
  pso
  AtomPert 1
  AtomResp 2
  scf iterations=15 converge=1.0e-05
 End
eor
rm logfile


# NH3-H2O CPL calculation J(O-H) in water H-bond donor
# tests spin-orbit code branch (sd term)


"$AMSBIN/cpl" <<eor
 tape10file FDE1.results/TAPE10
 adffile FDE1.results/adf.rkf
 GGA
 NMRCoupling
  fc
  sd
  dso
  pso
  AtomPert 1
  AtomResp 2
  scf iterations=15 converge=1.0e-05
 End
eor
rm t21.H t21.N t21.O t21.water t21.ammonia
```

Example: Subsystem TDDFT, coupled FDE excitation energies

Download SUBEXCI_dimer.run

```
#! /bin/sh

# This is example for coupled FDE calculation of excitation energies. The
# subsystem TDDFT code couples the monomer excitations to obtain the excited
# states of the total system (often denoted as coupled frozen density embedding,
# FDEc).

# First the isolated fragments are calculated, the TAPe21's of these fragments
# are t21.iso.rho1 and t21.iso.rho2. Next uncoupled FDE excitation energies are
# calculated in which one fragment is frozen and the other active. The key ALLOW
# PARTIALSUPERFRAGS is necessary to be able to use subsystem information for
# only one subsystem from a TAPE21 file of a previous FDE calculation.

###################################################################
# PREPARATION OF ISOLATED FRAGMENTS
```

(continues on next page)

(continued from previous page)

```
###################################################################

AMS_JOBNAME=Iso1 $AMSBIN/ams <<eor
System
  atoms
     C          1.05754858422573     -1.70701086799077     -3.50000000000000
     O          2.28164544472573     -1.70701086799077     -3.50000000000000
     C          0.20221626882573     -0.49944933059077     -3.50000000000000
     H          0.49106545372573     -2.67285236319077     -3.50000000000000
     C         -1.19491351307427     -0.64892031589077     -3.50000000000000
     C          0.76942690052573      0.78743686120923     -3.50000000000000
     C         -2.02186702237427      0.47538393990923     -3.50000000000000
     H         -1.62606655117427     -1.65281003349077     -3.50000000000000
     C         -0.05719256647427      1.90851291410923     -3.50000000000000
     H          1.85606600152573      0.87752625020923     -3.50000000000000
     C         -1.45186603427427      1.75322106580923     -3.50000000000000
     H         -3.10608163477427      0.35931991730923     -3.50000000000000
     H          0.37787441672573      2.90894982040923     -3.50000000000000
     H         -2.09570824397427      2.63406412680923     -3.50000000000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core none
    type DZP
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
  title Fragment no.           1; (isolated)
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine

eor

mv Iso1.results/adf.rkf t21.iso.rho1

AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
System
  atoms
     C          2.00708906832899      0.06235850568037      3.50000000000000
     O          2.61913749857899      1.12245748356614      3.50000000000000
     C          0.53364394260760     -0.07460023943380      3.50000000000000
     H          2.56029077395134     -0.91115102374797      3.50000000000000
     C         -0.03547527794391     -1.35928561559301      3.50000000000000
     C         -0.29722687542061      1.06006167281502      3.50000000000000
```

(continues on next page)

(continued from previous page)

```
     C        -1.42262807969967     -1.51329623449550      3.50000000000000
     H         0.61834220104568     -2.23461995830645      3.50000000000000
     C        -1.68141695030640      0.90472624158027      3.50000000000000
     H         0.16807297559397      2.04616343352651      3.50000000000000
     C        -2.24426699857796     -0.38074233566867      3.50000000000000
     H        -1.86422099386266     -2.51028564328820      3.50000000000000
     H        -2.33028723444571      1.78172375452933      3.50000000000000
     H        -3.32902057100121     -0.49790451479758      3.50000000000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core none
    type DZP
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
  title Fragment no.          2;  (isolated)
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine

eor

mv Iso2.results/adf.rkf t21.iso.rho2

####################################################################
# FROZEN-DENSITY EMBEDDING CALCULATIONS
# CK: first TDDFT then TDA
####################################################################

AMS_JOBNAME=Emb1 $AMSBIN/ams <<eor
System
  atoms
     C         1.05754858422573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     O         2.28164544472573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     C         0.20221626882573     -0.49944933059077     -3.50000000000000    adf.
↪f=rho1
     H         0.49106545372573     -2.67285236319077     -3.50000000000000    adf.
↪f=rho1
     C        -1.19491351307427     -0.64892031589077     -3.50000000000000    adf.
↪f=rho1
     C         0.76942690052573      0.78743686120923     -3.50000000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     C        -2.02186702237427      0.47538393990923     -3.50000000000000    adf.
↪f=rho1
     H        -1.62606655117427     -1.65281003349077     -3.50000000000000    adf.
↪f=rho1
     C        -0.05719256647427      1.90851291410923     -3.50000000000000    adf.
↪f=rho1
     H         1.85606600152573      0.87752625020923     -3.50000000000000    adf.
↪f=rho1
     C        -1.45186603427427      1.75322106580923     -3.50000000000000    adf.
↪f=rho1
     H        -3.10608163477427      0.35931991730923     -3.50000000000000    adf.
↪f=rho1
     H         0.37787441672573      2.90894982040923     -3.50000000000000    adf.
↪f=rho1
     H        -2.09570824397427      2.63406412680923     -3.50000000000000    adf.
↪f=rho1
     C         2.00708906832899      0.06235850568037      3.50000000000000    adf.
↪f=rho2
     O         2.61913749857899      1.12245748356614      3.50000000000000    adf.
↪f=rho2
     C         0.53364394260760     -0.07460023943380      3.50000000000000    adf.
↪f=rho2
     H         2.56029077395134     -0.91115102374797      3.50000000000000    adf.
↪f=rho2
     C        -0.03547527794391     -1.35928561559301      3.50000000000000    adf.
↪f=rho2
     C        -0.29722687542061      1.06006167281502      3.50000000000000    adf.
↪f=rho2
     C        -1.42262807969967     -1.51329623449550      3.50000000000000    adf.
↪f=rho2
     H         0.61834220104568     -2.23461995830645      3.50000000000000    adf.
↪f=rho2
     C        -1.68141695030640      0.90472624158027      3.50000000000000    adf.
↪f=rho2
     H         0.16807297559397      2.04616343352651      3.50000000000000    adf.
↪f=rho2
     C        -2.24426699857796     -0.38074233566867      3.50000000000000    adf.
↪f=rho2
     H        -1.86422099386266     -2.51028564328820      3.50000000000000    adf.
↪f=rho2
     H        -2.33028723444571      1.78172375452933      3.50000000000000    adf.
↪f=rho2
     H        -3.32902057100121     -0.49790451479758      3.50000000000000    adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  excitations
    analytical
    cdspectrum
```

(continues on next page)

(continued from previous page)

```
    lowest 20
    onlysing
    velocity
  end
  fde
    pw91k
  end
  fragments
     rho1 t21.iso.rho1 subfrag=active
     rho2 t21.iso.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
  title Fragment no.          1; relaxed; TDDFT
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine

eor

mv Emb1.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=TDA1 $AMSBIN/ams <<eor
System
  atoms
     C         1.05754858422573     -1.70701086799077     -3.50000000000000    adf.
 ↪f=rho1
     O         2.28164544472573     -1.70701086799077     -3.50000000000000    adf.
 ↪f=rho1
     C         0.20221626882573     -0.49944933059077     -3.50000000000000    adf.
 ↪f=rho1
     H         0.49106545372573     -2.67285236319077     -3.50000000000000    adf.
 ↪f=rho1
     C        -1.19491351307427     -0.64892031589077     -3.50000000000000    adf.
 ↪f=rho1
     C         0.76942690052573      0.78743686120923     -3.50000000000000    adf.
 ↪f=rho1
     C        -2.02186702237427      0.47538393990923     -3.50000000000000    adf.
 ↪f=rho1
     H        -1.62606655117427     -1.65281003349077     -3.50000000000000    adf.
 ↪f=rho1
     C        -0.05719256647427      1.90851291410923     -3.50000000000000    adf.
 ↪f=rho1
     H         1.85606600152573      0.87752625020923     -3.50000000000000    adf.
 ↪f=rho1
     C        -1.45186603427427      1.75322106580923     -3.50000000000000    adf.
 ↪f=rho1
     H        -3.10608163477427      0.35931991730923     -3.50000000000000    adf.
 ↪f=rho1
     H         0.37787441672573      2.90894982040923     -3.50000000000000    adf.
 ↪f=rho1
     H        -2.09570824397427      2.63406412680923     -3.50000000000000    adf.
 ↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     C        2.00708906832899      0.06235850568037      3.50000000000000    adf.
↪f=rho2
     O        2.61913749857899      1.12245748356614      3.50000000000000    adf.
↪f=rho2
     C        0.53364394260760     -0.07460023943380      3.50000000000000    adf.
↪f=rho2
     H        2.56029077395134     -0.91115102374797      3.50000000000000    adf.
↪f=rho2
     C       -0.03547527794391     -1.35928561559301      3.50000000000000    adf.
↪f=rho2
     C       -0.29722687542061      1.06006167281502      3.50000000000000    adf.
↪f=rho2
     C       -1.42262807969967     -1.51329623449550      3.50000000000000    adf.
↪f=rho2
     H        0.61834220104568     -2.23461995830645      3.50000000000000    adf.
↪f=rho2
     C       -1.68141695030640      0.90472624158027      3.50000000000000    adf.
↪f=rho2
     H        0.16807297559397      2.04616343352651      3.50000000000000    adf.
↪f=rho2
     C       -2.24426699857796     -0.38074233566867      3.50000000000000    adf.
↪f=rho2
     H       -1.86422099386266     -2.51028564328820      3.50000000000000    adf.
↪f=rho2
     H       -2.33028723444571      1.78172375452933      3.50000000000000    adf.
↪f=rho2
     H       -3.32902057100121     -0.49790451479758      3.50000000000000    adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  excitations
    analytical
    cdspectrum
    lowest 20
    onlysing
    velocity
  end
  fde
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active
     rho2 t21.emb.rho1 subfrag=rho2 type=fde
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
```

(continues on next page)

(continued from previous page)

```
  tda
  title Fragment no.          1; relaxed; TDA
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine
eor

mv TDA1.results/adf.rkf t21.emb.rho1_TDA

AMS_JOBNAME=Emb2 $AMSBIN/ams <<eor
System
  atoms
     C         1.05754858422573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     O         2.28164544472573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     C         0.20221626882573     -0.49944933059077     -3.50000000000000    adf.
↪f=rho1
     H         0.49106545372573     -2.67285236319077     -3.50000000000000    adf.
↪f=rho1
     C        -1.19491351307427     -0.64892031589077     -3.50000000000000    adf.
↪f=rho1
     C         0.76942690052573      0.78743686120923     -3.50000000000000    adf.
↪f=rho1
     C        -2.02186702237427      0.47538393990923     -3.50000000000000    adf.
↪f=rho1
     H        -1.62606655117427     -1.65281003349077     -3.50000000000000    adf.
↪f=rho1
     C        -0.05719256647427      1.90851291410923     -3.50000000000000    adf.
↪f=rho1
     H         1.85606600152573      0.87752625020923     -3.50000000000000    adf.
↪f=rho1
     C        -1.45186603427427      1.75322106580923     -3.50000000000000    adf.
↪f=rho1
     H        -3.10608163477427      0.35931991730923     -3.50000000000000    adf.
↪f=rho1
     H         0.37787441672573      2.90894982040923     -3.50000000000000    adf.
↪f=rho1
     H        -2.09570824397427      2.63406412680923     -3.50000000000000    adf.
↪f=rho1
     C         2.00708906832899      0.06235850568037      3.50000000000000    adf.
↪f=rho2
     O         2.61913749857899      1.12245748356614      3.50000000000000    adf.
↪f=rho2
     C         0.53364394260760     -0.07460023943380      3.50000000000000    adf.
↪f=rho2
     H         2.56029077395134     -0.91115102374797      3.50000000000000    adf.
↪f=rho2
     C        -0.03547527794391     -1.35928561559301      3.50000000000000    adf.
↪f=rho2
     C        -0.29722687542061      1.06006167281502      3.50000000000000    adf.
↪f=rho2
     C        -1.42262807969967     -1.51329623449550      3.50000000000000    adf.
↪f=rho2
     H         0.61834220104568     -2.23461995830645      3.50000000000000    adf.
↪f=rho2
```

(continues on next page)

(continued from previous page)

```
     C        -1.68141695030640      0.90472624158027      3.50000000000000    adf.
↪f=rho2
     H         0.16807297559397      2.04616343352651      3.50000000000000    adf.
↪f=rho2
     C        -2.24426699857796     -0.38074233566867      3.50000000000000    adf.
↪f=rho2
     H        -1.86422099386266     -2.51028564328820      3.50000000000000    adf.
↪f=rho2
     H        -2.33028723444571      1.78172375452933      3.50000000000000    adf.
↪f=rho2
     H        -3.32902057100121     -0.49790451479758      3.50000000000000    adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  excitations
    analytical
    cdspectrum
    lowest 20
    onlysing
    velocity
  end
  fde
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2 subfrag=active
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
  title Fragment no.          2; relaxed; TDDFT
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine
eor

mv Emb2.results/adf.rkf t21.emb.rho2

AMS_JOBNAME=TDA2 $AMSBIN/ams <<eor
System
  atoms
     C         1.05754858422573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     O         2.28164544472573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     C         0.20221626882573     -0.49944933059077     -3.50000000000000    adf.
↪f=rho1
     H         0.49106545372573     -2.67285236319077     -3.50000000000000    adf.
↪f=rho1
     C        -1.19491351307427     -0.64892031589077     -3.50000000000000    adf.
↪f=rho1
     C         0.76942690052573      0.78743686120923     -3.50000000000000    adf.
↪f=rho1
     C        -2.02186702237427      0.47538393990923     -3.50000000000000    adf.
↪f=rho1
     H        -1.62606655117427     -1.65281003349077     -3.50000000000000    adf.
↪f=rho1
     C        -0.05719256647427      1.90851291410923     -3.50000000000000    adf.
↪f=rho1
     H         1.85606600152573      0.87752625020923     -3.50000000000000    adf.
↪f=rho1
     C        -1.45186603427427      1.75322106580923     -3.50000000000000    adf.
↪f=rho1
     H        -3.10608163477427      0.35931991730923     -3.50000000000000    adf.
↪f=rho1
     H         0.37787441672573      2.90894982040923     -3.50000000000000    adf.
↪f=rho1
     H        -2.09570824397427      2.63406412680923     -3.50000000000000    adf.
↪f=rho1
     C         2.00708906832899      0.06235850568037      3.50000000000000    adf.
↪f=rho2
     O         2.61913749857899      1.12245748356614      3.50000000000000    adf.
↪f=rho2
     C         0.53364394260760     -0.07460023943380      3.50000000000000    adf.
↪f=rho2
     H         2.56029077395134     -0.91115102374797      3.50000000000000    adf.
↪f=rho2
     C        -0.03547527794391     -1.35928561559301      3.50000000000000    adf.
↪f=rho2
     C        -0.29722687542061      1.06006167281502      3.50000000000000    adf.
↪f=rho2
     C        -1.42262807969967     -1.51329623449550      3.50000000000000    adf.
↪f=rho2
     H         0.61834220104568     -2.23461995830645      3.50000000000000    adf.
↪f=rho2
     C        -1.68141695030640      0.90472624158027      3.50000000000000    adf.
↪f=rho2
     H         0.16807297559397      2.04616343352651      3.50000000000000    adf.
↪f=rho2
     C        -2.24426699857796     -0.38074233566867      3.50000000000000    adf.
↪f=rho2
     H        -1.86422099386266     -2.51028564328820      3.50000000000000    adf.
↪f=rho2
     H        -2.33028723444571      1.78172375452933      3.50000000000000    adf.
↪f=rho2
     H        -3.32902057100121     -0.49790451479758      3.50000000000000    adf.
↪f=rho2
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  excitations
    analytical
    cdspectrum
    lowest 20
    onlysing
    velocity
  end
  fde
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.emb.rho2 subfrag=active
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
  tda
  title Fragment no.          2; relaxed; TDA
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine
eor

mv TDA2.results/adf.rkf t21.emb.rho2_TDA

###################################################################
# SUBSYS EXCITATIONS CALCULATION
###################################################################

AMS_JOBNAME=TDDFT $AMSBIN/ams <<eor
System
  atoms
     C         1.05754858422573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     O         2.28164544472573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     C         0.20221626882573     -0.49944933059077     -3.50000000000000    adf.
↪f=rho1
     H         0.49106545372573     -2.67285236319077     -3.50000000000000    adf.
↪f=rho1
     C        -1.19491351307427     -0.64892031589077     -3.50000000000000    adf.
↪f=rho1
     C         0.76942690052573      0.78743686120923     -3.50000000000000    adf.
↪f=rho1
     C        -2.02186702237427      0.47538393990923     -3.50000000000000    adf.
↪f=rho1
     H        -1.62606655117427     -1.65281003349077     -3.50000000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     C        -0.05719256647427      1.90851291410923     -3.50000000000000    adf.
↪f=rho1
     H         1.85606600152573      0.87752625020923     -3.50000000000000    adf.
↪f=rho1
     C        -1.45186603427427      1.75322106580923     -3.50000000000000    adf.
↪f=rho1
     H        -3.10608163477427      0.35931991730923     -3.50000000000000    adf.
↪f=rho1
     H         0.37787441672573      2.90894982040923     -3.50000000000000    adf.
↪f=rho1
     H        -2.09570824397427      2.63406412680923     -3.50000000000000    adf.
↪f=rho1
     C         2.00708906832899      0.06235850568037      3.50000000000000    adf.
↪f=rho2
     O         2.61913749857899      1.12245748356614      3.50000000000000    adf.
↪f=rho2
     C         0.53364394260760     -0.07460023943380      3.50000000000000    adf.
↪f=rho2
     H         2.56029077395134     -0.91115102374797      3.50000000000000    adf.
↪f=rho2
     C        -0.03547527794391     -1.35928561559301      3.50000000000000    adf.
↪f=rho2
     C        -0.29722687542061      1.06006167281502      3.50000000000000    adf.
↪f=rho2
     C        -1.42262807969967     -1.51329623449550      3.50000000000000    adf.
↪f=rho2
     H         0.61834220104568     -2.23461995830645      3.50000000000000    adf.
↪f=rho2
     C        -1.68141695030640      0.90472624158027      3.50000000000000    adf.
↪f=rho2
     H         0.16807297559397      2.04616343352651      3.50000000000000    adf.
↪f=rho2
     C        -2.24426699857796     -0.38074233566867      3.50000000000000    adf.
↪f=rho2
     H        -1.86422099386266     -2.51028564328820      3.50000000000000    adf.
↪f=rho2
     H        -2.33028723444571      1.78172375452933      3.50000000000000    adf.
↪f=rho2
     H        -3.32902057100121     -0.49790451479758      3.50000000000000    adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  diffuse
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active
     rho2 t21.emb.rho2 subfrag=active type=fde
```

(continues on next page)

(continued from previous page)

```
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  subexci
    couplblock
    cthres 10000.00
    sfthres 0.00010000
  end
  symmetry NOSYM
  title COUPLED SUBSYSTEM EXCITATIONS; TDDFT
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine
eor

AMS_JOBNAME=TDA $AMSBIN/ams <<eor
System
  atoms
     C         1.05754858422573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     O         2.28164544472573     -1.70701086799077     -3.50000000000000    adf.
↪f=rho1
     C         0.20221626882573     -0.49944933059077     -3.50000000000000    adf.
↪f=rho1
     H         0.49106545372573     -2.67285236319077     -3.50000000000000    adf.
↪f=rho1
     C        -1.19491351307427     -0.64892031589077     -3.50000000000000    adf.
↪f=rho1
     C         0.76942690052573      0.78743686120923     -3.50000000000000    adf.
↪f=rho1
     C        -2.02186702237427      0.47538393990923     -3.50000000000000    adf.
↪f=rho1
     H        -1.62606655117427     -1.65281003349077     -3.50000000000000    adf.
↪f=rho1
     C        -0.05719256647427      1.90851291410923     -3.50000000000000    adf.
↪f=rho1
     H         1.85606600152573      0.87752625020923     -3.50000000000000    adf.
↪f=rho1
     C        -1.45186603427427      1.75322106580923     -3.50000000000000    adf.
↪f=rho1
     H        -3.10608163477427      0.35931991730923     -3.50000000000000    adf.
↪f=rho1
     H         0.37787441672573      2.90894982040923     -3.50000000000000    adf.
↪f=rho1
     H        -2.09570824397427      2.63406412680923     -3.50000000000000    adf.
↪f=rho1
     C         2.00708906832899      0.06235850568037      3.50000000000000    adf.
↪f=rho2
     O         2.61913749857899      1.12245748356614      3.50000000000000    adf.
↪f=rho2
     C         0.53364394260760     -0.07460023943380      3.50000000000000    adf.
↪f=rho2
     H         2.56029077395134     -0.91115102374797      3.50000000000000    adf.
↪f=rho2
```

(continues on next page)

(continued from previous page)

```
     C        -0.03547527794391     -1.35928561559301      3.50000000000000    adf.
↪f=rho2
     C        -0.29722687542061      1.06006167281502      3.50000000000000    adf.
↪f=rho2
     C        -1.42262807969967     -1.51329623449550      3.50000000000000    adf.
↪f=rho2
     H         0.61834220104568     -2.23461995830645      3.50000000000000    adf.
↪f=rho2
     C        -1.68141695030640      0.90472624158027      3.50000000000000    adf.
↪f=rho2
     H         0.16807297559397      2.04616343352651      3.50000000000000    adf.
↪f=rho2
     C        -2.24426699857796     -0.38074233566867      3.50000000000000    adf.
↪f=rho2
     H        -1.86422099386266     -2.51028564328820      3.50000000000000    adf.
↪f=rho2
     H        -2.33028723444571      1.78172375452933      3.50000000000000    adf.
↪f=rho2
     H        -3.32902057100121     -0.49790451479758      3.50000000000000    adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  diffuse
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
  end
  fragments
     rho1 t21.emb.rho1_TDA subfrag=active
     rho2 t21.emb.rho2_TDA subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  scf
    converge 1.0e-8
  end
  subexci
    couplblock
    cthres 10000.00
    sfthres 0.00010000
    tda
  end
  symmetry NOSYM
  title COUPLED SUBSYSTEM EXCITATIONS; TDA
  xc
    gga BECKE88 PERDEW86
  end
  Relativity Level=None
EndEngine
eor
```

Example: FDE and COSMO: H2O-NH3

Download LoCOSMO_H2O-NH3.run

```
#!/bin/sh


# This is example for a calculation of FDE and (localized) COSMO in ADF in case
# of closed shell fragments.

# It performs single point runs for H2 O and NH3 in COSMO (water) and uses these
# fragments in a freeze-and-thaw calculations. Recommended is to limit the
# number of freeze-and-thaw cycles in such calculations where FDE and COSMO are
# combined, because the FT convergence criterion may not be met in a finite
# number of cycles, although the density is practically converged.

# In the freeze-and-thaw cycles LoCosmo is included in the SOLVATION key, thus
# the local COSMO will be used in the calculation. A cutoff radius (Angstrom)
# must be specified after the LoCOSMO subargument, in this case it is 3.0
# Angstrom. All surface charges within this radius from any atom of the active
# fragment are included in the active charge space.


AMS_JOBNAME=Iso1 $AMSBIN/ams <<eor
System
  atoms
     H        0.0201  0.6299  0.0000
     O       -0.0370  1.6130  0.0000
     H        0.8936  1.9007  0.0000
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    type DZP
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  solvation
  end
  symmetry NOSYM
  title H2O isolated
  xc
    gga BP86
  end
EndEngine
eor
mv Iso1.results/adf.rkf t21.iso.rho1


AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
```

(continues on next page)

(continued from previous page)

```
System
  atoms
     N       -0.0370  -1.4327  0.0000
     H        0.8110  -2.0086  0.0000
     H       -0.5850  -1.6987  0.8244
     H       -0.5850  -1.6987 -0.8244
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    type DZP
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  solvation
  end
  symmetry NOSYM
  title NH3 isolated
  xc
    gga BP86
  end
EndEngine
eor
mv Iso2.results/adf.rkf t21.iso.rho2


$AMSBIN/ams <<eor
System
  atoms
     H        0.0201   0.6299   0.0000  adf.f=frag1
     O       -0.0370   1.6130   0.0000  adf.f=frag1
     H        0.8936   1.9007   0.0000  adf.f=frag1
     N       -0.0370  -1.4327   0.0000  adf.f=frag2
     H        0.8110  -2.0086   0.0000  adf.f=frag2
     H       -0.5850  -1.6987   0.8244  adf.f=frag2
     H       -0.5850  -1.6987  -0.8244  adf.f=frag2
  end
end

Task SinglePoint

Engine ADF
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  fde
    pw91k
```

(continues on next page)

(continued from previous page)

```
    relaxcycles 3
  end
  fragments
     frag1  t21.iso.rho1
     frag2  t21.iso.rho2 type=FDE &
     FDEOPTIONS RELAX
     SubEnd
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  solvation
    charged LoCOSMO 3.0
  end
  symmetry NOSYM
  title FT cycles for H2O-NH3
  xc
    gga BP86
  end
EndEngine
eor
```

Example: FDE and COSMO: H2O-NH3

Download LoCOSMO_H2O-NH3_TDDFT.run

```
#!/bin/sh


# This is example for a calculation of excitation energies using FDE and
# localized COSMO, which is based on the LoCOSMO_H2O-NH3 example. The difference
# is that it includes the calculation of excitation energies (EXCITATION block
# key), and performs only 1 freeze-and-thaw cycle for H2 O and NH3 in COSMO
# (water). The input for the fragment calculations are not repeated here.

AMS_JOBNAME=Iso1 $AMSBIN/ams <<eor
System
  atoms
     H        0.0201   0.6299   0.0000
     O       -0.0370   1.6130   0.0000
     H        0.8936   1.9007   0.0000
  end
end

Task SinglePoint

Engine ADF
  title H2O isolated
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    type DZP
  end
```

(continues on next page)

(continued from previous page)

```
  noprint BAS FUNCTIONS
  numericalquality Good
  solvation
  end
  symmetry NOSYM
  xc
    gga BP86
  end
EndEngine
eor
mv Iso1.results/adf.rkf t21.iso.rho1

AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
System
  atoms
     N       -0.0370  -1.4327  0.0000
     H        0.8110  -2.0086  0.0000
     H       -0.5850  -1.6987  0.8244
     H       -0.5850  -1.6987 -0.8244
  end
end

Task SinglePoint

Engine ADF
  title NH3 isolated
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    type DZP
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  solvation
  end
  symmetry NOSYM
  xc
    gga BP86
  end
EndEngine

eor
mv Iso2.results/adf.rkf t21.iso.rho2

$AMSBIN/ams <<eor
System
  atoms
     H        0.0201   0.6299   0.0000  adf.f=frag1
     O       -0.0370   1.6130   0.0000  adf.f=frag1
     H        0.8936   1.9007   0.0000  adf.f=frag1
     N       -0.0370  -1.4327   0.0000  adf.f=frag2
     H        0.8110  -2.0086   0.0000  adf.f=frag2
     H       -0.5850  -1.6987   0.8244  adf.f=frag2
     H       -0.5850  -1.6987  -0.8244  adf.f=frag2
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  title FT cycles for H2O-NH3
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  excitations
  end
  fde
    pw91k
    relaxcycles 1
  end
  fragments
     frag1  t21.iso.rho1
     frag2  t21.iso.rho2 type=FDE &
     FDEOPTIONS RELAX
     SubEnd
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  solvation
    charged LoCOSMO 3.0
  end
  symmetry NOSYM
  xc
    gga BP86
  end
EndEngine
eor
```

14.2.5 QM/MM calculations

14.2.6 Quild: Quantum-regions Interconnected by Local Descriptions

- Quild manual: Quild examples

14.2.7 DIM/QM: Discrete Interaction Model/Quantum Mechanics

Example: DRF: H2O and H2O

Download `DIMQM_DRF_BE.run`

```
#! /bin/sh


$AMSBIN/ams <<eor
System
```

(continues on next page)

(continued from previous page)

```
  atoms
     O       2.698990442000       0.000000000000      -0.000314732000
     H       1.927785302000       0.000000000000      -0.591074192000
     H       3.470195581000       0.000000000000      -0.591074192000
  end
end

Task SinglePoint

Engine ADF
  dimpar
     GROUP mol2
     3
     O      -0.801009558000       0.000000000000      -0.000314732000 pol=5.7494␣
↪char=-0.7134114435520156
     H      -1.572214698000      -0.000000000000      -0.591074192000 pol=2.7927␣
↪char=0.3567057217727492
     H      -0.029804419000      -0.000000000000      -0.591074192000 pol=2.7927␣
↪char=0.3567057217727492
     SUBEND
  end
  dimqm
     drf
  end
  basis
    core Large
    type TZP
  end
  symmetry NOSYM
  Relativity
    Level None
  End
EndEngine

eor
```

Example: DRF: hyperpolarizability H2O in water

Download DIMQM_DRF_Hyperpol.run

```
#!/bin/sh

# == Expert option ==

# This example demonstrates how to use DRF. Here, the hyperpolarizability is
# calculated of a water molecule surrounded by an environment of 127 water
# molecules. See also: C.R. Jacob, J. Neugebauer, L. Jensen and L. Visscher,
# Comparison of frozen-density embedding and discrete reaction field solvent
# models for molecular properties, Physical Chemistry Chemical Physics 8, 2349
# (2006)


$AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     O       0.0000    0.0000    0.0000
     H      -0.7568    0.0000    0.5860
     H       0.7568    0.0000    0.5860
  end
end

Task SinglePoint

Engine ADF
  title DRF-TEST H2O in H2O
  dimqm
     DRF
  end
  externals
     O    4      water 2, -0.6690,     -6.0223,  -6.2499,  -2.3894,  9.3005,  3.0236
     H    5      water 2,  0.3345,     -6.9347,  -6.2642,  -2.1006,  0.0690,  1.5118
     H    6      water 2,  0.3345,     -5.5621,  -6.8017,  -1.7570,  0.0690,  1.5118
     ANAL 2
     O    7      water 3, -0.6690,     -0.5907,   4.8257,  -1.7097,  9.3005,  3.0236
     H    8      water 3,  0.3345,     -1.4937,   5.1420,  -1.6831,  0.0690,  1.5118
     H    9      water 3,  0.3345,     -0.0655,   5.5766,  -1.4333,  0.0690,  1.5118
     ANAL 2
     O   10      water 4, -0.6690,      2.3651,  -0.2666,   1.1699,  9.3005,  3.0236
     H   11      water 4,  0.3345,      2.2615,  -1.1034,   1.6229,  0.0690,  1.5118
     H   12      water 4,  0.3345,      3.1322,  -0.3900,   0.6111,  0.0690,  1.5118
     ANAL 2
     O   13      water 5, -0.6690,     -8.9796,  -1.9359,  -5.7076,  9.3005,  3.0236
     H   14      water 5,  0.3345,     -8.3124,  -2.2437,  -5.0941,  0.0690,  1.5118
     H   15      water 5,  0.3345,     -8.7198,  -2.3076,  -6.5504,  0.0690,  1.5118
     ANAL 2
     O   16      water 6, -0.6690,     -5.6969,  -2.2033,   0.2741,  9.3005,  3.0236
     H   17      water 6,  0.3345,     -4.8479,  -2.1094,  -0.1577,  0.0690,  1.5118
     H   18      water 6,  0.3345,     -6.3368,  -2.0090,  -0.4106,  0.0690,  1.5118
     ANAL 2
     O   19      water 7, -0.6690,     -3.4684,  -4.8700,  -5.9433,  9.3005,  3.0236
     H   20      water 7,  0.3345,     -4.3144,  -5.1084,  -5.5645,  0.0690,  1.5118
     H   21      water 7,  0.3345,     -3.6551,  -4.1011,  -6.4819,  0.0690,  1.5118
     ANAL 2
     O   22      water 8, -0.6690,     -7.4969,  -3.4704,  -3.5082,  9.3005,  3.0236
     H   23      water 8,  0.3345,     -7.2745,  -4.3985,  -3.5808,  0.0690,  1.5118
     H   24      water 8,  0.3345,     -8.1668,  -3.4372,  -2.8253,  0.0690,  1.5118
     ANAL 2
     O   25      water 9, -0.6690,     -0.5357,  -3.4594, -10.9181,  9.3005,  3.0236
     H   26      water 9,  0.3345,      0.2718,  -3.7131, -10.4712,  0.0690,  1.5118
     H   27      water 9,  0.3345,     -1.1951,  -3.4329, -10.2249,  0.0690,  1.5118
     ANAL 2
     O   28      water 10, -0.6690,    -4.2195,   2.2934,   4.6878,  9.3005,  3.0236
     H   29      water 10,  0.3345,    -3.9117,   1.4378,   4.9869,  0.0690,  1.5118
     H   30      water 10,  0.3345,    -3.8849,   2.3695,   3.7943,  0.0690,  1.5118
     ANAL 2
     O   31      water 11, -0.6690,     0.8931,  -5.9761,  -1.5221,  9.3005,  3.0236
     H   32      water 11,  0.3345,     0.8044,  -6.1702,  -2.4552,  0.0690,  1.5118
     H   33      water 11,  0.3345,     0.4898,  -5.1144,  -1.4172,  0.0690,  1.5118
     ANAL 2
     O   34      water 12, -0.6690,     0.5778,  -0.8986,  -9.9576,  9.3005,  3.0236
     H   35      water 12,  0.3345,     1.0965,  -1.2400, -10.6860,  0.0690,  1.5118
     H   36      water 12,  0.3345,     0.0709,  -1.6505,  -9.6513,  0.0690,  1.5118
```

(continues on next page)

(continued from previous page)

```
    ANAL 2
    O   37      water 13, -0.6690,    -7.3305,  -0.8007,  -1.8143,  9.3005,  3.0236
    H   38      water 13,  0.3345,    -7.0622,  -1.0105,  -2.7088,  0.0690,  1.5118
    H   39      water 13,  0.3345,    -8.1064,  -1.3406,  -1.6640,  0.0690,  1.5118
    ANAL 2
    O   40      water 14, -0.6690,     2.7991,  -3.6051,  -6.9353,  9.3005,  3.0236
    H   41      water 14,  0.3345,     3.7133,  -3.5041,  -6.6702,  0.0690,  1.5118
    H   42      water 14,  0.3345,     2.3002,  -3.1634,  -6.2483,  0.0690,  1.5118
    ANAL 2
    O   43      water 15, -0.6690,    -6.0718,  -5.7718,  -4.9959,  9.3005,  3.0236
    H   44      water 15,  0.3345,    -5.9353,  -6.5906,  -5.4724,  0.0690,  1.5118
    H   45      water 15,  0.3345,    -5.9143,  -5.9987,  -4.0795,  0.0690,  1.5118
    ANAL 2
    O   46      water 16, -0.6690,    -0.8710,  -6.0513,   0.7837,  9.3005,  3.0236
    H   47      water 16,  0.3345,    -0.1617,  -5.5424,   0.3911,  0.0690,  1.5118
    H   48      water 16,  0.3345,    -0.9154,  -6.8479,   0.2550,  0.0690,  1.5118
    ANAL 2
    O   49      water 17, -0.6690,    -3.8761,  -2.6555,   2.3219,  9.3005,  3.0236
    H   50      water 17,  0.3345,    -4.6665,  -2.5032,   1.8041,  0.0690,  1.5118
    H   51      water 17,  0.3345,    -4.1060,  -3.3810,   2.9024,  0.0690,  1.5118
    ANAL 2
    O   52      water 18, -0.6690,     2.7564,  -3.6188,   1.9534,  9.3005,  3.0236
    H   53      water 18,  0.3345,     2.7865,  -3.5376,   1.0002,  0.0690,  1.5118
    H   54      water 18,  0.3345,     3.3823,  -2.9672,   2.2693,  0.0690,  1.5118
    ANAL 2
    O   55      water 19, -0.6690,    -0.6609,  -3.6923,  -1.5511,  9.3005,  3.0236
    H   56      water 19,  0.3345,    -1.2586,  -3.4903,  -0.8312,  0.0690,  1.5118
    H   57      water 19,  0.3345,    -0.1127,  -2.9121,  -1.6334,  0.0690,  1.5118
    ANAL 2
    O   58      water 20, -0.6690,     3.5653,  -0.1379,  -3.4865,  9.3005,  3.0236
    H   59      water 20,  0.3345,     3.4730,  -1.0831,  -3.6058,  0.0690,  1.5118
    H   60      water 20,  0.3345,     4.3918,   0.0785,  -3.9181,  0.0690,  1.5118
    ANAL 2
    O   61      water 21, -0.6690,     2.8079,  -2.9333,  -3.4058,  9.3005,  3.0236
    H   62      water 21,  0.3345,     2.0176,  -3.2231,  -3.8614,  0.0690,  1.5118
    H   63      water 21,  0.3345,     3.5200,  -3.4054,  -3.8373,  0.0690,  1.5118
    ANAL 2
    O   64      water 22, -0.6690,    -3.0280,  -4.0184,  -2.9686,  9.3005,  3.0236
    H   65      water 22,  0.3345,    -2.9833,  -3.9686,  -3.9234,  0.0690,  1.5118
    H   66      water 22,  0.3345,    -2.1153,  -3.9764,  -2.6835,  0.0690,  1.5118
    ANAL 2
    O   67      water 23, -0.6690,    -2.6269,   2.2878,  -1.7246,  9.3005,  3.0236
    H   68      water 23,  0.3345,    -3.3175,   2.9420,  -1.6184,  0.0690,  1.5118
    H   69      water 23,  0.3345,    -1.8157,   2.7657,  -1.5521,  0.0690,  1.5118
    ANAL 2
    O   70      water 24, -0.6690,    -5.7904,   0.2934,  -6.9759,  9.3005,  3.0236
    H   71      water 24,  0.3345,    -5.2884,   1.1025,  -7.0732,  0.0690,  1.5118
    H   72      water 24,  0.3345,    -6.4866,   0.3633,  -7.6290,  0.0690,  1.5118
    ANAL 2
    O   73      water 25, -0.6690,    -4.9194,   2.8572,  -6.6807,  9.3005,  3.0236
    H   74      water 25,  0.3345,    -4.1208,   2.8574,  -6.1531,  0.0690,  1.5118
    H   75      water 25,  0.3345,    -4.6111,   2.9479,  -7.5822,  0.0690,  1.5118
    ANAL 2
    O   76      water 26, -0.6690,     1.5750,  -6.5856,   1.9188,  9.3005,  3.0236
    H   77      water 26,  0.3345,     1.9374,  -5.6997,   1.9225,  0.0690,  1.5118
    H   78      water 26,  0.3345,     0.6464,  -6.4639,   1.7216,  0.0690,  1.5118
    ANAL 2
```

(continues on next page)

(continued from previous page)

```
    O   79      water 27, -0.6690,    -3.9796,  -1.4339,  -8.6954,  9.3005,  3.0236
    H   80      water 27,  0.3345,    -4.5199,  -1.0671,  -7.9956,  0.0690,  1.5118
    H   81      water 27,  0.3345,    -4.3002,  -1.0113,  -9.4921,  0.0690,  1.5118
    ANAL 2
    O   82      water 28, -0.6690,    -0.8980,  -1.0815,  -4.3428,  9.3005,  3.0236
    H   83      water 28,  0.3345,    -0.6188,  -0.2290,  -4.6764,  0.0690,  1.5118
    H   84      water 28,  0.3345,    -0.3369,  -1.2371,  -3.5832,  0.0690,  1.5118
    ANAL 2
    O   85      water 29, -0.6690,     2.8053,  -5.0592,   4.4727,  9.3005,  3.0236
    H   86      water 29,  0.3345,     2.7908,  -4.5603,   3.6560,  0.0690,  1.5118
    H   87      water 29,  0.3345,     2.4078,  -4.4744,   5.1179,  0.0690,  1.5118
    ANAL 2
    O   88      water 30, -0.6690,     3.1724,  -3.0925,  -0.6766,  9.3005,  3.0236
    H   89      water 30,  0.3345,     3.2306,  -3.1480,  -1.6304,  0.0690,  1.5118
    H   90      water 30,  0.3345,     3.7164,  -2.3393,  -0.4468,  0.0690,  1.5118
    ANAL 2
    O   91      water 31, -0.6690,    -4.4977,   3.3377,   0.6679,  9.3005,  3.0236
    H   92      water 31,  0.3345,    -4.9024,   2.7604,   0.0205,  0.0690,  1.5118
    H   93      water 31,  0.3345,    -5.2078,   3.5562,   1.2712,  0.0690,  1.5118
    ANAL 2
    O   94      water 32, -0.6690,     0.7528,   3.7841,  -3.5468,  9.3005,  3.0236
    H   95      water 32,  0.3345,     0.2600,   4.0986,  -2.7890,  0.0690,  1.5118
    H   96      water 32,  0.3345,     1.5200,   3.3523,  -3.1712,  0.0690,  1.5118
    ANAL 2
    O   97      water 33, -0.6690,     0.6169,  -8.3945,  -4.2494,  9.3005,  3.0236
    H   98      water 33,  0.3345,    -0.2327,  -8.7666,  -4.4857,  0.0690,  1.5118
    H   99      water 33,  0.3345,     0.6722,  -7.5835,  -4.7547,  0.0690,  1.5118
    ANAL 2
    O  100      water 34, -0.6690,    -1.0012,  -5.6281,  -4.5428,  9.3005,  3.0236
    H  101      water 34,  0.3345,    -1.4754,  -5.6715,  -5.3731,  0.0690,  1.5118
    H  102      water 34,  0.3345,    -1.5387,  -6.1335,  -3.9331,  0.0690,  1.5118
    ANAL 2
    O  103      water 35, -0.6690,     0.6896,  -3.4643,  -4.9164,  9.3005,  3.0236
    H  104      water 35,  0.3345,     0.0219,  -4.1296,  -4.7498,  0.0690,  1.5118
    H  105      water 35,  0.3345,     0.2328,  -2.6316,  -4.7984,  0.0690,  1.5118
    ANAL 2
    O  106      water 36, -0.6690,    -2.1739,  -3.4893,  -8.6965,  9.3005,  3.0236
    H  107      water 36,  0.3345,    -2.5934,  -4.2938,  -9.0013,  0.0690,  1.5118
    H  108      water 36,  0.3345,    -2.8829,  -2.8473,  -8.6625,  0.0690,  1.5118
    ANAL 2
    O  109      water 37, -0.6690,    -3.1332,  -1.6040,  -1.3286,  9.3005,  3.0236
    H  110      water 37,  0.3345,    -3.0590,  -2.4135,  -1.8339,  0.0690,  1.5118
    H  111      water 37,  0.3345,    -3.2876,  -0.9250,  -1.9853,  0.0690,  1.5118
    ANAL 2
    O  112      water 38, -0.6690,    -2.7516,  -6.7877,  -2.6294,  9.3005,  3.0236
    H  113      water 38,  0.3345,    -3.5480,  -7.1086,  -3.0525,  0.0690,  1.5118
    H  114      water 38,  0.3345,    -2.8732,  -5.8397,  -2.5775,  0.0690,  1.5118
    ANAL 2
    O  115      water 39, -0.6690,    -2.5918,   2.9114,  -8.7408,  9.3005,  3.0236
    H  116      water 39,  0.3345,    -2.6283,   3.8661,  -8.6824,  0.0690,  1.5118
    H  117      water 39,  0.3345,    -1.7080,   2.6872,  -8.4499,  0.0690,  1.5118
    ANAL 2
    O  118      water 40, -0.6690,    -9.3024,  -2.6424,  -1.5753,  9.3005,  3.0236
    H  119      water 40,  0.3345,    -9.9294,  -2.0331,  -1.1858,  0.0690,  1.5118
    H  120      water 40,  0.3345,    -9.2070,  -3.3371,  -0.9237,  0.0690,  1.5118
    ANAL 2
    O  121      water 41, -0.6690,    -1.8562,  -9.1674,  -5.3470,  9.3005,  3.0236
```

(continues on next page)

(continued from previous page)

```
    H  122       water 41,  0.3345,    -2.3294,  -9.8602,  -5.8077,  0.0690,  1.5118
    H  123       water 41,  0.3345,    -2.4022,  -8.9693,  -4.5862,  0.0690,  1.5118
    ANAL 2
    O  124       water 42, -0.6690,    -5.7252,  -2.4694,  -5.8976,  9.3005,  3.0236
    H  125       water 42,  0.3345,    -6.1500,  -2.5840,  -5.0476,  0.0690,  1.5118
    H  126       water 42,  0.3345,    -5.6141,  -1.5226,  -5.9843,  0.0690,  1.5118
    ANAL 2
    O  127       water 43, -0.6690,    -3.3825,   0.0095,  -3.5627,  9.3005,  3.0236
    H  128       water 43,  0.3345,    -3.0805,   0.6489,  -2.9177,  0.0690,  1.5118
    H  129       water 43,  0.3345,    -2.5797,  -0.3852,  -3.9029,  0.0690,  1.5118
    ANAL 2
    O  130       water 44, -0.6690,    -5.9824,   0.2454,  -4.3661,  9.3005,  3.0236
    H  131       water 44,  0.3345,    -5.1025,   0.4269,  -4.0360,  0.0690,  1.5118
    H  132       water 44,  0.3345,    -5.9262,   0.4186,  -5.3057,  0.0690,  1.5118
    ANAL 2
    O  133       water 45, -0.6690,     4.7647,   0.6670,  -7.3969,  9.3005,  3.0236
    H  134       water 45,  0.3345,     4.6954,   0.5448,  -6.4501,  0.0690,  1.5118
    H  135       water 45,  0.3345,     3.8600,   0.7691,  -7.6922,  0.0690,  1.5118
    ANAL 2
    O  136       water 46, -0.6690,     0.6049,  -1.4649,  -2.0801,  9.3005,  3.0236
    H  137       water 46,  0.3345,     1.5609,  -1.4717,  -2.0332,  0.0690,  1.5118
    H  138       water 46,  0.3345,     0.3326,  -0.9204,  -1.3415,  0.0690,  1.5118
    ANAL 2
    O  139       water 47, -0.6690,    -2.7599,  -0.3793,   1.0454,  9.3005,  3.0236
    H  140       water 47,  0.3345,    -3.1126,  -0.7184,   0.2228,  0.0690,  1.5118
    H  141       water 47,  0.3345,    -3.0830,  -0.9851,   1.7124,  0.0690,  1.5118
    ANAL 2
    O  142       water 48, -0.6690,    -2.8683,   1.9384,   2.3701,  9.3005,  3.0236
    H  143       water 48,  0.3345,    -3.4277,   2.5156,   1.8504,  0.0690,  1.5118
    H  144       water 48,  0.3345,    -2.7961,   1.1400,   1.8470,  0.0690,  1.5118
    ANAL 2
    O  145       water 49, -0.6690,    -5.7630,   1.8173,  -1.3700,  9.3005,  3.0236
    H  146       water 49,  0.3345,    -5.8465,   1.0390,  -1.9209,  0.0690,  1.5118
    H  147       water 49,  0.3345,    -6.6060,   1.8853,  -0.9219,  0.0690,  1.5118
    ANAL 2
    O  148       water 50, -0.6690,    -3.0562,   6.2727,  -1.6414,  9.3005,  3.0236
    H  149       water 50,  0.3345,    -3.9270,   5.9633,  -1.3923,  0.0690,  1.5118
    H  150       water 50,  0.3345,    -2.9607,   7.1103,  -1.1881,  0.0690,  1.5118
    ANAL 2
    O  151       water 51, -0.6690,     2.4606,   1.9450,  -2.0022,  9.3005,  3.0236
    H  152       water 51,  0.3345,     2.8071,   1.1437,  -2.3947,  0.0690,  1.5118
    H  153       water 51,  0.3345,     1.7993,   1.6407,  -1.3806,  0.0690,  1.5118
    ANAL 2
    O  154       water 52, -0.6690,    -0.8297,   1.8046,  -5.0085,  9.3005,  3.0236
    H  155       water 52,  0.3345,    -0.3315,   2.4831,  -4.5529,  0.0690,  1.5118
    H  156       water 52,  0.3345,    -0.6508,   1.9580,  -5.9361,  0.0690,  1.5118
    ANAL 2
    O  157       water 53, -0.6690,    -3.0963,   3.1475,  -4.6319,  9.3005,  3.0236
    H  158       water 53,  0.3345,    -2.3095,   2.6128,  -4.7377,  0.0690,  1.5118
    H  159       water 53,  0.3345,    -2.7695,   4.0450,  -4.5698,  0.0690,  1.5118
    ANAL 2
    O  160       water 54, -0.6690,    -2.9391,  -4.1930,   0.1352,  9.3005,  3.0236
    H  161       water 54,  0.3345,    -2.3195,  -4.9152,   0.2382,  0.0690,  1.5118
    H  162       water 54,  0.3345,    -3.1844,  -3.9582,   1.0301,  0.0690,  1.5118
    ANAL 2
    O  163       water 55, -0.6690,    -5.7159,   2.6060,  -9.7518,  9.3005,  3.0236
    H  164       water 55,  0.3345,    -5.0101,   2.1773, -10.2358,  0.0690,  1.5118
```

(continues on next page)

(continued from previous page)

```
H  165      water 55,  0.3345,    -6.3940,   1.9351,  -9.6725,  0.0690,  1.5118
ANAL 2
O  166      water 56, -0.6690,     6.2695,   3.1997,  -2.9652,  9.3005,  3.0236
H  167      water 56,  0.3345,     6.4762,   3.9915,  -2.4687,  0.0690,  1.5118
H  168      water 56,  0.3345,     5.3686,   3.3305,  -3.2607,  0.0690,  1.5118
ANAL 2
O  169      water 57, -0.6690,     0.7339,  -1.8844,   2.7237,  9.3005,  3.0236
H  170      water 57,  0.3345,     1.2467,  -1.5058,   3.4378,  0.0690,  1.5118
H  171      water 57,  0.3345,     1.1959,  -2.6920,   2.4989,  0.0690,  1.5118
ANAL 2
O  172      water 58, -0.6690,     4.9752,  -1.1454,   0.0349,  9.3005,  3.0236
H  173      water 58,  0.3345,     5.2875,  -0.3193,  -0.3341,  0.0690,  1.5118
H  174      water 58,  0.3345,     5.7725,  -1.6403,   0.2233,  0.0690,  1.5118
ANAL 2
O  175      water 59, -0.6690,    -1.7496,  -1.8578,   3.9802,  9.3005,  3.0236
H  176      water 59,  0.3345,    -0.9639,  -1.9885,   3.4495,  0.0690,  1.5118
H  177      water 59,  0.3345,    -2.4606,  -2.2142,   3.4475,  0.0690,  1.5118
ANAL 2
O  178      water 60, -0.6690,    -2.0308,   5.8210,  -4.5295,  9.3005,  3.0236
H  179      water 60,  0.3345,    -1.9811,   5.7326,  -3.5777,  0.0690,  1.5118
H  180      water 60,  0.3345,    -2.5794,   6.5926,  -4.6702,  0.0690,  1.5118
ANAL 2
O  181      water 61, -0.6690,    -4.7217,   3.9768,  -2.7147,  9.3005,  3.0236
H  182      water 61,  0.3345,    -4.0442,   3.5964,  -3.2736,  0.0690,  1.5118
H  183      water 61,  0.3345,    -5.2852,   3.2381,  -2.4846,  0.0690,  1.5118
ANAL 2
O  184      water 62, -0.6690,    -0.1740,   2.5699,  -7.5593,  9.3005,  3.0236
H  185      water 62,  0.3345,    -0.1065,   3.5022,  -7.3532,  0.0690,  1.5118
H  186      water 62,  0.3345,     0.5851,   2.3916,  -8.1144,  0.0690,  1.5118
ANAL 2
O  187      water 63, -0.6690,    -1.5247, -11.1982,  -7.2697,  9.3005,  3.0236
H  188      water 63,  0.3345,    -0.5802, -11.0500,  -7.3172,  0.0690,  1.5118
H  189      water 63,  0.3345,    -1.6552, -12.0487,  -7.6888,  0.0690,  1.5118
ANAL 2
O  190      water 64, -0.6690,     2.1613,  -9.1672,   2.4984,  9.3005,  3.0236
H  191      water 64,  0.3345,     2.1295,  -8.2119,   2.4479,  0.0690,  1.5118
H  192      water 64,  0.3345,     1.2538,  -9.4294,   2.6528,  0.0690,  1.5118
ANAL 2
O  193      water 65, -0.6690,    -6.7304,   3.8130,   2.3606,  9.3005,  3.0236
H  194      water 65,  0.3345,    -7.6519,   4.0716,   2.3705,  0.0690,  1.5118
H  195      water 65,  0.3345,    -6.3024,   4.4286,   2.9555,  0.0690,  1.5118
ANAL 2
O  196      water 66, -0.6690,     1.9295,  -3.0572,   6.3163,  9.3005,  3.0236
H  197      water 66,  0.3345,     2.4859,  -2.7944,   7.0495,  0.0690,  1.5118
H  198      water 66,  0.3345,     1.0479,  -3.0968,   6.6868,  0.0690,  1.5118
ANAL 2
O  199      water 67, -0.6690,    -6.1294,  -4.2386,  -7.7881,  9.3005,  3.0236
H  200      water 67,  0.3345,    -5.9027,  -3.6840,  -7.0417,  0.0690,  1.5118
H  201      water 67,  0.3345,    -7.0434,  -4.0256,  -7.9761,  0.0690,  1.5118
ANAL 2
O  202      water 68, -0.6690,   -12.0077,  -2.4353,  -6.0001,  9.3005,  3.0236
H  203      water 68,  0.3345,   -11.7598,  -3.2400,  -6.4553,  0.0690,  1.5118
H  204      water 68,  0.3345,   -11.1844,  -1.9604,  -5.8871,  0.0690,  1.5118
ANAL 2
O  205      water 69, -0.6690,     3.1564,   2.3885,   1.9597,  9.3005,  3.0236
H  206      water 69,  0.3345,     2.7744,   1.6226,   1.5314,  0.0690,  1.5118
H  207      water 69,  0.3345,     2.5063,   2.6469,   2.6129,  0.0690,  1.5118
```

(continues on next page)

(continued from previous page)

```
    ANAL 2
    O  208      water 70, -0.6690,     2.4177,   8.1317,  -2.7903,  9.3005,  3.0236
    H  209      water 70,  0.3345,     3.2979,   7.7921,  -2.9516,  0.0690,  1.5118
    H  210      water 70,  0.3345,     2.4177,   8.9993,  -3.1946,  0.0690,  1.5118
    ANAL 2
    O  211      water 71, -0.6690,   -10.2768,   0.0535,  -7.3704,  9.3005,  3.0236
    H  212      water 71,  0.3345,    -9.6914,  -0.5671,  -6.9364,  0.0690,  1.5118
    H  213      water 71,  0.3345,   -10.9610,   0.2293,  -6.7246,  0.0690,  1.5118
    ANAL 2
    O  214      water 72, -0.6690,     2.1317,  -4.3290, -12.7197,  9.3005,  3.0236
    H  215      water 72,  0.3345,     2.9926,  -4.7445, -12.7694,  0.0690,  1.5118
    H  216      water 72,  0.3345,     1.6300,  -4.8877, -12.1262,  0.0690,  1.5118
    ANAL 2
    O  217      water 73, -0.6690,    -8.7835,  -6.3924,  -1.6020,  9.3005,  3.0236
    H  218      water 73,  0.3345,    -8.9261,  -5.5450,  -1.1804,  0.0690,  1.5118
    H  219      water 73,  0.3345,    -9.3978,  -6.4013,  -2.3359,  0.0690,  1.5118
    ANAL 2
    O  220      water 74, -0.6690,    -3.8469,   5.8279,  -7.3208,  9.3005,  3.0236
    H  221      water 74,  0.3345,    -4.5008,   6.2386,  -6.7552,  0.0690,  1.5118
    H  222      water 74,  0.3345,    -3.1746,   6.4993,  -7.4365,  0.0690,  1.5118
    ANAL 2
    O  223      water 75, -0.6690,     7.8985,   0.3862,  -0.4660,  9.3005,  3.0236
    H  224      water 75,  0.3345,     7.7708,   0.2351,   0.4704,  0.0690,  1.5118
    H  225      water 75,  0.3345,     7.1082,   0.8491,  -0.7440,  0.0690,  1.5118
    ANAL 2
    O  226      water 76, -0.6690,    -5.4240,  -8.2086,  -6.6502,  9.3005,  3.0236
    H  227      water 76,  0.3345,    -5.7234,  -8.8913,  -6.0498,  0.0690,  1.5118
    H  228      water 76,  0.3345,    -4.6627,  -8.5904,  -7.0868,  0.0690,  1.5118
    ANAL 2
    O  229      water 77, -0.6690,     3.2564,  -3.6039,  -9.7256,  9.3005,  3.0236
    H  230      water 77,  0.3345,     2.9998,  -4.0224,  -8.9040,  0.0690,  1.5118
    H  231      water 77,  0.3345,     4.1288,  -3.2504,  -9.5522,  0.0690,  1.5118
    ANAL 2
    O  232      water 78, -0.6690,     1.9939,  -6.2830,  -6.3501,  9.3005,  3.0236
    H  233      water 78,  0.3345,     2.9264,  -6.4297,  -6.5082,  0.0690,  1.5118
    H  234      water 78,  0.3345,     1.8631,  -5.3529,  -6.5343,  0.0690,  1.5118
    ANAL 2
    O  235      water 79, -0.6690,     2.2430,   1.1482,  -8.6661,  9.3005,  3.0236
    H  236      water 79,  0.3345,     1.6065,   0.4905,  -8.9464,  0.0690,  1.5118
    H  237      water 79,  0.3345,     2.6610,   1.4379,  -9.4769,  0.0690,  1.5118
    ANAL 2
    O  238      water 80, -0.6690,    -7.7698,   0.3830,  -8.8444,  9.3005,  3.0236
    H  239      water 80,  0.3345,    -8.0150,  -0.2954,  -9.4735,  0.0690,  1.5118
    H  240      water 80,  0.3345,    -8.5940,   0.6307,  -8.4256,  0.0690,  1.5118
    ANAL 2
    O  241      water 81, -0.6690,    -4.5307,  -0.6035, -11.1722,  9.3005,  3.0236
    H  242      water 81,  0.3345,    -3.8351,   0.0538, -11.1889,  0.0690,  1.5118
    H  243      water 81,  0.3345,    -4.2161,  -1.3000, -11.7484,  0.0690,  1.5118
    ANAL 2
    O  244      water 82, -0.6690,     1.6394, -10.2154,  -2.5027,  9.3005,  3.0236
    H  245      water 82,  0.3345,     2.3686,  -9.8106,  -2.0331,  0.0690,  1.5118
    H  246      water 82,  0.3345,     1.2256,  -9.4915,  -2.9727,  0.0690,  1.5118
    ANAL 2
    O  247      water 83, -0.6690,    -6.0760,  -1.3905,   4.7551,  9.3005,  3.0236
    H  248      water 83,  0.3345,    -6.4040,  -1.0391,   3.9274,  0.0690,  1.5118
    H  249      water 83,  0.3345,    -6.6344,  -2.1475,   4.9320,  0.0690,  1.5118
    ANAL 2
```

(continues on next page)

(continued from previous page)

```
    O  250      water 84, -0.6690,     -8.1106,    2.6426,    0.3785,   9.3005,  3.0236
    H  251      water 84,  0.3345,     -8.6162,    1.9182,    0.7470,   0.0690,  1.5118
    H  252      water 84,  0.3345,     -7.5392,    2.9239,    1.0930,   0.0690,  1.5118
    ANAL 2
    O  253      water 85, -0.6690,      2.4867,   -1.6369,  -11.7542,   9.3005,  3.0236
    H  254      water 85,  0.3345,      2.3496,   -1.6019,  -12.7008,   0.0690,  1.5118
    H  255      water 85,  0.3345,      2.5971,   -2.5681,  -11.5623,   0.0690,  1.5118
    ANAL 2
    O  256      water 86, -0.6690,      2.9006,   -7.8697,   -1.2172,   9.3005,  3.0236
    H  257      water 86,  0.3345,      3.2004,   -8.2218,   -0.3792,   0.0690,  1.5118
    H  258      water 86,  0.3345,      2.1597,   -7.3080,   -0.9897,   0.0690,  1.5118
    ANAL 2
    O  259      water 87, -0.6690,      4.0836,    3.9644,   -0.1489,   9.3005,  3.0236
    H  260      water 87,  0.3345,      3.7837,    3.9554,    0.7599,   0.0690,  1.5118
    H  261      water 87,  0.3345,      3.5478,    3.3017,   -0.5847,   0.0690,  1.5118
    ANAL 2
    O  262      water 88, -0.6690,     -0.0528,    5.3863,   -6.6165,   9.3005,  3.0236
    H  263      water 88,  0.3345,      0.7584,    5.5704,   -6.1430,   0.0690,  1.5118
    H  264      water 88,  0.3345,     -0.7337,    5.4118,   -5.9443,   0.0690,  1.5118
    ANAL 2
    O  265      water 89, -0.6690,      0.0921,   -7.0690,   -9.1123,   9.3005,  3.0236
    H  266      water 89,  0.3345,     -0.8143,   -7.0535,   -8.8052,   0.0690,  1.5118
    H  267      water 89,  0.3345,      0.6140,   -7.1527,   -8.3143,   0.0690,  1.5118
    ANAL 2
    O  268      water 90, -0.6690,     -6.8138,   -3.3648,    2.6731,   9.3005,  3.0236
    H  269      water 90,  0.3345,     -6.3319,   -4.1487,    2.9367,   0.0690,  1.5118
    H  270      water 90,  0.3345,     -6.3884,   -3.0831,    1.8633,   0.0690,  1.5118
    ANAL 2
    O  271      water 91, -0.6690,      5.4902,    1.5948,   -0.8761,   9.3005,  3.0236
    H  272      water 91,  0.3345,      5.5884,    2.4297,   -0.4186,   0.0690,  1.5118
    H  273      water 91,  0.3345,      5.5594,    1.8202,   -1.8038,   0.0690,  1.5118
    ANAL 2
    O  274      water 92, -0.6690,     -5.5811,    6.0788,   -0.6043,   9.3005,  3.0236
    H  275      water 92,  0.3345,     -5.7564,    5.2131,   -0.9729,   0.0690,  1.5118
    H  276      water 92,  0.3345,     -5.9128,    6.6907,   -1.2613,   0.0690,  1.5118
    ANAL 2
    O  277      water 93, -0.6690,      8.8753,    2.8928,   -3.7493,   9.3005,  3.0236
    H  278      water 93,  0.3345,      7.9691,    3.1073,   -3.5279,   0.0690,  1.5118
    H  279      water 93,  0.3345,      9.3254,    2.8541,   -2.9054,   0.0690,  1.5118
    ANAL 2
    O  280      water 94, -0.6690,      1.9868,    5.9861,   -4.9235,   9.3005,  3.0236
    H  281      water 94,  0.3345,      2.0426,    6.7481,   -4.3470,   0.0690,  1.5118
    H  282      water 94,  0.3345,      1.6201,    5.2949,   -4.3722,   0.0690,  1.5118
    ANAL 2
    O  283      water 95, -0.6690,     -6.5048,    1.8881,    8.2333,   9.3005,  3.0236
    H  284      water 95,  0.3345,     -6.1127,    1.0795,    7.9039,   0.0690,  1.5118
    H  285      water 95,  0.3345,     -7.3475,    1.6153,    8.5961,   0.0690,  1.5118
    ANAL 2
    O  286      water 96, -0.6690,     -3.2286,  -12.1274,   -4.3924,   9.3005,  3.0236
    H  287      water 96,  0.3345,     -2.3268,  -12.4460,   -4.3539,   0.0690,  1.5118
    H  288      water 96,  0.3345,     -3.6082,  -12.5733,   -5.1495,   0.0690,  1.5118
    ANAL 2
    O  289      water 97, -0.6690,     -8.7177,   -5.3273,   -5.8158,   9.3005,  3.0236
    H  290      water 97,  0.3345,     -8.7728,   -4.4507,   -6.1962,   0.0690,  1.5118
    H  291      water 97,  0.3345,     -7.8414,   -5.6346,   -6.0479,   0.0690,  1.5118
    ANAL 2
    O  292      water 98, -0.6690,    -10.4764,   -5.1159,   -3.3291,   9.3005,  3.0236
```

(continues on next page)

(continued from previous page)

```
    H  293      water 98,  0.3345,   -10.2133,  -5.2516,  -4.2394,  0.0690,  1.5118
    H  294      water 98,  0.3345,   -10.9495,  -4.2839,  -3.3380,  0.0690,  1.5118
    ANAL 2
    O  295      water 99, -0.6690,    -2.7377,  -6.7273,  -8.1468,  9.3005,  3.0236
    H  296      water 99,  0.3345,    -3.3330,  -6.3292,  -8.7818,  0.0690,  1.5118
    H  297      water 99,  0.3345,    -3.1152,  -6.5089,  -7.2948,  0.0690,  1.5118
    ANAL 2
    O  298      water 100, -0.6690,   -0.5217,   1.4280, -11.3190,  9.3005,  3.0236
    H  299      water 100,  0.3345,   -1.4669,   1.5705, -11.3684,  0.0690,  1.5118
    H  300      water 100,  0.3345,   -0.4260,   0.6368, -10.7889,  0.0690,  1.5118
    ANAL 2
    O  301      water 101, -0.6690,   -3.1946,   1.9538, -11.0821,  9.3005,  3.0236
    H  302      water 101,  0.3345,   -3.2495,   2.7344, -11.6332,  0.0690,  1.5118
    H  303      water 101,  0.3345,   -2.8520,   2.2733, -10.2474,  0.0690,  1.5118
    ANAL 2
    O  304      water 102, -0.6690,    4.5346,  -2.0080,   3.3549,  9.3005,  3.0236
    H  305      water 102,  0.3345,    5.0305,  -1.1959,   3.4577,  0.0690,  1.5118
    H  306      water 102,  0.3345,    5.1502,  -2.6977,   3.6031,  0.0690,  1.5118
    ANAL 2
    O  307      water 103, -0.6690,    0.8621,   6.9715,  -0.8485,  9.3005,  3.0236
    H  308      water 103,  0.3345,    1.4108,   6.8975,  -0.0677,  0.0690,  1.5118
    H  309      water 103,  0.3345,    1.4106,   7.4310,  -1.4842,  0.0690,  1.5118
    ANAL 2
    O  310      water 104, -0.6690,   -0.9493,   9.0170,  -1.1625,  9.3005,  3.0236
    H  311      water 104,  0.3345,   -0.2815,   8.3384,  -1.0636,  0.0690,  1.5118
    H  312      water 104,  0.3345,   -0.7553,   9.4228,  -2.0074,  0.0690,  1.5118
    ANAL 2
    O  313      water 105, -0.6690,   -4.0581,  -9.7236,  -3.3874,  9.3005,  3.0236
    H  314      water 105,  0.3345,   -4.4897,  -9.8495,  -2.5424,  0.0690,  1.5118
    H  315      water 105,  0.3345,   -3.7570, -10.5977,  -3.6353,  0.0690,  1.5118
    ANAL 2
    O  316      water 106, -0.6690,    5.5055,  -1.6390, -11.7299,  9.3005,  3.0236
    H  317      water 106,  0.3345,    4.6196,  -1.3194, -11.5594,  0.0690,  1.5118
    H  318      water 106,  0.3345,    5.8834,  -0.9915, -12.3250,  0.0690,  1.5118
    ANAL 2
    O  319      water 107, -0.6690,   -8.0985,   0.7307,   3.8635,  9.3005,  3.0236
    H  320      water 107,  0.3345,   -7.7375,   1.4208,   4.4200,  0.0690,  1.5118
    H  321      water 107,  0.3345,   -8.6955,   0.2490,   4.4359,  0.0690,  1.5118
    ANAL 2
    O  322      water 108, -0.6690,    4.5436,  -4.5512,  -4.7929,  9.3005,  3.0236
    H  323      water 108,  0.3345,    4.6064,  -5.1929,  -5.5003,  0.0690,  1.5118
    H  324      water 108,  0.3345,    4.5660,  -5.0746,  -3.9919,  0.0690,  1.5118
    ANAL 2
    O  325      water 109, -0.6690,    8.4875,   0.8021,  -5.4202,  9.3005,  3.0236
    H  326      water 109,  0.3345,    8.0204,   1.1624,  -6.1739,  0.0690,  1.5118
    H  327      water 109,  0.3345,    8.7010,   1.5653,  -4.8834,  0.0690,  1.5118
    ANAL 2
    O  328      water 110, -0.6690,   -7.9508,   2.4432,  -3.8256,  9.3005,  3.0236
    H  329      water 110,  0.3345,   -7.3888,   2.8104,  -4.5078,  0.0690,  1.5118
    H  330      water 110,  0.3345,   -7.5612,   1.5919,  -3.6267,  0.0690,  1.5118
    ANAL 2
    O  331      water 111, -0.6690,   -1.7889,  -8.4203,  -0.5295,  9.3005,  3.0236
    H  332      water 111,  0.3345,   -2.2353,  -7.9598,  -1.2400,  0.0690,  1.5118
    H  333      water 111,  0.3345,   -1.0075,  -8.7920,  -0.9386,  0.0690,  1.5118
    ANAL 2
    O  334      water 112, -0.6690,   -8.0755,   3.4368, -10.6340,  9.3005,  3.0236
    H  335      water 112,  0.3345,   -7.1592,   3.3044, -10.3910,  0.0690,  1.5118
```

(continues on next page)

(continued from previous page)

```
H  336      water 112,  0.3345,   -8.1904,   2.9154, -11.4284,   0.0690,  1.5118
ANAL 2
O  337      water 113, -0.6690,   -9.5459,  -4.4615,   0.4290,   9.3005,  3.0236
H  338      water 113,  0.3345,  -10.4955,  -4.5770,   0.3975,   0.0690,  1.5118
H  339      water 113,  0.3345,   -9.3158,  -4.6171,   1.3449,   0.0690,  1.5118
ANAL 2
O  340      water 114, -0.6690,   -0.8791, -13.1523,  -4.0194,   9.3005,  3.0236
H  341      water 114,  0.3345,   -0.4970, -13.8941,  -3.5505,   0.0690,  1.5118
H  342      water 114,  0.3345,   -1.1450, -13.5144,  -4.8646,   0.0690,  1.5118
ANAL 2
O  343      water 115, -0.6690,   -6.9397,   2.4469,   5.6391,   9.3005,  3.0236
H  344      water 115,  0.3345,   -6.7580,   2.1754,   6.5388,   0.0690,  1.5118
H  345      water 115,  0.3345,   -6.0782,   2.4962,   5.2250,   0.0690,  1.5118
ANAL 2
O  346      water 116, -0.6690,   -6.5700,   6.2697,  -3.7013,   9.3005,  3.0236
H  347      water 116,  0.3345,   -5.9806,   5.5298,  -3.5557,   0.0690,  1.5118
H  348      water 116,  0.3345,   -6.5995,   6.3690,  -4.6528,   0.0690,  1.5118
ANAL 2
O  349      water 117, -0.6690,   -3.6991,   7.9896,  -4.8411,   9.3005,  3.0236
H  350      water 117,  0.3345,   -4.4722,   8.3090,  -5.3064,   0.0690,  1.5118
H  351      water 117,  0.3345,   -3.7830,   8.3527,  -3.9595,   0.0690,  1.5118
ANAL 2
O  352      water 118, -0.6690,   -6.5334,   5.5889,  -6.4765,   9.3005,  3.0236
H  353      water 118,  0.3345,   -6.0589,   4.7600,  -6.5391,   0.0690,  1.5118
H  354      water 118,  0.3345,   -6.6138,   5.8880,  -7.3821,   0.0690,  1.5118
ANAL 2
O  355      water 119, -0.6690,    6.0369,  -0.5485,  -4.9230,   9.3005,  3.0236
H  356      water 119,  0.3345,    6.9450,  -0.2473,  -4.9487,   0.0690,  1.5118
H  357      water 119,  0.3345,    5.9883,  -1.2214,  -5.6020,   0.0690,  1.5118
ANAL 2
O  358      water 120, -0.6690,   -2.7088,  -0.0648,   5.8336,   9.3005,  3.0236
H  359      water 120,  0.3345,   -2.0780,   0.6506,   5.7543,   0.0690,  1.5118
H  360      water 120,  0.3345,   -2.3160,  -0.7863,   5.3424,   0.0690,  1.5118
ANAL 2
O  361      water 121, -0.6690,   -5.2890,  -0.1781,   7.0195,   9.3005,  3.0236
H  362      water 121,  0.3345,   -4.4159,   0.1300,   6.7767,   0.0690,  1.5118
H  363      water 121,  0.3345,   -5.6072,  -0.6345,   6.2406,   0.0690,  1.5118
ANAL 2
O  364      water 122, -0.6690,   -1.2183,   5.6595,   1.5861,   9.3005,  3.0236
H  365      water 122,  0.3345,   -1.0245,   5.3015,   0.7198,   0.0690,  1.5118
H  366      water 122,  0.3345,   -2.1428,   5.9036,   1.5432,   0.0690,  1.5118
ANAL 2
O  367      water 123, -0.6690,    6.8915,  -2.7911,   0.6554,   9.3005,  3.0236
H  368      water 123,  0.3345,    7.5727,  -2.4526,   0.0744,   0.0690,  1.5118
H  369      water 123,  0.3345,    7.3617,  -3.0616,   1.4439,   0.0690,  1.5118
ANAL 2
O  370      water 124, -0.6690,    4.0041,   3.6736,  -5.0404,   9.3005,  3.0236
H  371      water 124,  0.3345,    4.3507,   3.1112,  -5.7330,   0.0690,  1.5118
H  372      water 124,  0.3345,    3.5308,   4.3631,  -5.5061,   0.0690,  1.5118
ANAL 2
O  373      water 125, -0.6690,   -1.1523,   3.7231,   3.3874,   9.3005,  3.0236
H  374      water 125,  0.3345,   -1.5693,   2.9944,   2.9278,   0.0690,  1.5118
H  375      water 125,  0.3345,   -1.1728,   4.4454,   2.7598,   0.0690,  1.5118
ANAL 2
O  376      water 126, -0.6690,    4.7668,   6.6724,  -2.9741,   9.3005,  3.0236
H  377      water 126,  0.3345,    5.4307,   6.2764,  -2.4097,   0.0690,  1.5118
H  378      water 126,  0.3345,    4.2200,   5.9378,  -3.2526,   0.0690,  1.5118
```

(continues on next page)

(continued from previous page)

```
     ANAL 2
     O  379      water 127, -0.6690,   -0.7662,  -1.9711, -13.2247,  9.3005,  3.0236
     H  380      water 127,  0.3345,    0.1253,  -2.3133, -13.2901,  0.0690,  1.5118
     H  381      water 127,  0.3345,   -1.0164,  -2.1267, -12.3140,  0.0690,  1.5118
     ANAL 2
     O  382      water 128, -0.6690,   -4.0473,   6.6758,   1.6220,  9.3005,  3.0236
     H  383      water 128,  0.3345,   -4.1338,   7.6099,   1.8123,  0.0690,  1.5118
     H  384      water 128,  0.3345,   -4.6763,   6.5154,   0.9186,  0.0690,  1.5118
     ANAL 2
  end
  response
    allcomponents
    dynahyp
    hyperpol 0.0428 [Hartree]
  end
  Basis Core=None
  symmetry NOSYM
  Relativity
    Level None
  End
EndEngine

eor
```

Example: DRF: scripting tool

Download DIMQM_DRF_plams.run

```
#! /bin/sh

# The Discrete Reaction Field (DRF) method is a polarizable QM/MM method. For
# each type of solvent molecule,the DRF method requires two parameters, atomic
# polarizabilities and atomic charges. The main program reads the QM and DRF
# regions, splits the DRF region into unique molecules, computes the atomic
# charges of a single DRF molecule for each unique type, assigns the atomic
# charges and atomic polarizabilities for all the DRF molecules and defines the
# QM_DRF settings. This simple script works for initial configurations that come
# from a MD  simulation, for example.

# In this example a python script is used for DRF calculations, that uses PLAMS
# (Python Library for Automating Molecular Simulation). PLAMS takes care of
# input preparation, job execution, file management and output processing, see
# also the PLAMS documentation in the Scripting manual. The DRF python script
# has default settings which can be easily modified if this is required. In the
# simplest case the user just needs to provide the QM and DRF regions (in xyz
# format). Typically the user may also want to choose the functional, basis set
# and numerical quality for the QM and DRF region. By default, atomic charges
# that are used for the DRF are computed with LDA functional, DZP basis set and
# normal numerical quality. Also by default, atomic polarizabilities (computed
# with the Thole's model) are taken from an inner database for the elements
# H, C, N, O, F, S, Cl, Br, and I [J.Phys.Chem.A 102 (1998) 2399].
# For the other elements up to U (Z=92) the inner database also include atomic
# polarizabilities, however, those are not optimized and not tested,
# and no reference paper exists.
#
```

(continues on next page)

(continued from previous page)

```
# The main program defines a series of functions
# for general settings for a DRF calculation, reading QM and DRF regions,
# classifying DRF molecules of the same type (first by formula and then for
# configurations for a given formula), computing, for a given settings, the
# charges for a single molecule of unique DRF type and assigning the atomic
# charges for the rest of molecules that belong to a given DRF type (preserving
# atom orders).

# The DRF python script DIMQM_DRF_plams.py users will probably want to change or
# extend:

# ================================================================================
# from DIMQM_DRF_plams_module import full_DIMQM_DRF

# qm_region_filename = 'qm.xyz'
# drf_region_filename = 'drf.xyz'

# settings.input.AMS.Task = 'SinglePoint'
# settings.input.ADF.Basis.Type = 'TZP'
# settings.input.ADF.XC['_1'] = 'GGA BLYP'
# settings.input.ADF.NumericalQuality = 'Normal'

# full_DIMQM_DRF(qm_region_filename, drf_region_filename,
#                settings, drf_charges_type='MDC-q charges',
#                drf_fragment_settings=None)
# ================================================================================


# This DRF python script DIMQM_DRF_plams.py imports DIMQM_DRF_plams_module.py,
# which contains a script full_DIMQM_DRF that splits the DRF region into unique
# molecules, computes the atomic charges of a single DRF molecule for each
# unique type, assigns the atomic charges and atomic polarizabilities for all
# the DRF molecules and defines the QM_DRF settings.

# The run script DIMQM_DRF_plams.run contains an example, how one could use the
# DIMQM_DRF_plams.py script. In this simple example both the QM and DRF region
# consists of a Formic acid molecule.

cp $AMSHOME/examples/adf/DIMQM_DRF_plams/formic_acid_qm.xyz qm.xyz
cp $AMSHOME/examples/adf/DIMQM_DRF_plams/formic_acid_drf.xyz drf.xyz
cp $AMSHOME/examples/adf/DIMQM_DRF_plams/DIMQM_DRF_plams_module.py .
cp $AMSHOME/examples/adf/DIMQM_DRF_plams/DIMQM_DRF_plams.py .

$AMSBIN/plams DIMQM_DRF_plams.py

# Every time you run a PLAMS script, a uniquely named working directory is
# created (plams.*****). This folder will contain one subdirectory per job. Each
# job directory contains the job's input and results files. In this case there
# will be 1 directory for the ADF calculation of the atomic charges that are
# used for DRF. Next there is a directory which calculates the full system.
```

Example: DRF2: Polarizability N2 on Ag68 + H2O

Download DIMQM_DRF2.run

```
#! /bin/sh


# == Expert option ==

# This example shows how to run DRF2, which is a combination of the
# Discrete Reaction Field and the PIM method of DIM/QM. This allows
# for treatment of systems such as a solvated nanopartcle using
# DRF with gaussian screening for the solvent, and PIM for the
# nanoparticle. Allows for geometry optimizations in "DRF" as well,
# as long as done with gaussian screening.

#######################
# STATIC POLARIZABILITY
#######################

AMS_JOBNAME=first $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title DRF2 Static Polarizability, N2 on Ag68 + H2O
  aoresponse
    alda
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     char 0.000000
     SUBEND
     O
     rad 0.90
     char 0.6
     STATIC
     SUBEND
     H
     rad 0.45
     char -0.3
     STATIC
     SUBEND
     XYZ
     71
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
```

(continues on next page)

(continued from previous page)

```
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
     Ag      4.3708278508    0.0708740258   -2.8525785565
     Ag     -4.0450216512   -1.6573690296   -2.8525892410
     Ag      2.5125751829    2.2311280445   -7.5634444475
     Ag      0.6759414795   -3.2915128984   -7.5634430672
     Ag      0.3339235029   -1.6260239766   -9.9892067983
     Ag      1.0324285615   -5.0274542660   -5.1718100987
     Ag     -2.2484187072   -3.9433040680   -7.6050830302
     Ag      3.6204971713   -2.7380910382   -7.6050765656
     Ag     -2.4569103660   -2.1817169842   -9.8785349524
     Ag     -1.7752961456   -5.4191572521   -5.2192070180
     Ag      3.7672926986   -4.2809622136   -5.2191965533
     Ag      3.1179002480   -1.0369024811   -9.8785348598
     Ag     -1.9491576017    2.1440839896   -5.1959915042
     Ag      2.5822650861    2.2930127465   -2.8061239177
     Ag     -0.3326165571    1.6197266638   -2.7987957984
     Ag      0.9464626101    2.7387205045   -5.1959886567
     Ag     -0.5598334280   -4.3819335958   -9.8593641366
     Ag     -0.2204924851   -6.0299810786   -7.5512447020
     Ag      2.5790329655   -5.4550834512   -7.5512427315
     Ag      2.2414383913   -3.8066764388   -9.8593649949
     Ag     -2.2907828197    3.9188443502   -7.6050826773
     Ag      0.5610225957    4.5044771077   -7.6050819796
     Ag     -4.8701069022    1.6196298364   -5.1718125544
     Ag     -3.2769317991    1.0897966386   -2.8061340012
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
```

(continues on next page)

(continued from previous page)

```
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     O      -4.069283       -3.055450        0.358275
     H      -5.031565       -3.283620        0.404829
     H      -3.843952       -4.017650        0.299391
     SUBEND
  end
  dimqm
     DRF2
     ALGORITHM BRUTE
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

####################################
# FREQUENCY DEPENDENT POLARIZABILITY
####################################

AMS_JOBNAME=second $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title DRF2 Frequency-Dependent Polarizability, N2 on Ag68 + H2O
  aoresponse
    alda
    frequencies 3.55 [eV]
    lifetime 0.0036749
  end
  beckegrid
    quality good
```

(continues on next page)

(continued from previous page)

```
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     char 0.000000
     SUBEND
     O
     rad 0.90
     char 0.6
     STATIC
     SUBEND
     H
     rad 0.45
     char -0.3
     STATIC
     SUBEND
     XYZ
     71
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
     Ag      4.3708278508    0.0708740258   -2.8525785565
     Ag     -4.0450216512   -1.6573690296   -2.8525892410
     Ag      2.5125751829    2.2311280445   -7.5634444475
     Ag      0.6759414795   -3.2915128984   -7.5634430672
     Ag      0.3339235029   -1.6260239766   -9.9892067983
     Ag      1.0324285615   -5.0274542660   -5.1718100987
     Ag     -2.2484187072   -3.9433040680   -7.6050830302
     Ag      3.6204971713   -2.7380910382   -7.6050765656
     Ag     -2.4569103660   -2.1817169842   -9.8785349524
     Ag     -1.7752961456   -5.4191572521   -5.2192070180
     Ag      3.7672926986   -4.2809622136   -5.2191965533
     Ag      3.1179002480   -1.0369024811   -9.8785348598
     Ag     -1.9491576017    2.1440839896   -5.1959915042
     Ag      2.5822650861    2.2930127465   -2.8061239177
     Ag     -0.3326165571    1.6197266638   -2.7987957984
     Ag      0.9464626101    2.7387205045   -5.1959886567
     Ag     -0.5598334280   -4.3819335958   -9.8593641366
```

(continues on next page)

(continued from previous page)

```
     Ag     -0.2204924851   -6.0299810786   -7.5512447020
     Ag      2.5790329655   -5.4550834512   -7.5512427315
     Ag      2.2414383913   -3.8066764388   -9.8593649949
     Ag     -2.2907828197    3.9188443502   -7.6050826773
     Ag      0.5610225957    4.5044771077   -7.6050819796
     Ag     -4.8701069022    1.6196298364   -5.1718125544
     Ag     -3.2769317991    1.0897966386   -2.8061340012
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     O      -4.069283       -3.055450        0.358275
     H      -5.031565       -3.283620        0.404829
     H      -3.843952       -4.017650        0.299391
     SUBEND
  end
  dimqm
     DRF2
     ALGORITHM BRUTE
     FREQUENCY
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

#######################################################
```

(continues on next page)

(continued from previous page)

```
# FREQUENCY DEPENDENT POLARIZABILITY WITH LOCAL FIELDS
######################################################

AMS_JOBNAME=third $AMSBIN/ams <<eor
System
  atoms
     N       -0.000006    -0.000060    2.302488
     N        0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title DRF2 Frequency-Dependent Polarizability w/ Local Fields, N2 on Ag68 + H2O
  aoresponse
    alda
    frequencies 3.55 [eV]
    lifetime 0.0036749
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     char 0.000000
     SUBEND
     O
     rad 0.90
     char 0.6
     STATIC
     SUBEND
     H
     rad 0.45
     char -0.3
     STATIC
     SUBEND
     XYZ
     71
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
```

(continues on next page)

(continued from previous page)

```
    Ag     -2.8450227342   -0.5497019461   -5.1959955618
    Ag      2.8314207151    0.6159878635   -5.1959875699
    Ag     -3.1884901403    1.0603809604   -7.5634515357
    Ag     -1.2364094731   -1.0979217456   -2.7987978362
    Ag      1.5690420034   -0.5218070047   -2.7987993856
    Ag      4.3708278508    0.0708740258   -2.8525785565
    Ag     -4.0450216512   -1.6573690296   -2.8525892410
    Ag      2.5125751829    2.2311280445   -7.5634444475
    Ag      0.6759414795   -3.2915128984   -7.5634430672
    Ag      0.3339235029   -1.6260239766   -9.9892067983
    Ag      1.0324285615   -5.0274542660   -5.1718100987
    Ag     -2.2484187072   -3.9433040680   -7.6050830302
    Ag      3.6204971713   -2.7380910382   -7.6050765656
    Ag     -2.4569103660   -2.1817169842   -9.8785349524
    Ag     -1.7752961456   -5.4191572521   -5.2192070180
    Ag      3.7672926986   -4.2809622136   -5.2191965533
    Ag      3.1179002480   -1.0369024811   -9.8785348598
    Ag     -1.9491576017    2.1440839896   -5.1959915042
    Ag      2.5822650861    2.2930127465   -2.8061239177
    Ag     -0.3326165571    1.6197266638   -2.7987957984
    Ag      0.9464626101    2.7387205045   -5.1959886567
    Ag     -0.5598334280   -4.3819335958   -9.8593641366
    Ag     -0.2204924851   -6.0299810786   -7.5512447020
    Ag      2.5790329655   -5.4550834512   -7.5512427315
    Ag      2.2414383913   -3.8066764388   -9.8593649949
    Ag     -2.2907828197    3.9188443502   -7.6050826773
    Ag      0.5610225957    4.5044771077   -7.6050819796
    Ag     -4.8701069022    1.6196298364   -5.1718125544
    Ag     -3.2769317991    1.0897966386   -2.8061340012
    Ag     -0.6609450333    3.2186205924   -9.8785392784
    Ag     -5.5910570566   -1.1220840757   -5.2192067211
    Ag     -2.2467815447    3.7498049589   -2.8525803584
    Ag     -0.9993904347    4.8666585643   -5.2034637574
    Ag     -3.8054729020    4.2470446290   -5.2192106249
    Ag     -3.5149221622    2.6758039407   -9.8593694452
    Ag     -4.4173760657   -0.0377981351   -9.8593712497
    Ag     -6.0137475245    0.4940463872   -7.5512522368
    Ag     -5.1118586671    3.2059431066   -7.5512501894
    Ag      0.5871922586    4.3317735603   -2.8525794711
    Ag      1.8237836111    5.4030432621   -5.2192000459
    Ag      3.8377073288    3.4078339914   -5.1718059272
    Ag      5.5807980727    1.1721205307   -5.2191937875
    Ag      3.4347425354    4.9610463596   -7.5512390021
    Ag      4.0747942310    1.7061247432   -9.8593564188
    Ag      5.3323750373    2.8240364209   -7.5512331190
    Ag      2.1759755136    3.8444690680   -9.8593641559
    Ag     -1.8817544351    2.1073913325   -0.4836595729
    Ag      0.8989635719    2.6784284394   -0.4836553279
    Ag      0.0000000000    0.0000000000    0.0000000000
    Ag      2.7659350940    0.5759533343   -0.4836497308
    Ag     -2.7690678527   -0.5606913044   -0.4836604335
    Ag     -0.8841719671   -2.6833451220   -0.4836610364
    Ag      1.8701101821   -2.1177433283   -0.4836551986
    O      -4.069283       -3.055450        0.358275
    H      -5.031565       -3.283620        0.404829
    H      -3.843952       -4.017650        0.299391
    SUBEND
```

(continues on next page)

(continued from previous page)

```
  end
  dimqm
     DRF2
     ALGORITHM DIRECT
     FREQUENCY
     LOCALFIELD
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: CPIM: excitation energies N2 on silver cluster Ag68

`Download DIMQM_CPIM_Excitations.run`

```
#! /bin/sh


# == Expert option ==

# This example demonstrates how to calculate excitation energies with the
# Capacitance Polarizability Interaction Model (CPIM). Key EXCITATIONS is used.


$AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title DIM/QM Excitations Test, N2 on Ag68
  beckegrid
    quality good
  end
  dimpar
     Ag
     pol 49.9843
     cap 2.7529
     SUBEND
```

(continues on next page)

(continued from previous page)

```
XYZ
68
Ag     -1.2626801699   -1.1212553334   -7.5641602542
Ag     -1.5751210922    0.5238327185   -9.9892089940
Ag     -4.1814870442   -1.7663886735   -7.6050834452
Ag     -0.8822430837   -2.7600722440   -5.1959892962
Ag     -2.1240266195   -3.8206731623   -2.8525807753
Ag      3.4578336576   -2.6744062081   -2.8525742581
Ag     -3.7149452090   -3.2988294197   -5.2034646499
Ag      1.8985756093   -2.1890139391   -5.1959881717
Ag      0.6946811325   -3.3828052555   -2.8061234013
Ag      4.7143519584   -1.5678241605   -5.2034549930
Ag      1.6023915608   -0.5328957432   -7.5641613132
Ag      1.2412345786    1.1021928389   -9.9892023582
Ag      4.5392237030    0.0244593089   -7.6050682929
Ag      0.0000106997    0.0000013921   -5.1462708001
Ag     -0.3396832259    1.6541471619   -7.5641597129
Ag     -2.8450227342   -0.5497019461   -5.1959955618
Ag      2.8314207151    0.6159878635   -5.1959875699
Ag     -3.1884901403    1.0603809604   -7.5634515357
Ag     -1.2364094731   -1.0979217456   -2.7987978362
Ag      1.5690420034   -0.5218070047   -2.7987993856
Ag      4.3708278508    0.0708740258   -2.8525785565
Ag     -4.0450216512   -1.6573690296   -2.8525892410
Ag      2.5125751829    2.2311280445   -7.5634444475
Ag      0.6759414795   -3.2915128984   -7.5634430672
Ag      0.3339235029   -1.6260239766   -9.9892067983
Ag      1.0324285615   -5.0274542660   -5.1718100987
Ag     -2.2484187072   -3.9433040680   -7.6050830302
Ag      3.6204971713   -2.7380910382   -7.6050765656
Ag     -2.4569103660   -2.1817169842   -9.8785349524
Ag     -1.7752961456   -5.4191572521   -5.2192070180
Ag      3.7672926986   -4.2809622136   -5.2191965533
Ag      3.1179002480   -1.0369024811   -9.8785348598
Ag     -1.9491576017    2.1440839896   -5.1959915042
Ag      2.5822650861    2.2930127465   -2.8061239177
Ag     -0.3326165571    1.6197266638   -2.7987957984
Ag      0.9464626101    2.7387205045   -5.1959886567
Ag     -0.5598334280   -4.3819335958   -9.8593641366
Ag     -0.2204924851   -6.0299810786   -7.5512447020
Ag      2.5790329655   -5.4550834512   -7.5512427315
Ag      2.2414383913   -3.8066764388   -9.8593649949
Ag     -2.2907828197    3.9188443502   -7.6050826773
Ag      0.5610225957    4.5044771077   -7.6050819796
Ag     -4.8701069022    1.6196298364   -5.1718125544
Ag     -3.2769317991    1.0897966386   -2.8061340012
Ag     -0.6609450333    3.2186205924   -9.8785392784
Ag     -5.5910570566   -1.1220840757   -5.2192067211
Ag     -2.2467815447    3.7498049589   -2.8525803584
Ag     -0.9993904347    4.8666585643   -5.2034637574
Ag     -3.8054729020    4.2470446290   -5.2192106249
Ag     -3.5149221622    2.6758039407   -9.8593694452
Ag     -4.4173760657   -0.0377981351   -9.8593712497
Ag     -6.0137475245    0.4940463872   -7.5512522368
Ag     -5.1118586671    3.2059431066   -7.5512501894
Ag      0.5871922586    4.3317735603   -2.8525794711
Ag      1.8237836111    5.4030432621   -5.2192000459
```

(continues on next page)

(continued from previous page)

```
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
  end
  dimqm
     CPIM
  end
  excitations
    allowed
    davidson
    end
    lowest 10
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: CPIM: polarizability N2 on silver cluster Ag68

Download DIMQM_JACOBI.run

```
#! /bin/sh


# == Expert option ==

# This example demonstrates how to do calculate static and frequency dependent
# polarizabilities with the Polarizability Interaction Model (PIM). The example
# consists of 3 calculations: static polarizability, frequency dependent, and
# with inclusion of effect of the local field.

# When the molecule interacts with a metal nanoparticle, there are two types of
# interactions: the image field and the local field. The image field is caused
```

(continues on next page)

(continued from previous page)

```
# by the dipoles induced into the nanoparticle by the molecule's electron
# density. This is always taken into account in a DIM/QM calculation. The local
# field arises by direct interactions of the nanoparticle with an external
# field. Addition of the LOCALFIELD key causes the DIM/QM calculation to include
# this effect, but by default this is not included in a DIM/QM calculation.


#######################
# STATIC POLARIZABILITY
#######################

AMS_JOBNAME=first $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Static Polarizability, N2 on Ag68
  aoresponse
    alda
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
```

(continues on next page)

(continued from previous page)

```
     Ag      4.3708278508    0.0708740258   -2.8525785565
     Ag     -4.0450216512   -1.6573690296   -2.8525892410
     Ag      2.5125751829    2.2311280445   -7.5634444475
     Ag      0.6759414795   -3.2915128984   -7.5634430672
     Ag      0.3339235029   -1.6260239766   -9.9892067983
     Ag      1.0324285615   -5.0274542660   -5.1718100987
     Ag     -2.2484187072   -3.9433040680   -7.6050830302
     Ag      3.6204971713   -2.7380910382   -7.6050765656
     Ag     -2.4569103660   -2.1817169842   -9.8785349524
     Ag     -1.7752961456   -5.4191572521   -5.2192070180
     Ag      3.7672926986   -4.2809622136   -5.2191965533
     Ag      3.1179002480   -1.0369024811   -9.8785348598
     Ag     -1.9491576017    2.1440839896   -5.1959915042
     Ag      2.5822650861    2.2930127465   -2.8061239177
     Ag     -0.3326165571    1.6197266638   -2.7987957984
     Ag      0.9464626101    2.7387205045   -5.1959886567
     Ag     -0.5598334280   -4.3819335958   -9.8593641366
     Ag     -0.2204924851   -6.0299810786   -7.5512447020
     Ag      2.5790329655   -5.4550834512   -7.5512427315
     Ag      2.2414383913   -3.8066764388   -9.8593649949
     Ag     -2.2907828197    3.9188443502   -7.6050826773
     Ag      0.5610225957    4.5044771077   -7.6050819796
     Ag     -4.8701069022    1.6196298364   -5.1718125544
     Ag     -3.2769317991    1.0897966386   -2.8061340012
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
  end
  dimqm
     PIM
     JACOBI
     ALGORITHM BRUTE
  end
  basis
    core none
```

(continues on next page)

(continued from previous page)

```
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

####################################
# FREQUENCY DEPENDENT POLARIZABILITY
####################################

AMS_JOBNAME=second $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Frequency-Dependent Polarizability, N2 on Ag68
  aoresponse
    alda
    frequencies 3.55 [eV]
    lifetime 0.0036749
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
```

(continues on next page)

(continued from previous page)

```
    Ag      4.5392237030    0.0244593089   -7.6050682929
    Ag      0.0000106997    0.0000013921   -5.1462708001
    Ag     -0.3396832259    1.6541471619   -7.5641597129
    Ag     -2.8450227342   -0.5497019461   -5.1959955618
    Ag      2.8314207151    0.6159878635   -5.1959875699
    Ag     -3.1884901403    1.0603809604   -7.5634515357
    Ag     -1.2364094731   -1.0979217456   -2.7987978362
    Ag      1.5690420034   -0.5218070047   -2.7987993856
    Ag      4.3708278508    0.0708740258   -2.8525785565
    Ag     -4.0450216512   -1.6573690296   -2.8525892410
    Ag      2.5125751829    2.2311280445   -7.5634444475
    Ag      0.6759414795   -3.2915128984   -7.5634430672
    Ag      0.3339235029   -1.6260239766   -9.9892067983
    Ag      1.0324285615   -5.0274542660   -5.1718100987
    Ag     -2.2484187072   -3.9433040680   -7.6050830302
    Ag      3.6204971713   -2.7380910382   -7.6050765656
    Ag     -2.4569103660   -2.1817169842   -9.8785349524
    Ag     -1.7752961456   -5.4191572521   -5.2192070180
    Ag      3.7672926986   -4.2809622136   -5.2191965533
    Ag      3.1179002480   -1.0369024811   -9.8785348598
    Ag     -1.9491576017    2.1440839896   -5.1959915042
    Ag      2.5822650861    2.2930127465   -2.8061239177
    Ag     -0.3326165571    1.6197266638   -2.7987957984
    Ag      0.9464626101    2.7387205045   -5.1959886567
    Ag     -0.5598334280   -4.3819335958   -9.8593641366
    Ag     -0.2204924851   -6.0299810786   -7.5512447020
    Ag      2.5790329655   -5.4550834512   -7.5512427315
    Ag      2.2414383913   -3.8066764388   -9.8593649949
    Ag     -2.2907828197    3.9188443502   -7.6050826773
    Ag      0.5610225957    4.5044771077   -7.6050819796
    Ag     -4.8701069022    1.6196298364   -5.1718125544
    Ag     -3.2769317991    1.0897966386   -2.8061340012
    Ag     -0.6609450333    3.2186205924   -9.8785392784
    Ag     -5.5910570566   -1.1220840757   -5.2192067211
    Ag     -2.2467815447    3.7498049589   -2.8525803584
    Ag     -0.9993904347    4.8666585643   -5.2034637574
    Ag     -3.8054729020    4.2470446290   -5.2192106249
    Ag     -3.5149221622    2.6758039407   -9.8593694452
    Ag     -4.4173760657   -0.0377981351   -9.8593712497
    Ag     -6.0137475245    0.4940463872   -7.5512522368
    Ag     -5.1118586671    3.2059431066   -7.5512501894
    Ag      0.5871922586    4.3317735603   -2.8525794711
    Ag      1.8237836111    5.4030432621   -5.2192000459
    Ag      3.8377073288    3.4078339914   -5.1718059272
    Ag      5.5807980727    1.1721205307   -5.2191937875
    Ag      3.4347425354    4.9610463596   -7.5512390021
    Ag      4.0747942310    1.7061247432   -9.8593564188
    Ag      5.3323750373    2.8240364209   -7.5512331190
    Ag      2.1759755136    3.8444690680   -9.8593641559
    Ag     -1.8817544351    2.1073913325   -0.4836595729
    Ag      0.8989635719    2.6784284394   -0.4836553279
    Ag      0.0000000000    0.0000000000    0.0000000000
    Ag      2.7659350940    0.5759533343   -0.4836497308
    Ag     -2.7690678527   -0.5606913044   -0.4836604335
    Ag     -0.8841719671   -2.6833451220   -0.4836610364
    Ag      1.8701101821   -2.1177433283   -0.4836551986
    SUBEND
```

(continues on next page)

(continued from previous page)

```
  end
  dimqm
     PIM
     JACOBI
     ALGORITHM BRUTE
     FREQUENCY
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

######################################################
# FREQUENCY DEPENDENT POLARIZABILITY WITH LOCAL FIELDS
######################################################

AMS_JOBNAME=third $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Frequency-Dependent Polarizability w/ Local Fields, N2 on Ag68
  aoresponse
    alda
    frequencies 3.55 [eV]
    lifetime 0.0036749
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
```

(continues on next page)

(continued from previous page)

```
     Ag      -0.8822430837   -2.7600722440   -5.1959892962
     Ag      -2.1240266195   -3.8206731623   -2.8525807753
     Ag       3.4578336576   -2.6744062081   -2.8525742581
     Ag      -3.7149452090   -3.2988294197   -5.2034646499
     Ag       1.8985756093   -2.1890139391   -5.1959881717
     Ag       0.6946811325   -3.3828052555   -2.8061234013
     Ag       4.7143519584   -1.5678241605   -5.2034549930
     Ag       1.6023915608   -0.5328957432   -7.5641613132
     Ag       1.2412345786    1.1021928389   -9.9892023582
     Ag       4.5392237030    0.0244593089   -7.6050682929
     Ag       0.0000106997    0.0000013921   -5.1462708001
     Ag      -0.3396832259    1.6541471619   -7.5641597129
     Ag      -2.8450227342   -0.5497019461   -5.1959955618
     Ag       2.8314207151    0.6159878635   -5.1959875699
     Ag      -3.1884901403    1.0603809604   -7.5634515357
     Ag      -1.2364094731   -1.0979217456   -2.7987978362
     Ag       1.5690420034   -0.5218070047   -2.7987993856
     Ag       4.3708278508    0.0708740258   -2.8525785565
     Ag      -4.0450216512   -1.6573690296   -2.8525892410
     Ag       2.5125751829    2.2311280445   -7.5634444475
     Ag       0.6759414795   -3.2915128984   -7.5634430672
     Ag       0.3339235029   -1.6260239766   -9.9892067983
     Ag       1.0324285615   -5.0274542660   -5.1718100987
     Ag      -2.2484187072   -3.9433040680   -7.6050830302
     Ag       3.6204971713   -2.7380910382   -7.6050765656
     Ag      -2.4569103660   -2.1817169842   -9.8785349524
     Ag      -1.7752961456   -5.4191572521   -5.2192070180
     Ag       3.7672926986   -4.2809622136   -5.2191965533
     Ag       3.1179002480   -1.0369024811   -9.8785348598
     Ag      -1.9491576017    2.1440839896   -5.1959915042
     Ag       2.5822650861    2.2930127465   -2.8061239177
     Ag      -0.3326165571    1.6197266638   -2.7987957984
     Ag       0.9464626101    2.7387205045   -5.1959886567
     Ag      -0.5598334280   -4.3819335958   -9.8593641366
     Ag      -0.2204924851   -6.0299810786   -7.5512447020
     Ag       2.5790329655   -5.4550834512   -7.5512427315
     Ag       2.2414383913   -3.8066764388   -9.8593649949
     Ag      -2.2907828197    3.9188443502   -7.6050826773
     Ag       0.5610225957    4.5044771077   -7.6050819796
     Ag      -4.8701069022    1.6196298364   -5.1718125544
     Ag      -3.2769317991    1.0897966386   -2.8061340012
     Ag      -0.6609450333    3.2186205924   -9.8785392784
     Ag      -5.5910570566   -1.1220840757   -5.2192067211
     Ag      -2.2467815447    3.7498049589   -2.8525803584
     Ag      -0.9993904347    4.8666585643   -5.2034637574
     Ag      -3.8054729020    4.2470446290   -5.2192106249
     Ag      -3.5149221622    2.6758039407   -9.8593694452
     Ag      -4.4173760657   -0.0377981351   -9.8593712497
     Ag      -6.0137475245    0.4940463872   -7.5512522368
     Ag      -5.1118586671    3.2059431066   -7.5512501894
     Ag       0.5871922586    4.3317735603   -2.8525794711
     Ag       1.8237836111    5.4030432621   -5.2192000459
     Ag       3.8377073288    3.4078339914   -5.1718059272
     Ag       5.5807980727    1.1721205307   -5.2191937875
     Ag       3.4347425354    4.9610463596   -7.5512390021
     Ag       4.0747942310    1.7061247432   -9.8593564188
     Ag       5.3323750373    2.8240364209   -7.5512331190
```

(continues on next page)

(continued from previous page)

```
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
  end
  dimqm
     PIM
     JACOBI
     ALGORITHM DIRECT
     FREQUENCY
     LOCALFIELD
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: PIM: H2O on Ag2689

Download DIMQM_LOCAL.run

```
#! /bin/sh


$AMSBIN/ams <<eor
System
  atoms
     O         0.016560    0.127604    0.000000
     H         0.650009    0.566164    0.000000
     H        -0.666568    0.484010    0.000000
  end
end

Task SinglePoint

Engine ADF
  title ['Local Dim Test']
  allpoints
  beckegrid
    quality good
```

(continues on next page)

(continued from previous page)

```
  end
  dimpar
     Ag
     rad=1.4445
     SUBEND
     XYZ
     $AMSHOME/examples/adf/DIMQM_LOCAL/ag2869.xyz
     SUBEND
  end
  dimqm
     PIM
     LOCALDIM
     debug
     PRINTLJPAR
     niter 2000
     CutOffDist 50.0
  end
  basis
    core None
    type TZP
  end
  symmetry nosym
  xc
    model SAOP
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: PIM: Polarizability with local fields

Download DIMQM_READLCLFLD.run

```
#!/bin/sh


#####################################################
# FREQUENCY DEPENDENT POLARIZABILITY WITH LOCAL FIELDS
#####################################################

$AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Frequency-Dependent Polarizability w/ Local Fields, N2 on Ag68
```

(continues on next page)

(continued from previous page)

```
aoresponse
  alda
  frequencies 3.55 [eV]
  lifetime 0.0036749
end
beckegrid
  quality good
end
dimpar
   Ag
   rad 1.4445
   drude 9.2 0.17 eV
   bound 5.0
   SUBEND
   XYZ
   68
   Ag     -1.2626801699   -1.1212553334   -7.5641602542
   Ag     -1.5751210922    0.5238327185   -9.9892089940
   Ag     -4.1814870442   -1.7663886735   -7.6050834452
   Ag     -0.8822430837   -2.7600722440   -5.1959892962
   Ag     -2.1240266195   -3.8206731623   -2.8525807753
   Ag      3.4578336576   -2.6744062081   -2.8525742581
   Ag     -3.7149452090   -3.2988294197   -5.2034646499
   Ag      1.8985756093   -2.1890139391   -5.1959881717
   Ag      0.6946811325   -3.3828052555   -2.8061234013
   Ag      4.7143519584   -1.5678241605   -5.2034549930
   Ag      1.6023915608   -0.5328957432   -7.5641613132
   Ag      1.2412345786    1.1021928389   -9.9892023582
   Ag      4.5392237030    0.0244593089   -7.6050682929
   Ag      0.0000106997    0.0000013921   -5.1462708001
   Ag     -0.3396832259    1.6541471619   -7.5641597129
   Ag     -2.8450227342   -0.5497019461   -5.1959955618
   Ag      2.8314207151    0.6159878635   -5.1959875699
   Ag     -3.1884901403    1.0603809604   -7.5634515357
   Ag     -1.2364094731   -1.0979217456   -2.7987978362
   Ag      1.5690420034   -0.5218070047   -2.7987993856
   Ag      4.3708278508    0.0708740258   -2.8525785565
   Ag     -4.0450216512   -1.6573690296   -2.8525892410
   Ag      2.5125751829    2.2311280445   -7.5634444475
   Ag      0.6759414795   -3.2915128984   -7.5634430672
   Ag      0.3339235029   -1.6260239766   -9.9892067983
   Ag      1.0324285615   -5.0274542660   -5.1718100987
   Ag     -2.2484187072   -3.9433040680   -7.6050830302
   Ag      3.6204971713   -2.7380910382   -7.6050765656
   Ag     -2.4569103660   -2.1817169842   -9.8785349524
   Ag     -1.7752961456   -5.4191572521   -5.2192070180
   Ag      3.7672926986   -4.2809622136   -5.2191965533
   Ag      3.1179002480   -1.0369024811   -9.8785348598
   Ag     -1.9491576017    2.1440839896   -5.1959915042
   Ag      2.5822650861    2.2930127465   -2.8061239177
   Ag     -0.3326165571    1.6197266638   -2.7987957984
   Ag      0.9464626101    2.7387205045   -5.1959886567
   Ag     -0.5598334280   -4.3819335958   -9.8593641366
   Ag     -0.2204924851   -6.0299810786   -7.5512447020
   Ag      2.5790329655   -5.4550834512   -7.5512427315
   Ag      2.2414383913   -3.8066764388   -9.8593649949
   Ag     -2.2907828197    3.9188443502   -7.6050826773
```

(continues on next page)

(continued from previous page)

```
     Ag      0.5610225957    4.5044771077   -7.6050819796
     Ag     -4.8701069022    1.6196298364   -5.1718125544
     Ag     -3.2769317991    1.0897966386   -2.8061340012
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
     LCLFLD
     $AMSHOME/examples/adf/DIMQM_READLCLFLD/lclfld.txt
     SUBEND
  end
  dimqm
     PIM
     FREQUENCY
     LocalField RESP
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: PIM: optimization N2 on silver cluster Ag68

Download DIMQM_Gradients.run

```
#! /bin/sh

# == Expert option ==

# This example demonstrates how to do a geometry optimization with the
# Polarizability Interaction Model (PIM).


$AMSBIN/ams <<eor
System
  atoms
     N       0.0000000000    0.0000000000    2.5000000000
     N       0.0000000000    0.0000000000    3.5000000000
  end
end

Task SinglePoint

Properties
    Gradients Yes
End

Engine ADF
  title DIM/QM Gradient Test, N2 on Ag68
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad=1.4445
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
     Ag      4.3708278508    0.0708740258   -2.8525785565
```

(continues on next page)

(continued from previous page)

```
   Ag     -4.0450216512   -1.6573690296   -2.8525892410
   Ag      2.5125751829    2.2311280445   -7.5634444475
   Ag      0.6759414795   -3.2915128984   -7.5634430672
   Ag      0.3339235029   -1.6260239766   -9.9892067983
   Ag      1.0324285615   -5.0274542660   -5.1718100987
   Ag     -2.2484187072   -3.9433040680   -7.6050830302
   Ag      3.6204971713   -2.7380910382   -7.6050765656
   Ag     -2.4569103660   -2.1817169842   -9.8785349524
   Ag     -1.7752961456   -5.4191572521   -5.2192070180
   Ag      3.7672926986   -4.2809622136   -5.2191965533
   Ag      3.1179002480   -1.0369024811   -9.8785348598
   Ag     -1.9491576017    2.1440839896   -5.1959915042
   Ag      2.5822650861    2.2930127465   -2.8061239177
   Ag     -0.3326165571    1.6197266638   -2.7987957984
   Ag      0.9464626101    2.7387205045   -5.1959886567
   Ag     -0.5598334280   -4.3819335958   -9.8593641366
   Ag     -0.2204924851   -6.0299810786   -7.5512447020
   Ag      2.5790329655   -5.4550834512   -7.5512427315
   Ag      2.2414383913   -3.8066764388   -9.8593649949
   Ag     -2.2907828197    3.9188443502   -7.6050826773
   Ag      0.5610225957    4.5044771077   -7.6050819796
   Ag     -4.8701069022    1.6196298364   -5.1718125544
   Ag     -3.2769317991    1.0897966386   -2.8061340012
   Ag     -0.6609450333    3.2186205924   -9.8785392784
   Ag     -5.5910570566   -1.1220840757   -5.2192067211
   Ag     -2.2467815447    3.7498049589   -2.8525803584
   Ag     -0.9993904347    4.8666585643   -5.2034637574
   Ag     -3.8054729020    4.2470446290   -5.2192106249
   Ag     -3.5149221622    2.6758039407   -9.8593694452
   Ag     -4.4173760657   -0.0377981351   -9.8593712497
   Ag     -6.0137475245    0.4940463872   -7.5512522368
   Ag     -5.1118586671    3.2059431066   -7.5512501894
   Ag      0.5871922586    4.3317735603   -2.8525794711
   Ag      1.8237836111    5.4030432621   -5.2192000459
   Ag      3.8377073288    3.4078339914   -5.1718059272
   Ag      5.5807980727    1.1721205307   -5.2191937875
   Ag      3.4347425354    4.9610463596   -7.5512390021
   Ag      4.0747942310    1.7061247432   -9.8593564188
   Ag      5.3323750373    2.8240364209   -7.5512331190
   Ag      2.1759755136    3.8444690680   -9.8593641559
   Ag     -1.8817544351    2.1073913325   -0.4836595729
   Ag      0.8989635719    2.6784284394   -0.4836553279
   Ag      0.0000000000    0.0000000000    0.0000000000
   Ag      2.7659350940    0.5759533343   -0.4836497308
   Ag     -2.7690678527   -0.5606913044   -0.4836604335
   Ag     -0.8841719671   -2.6833451220   -0.4836610364
   Ag      1.8701101821   -2.1177433283   -0.4836551986
   SUBEND
end
dimqm
   PIM
   ALGORITHM DIRECT
   SCREEN EXP 1.0
   FORCEFIELD LJ12-6
   COORDDEPEND
   CHEMISORPTION
   COORDPAR  Ag   0.003000  0.000800  3.250000  3.650000  12.000000  9.448600  5.
↪669200
```

(continues on next page)

(continued from previous page)

```
     CHEMPAR   N     0.275000  0.052500  0.800000  0.900000  4.913000
     DEBUG
  end
  basis
    core none
    type TZP
  end
  save TAPE21
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: PIM: polarizability N2 on silver cluster Ag68

Download DIMQM_PIM_Pol.run

```
#! /bin/sh


# == Expert option ==

# This example demonstrates how to do calculate static and frequency dependent
# polarizabilities with the Polarizability Interaction Model (PIM). The example
# consists of 3 calculations: static polarizability, frequency dependent, and
# with inclusion of effect of the local field.

# When the molecule interacts with a metal nanoparticle, there are two types of
# interactions: the image field and the local field. The image field is caused
# by the dipoles induced into the nanoparticle by the molecule's electron
# density. This is always taken into account in a DIM/QM calculation. The local
# field arises by direct interactions of the nanoparticle with an external
# field. Addition of the LOCALFIELD key causes the DIM/QM calculation to include
# this effect, but by default this is not included in a DIM/QM calculation.


#######################
# STATIC POLARIZABILITY
#######################

AMS_JOBNAME=first $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  title PIM Static Polarizability, N2 on Ag68
  aoresponse
    alda
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
     Ag      4.3708278508    0.0708740258   -2.8525785565
     Ag     -4.0450216512   -1.6573690296   -2.8525892410
     Ag      2.5125751829    2.2311280445   -7.5634444475
     Ag      0.6759414795   -3.2915128984   -7.5634430672
     Ag      0.3339235029   -1.6260239766   -9.9892067983
     Ag      1.0324285615   -5.0274542660   -5.1718100987
     Ag     -2.2484187072   -3.9433040680   -7.6050830302
     Ag      3.6204971713   -2.7380910382   -7.6050765656
     Ag     -2.4569103660   -2.1817169842   -9.8785349524
     Ag     -1.7752961456   -5.4191572521   -5.2192070180
     Ag      3.7672926986   -4.2809622136   -5.2191965533
     Ag      3.1179002480   -1.0369024811   -9.8785348598
     Ag     -1.9491576017    2.1440839896   -5.1959915042
     Ag      2.5822650861    2.2930127465   -2.8061239177
     Ag     -0.3326165571    1.6197266638   -2.7987957984
     Ag      0.9464626101    2.7387205045   -5.1959886567
     Ag     -0.5598334280   -4.3819335958   -9.8593641366
     Ag     -0.2204924851   -6.0299810786   -7.5512447020
     Ag      2.5790329655   -5.4550834512   -7.5512427315
     Ag      2.2414383913   -3.8066764388   -9.8593649949
```

(continues on next page)

(continued from previous page)

```
     Ag     -2.2907828197    3.9188443502   -7.6050826773
     Ag      0.5610225957    4.5044771077   -7.6050819796
     Ag     -4.8701069022    1.6196298364   -5.1718125544
     Ag     -3.2769317991    1.0897966386   -2.8061340012
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
  end
  dimqm
     PIM
     ALGORITHM BRUTE
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

####################################
# FREQUENCY DEPENDENT POLARIZABILITY
####################################

AMS_JOBNAME=second $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
```

(continues on next page)

(continued from previous page)

```
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Frequency-Dependent Polarizability, N2 on Ag68
  aoresponse
    alda
    frequencies 3.55 [eV]
    lifetime 0.0036749
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
     Ag      4.3708278508    0.0708740258   -2.8525785565
     Ag     -4.0450216512   -1.6573690296   -2.8525892410
     Ag      2.5125751829    2.2311280445   -7.5634444475
     Ag      0.6759414795   -3.2915128984   -7.5634430672
     Ag      0.3339235029   -1.6260239766   -9.9892067983
     Ag      1.0324285615   -5.0274542660   -5.1718100987
     Ag     -2.2484187072   -3.9433040680   -7.6050830302
     Ag      3.6204971713   -2.7380910382   -7.6050765656
     Ag     -2.4569103660   -2.1817169842   -9.8785349524
     Ag     -1.7752961456   -5.4191572521   -5.2192070180
     Ag      3.7672926986   -4.2809622136   -5.2191965533
     Ag      3.1179002480   -1.0369024811   -9.8785348598
     Ag     -1.9491576017    2.1440839896   -5.1959915042
```

(continues on next page)

(continued from previous page)

```
     Ag      2.5822650861    2.2930127465   -2.8061239177
     Ag     -0.3326165571    1.6197266638   -2.7987957984
     Ag      0.9464626101    2.7387205045   -5.1959886567
     Ag     -0.5598334280   -4.3819335958   -9.8593641366
     Ag     -0.2204924851   -6.0299810786   -7.5512447020
     Ag      2.5790329655   -5.4550834512   -7.5512427315
     Ag      2.2414383913   -3.8066764388   -9.8593649949
     Ag     -2.2907828197    3.9188443502   -7.6050826773
     Ag      0.5610225957    4.5044771077   -7.6050819796
     Ag     -4.8701069022    1.6196298364   -5.1718125544
     Ag     -3.2769317991    1.0897966386   -2.8061340012
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
  end
  dimqm
     PIM
     ALGORITHM BRUTE
     FREQUENCY
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

(continues on next page)

(continued from previous page)

```
######################################################
# FREQUENCY DEPENDENT POLARIZABILITY WITH LOCAL FIELDS
######################################################

AMS_JOBNAME=third $AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Frequency-Dependent Polarizability w/ Local Fields, N2 on Ag68
  aoresponse
    alda
    frequencies 3.55 [eV]
    lifetime 0.0036749
  end
  beckegrid
    quality good
  end
  dimpar
     Ag
     rad 1.4445
     drude 9.2 0.17 eV
     bound 5.0
     SUBEND
     XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
     Ag     -0.8822430837   -2.7600722440   -5.1959892962
     Ag     -2.1240266195   -3.8206731623   -2.8525807753
     Ag      3.4578336576   -2.6744062081   -2.8525742581
     Ag     -3.7149452090   -3.2988294197   -5.2034646499
     Ag      1.8985756093   -2.1890139391   -5.1959881717
     Ag      0.6946811325   -3.3828052555   -2.8061234013
     Ag      4.7143519584   -1.5678241605   -5.2034549930
     Ag      1.6023915608   -0.5328957432   -7.5641613132
     Ag      1.2412345786    1.1021928389   -9.9892023582
     Ag      4.5392237030    0.0244593089   -7.6050682929
     Ag      0.0000106997    0.0000013921   -5.1462708001
     Ag     -0.3396832259    1.6541471619   -7.5641597129
     Ag     -2.8450227342   -0.5497019461   -5.1959955618
     Ag      2.8314207151    0.6159878635   -5.1959875699
     Ag     -3.1884901403    1.0603809604   -7.5634515357
     Ag     -1.2364094731   -1.0979217456   -2.7987978362
     Ag      1.5690420034   -0.5218070047   -2.7987993856
     Ag      4.3708278508    0.0708740258   -2.8525785565
     Ag     -4.0450216512   -1.6573690296   -2.8525892410
     Ag      2.5125751829    2.2311280445   -7.5634444475
     Ag      0.6759414795   -3.2915128984   -7.5634430672
     Ag      0.3339235029   -1.6260239766   -9.9892067983
```

(continues on next page)

(continued from previous page)

```
   Ag      1.0324285615   -5.0274542660   -5.1718100987
   Ag     -2.2484187072   -3.9433040680   -7.6050830302
   Ag      3.6204971713   -2.7380910382   -7.6050765656
   Ag     -2.4569103660   -2.1817169842   -9.8785349524
   Ag     -1.7752961456   -5.4191572521   -5.2192070180
   Ag      3.7672926986   -4.2809622136   -5.2191965533
   Ag      3.1179002480   -1.0369024811   -9.8785348598
   Ag     -1.9491576017    2.1440839896   -5.1959915042
   Ag      2.5822650861    2.2930127465   -2.8061239177
   Ag     -0.3326165571    1.6197266638   -2.7987957984
   Ag      0.9464626101    2.7387205045   -5.1959886567
   Ag     -0.5598334280   -4.3819335958   -9.8593641366
   Ag     -0.2204924851   -6.0299810786   -7.5512447020
   Ag      2.5790329655   -5.4550834512   -7.5512427315
   Ag      2.2414383913   -3.8066764388   -9.8593649949
   Ag     -2.2907828197    3.9188443502   -7.6050826773
   Ag      0.5610225957    4.5044771077   -7.6050819796
   Ag     -4.8701069022    1.6196298364   -5.1718125544
   Ag     -3.2769317991    1.0897966386   -2.8061340012
   Ag     -0.6609450333    3.2186205924   -9.8785392784
   Ag     -5.5910570566   -1.1220840757   -5.2192067211
   Ag     -2.2467815447    3.7498049589   -2.8525803584
   Ag     -0.9993904347    4.8666585643   -5.2034637574
   Ag     -3.8054729020    4.2470446290   -5.2192106249
   Ag     -3.5149221622    2.6758039407   -9.8593694452
   Ag     -4.4173760657   -0.0377981351   -9.8593712497
   Ag     -6.0137475245    0.4940463872   -7.5512522368
   Ag     -5.1118586671    3.2059431066   -7.5512501894
   Ag      0.5871922586    4.3317735603   -2.8525794711
   Ag      1.8237836111    5.4030432621   -5.2192000459
   Ag      3.8377073288    3.4078339914   -5.1718059272
   Ag      5.5807980727    1.1721205307   -5.2191937875
   Ag      3.4347425354    4.9610463596   -7.5512390021
   Ag      4.0747942310    1.7061247432   -9.8593564188
   Ag      5.3323750373    2.8240364209   -7.5512331190
   Ag      2.1759755136    3.8444690680   -9.8593641559
   Ag     -1.8817544351    2.1073913325   -0.4836595729
   Ag      0.8989635719    2.6784284394   -0.4836553279
   Ag      0.0000000000    0.0000000000    0.0000000000
   Ag      2.7659350940    0.5759533343   -0.4836497308
   Ag     -2.7690678527   -0.5606913044   -0.4836604335
   Ag     -0.8841719671   -2.6833451220   -0.4836610364
   Ag      1.8701101821   -2.1177433283   -0.4836551986
   SUBEND
end
dimqm
   PIM
   ALGORITHM DIRECT
   FREQUENCY
   LOCALFIELD
end
basis
  core none
  type TZP
end
symmetry NOSYM
xc
```

(continues on next page)

(continued from previous page)

```
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: PIM: Raman scattering N2 on silver cluster Ag68

Download DIMQM_Raman.run

```
#! /bin/sh

# == Expert option ==

# This example demonstrates how to calculate Raman scattering with the
# Polarizability Interaction Model (PIM). AORESPONSE is used and numerical
# frequencies.


# Allowing for all rigid motions gives a very different result, suggesting to the non-
↪expert that the example may be wrong


$AMSBIN/ams <<eor

# RigidMotions AllowRotations=all AllowTranslations=all

# log
# debug RigidMotionsConfigModule
# debug NormalModes
# debug RigidMotionsProjectorModule
# end

System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint
Properties
   NormalModes True
   Raman True
End
Raman
   IncidentFrequency 3.55 [eV]
   Lifetime 0.0036749
End

Engine ADF
  title DIM/QM Raman Calculation, N2 on Ag68
```

(continues on next page)

(continued from previous page)

```
responseformalism AORESPONSE
allpoints
beckegrid
  quality good
end
dimpar
   Ag
   rad=1.4445
   drude 9.2 0.17 eV
   bound 5.0
   SUBEND
   XYZ
   68
   Ag     -1.2626801699   -1.1212553334   -7.5641602542
   Ag     -1.5751210922    0.5238327185   -9.9892089940
   Ag     -4.1814870442   -1.7663886735   -7.6050834452
   Ag     -0.8822430837   -2.7600722440   -5.1959892962
   Ag     -2.1240266195   -3.8206731623   -2.8525807753
   Ag      3.4578336576   -2.6744062081   -2.8525742581
   Ag     -3.7149452090   -3.2988294197   -5.2034646499
   Ag      1.8985756093   -2.1890139391   -5.1959881717
   Ag      0.6946811325   -3.3828052555   -2.8061234013
   Ag      4.7143519584   -1.5678241605   -5.2034549930
   Ag      1.6023915608   -0.5328957432   -7.5641613132
   Ag      1.2412345786    1.1021928389   -9.9892023582
   Ag      4.5392237030    0.0244593089   -7.6050682929
   Ag      0.0000106997    0.0000013921   -5.1462708001
   Ag     -0.3396832259    1.6541471619   -7.5641597129
   Ag     -2.8450227342   -0.5497019461   -5.1959955618
   Ag      2.8314207151    0.6159878635   -5.1959875699
   Ag     -3.1884901403    1.0603809604   -7.5634515357
   Ag     -1.2364094731   -1.0979217456   -2.7987978362
   Ag      1.5690420034   -0.5218070047   -2.7987993856
   Ag      4.3708278508    0.0708740258   -2.8525785565
   Ag     -4.0450216512   -1.6573690296   -2.8525892410
   Ag      2.5125751829    2.2311280445   -7.5634444475
   Ag      0.6759414795   -3.2915128984   -7.5634430672
   Ag      0.3339235029   -1.6260239766   -9.9892067983
   Ag      1.0324285615   -5.0274542660   -5.1718100987
   Ag     -2.2484187072   -3.9433040680   -7.6050830302
   Ag      3.6204971713   -2.7380910382   -7.6050765656
   Ag     -2.4569103660   -2.1817169842   -9.8785349524
   Ag     -1.7752961456   -5.4191572521   -5.2192070180
   Ag      3.7672926986   -4.2809622136   -5.2191965533
   Ag      3.1179002480   -1.0369024811   -9.8785348598
   Ag     -1.9491576017    2.1440839896   -5.1959915042
   Ag      2.5822650861    2.2930127465   -2.8061239177
   Ag     -0.3326165571    1.6197266638   -2.7987957984
   Ag      0.9464626101    2.7387205045   -5.1959886567
   Ag     -0.5598334280   -4.3819335958   -9.8593641366
   Ag     -0.2204924851   -6.0299810786   -7.5512447020
   Ag      2.5790329655   -5.4550834512   -7.5512427315
   Ag      2.2414383913   -3.8066764388   -9.8593649949
   Ag     -2.2907828197    3.9188443502   -7.6050826773
   Ag      0.5610225957    4.5044771077   -7.6050819796
   Ag     -4.8701069022    1.6196298364   -5.1718125544
   Ag     -3.2769317991    1.0897966386   -2.8061340012
```

(continues on next page)

(continued from previous page)

```
     Ag     -0.6609450333    3.2186205924   -9.8785392784
     Ag     -5.5910570566   -1.1220840757   -5.2192067211
     Ag     -2.2467815447    3.7498049589   -2.8525803584
     Ag     -0.9993904347    4.8666585643   -5.2034637574
     Ag     -3.8054729020    4.2470446290   -5.2192106249
     Ag     -3.5149221622    2.6758039407   -9.8593694452
     Ag     -4.4173760657   -0.0377981351   -9.8593712497
     Ag     -6.0137475245    0.4940463872   -7.5512522368
     Ag     -5.1118586671    3.2059431066   -7.5512501894
     Ag      0.5871922586    4.3317735603   -2.8525794711
     Ag      1.8237836111    5.4030432621   -5.2192000459
     Ag      3.8377073288    3.4078339914   -5.1718059272
     Ag      5.5807980727    1.1721205307   -5.2191937875
     Ag      3.4347425354    4.9610463596   -7.5512390021
     Ag      4.0747942310    1.7061247432   -9.8593564188
     Ag      5.3323750373    2.8240364209   -7.5512331190
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
     SUBEND
  end
  dimqm
     PIM
     ALGORITHM DIRECT
     SCREEN EXP 1.0
     FORCEFIELD LJ12-6
     COORDDEPEND
     CHEMISORPTION
     COORDPAR  Ag   0.003000  0.000800  3.250000  3.650000  12.000000  9.448600  5.
↪669200
     CHEMPAR   N     0.275000  0.052500  0.800000  0.900000  4.913000
     FREQUENCY
     LOCALFIELD
  end
  basis
    core none
    type TZP
  end
  save TAPE21
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine
eor
```

Example: PIM: SEROA calculation N2 on silver cluster Ag68

Download DIMQM_SEROA.run

```
#! /bin/sh

# == Expert option ==


# This example demonstrates how to calculate Surface-Enhanced Raman Optical
# Activity (SEROA) with the Polarizability Interaction Model (PIM). AORESPONSE
# is used and numerical frequencies.


$AMSBIN/ams <<eor
System
  atoms
     N        -0.000097   -0.000310    2.997901
     N         0.000076    0.000133    4.095604
  end
end

Task SinglePoint
Properties
  VROA
End
Raman
  IncidentFrequency 5.0 [eV]
  Lifetime 0.0037
End

Engine ADF
  title DIM/QM SEROA Calculation, N2 on Ag68
  dimqm
     PIM
     ALGORITHM DIRECT
     SCREEN GAU 1.0
     FORCEFIELD LJ12-6
     FREQUENCY
     LOCALFIELD
  end
  exactdensity
  NumericalQuality Good
  basis
    type TZP
  end
  symmetry NoSym
  dimpar
    Ag
     rad=1.4445
     drude 5.0 0.1 eV
     bound 5.0
    SUBEND
    XYZ
     68
     Ag     -1.2626801699   -1.1212553334   -7.5641602542
     Ag     -1.5751210922    0.5238327185   -9.9892089940
     Ag     -4.1814870442   -1.7663886735   -7.6050834452
```

(continues on next page)

(continued from previous page)

```
    Ag      -0.8822430837   -2.7600722440   -5.1959892962
    Ag      -2.1240266195   -3.8206731623   -2.8525807753
    Ag       3.4578336576   -2.6744062081   -2.8525742581
    Ag      -3.7149452090   -3.2988294197   -5.2034646499
    Ag       1.8985756093   -2.1890139391   -5.1959881717
    Ag       0.6946811325   -3.3828052555   -2.8061234013
    Ag       4.7143519584   -1.5678241605   -5.2034549930
    Ag       1.6023915608   -0.5328957432   -7.5641613132
    Ag       1.2412345786    1.1021928389   -9.9892023582
    Ag       4.5392237030    0.0244593089   -7.6050682929
    Ag       0.0000106997    0.0000013921   -5.1462708001
    Ag      -0.3396832259    1.6541471619   -7.5641597129
    Ag      -2.8450227342   -0.5497019461   -5.1959955618
    Ag       2.8314207151    0.6159878635   -5.1959875699
    Ag      -3.1884901403    1.0603809604   -7.5634515357
    Ag      -1.2364094731   -1.0979217456   -2.7987978362
    Ag       1.5690420034   -0.5218070047   -2.7987993856
    Ag       4.3708278508    0.0708740258   -2.8525785565
    Ag      -4.0450216512   -1.6573690296   -2.8525892410
    Ag       2.5125751829    2.2311280445   -7.5634444475
    Ag       0.6759414795   -3.2915128984   -7.5634430672
    Ag       0.3339235029   -1.6260239766   -9.9892067983
    Ag       1.0324285615   -5.0274542660   -5.1718100987
    Ag      -2.2484187072   -3.9433040680   -7.6050830302
    Ag       3.6204971713   -2.7380910382   -7.6050765656
    Ag      -2.4569103660   -2.1817169842   -9.8785349524
    Ag      -1.7752961456   -5.4191572521   -5.2192070180
    Ag       3.7672926986   -4.2809622136   -5.2191965533
    Ag       3.1179002480   -1.0369024811   -9.8785348598
    Ag      -1.9491576017    2.1440839896   -5.1959915042
    Ag       2.5822650861    2.2930127465   -2.8061239177
    Ag      -0.3326165571    1.6197266638   -2.7987957984
    Ag       0.9464626101    2.7387205045   -5.1959886567
    Ag      -0.5598334280   -4.3819335958   -9.8593641366
    Ag      -0.2204924851   -6.0299810786   -7.5512447020
    Ag       2.5790329655   -5.4550834512   -7.5512427315
    Ag       2.2414383913   -3.8066764388   -9.8593649949
    Ag      -2.2907828197    3.9188443502   -7.6050826773
    Ag       0.5610225957    4.5044771077   -7.6050819796
    Ag      -4.8701069022    1.6196298364   -5.1718125544
    Ag      -3.2769317991    1.0897966386   -2.8061340012
    Ag      -0.6609450333    3.2186205924   -9.8785392784
    Ag      -5.5910570566   -1.1220840757   -5.2192067211
    Ag      -2.2467815447    3.7498049589   -2.8525803584
    Ag      -0.9993904347    4.8666585643   -5.2034637574
    Ag      -3.8054729020    4.2470446290   -5.2192106249
    Ag      -3.5149221622    2.6758039407   -9.8593694452
    Ag      -4.4173760657   -0.0377981351   -9.8593712497
    Ag      -6.0137475245    0.4940463872   -7.5512522368
    Ag      -5.1118586671    3.2059431066   -7.5512501894
    Ag       0.5871922586    4.3317735603   -2.8525794711
    Ag       1.8237836111    5.4030432621   -5.2192000459
    Ag       3.8377073288    3.4078339914   -5.1718059272
    Ag       5.5807980727    1.1721205307   -5.2191937875
    Ag       3.4347425354    4.9610463596   -7.5512390021
    Ag       4.0747942310    1.7061247432   -9.8593564188
    Ag       5.3323750373    2.8240364209   -7.5512331190
```

(continues on next page)

(continued from previous page)

```
     Ag      2.1759755136    3.8444690680   -9.8593641559
     Ag     -1.8817544351    2.1073913325   -0.4836595729
     Ag      0.8989635719    2.6784284394   -0.4836553279
     Ag      0.0000000000    0.0000000000    0.0000000000
     Ag      2.7659350940    0.5759533343   -0.4836497308
     Ag     -2.7690678527   -0.5606913044   -0.4836604335
     Ag     -0.8841719671   -2.6833451220   -0.4836610364
     Ag      1.8701101821   -2.1177433283   -0.4836551986
    SUBEND
  end
  Relativity
    Level None
  End
EndEngine
eor
```

Example: PIM: Multipole Method N2 on silver cluster Ag1415

Download DIMQM_PIM_CMM.run

```
#! /bin/sh


# == Expert option ==

# This example demonstrates how to do a single point calculation with the
# Polarizability Interaction Model (PIM), for a large metal cluster.


$AMSBIN/ams <<eor
System
  atoms
     N      -0.000006    -0.000060    2.302488
     N       0.000019     0.000041    3.408628
  end
end

Task SinglePoint

Engine ADF
  title PIM Cell Multipole Method Test, N2 on Ag1415
  dimqm
     PIM
     ALGORITHM MULTI
     MULTIPLIER 0.683 0.517 1.1301
  end
  basis
    core none
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  beckegrid
```

(continues on next page)

(continued from previous page)

```
  quality good
end
  Relativity
  Level None
End
dimpar
   Ag
   rad 1.4445
   SUBEND
   XYZ
   1415
   Ag    0.00000000    0.00000000  -17.21195420
   Ag    2.88900000    0.00000000  -17.21195420
   Ag    5.77800000    0.00000000  -17.21195420
   Ag    8.66700000    0.00000000  -17.21195420
   Ag   11.55600000    0.00000000  -17.21195420
   Ag   14.44500000    0.00000000  -17.21195420
   Ag   17.33400000    0.00000000  -17.21195420
   Ag   20.22300000    0.00000000  -17.21195420
   Ag    1.29138300    2.58276600  -17.21195420
   Ag    4.18038300    2.58276600  -17.21195420
   Ag    7.06938300    2.58276600  -17.21195420
   Ag    9.95838300    2.58276600  -17.21195420
   Ag   12.84738300    2.58276600  -17.21195420
   Ag   15.73638300    2.58276600  -17.21195420
   Ag   18.62538300    2.58276600  -17.21195420
   Ag    2.58276600    5.16553200  -17.21195420
   Ag    5.47176600    5.16553200  -17.21195420
   Ag    8.36076600    5.16553200  -17.21195420
   Ag   11.24976600    5.16553200  -17.21195420
   Ag   14.13876600    5.16553200  -17.21195420
   Ag   17.02776600    5.16553200  -17.21195420
   Ag    3.87414900    7.74829800  -17.21195420
   Ag    6.76314900    7.74829800  -17.21195420
   Ag    9.65214900    7.74829800  -17.21195420
   Ag   12.54114900    7.74829800  -17.21195420
   Ag   15.43014900    7.74829800  -17.21195420
   Ag    5.16553200   10.33106400  -17.21195420
   Ag    8.05453200   10.33106400  -17.21195420
   Ag   10.94353200   10.33106400  -17.21195420
   Ag   13.83253200   10.33106400  -17.21195420
   Ag    6.45691500   12.91383000  -17.21195420
   Ag    9.34591500   12.91383000  -17.21195420
   Ag   12.23491500   12.91383000  -17.21195420
   Ag    7.74829800   15.49659600  -17.21195420
   Ag   10.63729800   15.49659600  -17.21195420
   Ag    9.03968100   18.07936200  -17.21195420
   Ag    1.29716100    0.80314200  -14.75919320
   Ag    4.18616100    0.80314200  -14.75919320
   Ag    7.07516100    0.80314200  -14.75919320
   Ag    9.96416100    0.80314200  -14.75919320
   Ag   12.85316100    0.80314200  -14.75919320
   Ag   15.74216100    0.80314200  -14.75919320
   Ag   18.63116100    0.80314200  -14.75919320
   Ag    2.58854400    3.38590800  -14.75919320
   Ag    5.47754400    3.38590800  -14.75919320
   Ag    8.36654400    3.38590800  -14.75919320
```

(continues on next page)

(continued from previous page)

```
     Ag   11.25554400    3.38590800  -14.75919320
     Ag   14.14454400    3.38590800  -14.75919320
     Ag   17.03354400    3.38590800  -14.75919320
     Ag    3.87992700    5.96867400  -14.75919320
     Ag    6.76892700    5.96867400  -14.75919320
     Ag    9.65792700    5.96867400  -14.75919320
     Ag   12.54692700    5.96867400  -14.75919320
     Ag   15.43592700    5.96867400  -14.75919320
     Ag    5.17131000    8.55144000  -14.75919320
     Ag    8.06031000    8.55144000  -14.75919320
     Ag   10.94931000    8.55144000  -14.75919320
     Ag   13.83831000    8.55144000  -14.75919320
     Ag    6.46269300   11.13420600  -14.75919320
     Ag    9.35169300   11.13420600  -14.75919320
     Ag   12.24069300   11.13420600  -14.75919320
     Ag    7.75407600   13.71697200  -14.75919320
     Ag   10.64307600   13.71697200  -14.75919320
     Ag    9.04545900   16.29973800  -14.75919320
     Ag    2.59432200    1.60628400  -12.30643220
     Ag    5.48332200    1.60628400  -12.30643220
     Ag    8.37232200    1.60628400  -12.30643220
     Ag   11.26132200    1.60628400  -12.30643220
     Ag   14.15032200    1.60628400  -12.30643220
     Ag   17.03932200    1.60628400  -12.30643220
     Ag    3.88570500    4.18905000  -12.30643220
     Ag    6.77470500    4.18905000  -12.30643220
     Ag    9.66370500    4.18905000  -12.30643220
     Ag   12.55270500    4.18905000  -12.30643220
     Ag   15.44170500    4.18905000  -12.30643220
     Ag    5.17708800    6.77181600  -12.30643220
     Ag    8.06608800    6.77181600  -12.30643220
     Ag   10.95508800    6.77181600  -12.30643220
     Ag   13.84408800    6.77181600  -12.30643220
     Ag    6.46847100    9.35458200  -12.30643220
     Ag    9.35747100    9.35458200  -12.30643220
     Ag   12.24647100    9.35458200  -12.30643220
     Ag    7.75985400   11.93734800  -12.30643220
     Ag   10.64885400   11.93734800  -12.30643220
     Ag    9.05123700   14.52011400  -12.30643220
     Ag    3.89148300    2.40942600   -9.85367120
     Ag    6.78048300    2.40942600   -9.85367120
     Ag    9.66948300    2.40942600   -9.85367120
     Ag   12.55848300    2.40942600   -9.85367120
     Ag   15.44748300    2.40942600   -9.85367120
     Ag    5.18286600    4.99219200   -9.85367120
     Ag    8.07186600    4.99219200   -9.85367120
     Ag   10.96086600    4.99219200   -9.85367120
     Ag   13.84986600    4.99219200   -9.85367120
     Ag    6.47424900    7.57495800   -9.85367120
     Ag    9.36324900    7.57495800   -9.85367120
     Ag   12.25224900    7.57495800   -9.85367120
     Ag    7.76563200   10.15772400   -9.85367120
     Ag   10.65463200   10.15772400   -9.85367120
     Ag    9.05701500   12.74049000   -9.85367120
     Ag    5.18864400    3.21256800   -7.40091020
     Ag    8.07764400    3.21256800   -7.40091020
     Ag   10.96664400    3.21256800   -7.40091020
```

(continues on next page)

(continued from previous page)

```
    Ag   13.85564400    3.21256800   -7.40091020
    Ag    6.48002700    5.79533400   -7.40091020
    Ag    9.36902700    5.79533400   -7.40091020
    Ag   12.25802700    5.79533400   -7.40091020
    Ag    7.77141000    8.37810000   -7.40091020
    Ag   10.66041000    8.37810000   -7.40091020
    Ag    9.06279300   10.96086600   -7.40091020
    Ag    6.48580500    4.01571000   -4.94814920
    Ag    9.37480500    4.01571000   -4.94814920
    Ag   12.26380500    4.01571000   -4.94814920
    Ag    7.77718800    6.59847600   -4.94814920
    Ag   10.66618800    6.59847600   -4.94814920
    Ag    9.06857100    9.18124200   -4.94814920
    Ag    7.78296600    4.81885200   -2.49538820
    Ag   10.67196600    4.81885200   -2.49538820
    Ag    9.07434900    7.40161800   -2.49538820
    Ag    9.08012700    5.62199400   -0.04262720
    Ag   -1.28560500    2.09163600  -15.69234020
    Ag    0.01155600    2.89188900  -13.23957920
    Ag    1.30871700    3.69214200  -10.78681820
    Ag    2.60587800    4.49239500   -8.33405720
    Ag    3.90303900    5.29264800   -5.88129620
    Ag    5.20020000    6.09290100   -3.42853520
    Ag    6.49736100    6.89315400   -0.97577420
    Ag    0.00577800    4.67440200  -15.69234020
    Ag    1.30293900    5.47465500  -13.23957920
    Ag    2.60010000    6.27490800  -10.78681820
    Ag    3.89726100    7.07516100   -8.33405720
    Ag    5.19442200    7.87541400   -5.88129620
    Ag    6.49158300    8.67566700   -3.42853520
    Ag    1.29716100    7.25716800  -15.69234020
    Ag    2.59432200    8.05742100  -13.23957920
    Ag    3.89148300    8.85767400  -10.78681820
    Ag    5.18864400    9.65792700   -8.33405720
    Ag    6.48580500   10.45818000   -5.88129620
    Ag    2.58854400    9.83993400  -15.69234020
    Ag    3.88570500   10.64018700  -13.23957920
    Ag    5.18286600   11.44044000  -10.78681820
    Ag    6.48002700   12.24069300   -8.33405720
    Ag    3.87992700   12.42270000  -15.69234020
    Ag    5.17708800   13.22295300  -13.23957920
    Ag    6.47424900   14.02320600  -10.78681820
    Ag    5.17131000   15.00546600  -15.69234020
    Ag    6.46847100   15.80571900  -13.23957920
    Ag    6.46269300   17.58823200  -15.69234020
    Ag   -2.57121000    4.18327200  -14.17272620
    Ag   -1.27404900    4.98352500  -11.71996520
    Ag    0.02311200    5.78377800   -9.26720420
    Ag    1.32027300    6.58403100   -6.81444320
    Ag    2.61743400    7.38428400   -4.36168220
    Ag    3.91459500    8.18453700   -1.90892120
    Ag   -1.27982700    6.76603800  -14.17272620
    Ag    0.01733400    7.56629100  -11.71996520
    Ag    1.31449500    8.36654400   -9.26720420
    Ag    2.61165600    9.16679700   -6.81444320
    Ag    3.90881700    9.96705000   -4.36168220
    Ag    0.01155600    9.34880400  -14.17272620
```

(continues on next page)

(continued from previous page)

```
     Ag    1.30871700   10.14905700  -11.71996520
     Ag    2.60587800   10.94931000   -9.26720420
     Ag    3.90303900   11.74956300   -6.81444320
     Ag    1.30293900   11.93157000  -14.17272620
     Ag    2.60010000   12.73182300  -11.71996520
     Ag    3.89726100   13.53207600   -9.26720420
     Ag    2.59432200   14.51433600  -14.17272620
     Ag    3.89148300   15.31458900  -11.71996520
     Ag    3.88570500   17.09710200  -14.17272620
     Ag   -3.85681500    6.27490800  -12.65311220
     Ag   -2.55965400    7.07516100  -10.20035120
     Ag   -1.26249300    7.87541400   -7.74759020
     Ag    0.03466800    8.67566700   -5.29482920
     Ag    1.33182900    9.47592000   -2.84206820
     Ag   -2.56543200    8.85767400  -12.65311220
     Ag   -1.26827100    9.65792700  -10.20035120
     Ag    0.02889000   10.45818000   -7.74759020
     Ag    1.32605100   11.25843300   -5.29482920
     Ag   -1.27404900   11.44044000  -12.65311220
     Ag    0.02311200   12.24069300  -10.20035120
     Ag    1.32027300   13.04094600   -7.74759020
     Ag    0.01733400   14.02320600  -12.65311220
     Ag    1.31449500   14.82345900  -10.20035120
     Ag    1.30871700   16.60597200  -12.65311220
     Ag   -5.14242000    8.36654400  -11.13349820
     Ag   -3.84525900    9.16679700   -8.68073720
     Ag   -2.54809800    9.96705000   -6.22797620
     Ag   -1.25093700   10.76730300   -3.77521520
     Ag   -3.85103700   10.94931000  -11.13349820
     Ag   -2.55387600   11.74956300   -8.68073720
     Ag   -1.25671500   12.54981600   -6.22797620
     Ag   -2.55965400   13.53207600  -11.13349820
     Ag   -1.26249300   14.33232900   -8.68073720
     Ag   -1.26827100   16.11484200  -11.13349820
     Ag   -6.42802500   10.45818000   -9.61388420
     Ag   -5.13086400   11.25843300   -7.16112320
     Ag   -3.83370300   12.05868600   -4.70836220
     Ag   -5.13664200   13.04094600   -9.61388420
     Ag   -3.83948100   13.84119900   -7.16112320
     Ag   -3.84525900   15.62371200   -9.61388420
     Ag   -7.71363000   12.54981600   -8.09427020
     Ag   -6.41646900   13.35006900   -5.64150920
     Ag   -6.42224700   15.13258200   -8.09427020
     Ag   -8.99923500   14.64145200   -6.57465620
     Ag    1.29138300    0.79811860  -19.66831060
     Ag    4.18038300    0.79811860  -19.66831060
     Ag    7.06938300    0.79811860  -19.66831060
     Ag    9.95838300    0.79811860  -19.66831060
     Ag   12.84738300    0.79811860  -19.66831060
     Ag   15.73638300    0.79811860  -19.66831060
     Ag   18.62538300    0.79811860  -19.66831060
     Ag    2.58276600    1.59623720  -22.12466710
     Ag    5.47176600    1.59623720  -22.12466710
     Ag    8.36076600    1.59623720  -22.12466710
     Ag   11.24976600    1.59623720  -22.12466710
     Ag   14.13876600    1.59623720  -22.12466710
     Ag   17.02776600    1.59623720  -22.12466710
```

(continues on next page)

(continued from previous page)

```
    Ag    3.87414900    2.39435580  -24.58102350
    Ag    6.76314900    2.39435580  -24.58102350
    Ag    9.65214900    2.39435580  -24.58102350
    Ag   12.54114900    2.39435580  -24.58102350
    Ag   15.43014900    2.39435580  -24.58102350
    Ag    5.16553200    3.19247430  -27.03737990
    Ag    8.05453200    3.19247430  -27.03737990
    Ag   10.94353200    3.19247430  -27.03737990
    Ag   13.83253200    3.19247430  -27.03737990
    Ag    6.45691500    3.99059290  -29.49373640
    Ag    9.34591500    3.99059290  -29.49373640
    Ag   12.23491500    3.99059290  -29.49373640
    Ag    7.74829800    4.78871150  -31.95009280
    Ag   10.63729800    4.78871150  -31.95009280
    Ag    9.03968100    5.58683010  -34.40644920
    Ag    2.58854400    3.37901740  -19.67419920
    Ag    5.47754400    3.37901740  -19.67419920
    Ag    8.36654400    3.37901740  -19.67419920
    Ag   11.25554400    3.37901740  -19.67419920
    Ag   14.14454400    3.37901740  -19.67419920
    Ag   17.03354400    3.37901740  -19.67419920
    Ag    3.87992700    4.17713600  -22.13055570
    Ag    6.76892700    4.17713600  -22.13055570
    Ag    9.65792700    4.17713600  -22.13055570
    Ag   12.54692700    4.17713600  -22.13055570
    Ag   15.43592700    4.17713600  -22.13055570
    Ag    5.17131000    4.97525460  -24.58691210
    Ag    8.06031000    4.97525460  -24.58691210
    Ag   10.94931000    4.97525460  -24.58691210
    Ag   13.83831000    4.97525460  -24.58691210
    Ag    6.46269300    5.77337320  -27.04326850
    Ag    9.35169300    5.77337320  -27.04326850
    Ag   12.24069300    5.77337320  -27.04326850
    Ag    7.75407600    6.57149180  -29.49962500
    Ag   10.64307600    6.57149180  -29.49962500
    Ag    9.04545900    7.36961040  -31.95598140
    Ag    3.88570500    5.95991630  -19.68008780
    Ag    6.77470500    5.95991630  -19.68008780
    Ag    9.66370500    5.95991630  -19.68008780
    Ag   12.55270500    5.95991630  -19.68008780
    Ag   15.44170500    5.95991630  -19.68008780
    Ag    5.17708800    6.75803490  -22.13644430
    Ag    8.06608800    6.75803490  -22.13644430
    Ag   10.95508800    6.75803490  -22.13644430
    Ag   13.84408800    6.75803490  -22.13644430
    Ag    6.46847100    7.55615350  -24.59280070
    Ag    9.35747100    7.55615350  -24.59280070
    Ag   12.24647100    7.55615350  -24.59280070
    Ag    7.75985400    8.35427210  -27.04915710
    Ag   10.64885400    8.35427210  -27.04915710
    Ag    9.05123700    9.15239070  -29.50551360
    Ag    5.18286600    8.54081520  -19.68597640
    Ag    8.07186600    8.54081520  -19.68597640
    Ag   10.96086600    8.54081520  -19.68597640
    Ag   13.84986600    8.54081520  -19.68597640
    Ag    6.47424900    9.33893370  -22.14233290
    Ag    9.36324900    9.33893370  -22.14233290
```

(continues on next page)

(continued from previous page)

```
    Ag   12.25224900    9.33893370  -22.14233290
    Ag    7.76563200   10.13705230  -24.59868930
    Ag   10.65463200   10.13705230  -24.59868930
    Ag    9.05701500   10.93517090  -27.05504570
    Ag    6.48002700   11.12171400  -19.69186500
    Ag    9.36902700   11.12171400  -19.69186500
    Ag   12.25802700   11.12171400  -19.69186500
    Ag    7.77141000   11.91983260  -22.14822150
    Ag   10.66041000   11.91983260  -22.14822150
    Ag    9.06279300   12.71795120  -24.60457790
    Ag    7.77718800   13.70261290  -19.69775360
    Ag   10.66618800   13.70261290  -19.69775360
    Ag    9.06857100   14.50073150  -22.15411010
    Ag    9.07434900   16.28351170  -19.70364220
    Ag   -1.28560500    2.09158990  -18.73163170
    Ag    0.01155600    4.67159600  -18.73477270
    Ag    1.30871700    7.25160210  -18.73791370
    Ag    2.60587800    9.83160820  -18.74105470
    Ag    3.90303900   12.41161430  -18.74419570
    Ag    5.20020000   14.99162040  -18.74733670
    Ag    6.49736100   17.57162650  -18.75047770
    Ag    0.00577800    2.88970850  -21.18798810
    Ag    1.30293900    5.46971460  -21.19112910
    Ag    2.60010000    8.04972070  -21.19427010
    Ag    3.89726100   10.62972680  -21.19741110
    Ag    5.19442200   13.20973290  -21.20055210
    Ag    6.49158300   15.78973900  -21.20369310
    Ag    1.29716100    3.68782700  -23.64434460
    Ag    2.59432200    6.26783310  -23.64748560
    Ag    3.89148300    8.84783930  -23.65062660
    Ag    5.18864400   11.42784540  -23.65376760
    Ag    6.48580500   14.00785150  -23.65690860
    Ag    2.58854400    4.48594560  -26.10070100
    Ag    3.88570500    7.06595170  -26.10384200
    Ag    5.18286600    9.64595780  -26.10698300
    Ag    6.48002700   12.22596400  -26.11012400
    Ag    3.87992700    5.28406420  -28.55705740
    Ag    5.17708800    7.86407030  -28.56019840
    Ag    6.47424900   10.44407640  -28.56333940
    Ag    5.17131000    6.08218280  -31.01341390
    Ag    6.46847100    8.66218890  -31.01655490
    Ag    6.46269300    6.88030140  -33.46977030
    Ag   -2.57121000    4.18317970  -20.25130920
    Ag   -1.27404900    6.76318580  -20.25445020
    Ag    0.02311200    9.34319200  -20.25759120
    Ag    1.32027300   11.92319810  -20.26073220
    Ag    2.61743400   14.50320420  -20.26387320
    Ag    3.91459500   17.08321030  -20.26701420
    Ag   -1.27982700    4.98129830  -22.70766560
    Ag    0.01733400    7.56130440  -22.71080660
    Ag    1.31449500   10.14131050  -22.71394760
    Ag    2.61165600   12.72131660  -22.71708860
    Ag    3.90881700   15.30132280  -22.72022960
    Ag    0.01155600    5.77941690  -25.16402210
    Ag    1.30871700    8.35942300  -25.16716310
    Ag    2.60587800   10.93942910  -25.17030410
    Ag    3.90303900   13.51943520  -25.17344510
```

(continues on next page)

(continued from previous page)

```
    Ag    1.30293900    6.57753550  -27.62037850
    Ag    2.60010000    9.15754160  -27.62351950
    Ag    3.89726100   11.73754770  -27.62666050
    Ag    2.59432200    7.37565410  -30.07673490
    Ag    3.89148300    9.95566020  -30.07987590
    Ag    3.88570500    8.17377270  -32.53309140
    Ag   -3.85681500    6.27476960  -21.77098670
    Ag   -2.55965400    8.85477570  -21.77412770
    Ag   -1.26249300   11.43478180  -21.77726870
    Ag    0.03466800   14.01478790  -21.78040970
    Ag    1.33182900   16.59479400  -21.78355070
    Ag   -2.56543200    7.07288820  -24.22734310
    Ag   -1.26827100    9.65289430  -24.23048410
    Ag    0.02889000   12.23290040  -24.23362510
    Ag    1.32605100   14.81290650  -24.23676610
    Ag   -1.27404900    7.87100680  -26.68369960
    Ag    0.02311200   10.45101290  -26.68684060
    Ag    1.32027300   13.03101900  -26.68998160
    Ag    0.01733400    8.66912540  -29.14005600
    Ag    1.31449500   11.24913150  -29.14319700
    Ag    1.30871700    9.46724390  -31.59641240
    Ag   -5.14242000    8.36635950  -23.29066420
    Ag   -3.84525900   10.94636560  -23.29380520
    Ag   -2.54809800   13.52637170  -23.29694620
    Ag   -1.25093700   16.10637780  -23.30008720
    Ag   -3.85103700    9.16447810  -25.74702060
    Ag   -2.55387600   11.74448420  -25.75016160
    Ag   -1.25671500   14.32449030  -25.75330260
    Ag   -2.55965400    9.96259660  -28.20337710
    Ag   -1.26249300   12.54260270  -28.20651810
    Ag   -1.26827100   10.76071520  -30.65973350
    Ag   -6.42802500   10.45794930  -24.81034170
    Ag   -5.13086400   13.03795540  -24.81348270
    Ag   -3.83370300   15.61796160  -24.81662370
    Ag   -5.13664200   11.25606790  -27.26669810
    Ag   -3.83948100   13.83607400  -27.26983910
    Ag   -3.84525900   12.05418650  -29.72305460
    Ag   -7.71363000   12.54953920  -26.33001920
    Ag   -6.41646900   15.12954530  -26.33316020
    Ag   -6.42224700   13.34765780  -28.78637560
    Ag   -8.99923500   14.64112910  -27.84969670
    Ag    1.29138300   -2.08950160  -18.73006600
    Ag    4.18038300   -2.08950160  -18.73006600
    Ag    7.06938300   -2.08950160  -18.73006600
    Ag    9.95838300   -2.08950160  -18.73006600
    Ag   12.84738300   -2.08950160  -18.73006600
    Ag   15.73638300   -2.08950160  -18.73006600
    Ag   18.62538300   -2.08950160  -18.73006600
    Ag    2.58276600   -4.17900320  -20.24817770
    Ag    5.47176600   -4.17900320  -20.24817770
    Ag    8.36076600   -4.17900320  -20.24817770
    Ag   11.24976600   -4.17900320  -20.24817770
    Ag   14.13876600   -4.17900320  -20.24817770
    Ag   17.02776600   -4.17900320  -20.24817770
    Ag    3.87414900   -6.26850480  -21.76628950
    Ag    6.76314900   -6.26850480  -21.76628950
    Ag    9.65214900   -6.26850480  -21.76628950
```

(continues on next page)

(continued from previous page)

```
    Ag   12.54114900   -6.26850480  -21.76628950
    Ag   15.43014900   -6.26850480  -21.76628950
    Ag    5.16553200   -8.35800630  -23.28440130
    Ag    8.05453200   -8.35800630  -23.28440130
    Ag   10.94353200   -8.35800630  -23.28440130
    Ag   13.83253200   -8.35800630  -23.28440130
    Ag    6.45691500  -10.44750790  -24.80251300
    Ag    9.34591500  -10.44750790  -24.80251300
    Ag   12.23491500  -10.44750790  -24.80251300
    Ag    7.74829800  -12.53700950  -26.32062480
    Ag   10.63729800  -12.53700950  -26.32062480
    Ag    9.03968100  -14.62651110  -27.83873660
    Ag    2.58854400   -1.29756040  -21.18646630
    Ag    5.47754400   -1.29756040  -21.18646630
    Ag    8.36654400   -1.29756040  -21.18646630
    Ag   11.25554400   -1.29756040  -21.18646630
    Ag   14.14454400   -1.29756040  -21.18646630
    Ag   17.03354400   -1.29756040  -21.18646630
    Ag    3.87992700   -3.38706200  -22.70457810
    Ag    6.76892700   -3.38706200  -22.70457810
    Ag    9.65792700   -3.38706200  -22.70457810
    Ag   12.54692700   -3.38706200  -22.70457810
    Ag   15.43592700   -3.38706200  -22.70457810
    Ag    5.17131000   -5.47656350  -24.22268990
    Ag    8.06031000   -5.47656350  -24.22268990
    Ag   10.94931000   -5.47656350  -24.22268990
    Ag   13.83831000   -5.47656350  -24.22268990
    Ag    6.46269300   -7.56606510  -25.74080160
    Ag    9.35169300   -7.56606510  -25.74080160
    Ag   12.24069300   -7.56606510  -25.74080160
    Ag    7.75407600   -9.65556670  -27.25891340
    Ag   10.64307600   -9.65556670  -27.25891340
    Ag    9.04545900  -11.74506830  -28.77702510
    Ag    3.88570500   -0.50561920  -23.64286670
    Ag    6.77470500   -0.50561920  -23.64286670
    Ag    9.66370500   -0.50561920  -23.64286670
    Ag   12.55270500   -0.50561920  -23.64286670
    Ag   15.44170500   -0.50561920  -23.64286670
    Ag    5.17708800   -2.59512070  -25.16097840
    Ag    8.06608800   -2.59512070  -25.16097840
    Ag   10.95508800   -2.59512070  -25.16097840
    Ag   13.84408800   -2.59512070  -25.16097840
    Ag    6.46847100   -4.68462230  -26.67909020
    Ag    9.35747100   -4.68462230  -26.67909020
    Ag   12.24647100   -4.68462230  -26.67909020
    Ag    7.75985400   -6.77412390  -28.19720200
    Ag   10.64885400   -6.77412390  -28.19720200
    Ag    9.05123700   -8.86362550  -29.71531370
    Ag    5.18286600    0.28632210  -26.09926700
    Ag    8.07186600    0.28632210  -26.09926700
    Ag   10.96086600    0.28632210  -26.09926700
    Ag   13.84986600    0.28632210  -26.09926700
    Ag    6.47424900   -1.80317950  -27.61737880
    Ag    9.36324900   -1.80317950  -27.61737880
    Ag   12.25224900   -1.80317950  -27.61737880
    Ag    7.76563200   -3.89268110  -29.13549060
    Ag   10.65463200   -3.89268110  -29.13549060
```

(continues on next page)

(continued from previous page)

```
    Ag    9.05701500   -5.98218270  -30.65360230
    Ag    6.48002700    1.07826330  -28.55566740
    Ag    9.36902700    1.07826330  -28.55566740
    Ag   12.25802700    1.07826330  -28.55566740
    Ag    7.77141000   -1.01123830  -30.07377920
    Ag   10.66041000   -1.01123830  -30.07377920
    Ag    9.06279300   -3.10073990  -31.59189090
    Ag    7.77718800    1.87020450  -31.01206770
    Ag   10.66618800    1.87020450  -31.01206770
    Ag    9.06857100   -0.21929710  -32.53017950
    Ag    9.07434900    2.66214570  -33.46846810
    Ag   -1.28560500   -0.79896240  -19.67078050
    Ag    0.01155600   -0.00468390  -22.12548280
    Ag    1.30871700    0.78959460  -24.58018500
    Ag    2.60587800    1.58387300  -27.03488730
    Ag    3.90303900    2.37815150  -29.48958950
    Ag    5.20020000    3.17243000  -31.94429180
    Ag    6.49736100    3.96670840  -34.39899400
    Ag    0.00577800   -2.88846400  -21.18889230
    Ag    1.30293900   -2.09418550  -23.64359460
    Ag    2.60010000   -1.29990700  -26.09829680
    Ag    3.89726100   -0.50562860  -28.55299900
    Ag    5.19442200    0.28864990  -31.00770130
    Ag    6.49158300    1.08292840  -33.46240350
    Ag    1.29716100   -4.97796550  -22.70700410
    Ag    2.59432200   -4.18368710  -25.16170630
    Ag    3.89148300   -3.38940860  -27.61640860
    Ag    5.18864400   -2.59513010  -30.07111080
    Ag    6.48580500   -1.80085170  -32.52581300
    Ag    2.58854400   -7.06746710  -24.22511580
    Ag    3.88570500   -6.27318870  -26.67981810
    Ag    5.18286600   -5.47891020  -29.13452030
    Ag    6.48002700   -4.68463170  -31.58922260
    Ag    3.87992700   -9.15696870  -25.74322760
    Ag    5.17708800   -8.36269030  -28.19792980
    Ag    6.47424900   -7.56841180  -30.65263210
    Ag    5.17131000  -11.24647030  -27.26133940
    Ag    6.46847100  -10.45219180  -29.71604160
    Ag    6.46269300  -13.33597190  -28.77945110
    Ag   -2.57121000   -1.59792470  -22.12960690
    Ag   -1.27404900   -0.80364630  -24.58430910
    Ag    0.02311200   -0.00936780  -27.03901140
    Ag    1.32027300    0.78491070  -29.49371360
    Ag    2.61743400    1.57918910  -31.94841590
    Ag    3.91459500    2.37346760  -34.40311810
    Ag   -1.27982700   -3.68742630  -23.64771870
    Ag    0.01733400   -2.89314790  -26.10242090
    Ag    1.31449500   -2.09886940  -28.55712310
    Ag    2.61165600   -1.30459090  -31.01182540
    Ag    3.90881700   -0.51031250  -33.46652760
    Ag    0.01155600   -5.77692790  -25.16583040
    Ag    1.30871700   -4.98264940  -27.62053270
    Ag    2.60587800   -4.18837100  -30.07523490
    Ag    3.90303900   -3.39409250  -32.52993720
    Ag    1.30293900   -7.86642950  -26.68394220
    Ag    2.60010000   -7.07215100  -29.13864440
    Ag    3.89726100   -6.27787260  -31.59334670
```

(continues on next page)

(continued from previous page)

```
    Ag    2.59432200   -9.95593110  -28.20205390
    Ag    3.89148300   -9.16165260  -30.65675620
    Ag    3.88570500  -12.04543270  -29.72016570
    Ag   -3.85681500   -2.39688710  -24.58843320
    Ag   -2.55965400   -1.60260860  -27.04313550
    Ag   -1.26249300   -0.80833020  -29.49783770
    Ag    0.03466800   -0.01405170  -31.95254000
    Ag    1.33182900    0.78022680  -34.40724220
    Ag   -2.56543200   -4.48638870  -26.10654500
    Ag   -1.26827100   -3.69211020  -28.56124720
    Ag    0.02889000   -2.89783180  -31.01594950
    Ag    1.32605100   -2.10355330  -33.47065170
    Ag   -1.27404900   -6.57589030  -27.62465680
    Ag    0.02311200   -5.78161180  -30.07935900
    Ag    1.32027300   -4.98733340  -32.53406130
    Ag    0.01733400   -8.66539190  -29.14276850
    Ag    1.31449500   -7.87111340  -31.59747080
    Ag    1.30871700  -10.75489350  -30.66088030
    Ag   -5.14242000   -3.19584950  -27.04725960
    Ag   -3.84525900   -2.40157100  -29.50196180
    Ag   -2.54809800   -1.60729260  -31.95666410
    Ag   -1.25093700   -0.81301410  -34.41136630
    Ag   -3.85103700   -5.28535110  -28.56537130
    Ag   -2.55387600   -4.49107260  -31.02007360
    Ag   -1.25671500   -3.69679410  -33.47477580
    Ag   -2.55965400   -7.37485270  -30.08348310
    Ag   -1.26249300   -6.58057420  -32.53818540
    Ag   -1.26827100   -9.46435420  -31.60159490
    Ag   -6.42802500   -3.99481190  -29.50608590
    Ag   -5.13086400   -3.20053340  -31.96078820
    Ag   -3.83370300   -2.40625490  -34.41549040
    Ag   -5.13664200   -6.08431340  -31.02419770
    Ag   -3.83948100   -5.29003500  -33.47889990
    Ag   -3.84525900   -8.17381500  -32.54230950
    Ag   -7.71363000   -4.79377420  -31.96491230
    Ag   -6.41646900   -3.99949580  -34.41961450
    Ag   -6.42224700   -6.88327580  -33.48302400
    Ag   -8.99923500   -5.59273660  -34.42373860
    Ag    1.29138300   -2.08950160  -15.69384240
    Ag    4.18038300   -2.08950160  -15.69384240
    Ag    7.06938300   -2.08950160  -15.69384240
    Ag    9.95838300   -2.08950160  -15.69384240
    Ag   12.84738300   -2.08950160  -15.69384240
    Ag   15.73638300   -2.08950160  -15.69384240
    Ag   18.62538300   -2.08950160  -15.69384240
    Ag    2.58276600   -4.17900320  -14.17573070
    Ag    5.47176600   -4.17900320  -14.17573070
    Ag    8.36076600   -4.17900320  -14.17573070
    Ag   11.24976600   -4.17900320  -14.17573070
    Ag   14.13876600   -4.17900320  -14.17573070
    Ag   17.02776600   -4.17900320  -14.17573070
    Ag    3.87414900   -6.26850480  -12.65761890
    Ag    6.76314900   -6.26850480  -12.65761890
    Ag    9.65214900   -6.26850480  -12.65761890
    Ag   12.54114900   -6.26850480  -12.65761890
    Ag   15.43014900   -6.26850480  -12.65761890
    Ag    5.16553200   -8.35800630  -11.13950710
```

(continues on next page)

(continued from previous page)

```
    Ag    8.05453200   -8.35800630  -11.13950710
    Ag   10.94353200   -8.35800630  -11.13950710
    Ag   13.83253200   -8.35800630  -11.13950710
    Ag    6.45691500  -10.44750790   -9.62139540
    Ag    9.34591500  -10.44750790   -9.62139540
    Ag   12.23491500  -10.44750790   -9.62139540
    Ag    7.74829800  -12.53700950   -8.10328360
    Ag   10.63729800  -12.53700950   -8.10328360
    Ag    9.03968100  -14.62651110   -6.58517180
    Ag    2.58854400   -4.18095390  -17.20609270
    Ag    5.47754400   -4.18095390  -17.20609270
    Ag    8.36654400   -4.18095390  -17.20609270
    Ag   11.25554400   -4.18095390  -17.20609270
    Ag   14.14454400   -4.18095390  -17.20609270
    Ag   17.03354400   -4.18095390  -17.20609270
    Ag    3.87992700   -6.27045540  -15.68798100
    Ag    6.76892700   -6.27045540  -15.68798100
    Ag    9.65792700   -6.27045540  -15.68798100
    Ag   12.54692700   -6.27045540  -15.68798100
    Ag   15.43592700   -6.27045540  -15.68798100
    Ag    5.17131000   -8.35995700  -14.16986920
    Ag    8.06031000   -8.35995700  -14.16986920
    Ag   10.94931000   -8.35995700  -14.16986920
    Ag   13.83831000   -8.35995700  -14.16986920
    Ag    6.46269300  -10.44945860  -12.65175740
    Ag    9.35169300  -10.44945860  -12.65175740
    Ag   12.24069300  -10.44945860  -12.65175740
    Ag    7.75407600  -12.53896020  -11.13364570
    Ag   10.64307600  -12.53896020  -11.13364570
    Ag    9.04545900  -14.62846180   -9.61553390
    Ag    3.88570500   -6.27240610  -18.71834310
    Ag    6.77470500   -6.27240610  -18.71834310
    Ag    9.66370500   -6.27240610  -18.71834310
    Ag   12.55270500   -6.27240610  -18.71834310
    Ag   15.44170500   -6.27240610  -18.71834310
    Ag    5.17708800   -8.36190770  -17.20023130
    Ag    8.06608800   -8.36190770  -17.20023130
    Ag   10.95508800   -8.36190770  -17.20023130
    Ag   13.84408800   -8.36190770  -17.20023130
    Ag    6.46847100  -10.45140930  -15.68211950
    Ag    9.35747100  -10.45140930  -15.68211950
    Ag   12.24647100  -10.45140930  -15.68211950
    Ag    7.75985400  -12.54091090  -14.16400780
    Ag   10.64885400  -12.54091090  -14.16400780
    Ag    9.05123700  -14.63041250  -12.64589600
    Ag    5.18286600   -8.36385840  -20.23059340
    Ag    8.07186600   -8.36385840  -20.23059340
    Ag   10.96086600   -8.36385840  -20.23059340
    Ag   13.84986600   -8.36385840  -20.23059340
    Ag    6.47424900  -10.45336000  -18.71248160
    Ag    9.36324900  -10.45336000  -18.71248160
    Ag   12.25224900  -10.45336000  -18.71248160
    Ag    7.76563200  -12.54286160  -17.19436980
    Ag   10.65463200  -12.54286160  -17.19436980
    Ag    9.05701500  -14.63236320  -15.67625810
    Ag    6.48002700  -10.45531070  -21.74284370
    Ag    9.36902700  -10.45531070  -21.74284370
```

(continues on next page)

(continued from previous page)

```
    Ag   12.25802700  -10.45531070  -21.74284370
    Ag    7.77141000  -12.54481230  -20.22473190
    Ag   10.66041000  -12.54481230  -20.22473190
    Ag    9.06279300  -14.63431380  -18.70662010
    Ag    7.77718800  -12.54676290  -23.25509400
    Ag   10.66618800  -12.54676290  -23.25509400
    Ag    9.06857100  -14.63626450  -21.73698220
    Ag    9.07434900  -14.63821520  -24.76734430
    Ag   -1.28560500   -2.58537580  -17.21191500
    Ag    0.01155600   -4.67449080  -18.72586340
    Ag    1.30871700   -6.76360580  -20.23981180
    Ag    2.60587800   -8.85272080  -21.75376020
    Ag    3.90303900  -10.94183580  -23.26770860
    Ag    5.20020000  -13.03095090  -24.78165710
    Ag    6.49736100  -15.12006590  -26.29560550
    Ag    0.00577800   -4.67487740  -15.69380320
    Ag    1.30293900   -6.76399240  -17.20775160
    Ag    2.60010000   -8.85310740  -18.72170000
    Ag    3.89726100  -10.94222240  -20.23564850
    Ag    5.19442200  -13.03133740  -21.74959690
    Ag    6.49158300  -15.12045250  -23.26354530
    Ag    1.29716100   -6.76437890  -14.17569140
    Ag    2.59432200   -8.85349400  -15.68963980
    Ag    3.89148300  -10.94260900  -17.20358830
    Ag    5.18864400  -13.03172400  -18.71753670
    Ag    6.48580500  -15.12083900  -20.23148510
    Ag    2.58854400   -8.85388050  -12.65757970
    Ag    3.88570500  -10.94299550  -14.17152810
    Ag    5.18286600  -13.03211060  -15.68547650
    Ag    6.48002700  -15.12122560  -17.19942490
    Ag    3.87992700  -10.94338210  -11.13946790
    Ag    5.17708800  -13.03249710  -12.65341630
    Ag    6.47424900  -15.12161220  -14.16736470
    Ag    5.17131000  -13.03288370   -9.62135610
    Ag    6.46847100  -15.12199870  -11.13530460
    Ag    6.46269300  -15.12238530   -8.10324440
    Ag   -2.57121000   -5.17075150  -17.21187570
    Ag   -1.27404900   -7.25986660  -18.72582410
    Ag    0.02311200   -9.34898160  -20.23977260
    Ag    1.32027300  -11.43809660  -21.75372100
    Ag    2.61743400  -13.52721160  -23.26766940
    Ag    3.91459500  -15.61632660  -24.78161780
    Ag   -1.27982700   -7.26025310  -15.69376390
    Ag    0.01733400   -9.34936810  -17.20771240
    Ag    1.31449500  -11.43848320  -18.72166080
    Ag    2.61165600  -13.52759820  -20.23560920
    Ag    3.90881700  -15.61671320  -21.74955760
    Ag    0.01155600   -9.34975470  -14.17565220
    Ag    1.30871700  -11.43886970  -15.68960060
    Ag    2.60587800  -13.52798470  -17.20354900
    Ag    3.90303900  -15.61709980  -18.71749740
    Ag    1.30293900  -11.43925630  -12.65754040
    Ag    2.60010000  -13.52837130  -14.17148880
    Ag    3.89726100  -15.61748630  -15.68543730
    Ag    2.59432200  -13.52875790  -11.13942870
    Ag    3.89148300  -15.61787290  -12.65337710
    Ag    3.88570500  -15.61825950   -9.62131690
```

(continues on next page)

(continued from previous page)

```
    Ag   -3.85681500   -7.75612730  -17.21183650
    Ag   -2.55965400   -9.84524230  -18.72578490
    Ag   -1.26249300  -11.93435730  -20.23973330
    Ag    0.03466800  -14.02347240  -21.75368170
    Ag    1.33182900  -16.11258740  -23.26763020
    Ag   -2.56543200   -9.84562890  -15.69372470
    Ag   -1.26827100  -11.93474390  -17.20767310
    Ag    0.02889000  -14.02385890  -18.72162150
    Ag    1.32605100  -16.11297400  -20.23557000
    Ag   -1.27404900  -11.93513050  -14.17561290
    Ag    0.02311200  -14.02424550  -15.68956140
    Ag    1.32027300  -16.11336050  -17.20350980
    Ag    0.01733400  -14.02463210  -12.65750120
    Ag    1.31449500  -16.11374710  -14.17144960
    Ag    1.30871700  -16.11413370  -11.13938940
    Ag   -5.14242000  -10.34150310  -17.21179720
    Ag   -3.84525900  -12.43061810  -18.72574560
    Ag   -2.54809800  -14.51973310  -20.23969410
    Ag   -1.25093700  -16.60884810  -21.75364250
    Ag   -3.85103700  -12.43100470  -15.69368550
    Ag   -2.55387600  -14.52011970  -17.20763390
    Ag   -1.25671500  -16.60923470  -18.72158230
    Ag   -2.55965400  -14.52050620  -14.17557370
    Ag   -1.26249300  -16.60962130  -15.68952210
    Ag   -1.26827100  -16.61000780  -12.65746190
    Ag   -6.42802500  -12.92687880  -17.21175800
    Ag   -5.13086400  -15.01599390  -18.72570640
    Ag   -3.83370300  -17.10510890  -20.23965480
    Ag   -5.13664200  -15.01638040  -15.69364620
    Ag   -3.83948100  -17.10549540  -17.20759460
    Ag   -3.84525900  -17.10588200  -14.17553450
    Ag   -7.71363000  -15.51225460  -17.21171870
    Ag   -6.41646900  -17.60136960  -18.72566720
    Ag   -6.42224700  -17.60175620  -15.69360700
    Ag   -8.99923500  -18.09763040  -17.21167950
    Ag    2.58854400   -1.28641120  -13.23381950
    Ag    5.47754400   -1.28641120  -13.23381950
    Ag    8.36654400   -1.28641120  -13.23381950
    Ag   11.25554400   -1.28641120  -13.23381950
    Ag   14.14454400   -1.28641120  -13.23381950
    Ag   17.03354400   -1.28641120  -13.23381950
    Ag    3.87992700   -0.48829260  -10.77746310
    Ag    6.76892700   -0.48829260  -10.77746310
    Ag    9.65792700   -0.48829260  -10.77746310
    Ag   12.54692700   -0.48829260  -10.77746310
    Ag   15.43592700   -0.48829260  -10.77746310
    Ag    5.17131000    0.30982600   -8.32110660
    Ag    8.06031000    0.30982600   -8.32110660
    Ag   10.94931000    0.30982600   -8.32110660
    Ag   13.83831000    0.30982600   -8.32110660
    Ag    6.46269300    1.10794450   -5.86475020
    Ag    9.35169300    1.10794450   -5.86475020
    Ag   12.24069300    1.10794450   -5.86475020
    Ag    7.75407600    1.90606310   -3.40839380
    Ag   10.64307600    1.90606310   -3.40839380
    Ag    9.04545900    2.70418170   -0.95203730
    Ag    3.88570500   -3.37094100  -11.71204120
```

(continues on next page)

(continued from previous page)

```
    Ag    6.77470500   -3.37094100  -11.71204120
    Ag    9.66370500   -3.37094100  -11.71204120
    Ag   12.55270500   -3.37094100  -11.71204120
    Ag   15.44170500   -3.37094100  -11.71204120
    Ag    5.17708800   -2.57282240   -9.25568480
    Ag    8.06608800   -2.57282240   -9.25568480
    Ag   10.95508800   -2.57282240   -9.25568480
    Ag   13.84408800   -2.57282240   -9.25568480
    Ag    6.46847100   -1.77470390   -6.79932840
    Ag    9.35747100   -1.77470390   -6.79932840
    Ag   12.24647100   -1.77470390   -6.79932840
    Ag    7.75985400   -0.97658530   -4.34297190
    Ag   10.64885400   -0.97658530   -4.34297190
    Ag    9.05123700   -0.17846670   -1.88661550
    Ag    5.18286600   -5.45547080  -10.19026300
    Ag    8.07186600   -5.45547080  -10.19026300
    Ag   10.96086600   -5.45547080  -10.19026300
    Ag   13.84986600   -5.45547080  -10.19026300
    Ag    6.47424900   -4.65735220   -7.73390650
    Ag    9.36324900   -4.65735220   -7.73390650
    Ag   12.25224900   -4.65735220   -7.73390650
    Ag    7.76563200   -3.85923370   -5.27755010
    Ag   10.65463200   -3.85923370   -5.27755010
    Ag    9.05701500   -3.06111510   -2.82119370
    Ag    6.48002700   -7.54000060   -8.66848470
    Ag    9.36902700   -7.54000060   -8.66848470
    Ag   12.25802700   -7.54000060   -8.66848470
    Ag    7.77141000   -6.74188200   -6.21212830
    Ag   10.66041000   -6.74188200   -6.21212830
    Ag    9.06279300   -5.94376350   -3.75577180
    Ag    7.77718800   -9.62453040   -7.14670640
    Ag   10.66618800   -9.62453040   -7.14670640
    Ag    9.06857100   -8.82641190   -4.69035000
    Ag    9.07434900  -11.70906020   -5.62492820
    Ag   -1.28560500   -0.79888770  -14.75310360
    Ag    0.01155600   -2.88431030  -13.23407290
    Ag    1.30871700   -4.96973280  -11.71504230
    Ag    2.60587800   -7.05515540  -10.19601160
    Ag    3.90303900   -9.14057790   -8.67698100
    Ag    5.20020000  -11.22600050   -7.15795030
    Ag    6.49736100  -13.31142310   -5.63891960
    Ag    0.00577800   -0.00076910  -12.29674720
    Ag    1.30293900   -2.08619170  -10.77771650
    Ag    2.60010000   -4.17161430   -9.25868580
    Ag    3.89726100   -6.25703680   -7.73965520
    Ag    5.19442200   -8.34245940   -6.22062450
    Ag    6.49158300  -10.42788190   -4.70159390
    Ag    1.29716100    0.79734940   -9.84039070
    Ag    2.59432200   -1.28807310   -8.32136010
    Ag    3.89148300   -3.37349570   -6.80232940
    Ag    5.18864400   -5.45891820   -5.28329870
    Ag    6.48580500   -7.54434080   -3.76426810
    Ag    2.58854400    1.59546800   -7.38403430
    Ag    3.88570500   -0.48995450   -5.86500360
    Ag    5.18286600   -2.57537710   -4.34597300
    Ag    6.48002700   -4.66079960   -2.82694230
    Ag    3.87992700    2.39358660   -4.92767790
```

(continues on next page)

(continued from previous page)

```
     Ag    5.17708800    0.30816410   -3.40864720
     Ag    6.47424900   -1.77725850   -1.88961650
     Ag    5.17131000    3.19170520   -2.47132140
     Ag    6.46847100    1.10628270   -0.95229080
     Ag    6.46269300    3.98982380   -0.01496500
     Ag   -2.57121000   -1.59777550  -12.29425300
     Ag   -1.27404900   -3.68319800  -10.77522230
     Ag    0.02311200   -5.76862060   -9.25619170
     Ag    1.32027300   -7.85404310   -7.73716100
     Ag    2.61743400   -9.93946570   -6.21813040
     Ag    3.91459500  -12.02488820   -4.69909970
     Ag   -1.27982700   -0.79965690   -9.83789660
     Ag    0.01733400   -2.88507940   -8.31886590
     Ag    1.31449500   -4.97050200   -6.79983520
     Ag    2.61165600   -7.05592450   -5.28080460
     Ag    3.90881700   -9.14134710   -3.76177390
     Ag    0.01155600   -0.00153830   -7.38154010
     Ag    1.30871700   -2.08696080   -5.86250950
     Ag    2.60587800   -4.17238340   -4.34347880
     Ag    3.90303900   -6.25780590   -2.82444810
     Ag    1.30293900    0.79658030   -4.92518370
     Ag    2.60010000   -1.28884220   -3.40615300
     Ag    3.89726100   -3.37426480   -1.88712240
     Ag    2.59432200    1.59469890   -2.46882730
     Ag    3.89148300   -0.49072370   -0.94979660
     Ag    3.88570500    2.39281750   -0.01247080
     Ag   -3.85681500   -2.39666320   -9.83540240
     Ag   -2.55965400   -4.48208570   -8.31637170
     Ag   -1.26249300   -6.56750830   -6.79734110
     Ag    0.03466800   -8.65293080   -5.27831040
     Ag    1.33182900  -10.73835340   -3.75927980
     Ag   -2.56543200   -1.59854460   -7.37904600
     Ag   -1.26827100   -3.68396710   -5.86001530
     Ag    0.02889000   -5.76938970   -4.34098460
     Ag    1.32605100   -7.85481230   -2.82195400
     Ag   -1.27404900   -0.80042600   -4.92268950
     Ag    0.02311200   -2.88584860   -3.40365890
     Ag    1.32027300   -4.97127110   -1.88462820
     Ag    0.01733400   -0.00230740   -2.46633310
     Ag    1.31449500   -2.08773000   -0.94730240
     Ag    1.30871700    0.79581120   -0.00997670
     Ag   -5.14242000   -3.19555090   -7.37655180
     Ag   -3.84525900   -5.28097350   -5.85752110
     Ag   -2.54809800   -7.36639600   -4.33849050
     Ag   -1.25093700   -9.45181860   -2.81945980
     Ag   -3.85103700   -2.39743230   -4.92019540
     Ag   -2.55387600   -4.48285490   -3.40116470
     Ag   -1.25671500   -6.56827740   -1.88213400
     Ag   -2.55965400   -1.59931370   -2.46383890
     Ag   -1.26249300   -3.68473630   -0.94480830
     Ag   -1.26827100   -0.80119510   -0.00748250
     Ag   -6.42802500   -3.99443860   -4.91770120
     Ag   -5.13086400   -6.07986120   -3.39867050
     Ag   -3.83370300   -8.16528370   -1.87963990
     Ag   -5.13664200   -3.19632000   -2.46134480
     Ag   -3.83948100   -5.28174260   -0.94231410
     Ag   -3.84525900   -2.39820150   -0.00498830
```

(continues on next page)

(continued from previous page)

```
    Ag   -7.71363000   -4.79332640   -2.45885060
    Ag   -6.41646900   -6.87874890   -0.93981990
    Ag   -6.42224700   -3.99520780   -0.00249420
    Ag   -8.99923500   -5.59221410    0.00000000
    Ag   -2.88900000    0.00000000  -17.21195420
    Ag   -5.77800000    0.00000000  -17.21195420
    Ag   -8.66700000    0.00000000  -17.21195420
    Ag  -11.55600000    0.00000000  -17.21195420
    Ag  -14.44500000    0.00000000  -17.21195420
    Ag  -17.33400000    0.00000000  -17.21195420
    Ag  -20.22300000    0.00000000  -17.21195420
    Ag   -4.18038300    2.08950160  -18.73006600
    Ag   -7.06938300    2.08950160  -18.73006600
    Ag   -9.95838300    2.08950160  -18.73006600
    Ag  -12.84738300    2.08950160  -18.73006600
    Ag  -15.73638300    2.08950160  -18.73006600
    Ag  -18.62538300    2.08950160  -18.73006600
    Ag   -5.47176600    4.17900320  -20.24817770
    Ag   -8.36076600    4.17900320  -20.24817770
    Ag  -11.24976600    4.17900320  -20.24817770
    Ag  -14.13876600    4.17900320  -20.24817770
    Ag  -17.02776600    4.17900320  -20.24817770
    Ag   -6.76314900    6.26850480  -21.76628950
    Ag   -9.65214900    6.26850480  -21.76628950
    Ag  -12.54114900    6.26850480  -21.76628950
    Ag  -15.43014900    6.26850480  -21.76628950
    Ag   -8.05453200    8.35800630  -23.28440130
    Ag  -10.94353200    8.35800630  -23.28440130
    Ag  -13.83253200    8.35800630  -23.28440130
    Ag   -9.34591500   10.44750790  -24.80251300
    Ag  -12.23491500   10.44750790  -24.80251300
    Ag  -10.63729800   12.53700950  -26.32062480
    Ag   -4.18616100    2.09145230  -15.69970390
    Ag   -7.07516100    2.09145230  -15.69970390
    Ag   -9.96416100    2.09145230  -15.69970390
    Ag  -12.85316100    2.09145230  -15.69970390
    Ag  -15.74216100    2.09145230  -15.69970390
    Ag  -18.63116100    2.09145230  -15.69970390
    Ag   -2.58854400    4.18095390  -17.21781570
    Ag   -5.47754400    4.18095390  -17.21781570
    Ag   -8.36654400    4.18095390  -17.21781570
    Ag  -11.25554400    4.18095390  -17.21781570
    Ag  -14.14454400    4.18095390  -17.21781570
    Ag  -17.03354400    4.18095390  -17.21781570
    Ag   -3.87992700    6.27045540  -18.73592740
    Ag   -6.76892700    6.27045540  -18.73592740
    Ag   -9.65792700    6.27045540  -18.73592740
    Ag  -12.54692700    6.27045540  -18.73592740
    Ag  -15.43592700    6.27045540  -18.73592740
    Ag   -5.17131000    8.35995700  -20.25403920
    Ag   -8.06031000    8.35995700  -20.25403920
    Ag  -10.94931000    8.35995700  -20.25403920
    Ag  -13.83831000    8.35995700  -20.25403920
    Ag   -6.46269300   10.44945860  -21.77215100
    Ag   -9.35169300   10.44945860  -21.77215100
    Ag  -12.24069300   10.44945860  -21.77215100
    Ag   -7.75407600   12.53896020  -23.29026270
```

(continues on next page)

(continued from previous page)

```
     Ag  -10.64307600   12.53896020  -23.29026270
     Ag   -9.04545900   14.62846180  -24.80837450
     Ag   -5.48332200    4.18290450  -14.18745360
     Ag   -8.37232200    4.18290450  -14.18745360
     Ag  -11.26132200    4.18290450  -14.18745360
     Ag  -14.15032200    4.18290450  -14.18745360
     Ag  -17.03932200    4.18290450  -14.18745360
     Ag   -3.88570500    6.27240610  -15.70556530
     Ag   -6.77470500    6.27240610  -15.70556530
     Ag   -9.66370500    6.27240610  -15.70556530
     Ag  -12.55270500    6.27240610  -15.70556530
     Ag  -15.44170500    6.27240610  -15.70556530
     Ag   -5.17708800    8.36190770  -17.22367710
     Ag   -8.06608800    8.36190770  -17.22367710
     Ag  -10.95508800    8.36190770  -17.22367710
     Ag  -13.84408800    8.36190770  -17.22367710
     Ag   -6.46847100   10.45140930  -18.74178890
     Ag   -9.35747100   10.45140930  -18.74178890
     Ag  -12.24647100   10.45140930  -18.74178890
     Ag   -7.75985400   12.54091090  -20.25990060
     Ag  -10.64885400   12.54091090  -20.25990060
     Ag   -9.05123700   14.63041250  -21.77801240
     Ag   -6.78048300    6.27435680  -12.67520330
     Ag   -9.66948300    6.27435680  -12.67520330
     Ag  -12.55848300    6.27435680  -12.67520330
     Ag  -15.44748300    6.27435680  -12.67520330
     Ag   -5.18286600    8.36385840  -14.19331500
     Ag   -8.07186600    8.36385840  -14.19331500
     Ag  -10.96086600    8.36385840  -14.19331500
     Ag  -13.84986600    8.36385840  -14.19331500
     Ag   -6.47424900   10.45336000  -15.71142680
     Ag   -9.36324900   10.45336000  -15.71142680
     Ag  -12.25224900   10.45336000  -15.71142680
     Ag   -7.76563200   12.54286160  -17.22953860
     Ag  -10.65463200   12.54286160  -17.22953860
     Ag   -9.05701500   14.63236320  -18.74765030
     Ag   -8.07764400    8.36580910  -11.16295300
     Ag  -10.96664400    8.36580910  -11.16295300
     Ag  -13.85564400    8.36580910  -11.16295300
     Ag   -6.48002700   10.45531070  -12.68106470
     Ag   -9.36902700   10.45531070  -12.68106470
     Ag  -12.25802700   10.45531070  -12.68106470
     Ag   -7.77141000   12.54481230  -14.19917650
     Ag  -10.66041000   12.54481230  -14.19917650
     Ag   -9.06279300   14.63431380  -15.71728830
     Ag   -9.37480500   10.45726140   -9.65070270
     Ag  -12.26380500   10.45726140   -9.65070270
     Ag   -7.77718800   12.54676290  -11.16881440
     Ag  -10.66618800   12.54676290  -11.16881440
     Ag   -9.06857100   14.63626450  -12.68692620
     Ag  -10.67196600   12.54871360   -8.13845230
     Ag   -9.07434900   14.63821520   -9.65656410
     Ag   -1.30293900    6.76399240  -17.21615680
     Ag   -2.60010000    8.85310740  -15.70220840
     Ag   -3.89726100   10.94222240  -14.18825990
     Ag   -5.19442200   13.03133740  -12.67431150
     Ag   -6.49158300   15.12045250  -11.16036310
```

(continues on next page)

(continued from previous page)

```
    Ag   -2.59432200    8.85349400  -18.73426860
    Ag   -3.89148300   10.94260900  -17.22032010
    Ag   -5.18864400   13.03172400  -15.70637170
    Ag   -6.48580500   15.12083900  -14.19242330
    Ag   -3.88570500   10.94299550  -20.25238030
    Ag   -5.18286600   13.03211060  -18.73843190
    Ag   -6.48002700   15.12122560  -17.22448350
    Ag   -5.17708800   13.03249710  -21.77049210
    Ag   -6.47424900   15.12161220  -20.25654370
    Ag   -6.46847100   15.12199870  -23.28860380
    Ag   -0.01733400    9.34936810  -17.21619600
    Ag   -1.31449500   11.43848320  -15.70224760
    Ag   -2.61165600   13.52759820  -14.18829920
    Ag   -3.90881700   15.61671320  -12.67435080
    Ag   -1.30871700   11.43886970  -18.73430780
    Ag   -2.60587800   13.52798470  -17.22035940
    Ag   -3.90303900   15.61709980  -15.70641100
    Ag   -2.60010000   13.52837130  -20.25241960
    Ag   -3.89726100   15.61748630  -18.73847110
    Ag   -3.89148300   15.61787290  -21.77053130
    Ag    1.26827100   11.93474390  -17.21623530
    Ag   -0.02889000   14.02385890  -15.70228690
    Ag   -1.32605100   16.11297400  -14.18833840
    Ag   -0.02311200   14.02424550  -18.73434700
    Ag   -1.32027300   16.11336050  -17.22039860
    Ag   -1.31449500   16.11374710  -20.25245880
    Ag    2.55387600   14.52011970  -17.21627450
    Ag    1.25671500   16.60923470  -15.70232610
    Ag    1.26249300   16.60962130  -18.73438630
    Ag    3.83948100   17.10549540  -17.21631380
    Ag   -4.18038300   -0.79811860  -19.66831060
    Ag   -7.06938300   -0.79811860  -19.66831060
    Ag   -9.95838300   -0.79811860  -19.66831060
    Ag  -12.84738300   -0.79811860  -19.66831060
    Ag  -15.73638300   -0.79811860  -19.66831060
    Ag  -18.62538300   -0.79811860  -19.66831060
    Ag   -5.47176600   -1.59623720  -22.12466710
    Ag   -8.36076600   -1.59623720  -22.12466710
    Ag  -11.24976600   -1.59623720  -22.12466710
    Ag  -14.13876600   -1.59623720  -22.12466710
    Ag  -17.02776600   -1.59623720  -22.12466710
    Ag   -6.76314900   -2.39435580  -24.58102350
    Ag   -9.65214900   -2.39435580  -24.58102350
    Ag  -12.54114900   -2.39435580  -24.58102350
    Ag  -15.43014900   -2.39435580  -24.58102350
    Ag   -8.05453200   -3.19247430  -27.03737990
    Ag  -10.94353200   -3.19247430  -27.03737990
    Ag  -13.83253200   -3.19247430  -27.03737990
    Ag   -9.34591500   -3.99059290  -29.49373640
    Ag  -12.23491500   -3.99059290  -29.49373640
    Ag  -10.63729800   -4.78871150  -31.95009280
    Ag   -2.58854400    1.28641120  -21.19008890
    Ag   -5.47754400    1.28641120  -21.19008890
    Ag   -8.36654400    1.28641120  -21.19008890
    Ag  -11.25554400    1.28641120  -21.19008890
    Ag  -14.14454400    1.28641120  -21.19008890
    Ag  -17.03354400    1.28641120  -21.19008890
```

(continues on next page)

(continued from previous page)

```
    Ag   -3.87992700    0.48829260  -23.64644530
    Ag   -6.76892700    0.48829260  -23.64644530
    Ag   -9.65792700    0.48829260  -23.64644530
    Ag  -12.54692700    0.48829260  -23.64644530
    Ag  -15.43592700    0.48829260  -23.64644530
    Ag   -5.17131000   -0.30982600  -26.10280180
    Ag   -8.06031000   -0.30982600  -26.10280180
    Ag  -10.94931000   -0.30982600  -26.10280180
    Ag  -13.83831000   -0.30982600  -26.10280180
    Ag   -6.46269300   -1.10794450  -28.55915820
    Ag   -9.35169300   -1.10794450  -28.55915820
    Ag  -12.24069300   -1.10794450  -28.55915820
    Ag   -7.75407600   -1.90606310  -31.01551460
    Ag  -10.64307600   -1.90606310  -31.01551460
    Ag   -9.04545900   -2.70418170  -33.47187110
    Ag   -3.88570500    3.37094100  -22.71186720
    Ag   -6.77470500    3.37094100  -22.71186720
    Ag   -9.66370500    3.37094100  -22.71186720
    Ag  -12.55270500    3.37094100  -22.71186720
    Ag  -15.44170500    3.37094100  -22.71186720
    Ag   -5.17708800    2.57282240  -25.16822360
    Ag   -8.06608800    2.57282240  -25.16822360
    Ag  -10.95508800    2.57282240  -25.16822360
    Ag  -13.84408800    2.57282240  -25.16822360
    Ag   -6.46847100    1.77470390  -27.62458000
    Ag   -9.35747100    1.77470390  -27.62458000
    Ag  -12.24647100    1.77470390  -27.62458000
    Ag   -7.75985400    0.97658530  -30.08093650
    Ag  -10.64885400    0.97658530  -30.08093650
    Ag   -9.05123700    0.17846670  -32.53729290
    Ag   -5.18286600    5.45547080  -24.23364540
    Ag   -8.07186600    5.45547080  -24.23364540
    Ag  -10.96086600    5.45547080  -24.23364540
    Ag  -13.84986600    5.45547080  -24.23364540
    Ag   -6.47424900    4.65735220  -26.69000190
    Ag   -9.36324900    4.65735220  -26.69000190
    Ag  -12.25224900    4.65735220  -26.69000190
    Ag   -7.76563200    3.85923370  -29.14635830
    Ag  -10.65463200    3.85923370  -29.14635830
    Ag   -9.05701500    3.06111510  -31.60271470
    Ag   -6.48002700    7.54000060  -25.75542370
    Ag   -9.36902700    7.54000060  -25.75542370
    Ag  -12.25802700    7.54000060  -25.75542370
    Ag   -7.77141000    6.74188200  -28.21178010
    Ag  -10.66041000    6.74188200  -28.21178010
    Ag   -9.06279300    5.94376350  -30.66813660
    Ag   -7.77718800    9.62453040  -27.27720200
    Ag  -10.66618800    9.62453040  -27.27720200
    Ag   -9.06857100    8.82641190  -29.73355840
    Ag   -9.07434900   11.70906020  -28.79898020
    Ag   -1.30293900    2.08619170  -23.64619190
    Ag   -2.60010000    4.17161430  -25.16522260
    Ag   -3.89726100    6.25703680  -26.68425320
    Ag   -5.19442200    8.34245940  -28.20328390
    Ag   -6.49158300   10.42788190  -29.72231450
    Ag   -2.59432200    1.28807310  -26.10254830
    Ag   -3.89148300    3.37349570  -27.62157900
```

(continues on next page)

(continued from previous page)

```
    Ag   -5.18864400    5.45891820  -29.14060970
    Ag   -6.48580500    7.54434080  -30.65964030
    Ag   -3.88570500    0.48995450  -28.55890480
    Ag   -5.18286600    2.57537710  -30.07793540
    Ag   -6.48002700    4.66079960  -31.59696610
    Ag   -5.17708800   -0.30816410  -31.01526120
    Ag   -6.47424900    1.77725850  -32.53429190
    Ag   -6.46847100   -1.10628270  -33.47161760
    Ag   -0.01733400    2.88507940  -26.10504250
    Ag   -1.31449500    4.97050200  -27.62407320
    Ag   -2.61165600    7.05592450  -29.14310380
    Ag   -3.90881700    9.14134710  -30.66213450
    Ag   -1.30871700    2.08696080  -28.56139890
    Ag   -2.60587800    4.17238340  -30.08042960
    Ag   -3.90303900    6.25780590  -31.59946030
    Ag   -2.60010000    1.28884220  -31.01775540
    Ag   -3.89726100    3.37426480  -32.53678600
    Ag   -3.89148300    0.49072370  -33.47411180
    Ag    1.26827100    3.68396710  -28.56389310
    Ag   -0.02889000    5.76938970  -30.08292380
    Ag   -1.32605100    7.85481230  -31.60195440
    Ag   -0.02311200    2.88584860  -31.02024950
    Ag   -1.32027300    4.97127110  -32.53928020
    Ag   -1.31449500    2.08773000  -33.47660600
    Ag    2.55387600    4.48285490  -31.02274370
    Ag    1.25671500    6.56827740  -32.54177440
    Ag    1.26249300    3.68473630  -33.47910010
    Ag    3.83948100    5.28174260  -33.48159430
    Ag   -4.18038300   -2.58276600  -17.21195420
    Ag   -7.06938300   -2.58276600  -17.21195420
    Ag   -9.95838300   -2.58276600  -17.21195420
    Ag  -12.84738300   -2.58276600  -17.21195420
    Ag  -15.73638300   -2.58276600  -17.21195420
    Ag  -18.62538300   -2.58276600  -17.21195420
    Ag   -5.47176600   -5.16553200  -17.21195420
    Ag   -8.36076600   -5.16553200  -17.21195420
    Ag  -11.24976600   -5.16553200  -17.21195420
    Ag  -14.13876600   -5.16553200  -17.21195420
    Ag  -17.02776600   -5.16553200  -17.21195420
    Ag   -6.76314900   -7.74829800  -17.21195420
    Ag   -9.65214900   -7.74829800  -17.21195420
    Ag  -12.54114900   -7.74829800  -17.21195420
    Ag  -15.43014900   -7.74829800  -17.21195420
    Ag   -8.05453200  -10.33106400  -17.21195420
    Ag  -10.94353200  -10.33106400  -17.21195420
    Ag  -13.83253200  -10.33106400  -17.21195420
    Ag   -9.34591500  -12.91383000  -17.21195420
    Ag  -12.23491500  -12.91383000  -17.21195420
    Ag  -10.63729800  -15.49659600  -17.21195420
    Ag   -2.58854400   -3.38590800  -19.66471520
    Ag   -5.47754400   -3.38590800  -19.66471520
    Ag   -8.36654400   -3.38590800  -19.66471520
    Ag  -11.25554400   -3.38590800  -19.66471520
    Ag  -14.14454400   -3.38590800  -19.66471520
    Ag  -17.03354400   -3.38590800  -19.66471520
    Ag   -3.87992700   -5.96867400  -19.66471520
    Ag   -6.76892700   -5.96867400  -19.66471520
```

(continues on next page)

(continued from previous page)

```
     Ag   -9.65792700   -5.96867400  -19.66471520
     Ag  -12.54692700   -5.96867400  -19.66471520
     Ag  -15.43592700   -5.96867400  -19.66471520
     Ag   -5.17131000   -8.55144000  -19.66471520
     Ag   -8.06031000   -8.55144000  -19.66471520
     Ag  -10.94931000   -8.55144000  -19.66471520
     Ag  -13.83831000   -8.55144000  -19.66471520
     Ag   -6.46269300  -11.13420600  -19.66471520
     Ag   -9.35169300  -11.13420600  -19.66471520
     Ag  -12.24069300  -11.13420600  -19.66471520
     Ag   -7.75407600  -13.71697200  -19.66471520
     Ag  -10.64307600  -13.71697200  -19.66471520
     Ag   -9.04545900  -16.29973800  -19.66471520
     Ag   -3.88570500   -4.18905000  -22.11747620
     Ag   -6.77470500   -4.18905000  -22.11747620
     Ag   -9.66370500   -4.18905000  -22.11747620
     Ag  -12.55270500   -4.18905000  -22.11747620
     Ag  -15.44170500   -4.18905000  -22.11747620
     Ag   -5.17708800   -6.77181600  -22.11747620
     Ag   -8.06608800   -6.77181600  -22.11747620
     Ag  -10.95508800   -6.77181600  -22.11747620
     Ag  -13.84408800   -6.77181600  -22.11747620
     Ag   -6.46847100   -9.35458200  -22.11747620
     Ag   -9.35747100   -9.35458200  -22.11747620
     Ag  -12.24647100   -9.35458200  -22.11747620
     Ag   -7.75985400  -11.93734800  -22.11747620
     Ag  -10.64885400  -11.93734800  -22.11747620
     Ag   -9.05123700  -14.52011400  -22.11747620
     Ag   -5.18286600   -4.99219200  -24.57023720
     Ag   -8.07186600   -4.99219200  -24.57023720
     Ag  -10.96086600   -4.99219200  -24.57023720
     Ag  -13.84986600   -4.99219200  -24.57023720
     Ag   -6.47424900   -7.57495800  -24.57023720
     Ag   -9.36324900   -7.57495800  -24.57023720
     Ag  -12.25224900   -7.57495800  -24.57023720
     Ag   -7.76563200  -10.15772400  -24.57023720
     Ag  -10.65463200  -10.15772400  -24.57023720
     Ag   -9.05701500  -12.74049000  -24.57023720
     Ag   -6.48002700   -5.79533400  -27.02299820
     Ag   -9.36902700   -5.79533400  -27.02299820
     Ag  -12.25802700   -5.79533400  -27.02299820
     Ag   -7.77141000   -8.37810000  -27.02299820
     Ag  -10.66041000   -8.37810000  -27.02299820
     Ag   -9.06279300  -10.96086600  -27.02299820
     Ag   -7.77718800   -6.59847600  -29.47575920
     Ag  -10.66618800   -6.59847600  -29.47575920
     Ag   -9.06857100   -9.18124200  -29.47575920
     Ag   -9.07434900   -7.40161800  -31.92852020
     Ag   -1.30293900   -5.47465500  -21.18432920
     Ag   -2.60010000   -6.27490800  -23.63709020
     Ag   -3.89726100   -7.07516100  -26.08985120
     Ag   -5.19442200   -7.87541400  -28.54261220
     Ag   -6.49158300   -8.67566700  -30.99537320
     Ag   -2.59432200   -8.05742100  -21.18432920
     Ag   -3.89148300   -8.85767400  -23.63709020
     Ag   -5.18864400   -9.65792700  -26.08985120
     Ag   -6.48580500  -10.45818000  -28.54261220
```

(continues on next page)

(continued from previous page)

```
     Ag   -3.88570500  -10.64018700  -21.18432920
     Ag   -5.18286600  -11.44044000  -23.63709020
     Ag   -6.48002700  -12.24069300  -26.08985120
     Ag   -5.17708800  -13.22295300  -21.18432920
     Ag   -6.47424900  -14.02320600  -23.63709020
     Ag   -6.46847100  -15.80571900  -21.18432920
     Ag   -0.01733400   -7.56629100  -22.70394320
     Ag   -1.31449500   -8.36654400  -25.15670420
     Ag   -2.61165600   -9.16679700  -27.60946520
     Ag   -3.90881700   -9.96705000  -30.06222620
     Ag   -1.30871700  -10.14905700  -22.70394320
     Ag   -2.60587800  -10.94931000  -25.15670420
     Ag   -3.90303900  -11.74956300  -27.60946520
     Ag   -2.60010000  -12.73182300  -22.70394320
     Ag   -3.89726100  -13.53207600  -25.15670420
     Ag   -3.89148300  -15.31458900  -22.70394320
     Ag    1.26827100   -9.65792700  -24.22355720
     Ag   -0.02889000  -10.45818000  -26.67631820
     Ag   -1.32605100  -11.25843300  -29.12907920
     Ag   -0.02311200  -12.24069300  -24.22355720
     Ag   -1.32027300  -13.04094600  -26.67631820
     Ag   -1.31449500  -14.82345900  -24.22355720
     Ag    2.55387600  -11.74956300  -25.74317120
     Ag    1.25671500  -12.54981600  -28.19593220
     Ag    1.26249300  -14.33232900  -25.74317120
     Ag    3.83948100  -13.84119900  -27.26278520
     Ag   -4.18038300   -0.79811860  -14.75559780
     Ag   -7.06938300   -0.79811860  -14.75559780
     Ag   -9.95838300   -0.79811860  -14.75559780
     Ag  -12.84738300   -0.79811860  -14.75559780
     Ag  -15.73638300   -0.79811860  -14.75559780
     Ag  -18.62538300   -0.79811860  -14.75559780
     Ag   -5.47176600   -1.59623720  -12.29924130
     Ag   -8.36076600   -1.59623720  -12.29924130
     Ag  -11.24976600   -1.59623720  -12.29924130
     Ag  -14.13876600   -1.59623720  -12.29924130
     Ag  -17.02776600   -1.59623720  -12.29924130
     Ag   -6.76314900   -2.39435580   -9.84288490
     Ag   -9.65214900   -2.39435580   -9.84288490
     Ag  -12.54114900   -2.39435580   -9.84288490
     Ag  -15.43014900   -2.39435580   -9.84288490
     Ag   -8.05453200   -3.19247430   -7.38652850
     Ag  -10.94353200   -3.19247430   -7.38652850
     Ag  -13.83253200   -3.19247430   -7.38652850
     Ag   -9.34591500   -3.99059290   -4.93017200
     Ag  -12.23491500   -3.99059290   -4.93017200
     Ag  -10.63729800   -4.78871150   -2.47381560
     Ag   -2.58854400   -3.37901740  -14.74970920
     Ag   -5.47754400   -3.37901740  -14.74970920
     Ag   -8.36654400   -3.37901740  -14.74970920
     Ag  -11.25554400   -3.37901740  -14.74970920
     Ag  -14.14454400   -3.37901740  -14.74970920
     Ag  -17.03354400   -3.37901740  -14.74970920
     Ag   -3.87992700   -4.17713600  -12.29335270
     Ag   -6.76892700   -4.17713600  -12.29335270
     Ag   -9.65792700   -4.17713600  -12.29335270
     Ag  -12.54692700   -4.17713600  -12.29335270
```

(continues on next page)

(continued from previous page)

```
    Ag  -15.43592700   -4.17713600  -12.29335270
    Ag   -5.17131000   -4.97525460   -9.83699630
    Ag   -8.06031000   -4.97525460   -9.83699630
    Ag  -10.94931000   -4.97525460   -9.83699630
    Ag  -13.83831000   -4.97525460   -9.83699630
    Ag   -6.46269300   -5.77337320   -7.38063990
    Ag   -9.35169300   -5.77337320   -7.38063990
    Ag  -12.24069300   -5.77337320   -7.38063990
    Ag   -7.75407600   -6.57149180   -4.92428340
    Ag  -10.64307600   -6.57149180   -4.92428340
    Ag   -9.04545900   -7.36961040   -2.46792700
    Ag   -3.88570500   -5.95991630  -14.74382060
    Ag   -6.77470500   -5.95991630  -14.74382060
    Ag   -9.66370500   -5.95991630  -14.74382060
    Ag  -12.55270500   -5.95991630  -14.74382060
    Ag  -15.44170500   -5.95991630  -14.74382060
    Ag   -5.17708800   -6.75803490  -12.28746410
    Ag   -8.06608800   -6.75803490  -12.28746410
    Ag  -10.95508800   -6.75803490  -12.28746410
    Ag  -13.84408800   -6.75803490  -12.28746410
    Ag   -6.46847100   -7.55615350   -9.83110770
    Ag   -9.35747100   -7.55615350   -9.83110770
    Ag  -12.24647100   -7.55615350   -9.83110770
    Ag   -7.75985400   -8.35427210   -7.37475130
    Ag  -10.64885400   -8.35427210   -7.37475130
    Ag   -9.05123700   -9.15239070   -4.91839480
    Ag   -5.18286600   -8.54081520  -14.73793200
    Ag   -8.07186600   -8.54081520  -14.73793200
    Ag  -10.96086600   -8.54081520  -14.73793200
    Ag  -13.84986600   -8.54081520  -14.73793200
    Ag   -6.47424900   -9.33893370  -12.28157550
    Ag   -9.36324900   -9.33893370  -12.28157550
    Ag  -12.25224900   -9.33893370  -12.28157550
    Ag   -7.76563200  -10.13705230   -9.82521910
    Ag  -10.65463200  -10.13705230   -9.82521910
    Ag   -9.05701500  -10.93517090   -7.36886270
    Ag   -6.48002700  -11.12171400  -14.73204340
    Ag   -9.36902700  -11.12171400  -14.73204340
    Ag  -12.25802700  -11.12171400  -14.73204340
    Ag   -7.77141000  -11.91983260  -12.27568690
    Ag  -10.66041000  -11.91983260  -12.27568690
    Ag   -9.06279300  -12.71795120   -9.81933050
    Ag   -7.77718800  -13.70261290  -14.72615480
    Ag  -10.66618800  -13.70261290  -14.72615480
    Ag   -9.06857100  -14.50073150  -12.26979830
    Ag   -9.07434900  -16.28351170  -14.72026620
    Ag   -1.30293900   -5.46971460  -13.23277930
    Ag   -2.60010000   -8.04972070  -13.22963830
    Ag   -3.89726100  -10.62972680  -13.22649730
    Ag   -5.19442200  -13.20973290  -13.22335630
    Ag   -6.49158300  -15.78973900  -13.22021530
    Ag   -2.59432200   -6.26783310  -10.77642280
    Ag   -3.89148300   -8.84783930  -10.77328180
    Ag   -5.18864400  -11.42784540  -10.77014080
    Ag   -6.48580500  -14.00785150  -10.76699980
    Ag   -3.88570500   -7.06595170   -8.32006640
    Ag   -5.18286600   -9.64595780   -8.31692540
```

(continues on next page)

(continued from previous page)

```
    Ag   -6.48002700  -12.22596400   -8.31378440
    Ag   -5.17708800   -7.86407030   -5.86371000
    Ag   -6.47424900  -10.44407640   -5.86056900
    Ag   -6.46847100   -8.66218890   -3.40735350
    Ag   -0.01733400   -7.56130440  -11.71310180
    Ag   -1.31449500  -10.14131050  -11.70996080
    Ag   -2.61165600  -12.72131660  -11.70681980
    Ag   -3.90881700  -15.30132280  -11.70367880
    Ag   -1.30871700   -8.35942300   -9.25674530
    Ag   -2.60587800  -10.93942910   -9.25360430
    Ag   -3.90303900  -13.51943520   -9.25046330
    Ag   -2.60010000   -9.15754160   -6.80038890
    Ag   -3.89726100  -11.73754770   -6.79724790
    Ag   -3.89148300   -9.95566020   -4.34403250
    Ag    1.26827100   -9.65289430  -10.19342430
    Ag   -0.02889000  -12.23290040  -10.19028330
    Ag   -1.32605100  -14.81290650  -10.18714230
    Ag   -0.02311200  -10.45101290   -7.73706780
    Ag   -1.32027300  -13.03101900   -7.73392680
    Ag   -1.31449500  -11.24913150   -5.28071140
    Ag    2.55387600  -11.74448420   -8.67374680
    Ag    1.25671500  -14.32449030   -8.67060580
    Ag    1.26249300  -12.54260270   -6.21739030
    Ag    3.83948100  -13.83607400   -7.15406930
    Ag   -2.58854400    1.29756040  -13.23744210
    Ag   -5.47754400    1.29756040  -13.23744210
    Ag   -8.36654400    1.29756040  -13.23744210
    Ag  -11.25554400    1.29756040  -13.23744210
    Ag  -14.14454400    1.29756040  -13.23744210
    Ag  -17.03354400    1.29756040  -13.23744210
    Ag   -3.87992700    3.38706200  -11.71933030
    Ag   -6.76892700    3.38706200  -11.71933030
    Ag   -9.65792700    3.38706200  -11.71933030
    Ag  -12.54692700    3.38706200  -11.71933030
    Ag  -15.43592700    3.38706200  -11.71933030
    Ag   -5.17131000    5.47656350  -10.20121850
    Ag   -8.06031000    5.47656350  -10.20121850
    Ag  -10.94931000    5.47656350  -10.20121850
    Ag  -13.83831000    5.47656350  -10.20121850
    Ag   -6.46269300    7.56606510   -8.68310680
    Ag   -9.35169300    7.56606510   -8.68310680
    Ag  -12.24069300    7.56606510   -8.68310680
    Ag   -7.75407600    9.65556670   -7.16499500
    Ag  -10.64307600    9.65556670   -7.16499500
    Ag   -9.04545900   11.74506830   -5.64688330
    Ag   -3.88570500    0.50561920  -10.78104170
    Ag   -6.77470500    0.50561920  -10.78104170
    Ag   -9.66370500    0.50561920  -10.78104170
    Ag  -12.55270500    0.50561920  -10.78104170
    Ag  -15.44170500    0.50561920  -10.78104170
    Ag   -5.17708800    2.59512070   -9.26293000
    Ag   -8.06608800    2.59512070   -9.26293000
    Ag  -10.95508800    2.59512070   -9.26293000
    Ag  -13.84408800    2.59512070   -9.26293000
    Ag   -6.46847100    4.68462230   -7.74481820
    Ag   -9.35747100    4.68462230   -7.74481820
    Ag  -12.24647100    4.68462230   -7.74481820
```

(continues on next page)

(continued from previous page)

```
    Ag   -7.75985400    6.77412390   -6.22670640
    Ag  -10.64885400    6.77412390   -6.22670640
    Ag   -9.05123700    8.86362550   -4.70859470
    Ag   -5.18286600   -0.28632210   -8.32464140
    Ag   -8.07186600   -0.28632210   -8.32464140
    Ag  -10.96086600   -0.28632210   -8.32464140
    Ag  -13.84986600   -0.28632210   -8.32464140
    Ag   -6.47424900    1.80317950   -6.80652960
    Ag   -9.36324900    1.80317950   -6.80652960
    Ag  -12.25224900    1.80317950   -6.80652960
    Ag   -7.76563200    3.89268110   -5.28841780
    Ag  -10.65463200    3.89268110   -5.28841780
    Ag   -9.05701500    5.98218270   -3.77030610
    Ag   -6.48002700   -1.07826330   -5.86824100
    Ag   -9.36902700   -1.07826330   -5.86824100
    Ag  -12.25802700   -1.07826330   -5.86824100
    Ag   -7.77141000    1.01123830   -4.35012920
    Ag  -10.66041000    1.01123830   -4.35012920
    Ag   -9.06279300    3.10073990   -2.83201750
    Ag   -7.77718800   -1.87020450   -3.41184070
    Ag  -10.66618800   -1.87020450   -3.41184070
    Ag   -9.06857100    0.21929710   -1.89372890
    Ag   -9.07434900   -2.66214570   -0.95544030
    Ag   -1.30293900    2.09418550  -10.78031380
    Ag   -2.60010000    1.29990700   -8.32561160
    Ag   -3.89726100    0.50562860   -5.87090940
    Ag   -5.19442200   -0.28864990   -3.41620710
    Ag   -6.49158300   -1.08292840   -0.96150490
    Ag   -2.59432200    4.18368710   -9.26220210
    Ag   -3.89148300    3.38940860   -6.80749980
    Ag   -5.18864400    2.59513010   -4.35279760
    Ag   -6.48580500    1.80085170   -1.89809540
    Ag   -3.88570500    6.27318870   -7.74409030
    Ag   -5.18286600    5.47891020   -5.28938810
    Ag   -6.48002700    4.68463170   -2.83468580
    Ag   -5.17708800    8.36269030   -6.22597860
    Ag   -6.47424900    7.56841180   -3.77127630
    Ag   -6.46847100   10.45219180   -4.70786680
    Ag   -0.01733400    2.89314790   -8.32148750
    Ag   -1.31449500    2.09886940   -5.86678530
    Ag   -2.61165600    1.30459090   -3.41208300
    Ag   -3.90881700    0.51031250   -0.95738080
    Ag   -1.30871700    4.98264940   -6.80337570
    Ag   -2.60587800    4.18837100   -4.34867350
    Ag   -3.90303900    3.39409250   -1.89397120
    Ag   -2.60010000    7.07215100   -5.28526400
    Ag   -3.89726100    6.27787260   -2.83056170
    Ag   -3.89148300    9.16165260   -3.76715220
    Ag    1.26827100    3.69211020   -5.86266120
    Ag   -0.02889000    2.89783180   -3.40795890
    Ag   -1.32605100    2.10355330   -0.95325670
    Ag   -0.02311200    5.78161180   -4.34454940
    Ag   -1.32027300    4.98733340   -1.88984710
    Ag   -1.31449500    7.87111340   -2.82643760
    Ag    2.55387600    4.49107260   -3.40383480
    Ag    1.25671500    3.69679410   -0.94913260
    Ag    1.26249300    6.58057420   -1.88572300
```

(continues on next page)

(continued from previous page)

```
    Ag    3.83948100    5.29003500   -0.94500850
    SUBEND
  end
EndEngine

eor
```

14.2.8 QM/FQ(Fμ): Quantum Mechanics / Fluctuating Charges (and Fluctuating Dipoles)

Example: QM/FQ(Fμ): 2-Methyloxirane (QM) in Water

Two examples: a QM/FQ calculation and a QM/FQFμ calculation.

Download FQQM_scf.run

```
#!/bin/bash

# QMFQ solvation calculation

$AMSBIN/ams << eor
system
  Atoms
    C     1.32246     0.46994    -0.44553
    C     0.03851     0.37791     0.32993
    O    -1.13145     0.98673    -0.28822
    C    -1.09292    -0.44743    -0.11660
    H    -0.99733    -1.01009    -1.04597
    H    -1.81522    -0.83104     0.60631
    H     0.10287     0.59921     1.39924
    H     2.07829    -0.18966     0.00080
    H     1.71275     1.49611    -0.42640
    H     1.17353     0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  basis
     type TZP
     core none
  End
  symmetry NOSYM
  title QM/FQ energy calculation
  xc
    gga PBE
  End
  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00
```

(continues on next page)

(continued from previous page)

```
   AtomType
     Symbol O
     Chi 0.189194
     Eta 0.523700
   SubEnd
   AtomType
     Symbol H
     Chi 0.012767
     Eta 0.537512
   SubEnd
   Coords
   O     6.179    -4.919    2.699   1
   H     7.008    -4.361    2.695   1
   H     5.687    -4.779    3.558   1
   O     3.341    2.501    5.245   2
   H     2.813    2.455    6.093   2
   H     3.93    1.695    5.175   2
   O     2.692    8.795    -4.87   3
   H     3.592    8.369    -4.781   3
   H     2.444    9.232    -4.005   3
   O     -2.039    5.881    6.772   4
   H     -3.025    6.045    6.742   4
   H     -1.597    6.604    7.301   4
   O     4.733    -3.969    5.277   5
   H     3.944    -3.36    5.362   5
   H     4.456    -4.904    5.497   5
   O     -0.143    3.368    -8.787   6
   H     0.262    2.454    -8.8   6
   H     -0.588    3.548    -9.665   6
   O     0.957    0.844    -8.505   7
   H     1.915    0.563    -8.573   7
   H     0.399    0.276    -9.11   7
   O     -2.971    -5.482    6.927   8
   H     -3.616    -6.141    7.313   8
   H     -3.473    -4.692    6.574   8
   O     3.049    3.883    -9.821   9
   H     2.387    4.418    -9.297   9
   H     2.574    3.376    -10.54   9
   O     6.079    8.112    2.853   10
   H     6.635    7.88    3.651   10
   H     5.245    7.56    2.855   10
   O     3.022    -6.207    8.919   11
   H     2.452    -6.216    9.74   11
   H     2.653    -5.546    8.265   11
   O     1.49    1.433    10.208   12
   H     2.352    1.441    10.715   12
   H     0.73    1.581    10.84   12
   O     -2.472    4.816    -5.151   13
   H     -1.53    5.086    -5.349   13
   H     -2.91    4.492    -5.99   13
   O     -6.138    7.963    -4.272   14
   H     -6.292    7.128    -3.744   14
   H     -5.371    7.824    -4.898   14
   O     6.384    -7.3    0.928   15
   H     5.462    -7.396    0.554   15
   H     6.379    -6.615    1.657   15
```

(continues on next page)

(continued from previous page)

```
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
    H     -10.69    -5.035    3.321   18
    O     -3.63    -2.187    -1.384   19
    H     -2.767    -2.462    -1.807   19
    H     -3.822    -1.231    -1.607   19
    O     1.942    -6.524    -8.737   20
    H     0.996    -6.22    -8.852   20
    H     2.391    -6.563    -9.63   20
    O     -3.547    2.764    0.052   21
    H     -3.23    1.901    -0.342   21
    H     -2.83    3.455    -0.038   21
    O     1.904    8.539    -7.933   22
    H     2.085    8.395    -6.96   22
    H     0.923    8.447    -8.106   22
    O     5.07    6.46    -6.979   23
    H     5.462    5.945    -6.216   23
    H     5.106    7.438    -6.774   23
    O     1.317    1.858    -4.009   24
    H     0.789    2.105    -4.821   24
    H     1.565    0.891    -4.054   24
    O     -3.466    6.667    1.722   25
    H     -2.758    6.783    1.026   25
    H     -3.233    5.891    2.308   25
    O     9.374    -0.284    -4.021   26
    H     9.226    -0.462    -3.048   26
    H     8.904    0.56    -4.28   26
    O     -1.567    -0.699    -9.148   27
    H     -1.459    -1.69    -9.234   27
    H     -2.068    -0.348    -9.939   27
    O     2.355    -6.756    5.05   28
    H     3.312    -6.948    5.27   28
    H     2.119    -5.837    5.367   28
    O     -11.241    2.669    -0.446   29
    H     -10.442    2.227    -0.039   29
    H     -12.069    2.351    0.017   29
    O     -0.519    2.829    -2.219   30
    H     0.293    2.675    -2.782   30
    H     -0.756    1.984    -1.738   30
    O     2.736    2.528    1.949   31
    H     3.367    2.379    2.711   31
    H     1.894    2.946    2.29   31
    O     -3.662    -3.579    1.114   32
    H     -2.722    -3.33    1.349   32
    H     -3.879    -3.238    0.2   32
    O     -4.354    -0.05    9.809   33
    H     -4.059    -0.99    9.982   33
    H     -4.437    0.098    8.824   33
    O     -8.16    -1.139    0.546   34
    H     -8.636    -1.849    1.066   34
    H     -7.667    -1.558    -0.216   34
```

(continues on next page)

(continued from previous page)

```
    O     -1.843    -1.194    6.421   35
    H     -1.115    -0.876    7.029   35
    H     -1.582    -1.022    5.471   35
    O     -1.16    -3.054    2.421   36
    H     -0.235    -2.945    2.057   36
    H     -1.147    -3.716    3.171   36
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
    H     0.061    1.437    -6.716   37
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
```

(continues on next page)

(continued from previous page)

```
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
    O     -0.792    1.114    -11.733   56
    H     -0.455    0.182    -11.865   56
    H     -0.023    1.721    -11.532   56
    O     9.721    -2.916    -1.66   57
    H     9.417    -2.736    -2.596   57
    H     9.34    -3.787    -1.348   57
    O     -7.087    -4.974    -5.981   58
    H     -7.388    -4.844    -6.925   58
    H     -6.088    -4.953    -5.943   58
    O     4.858    8.204    6.274   59
    H     4.557    8.898    6.929   59
    H     4.59    7.298    6.6   59
    O     -5.74    4.317    -0.952   60
    H     -4.795    3.992    -0.963   60
    H     -5.777    5.221    -0.525   60
    O     4.091    6.159    -2.019   61
    H     3.956    5.352    -1.444   61
    H     3.21    6.456    -2.388   61
    O     0.259    -1.583    -11.55   62
    H     0.456    -2.193    -10.783   62
    H     1.112    -1.18    -11.882   62
    O     0.167    8.19    -1.584   63
    H     -0.233    8.98    -2.049   63
    H     0.999    7.909    -2.062   63
    O     5.904    5.593    -4.57   64
    H     5.781    4.816    -3.953   64
    H     5.595    6.43    -4.118   64
    O     1.853    2.264    7.653   65
    H     2.031    1.992    8.599   65
    H     0.866    2.341    7.509   65
    O     -3.917    -8.529    3.889   66
    H     -3.102    -8.099    3.5   66
    H     -3.747    -8.76    4.847   66
    O     -5.078    7.76    3.832   67
    H     -4.485    7.317    3.159   67
    H     -5.5    7.064    4.413   67
    O     -5.333    -7.661    -6.49   68
    H     -5.485    -7.46    -7.457   68
    H     -4.51    -7.187    -6.176   68
    O     1.603    9.353    1.706   69
    H     1.306    8.406    1.576   69
    H     2.011    9.452    2.614   69
    O     -8.588    7.768    -2.072   70
    H     -8.096    8.409    -1.483   70
    H     -8.101    6.895    -2.096   70
    O     8.949    -5.62    0.186   71
    H     8.046    -5.232    0.372   71
    H     9.636    -5.141    0.731   71
    O     -2.56    -3.378    -9.638   72
    H     -2.169    -4.295    -9.725   72
    H     -2.903    -3.077    -10.528   72
```

(continues on next page)

(continued from previous page)

```
    O     4.132    5.755    7.273   73
    H     3.367    5.418    7.822   73
    H     4.613    4.982    6.859   73
    O     -8.001    -0.706    -7.091   74
    H     -8.965    -0.793    -7.341   74
    H     -7.795    0.252    -6.889   74
    O     1.985    -9.965    1.717   75
    H     1.164    -10.041    1.152   75
    H     1.726    -9.72    2.651   75
    O     -0.889    -3.751    7.35   76
    H     -1.637    -4.413    7.286   76
    H     -1.14    -2.918    6.857   76
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
```

(continues on next page)

(continued from previous page)

```
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
    H     -3.233    7.809    -4.93   94
    O     -2.693    4.73    3.854   95
    H     -2.342    3.925    3.376   95
    H     -3.179    4.443    4.679   95
    O     1.001    -0.574    4.81   96
    H     0.553    0.313    4.696   96
    H     1.648    -0.72    4.062   96
    O     -7.658    3.829    -5.601   97
    H     -7.06    3.029    -5.656   97
    H     -8.518    3.635    -6.073   97
    O     1.26    6.271    8.504   98
    H     1.441    6.387    7.527   98
    H     0.355    6.637    8.721   98
    O     -2.775    -9.109    0.291   99
    H     -2.86    -8.211    0.723   99
    H     -1.809    -9.316    0.139   99
    O     3.256    -1.013    2.987   100
    H     3.539    -1.259    2.06   100
    H     3.855    -0.293    3.338   100
    O     0.21    -0.312    8.228   101
    H     1.026    -0.659    7.765   101
    H     0.428    0.548    8.688   101
    O     5.422    -0.784    -1.655   102
    H     5.497    -0.764    -2.652   102
    H     4.972    0.05    -1.337   102
    O     -1.59    -4.154    -5.957   103
    H     -1.997    -4.976    -5.559   103
    H     -1.274    -3.551    -5.225   103
    O     0.51    -8.72    3.751   104
    H     1.317    -8.458    4.28   104
    H     -0.042    -9.364    4.281   104
    O     -2.594    -6.607    -5.182   105
    H     -3.233    -6.683    -4.416   105
    H     -1.761    -7.12    -4.974   105
    O     -5.857    7.025    -0.001   106
    H     -4.915    7.129    0.32   106
    H     -6.458    7.622    0.53   106
    O     -0.705    -2.368    10.335   107
    H     -0.302    -2.039    9.481   107
    H     -1.387    -3.071    10.133   107
    O     -0.613    6.667    3.919   108
    H     -0.003    6.582    4.707   108
    H     -1.404    6.068    4.044   108
    O     3.935    -9.377    -3.564   109
    H     4.784    -8.856    -3.476   109
    H     3.704    -9.789    -2.683   109
    O     6.435    -3.938    -2.832   110
    H     7.169    -3.555    -3.391   110
    H     6.591    -3.706    -1.872   110
```

(continues on next page)

(continued from previous page)

```
    O     6.896    8.206    0.362   111
    H     6.777    8.065    1.345   111
    H     6.022    8.469    -0.045   111
    O     -0.931    -3.296    -2.501   112
    H     -0.842    -2.655    -3.264   112
    H     -0.122    -3.232    -1.917   112
    O     -2.26    -0.566    -6.382   113
    H     -3.181    -0.898    -6.176   113
    H     -2.061    -0.717    -7.35   113
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
```

(continues on next page)

(continued from previous page)

```
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
```

(continues on next page)

(continued from previous page)

```
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
```

(continues on next page)

(continued from previous page)

```
    O    1.622    -4.606    7.267   168
    H    0.644    -4.448    7.405   168
    H    2.063    -3.751    6.996   168
    O    -9.426    -1.829    3.261   169
    H    -8.578    -1.754    3.786   169
    H    -9.786    -2.759    3.34   169
    O    10.397    -2.728    0.938   170
    H    9.987    -2.336    0.114   170
    H    11.372    -2.508    0.962   170
    O    -3.49    -10.263    -4.969   171
    H    -2.767    -9.841    -5.517   171
    H    -3.819    -9.607    -4.289   171
    O    -1.521    -7.2    2.797   172
    H    -1.357    -6.413    3.392   172
    H    -0.828    -7.899    2.97   172
    O    2.563    -4.792    -5.31   173
    H    1.916    -4.727    -4.55   173
    H    2.546    -5.719    -5.685   173
    O    2.837    -2.181    6.366   174
    H    2.273    -1.594    5.785   174
    H    3.201    -1.646    7.128   174
    O    -4.032    -6.368    -2.157   175
    H    -4.673    -5.671    -2.479   175
    H    -4.118    -6.467    -1.165   175
    O    -3.248    1.69    -4.498   176
    H    -2.674    2.482    -4.291   176
    H    -2.676    0.943    -4.835   176
    O    9.532    2.335    1.199   177
    H    9.808    3.202    0.786   177
    H    8.539    2.233    1.129   177
    O    4.296    6.036    3.312   178
    H    4.884    5.765    4.074   178
    H    3.344    6.057    3.617   178
    O    -4.003    0.618    -2.093   179
    H    -3.993    0.617    -3.093   179
    H    -4.888    0.953    -1.769   179
    O    6.312    5.31    5.167   180
    H    6.936    5.315    4.386   180
    H    6.368    6.187    5.644   180
    O    -6.267    -0.421    5.288   181
    H    -7.045    -1.049    5.313   181
    H    -6.538    0.461    5.671   181
    O    -4.077    0.335    6.986   182
    H    -4.824    -0.155    6.539   182
    H    -3.207    -0.105    6.762   182
    O    -1.592    10.18    -2.288   183
    H    -2.521    10.21    -2.657   183
    H    -1.428    10.996    -1.734   183
    O    -1.179    1.277    4.006   184
    H    -1.662    0.404    3.942   184
    H    -1.541    1.802    4.776   184
    O    -5.857    -7.312    2.548   185
    H    -5.09    -7.722    3.042   185
    H    -6.533    -8.017    2.337   185
    O    8.543    -2.902    -4.126   186
    H    8.959    -3.487    -4.823   186
    H    8.449    -1.973    -4.484   186
```

(continues on next page)

(continued from previous page)

```
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
    O     -0.872    -9.88    -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
```

(continues on next page)

(continued from previous page)

```
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
    H     2.47    4.016    4.736   208
    O     -3.182    3.268    -7.397   209
    H     -2.609    2.52    -7.062   209
    H     -3.481    3.066    -8.33   209
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
```

(continues on next page)

(continued from previous page)

```
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
eor
```

Download FQQM_FQFMU_scf.run

```
#!/bin/bash

# QMFQ solvation calculation

$AMSBIN/ams << eor
system
  Atoms
    C      1.32246      0.46994    -0.44553
    C      0.03851      0.37791     0.32993
    O     -1.13145      0.98673    -0.28822
    C     -1.09292     -0.44743    -0.11660
    H     -0.99733     -1.01009    -1.04597
    H     -1.81522     -0.83104     0.60631
    H      0.10287      0.59921     1.39924
    H      2.07829     -0.18966     0.00080
    H      1.71275      1.49611    -0.42640
    H      1.17353      0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  basis
     type TZP
     core none
  End
  symmetry NOSYM
  title QM/FQFMU energy calculation
  xc
    gga PBE
  End
  QMFQ
    FORCEFIELD FQFMU
    MolCharge 0.00
    Kernel GAUS

    AtomType
      Symbol O
      Chi 0.2908429850
      Eta 0.5625181140
      Alpha 2.2187983720
    SubEnd
    AtomType
```

(continues on next page)

(continued from previous page)

```
      Symbol H
      Chi 0.1675711970
      Eta 0.6093265770
      Alpha 1.1906416090
    SubEnd
    Coords
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
```

(continues on next page)

(continued from previous page)

```
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
    H     -10.69    -5.035    3.321   18
    O     -3.63    -2.187    -1.384   19
    H     -2.767    -2.462    -1.807   19
    H     -3.822    -1.231    -1.607   19
    O     1.942    -6.524    -8.737   20
    H     0.996    -6.22    -8.852   20
    H     2.391    -6.563    -9.63   20
    O     -3.547    2.764    0.052   21
    H     -3.23    1.901    -0.342   21
    H     -2.83    3.455    -0.038   21
    O     1.904    8.539    -7.933   22
    H     2.085    8.395    -6.96   22
    H     0.923    8.447    -8.106   22
    O     5.07    6.46    -6.979   23
    H     5.462    5.945    -6.216   23
    H     5.106    7.438    -6.774   23
    O     1.317    1.858    -4.009   24
    H     0.789    2.105    -4.821   24
    H     1.565    0.891    -4.054   24
    O     -3.466    6.667    1.722   25
    H     -2.758    6.783    1.026   25
    H     -3.233    5.891    2.308   25
    O     9.374    -0.284    -4.021   26
    H     9.226    -0.462    -3.048   26
    H     8.904    0.56    -4.28   26
    O     -1.567    -0.699    -9.148   27
    H     -1.459    -1.69    -9.234   27
    H     -2.068    -0.348    -9.939   27
    O     2.355    -6.756    5.05   28
    H     3.312    -6.948    5.27   28
    H     2.119    -5.837    5.367   28
    O     -11.241    2.669    -0.446   29
    H     -10.442    2.227    -0.039   29
    H     -12.069    2.351    0.017   29
    O     -0.519    2.829    -2.219   30
    H     0.293    2.675    -2.782   30
    H     -0.756    1.984    -1.738   30
    O     2.736    2.528    1.949   31
    H     3.367    2.379    2.711   31
    H     1.894    2.946    2.29   31
    O     -3.662    -3.579    1.114   32
    H     -2.722    -3.33    1.349   32
    H     -3.879    -3.238    0.2   32
    O     -4.354    -0.05    9.809   33
    H     -4.059    -0.99    9.982   33
    H     -4.437    0.098    8.824   33
    O     -8.16    -1.139    0.546   34
    H     -8.636    -1.849    1.066   34
    H     -7.667    -1.558    -0.216   34
    O     -1.843    -1.194    6.421   35
    H     -1.115    -0.876    7.029   35
    H     -1.582    -1.022    5.471   35
    O     -1.16    -3.054    2.421   36
    H     -0.235    -2.945    2.057   36
    H     -1.147    -3.716    3.171   36
```

(continues on next page)

(continued from previous page)

```
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
    H     0.061    1.437    -6.716   37
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
```

(continues on next page)

(continued from previous page)

```
O     -0.792    1.114    -11.733   56
H     -0.455    0.182    -11.865   56
H     -0.023    1.721    -11.532   56
O     9.721    -2.916    -1.66   57
H     9.417    -2.736    -2.596   57
H     9.34    -3.787    -1.348   57
O     -7.087    -4.974    -5.981   58
H     -7.388    -4.844    -6.925   58
H     -6.088    -4.953    -5.943   58
O     4.858    8.204    6.274   59
H     4.557    8.898    6.929   59
H     4.59    7.298    6.6   59
O     -5.74    4.317    -0.952   60
H     -4.795    3.992    -0.963   60
H     -5.777    5.221    -0.525   60
O     4.091    6.159    -2.019   61
H     3.956    5.352    -1.444   61
H     3.21    6.456    -2.388   61
O     0.259    -1.583    -11.55   62
H     0.456    -2.193    -10.783   62
H     1.112    -1.18    -11.882   62
O     0.167    8.19    -1.584   63
H     -0.233    8.98    -2.049   63
H     0.999    7.909    -2.062   63
O     5.904    5.593    -4.57   64
H     5.781    4.816    -3.953   64
H     5.595    6.43    -4.118   64
O     1.853    2.264    7.653   65
H     2.031    1.992    8.599   65
H     0.866    2.341    7.509   65
O     -3.917    -8.529    3.889   66
H     -3.102    -8.099    3.5   66
H     -3.747    -8.76    4.847   66
O     -5.078    7.76    3.832   67
H     -4.485    7.317    3.159   67
H     -5.5    7.064    4.413   67
O     -5.333    -7.661    -6.49   68
H     -5.485    -7.46    -7.457   68
H     -4.51    -7.187    -6.176   68
O     1.603    9.353    1.706   69
H     1.306    8.406    1.576   69
H     2.011    9.452    2.614   69
O     -8.588    7.768    -2.072   70
H     -8.096    8.409    -1.483   70
H     -8.101    6.895    -2.096   70
O     8.949    -5.62    0.186   71
H     8.046    -5.232    0.372   71
H     9.636    -5.141    0.731   71
O     -2.56    -3.378    -9.638   72
H     -2.169    -4.295    -9.725   72
H     -2.903    -3.077    -10.528   72
O     4.132    5.755    7.273   73
H     3.367    5.418    7.822   73
H     4.613    4.982    6.859   73
O     -8.001    -0.706    -7.091   74
H     -8.965    -0.793    -7.341   74
H     -7.795    0.252    -6.889   74
```

(continues on next page)

(continued from previous page)

```
O     1.985    -9.965    1.717   75
H     1.164    -10.041    1.152   75
H     1.726    -9.72    2.651   75
O     -0.889    -3.751    7.35   76
H     -1.637    -4.413    7.286   76
H     -1.14    -2.918    6.857   76
O     6.447    -8.648    -3.384   77
H     6.956    -8.025    -2.791   77
H     7.079    -9.101    -4.013   77
O     -10.087    1.376    -3.258   78
H     -10.346    1.756    -2.37   78
H     -10.584    1.853    -3.982   78
O     -7.721    -6.978    -4.007   79
H     -7.044    -7.461    -4.562   79
H     -7.673    -5.998    -4.2   79
O     3.742    -9.925    -0.53   80
H     4.013    -10.88    -0.645   80
H     3.147    -9.841    0.269   80
O     2.643    11.407    0.185   81
H     3.532    11.713    0.526   81
H     2.357    10.591    0.686   81
O     8.142    -2.725    7.39   82
H     7.867    -3.625    7.728   82
H     8.341    -2.124    8.164   82
O     6.161    3.092    -3.472   83
H     6.808    3.021    -2.713   83
H     6.61    2.81    -4.32   83
O     -0.296    -10.611    0.551   84
H     0.082    -11.185    -0.175   84
H     -1.124    -11.036    0.916   84
O     8.597    -7.332    -1.93   85
H     9.074    -6.978    -2.734   85
H     8.663    -6.67    -1.184   85
O     6.902    -2.876    0.076   86
H     7.286    -2.931    0.998   86
H     6.464    -1.986    -0.05   86
O     -6.753    2.76    8.111   87
H     -7.086    3.694    8.246   87
H     -6.953    2.467    7.176   87
O     -7.552    -4.988    2.842   88
H     -7.782    -4.906    1.872   88
H     -7.032    -5.829    2.991   88
O     -8.182    -7.7    -1.006   89
H     -7.682    -7.051    -1.581   89
H     -7.896    -8.632    -1.23   89
O     8.372    -3.157    2.438   90
H     8.557    -3.167    3.421   90
H     9.221    -2.98    1.941   90
O     -6.198    -1.22    -3.217   91
H     -6.53    -0.276    -3.211   91
H     -5.626    -1.367    -4.024   91
O     4.677    1.242    0.094   92
H     4.368    1.864    -0.625   92
H     4.105    1.361    0.906   92
O     -2.993    -1.195    3.495   93
H     -3.849    -1.308    2.99   93
H     -2.282    -1.76    3.075   93
```

(continues on next page)

(continued from previous page)

```
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
    H     -3.233    7.809    -4.93   94
    O     -2.693    4.73    3.854   95
    H     -2.342    3.925    3.376   95
    H     -3.179    4.443    4.679   95
    O     1.001    -0.574    4.81   96
    H     0.553    0.313    4.696   96
    H     1.648    -0.72    4.062   96
    O     -7.658    3.829    -5.601   97
    H     -7.06    3.029    -5.656   97
    H     -8.518    3.635    -6.073   97
    O     1.26    6.271    8.504   98
    H     1.441    6.387    7.527   98
    H     0.355    6.637    8.721   98
    O     -2.775    -9.109    0.291   99
    H     -2.86    -8.211    0.723   99
    H     -1.809    -9.316    0.139   99
    O     3.256    -1.013    2.987   100
    H     3.539    -1.259    2.06   100
    H     3.855    -0.293    3.338   100
    O     0.21    -0.312    8.228   101
    H     1.026    -0.659    7.765   101
    H     0.428    0.548    8.688   101
    O     5.422    -0.784    -1.655   102
    H     5.497    -0.764    -2.652   102
    H     4.972    0.05    -1.337   102
    O     -1.59    -4.154    -5.957   103
    H     -1.997    -4.976    -5.559   103
    H     -1.274    -3.551    -5.225   103
    O     0.51    -8.72    3.751   104
    H     1.317    -8.458    4.28   104
    H     -0.042    -9.364    4.281   104
    O     -2.594    -6.607    -5.182   105
    H     -3.233    -6.683    -4.416   105
    H     -1.761    -7.12    -4.974   105
    O     -5.857    7.025    -0.001   106
    H     -4.915    7.129    0.32   106
    H     -6.458    7.622    0.53   106
    O     -0.705    -2.368    10.335   107
    H     -0.302    -2.039    9.481   107
    H     -1.387    -3.071    10.133   107
    O     -0.613    6.667    3.919   108
    H     -0.003    6.582    4.707   108
    H     -1.404    6.068    4.044   108
    O     3.935    -9.377    -3.564   109
    H     4.784    -8.856    -3.476   109
    H     3.704    -9.789    -2.683   109
    O     6.435    -3.938    -2.832   110
    H     7.169    -3.555    -3.391   110
    H     6.591    -3.706    -1.872   110
    O     6.896    8.206    0.362   111
    H     6.777    8.065    1.345   111
    H     6.022    8.469    -0.045   111
    O     -0.931    -3.296    -2.501   112
    H     -0.842    -2.655    -3.264   112
    H     -0.122    -3.232    -1.917   112
```

(continues on next page)

(continued from previous page)

```
    O     -2.26    -0.566    -6.382   113
    H     -3.181    -0.898    -6.176   113
    H     -2.061    -0.717    -7.35   113
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
```

(continues on next page)

(continued from previous page)

```
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
```

(continues on next page)

(continued from previous page)

```
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
```

(continues on next page)

(continued from previous page)

```
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
    H     11.372    -2.508    0.962   170
    O     -3.49    -10.263    -4.969   171
    H     -2.767    -9.841    -5.517   171
    H     -3.819    -9.607    -4.289   171
    O     -1.521    -7.2    2.797   172
    H     -1.357    -6.413    3.392   172
    H     -0.828    -7.899    2.97   172
    O     2.563    -4.792    -5.31   173
    H     1.916    -4.727    -4.55   173
    H     2.546    -5.719    -5.685   173
    O     2.837    -2.181    6.366   174
    H     2.273    -1.594    5.785   174
    H     3.201    -1.646    7.128   174
    O     -4.032    -6.368    -2.157   175
    H     -4.673    -5.671    -2.479   175
    H     -4.118    -6.467    -1.165   175
    O     -3.248    1.69    -4.498   176
    H     -2.674    2.482    -4.291   176
    H     -2.676    0.943    -4.835   176
    O     9.532    2.335    1.199   177
    H     9.808    3.202    0.786   177
    H     8.539    2.233    1.129   177
    O     4.296    6.036    3.312   178
    H     4.884    5.765    4.074   178
    H     3.344    6.057    3.617   178
    O     -4.003    0.618    -2.093   179
    H     -3.993    0.617    -3.093   179
    H     -4.888    0.953    -1.769   179
    O     6.312    5.31    5.167   180
    H     6.936    5.315    4.386   180
    H     6.368    6.187    5.644   180
    O     -6.267    -0.421    5.288   181
    H     -7.045    -1.049    5.313   181
    H     -6.538    0.461    5.671   181
    O     -4.077    0.335    6.986   182
    H     -4.824    -0.155    6.539   182
    H     -3.207    -0.105    6.762   182
    O     -1.592    10.18    -2.288   183
    H     -2.521    10.21    -2.657   183
    H     -1.428    10.996    -1.734   183
    O     -1.179    1.277    4.006   184
    H     -1.662    0.404    3.942   184
    H     -1.541    1.802    4.776   184
    O     -5.857    -7.312    2.548   185
    H     -5.09    -7.722    3.042   185
    H     -6.533    -8.017    2.337   185
    O     8.543    -2.902    -4.126   186
    H     8.959    -3.487    -4.823   186
    H     8.449    -1.973    -4.484   186
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
```

(continues on next page)

(continued from previous page)

```
    O     -0.872    -9.88     -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
```

(continues on next page)

(continued from previous page)

```
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
    H     2.47    4.016    4.736   208
    O     -3.182    3.268    -7.397   209
    H     -2.609    2.52    -7.062   209
    H     -3.481    3.066    -8.33   209
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor
```

Example: QM/FQ(Fμ): polarizability 2-Methyloxirane (AORESPONSE)

Two examples: a QM/FQ calculation and a QM/FQFμ calculation.

Download FQQM_aoresponse.run

```
#!/bin/bash

# QMFQ polarizability calculation

$AMSBIN/ams << eor
system
  Atoms
    C      1.32246      0.46994     -0.44553
    C      0.03851      0.37791      0.32993
    O     -1.13145      0.98673     -0.28822
    C     -1.09292     -0.44743     -0.11660
    H     -0.99733     -1.01009     -1.04597
    H     -1.81522     -0.83104      0.60631
    H      0.10287      0.59921      1.39924
    H      2.07829     -0.18966      0.00080
    H      1.71275      1.49611     -0.42640
    H      1.17353      0.16408     -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  aoresponse
    Frequencies 0.0 0.5 [eV]
  End
  basis
     type TZP
     core none
  End
  symmetry NOSYM
  title QM/FQ molecular polarizability calculation
  xc
    gga PBE
  End
  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00

    AtomType
      Symbol O
      Chi 0.189194
      Eta 0.523700
    SubEnd
```

(continues on next page)

(continued from previous page)

```
    AtomType
      Symbol H
      Chi 0.012767
      Eta 0.537512
    SubEnd
    Coords
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
```

(continues on next page)

(continued from previous page)

```
    O     -10.018    -4.546    3.877    18
    H     -9.098    -4.855    3.637    18
    H     -10.69    -5.035    3.321    18
    O     -3.63    -2.187    -1.384    19
    H     -2.767    -2.462    -1.807    19
    H     -3.822    -1.231    -1.607    19
    O     1.942    -6.524    -8.737    20
    H     0.996    -6.22    -8.852    20
    H     2.391    -6.563    -9.63    20
    O     -3.547    2.764    0.052    21
    H     -3.23    1.901    -0.342    21
    H     -2.83    3.455    -0.038    21
    O     1.904    8.539    -7.933    22
    H     2.085    8.395    -6.96    22
    H     0.923    8.447    -8.106    22
    O     5.07    6.46    -6.979    23
    H     5.462    5.945    -6.216    23
    H     5.106    7.438    -6.774    23
    O     1.317    1.858    -4.009    24
    H     0.789    2.105    -4.821    24
    H     1.565    0.891    -4.054    24
    O     -3.466    6.667    1.722    25
    H     -2.758    6.783    1.026    25
    H     -3.233    5.891    2.308    25
    O     9.374    -0.284    -4.021    26
    H     9.226    -0.462    -3.048    26
    H     8.904    0.56    -4.28    26
    O     -1.567    -0.699    -9.148    27
    H     -1.459    -1.69    -9.234    27
    H     -2.068    -0.348    -9.939    27
    O     2.355    -6.756    5.05    28
    H     3.312    -6.948    5.27    28
    H     2.119    -5.837    5.367    28
    O     -11.241    2.669    -0.446    29
    H     -10.442    2.227    -0.039    29
    H     -12.069    2.351    0.017    29
    O     -0.519    2.829    -2.219    30
    H     0.293    2.675    -2.782    30
    H     -0.756    1.984    -1.738    30
    O     2.736    2.528    1.949    31
    H     3.367    2.379    2.711    31
    H     1.894    2.946    2.29    31
    O     -3.662    -3.579    1.114    32
    H     -2.722    -3.33    1.349    32
    H     -3.879    -3.238    0.2    32
    O     -4.354    -0.05    9.809    33
    H     -4.059    -0.99    9.982    33
    H     -4.437    0.098    8.824    33
    O     -8.16    -1.139    0.546    34
    H     -8.636    -1.849    1.066    34
    H     -7.667    -1.558    -0.216    34
    O     -1.843    -1.194    6.421    35
    H     -1.115    -0.876    7.029    35
    H     -1.582    -1.022    5.471    35
    O     -1.16    -3.054    2.421    36
    H     -0.235    -2.945    2.057    36
    H     -1.147    -3.716    3.171    36
```

(continues on next page)

(continued from previous page)

```
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
    H     0.061    1.437    -6.716   37
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
```

(continues on next page)

(continued from previous page)

```
    O     -0.792    1.114    -11.733   56
    H     -0.455    0.182    -11.865   56
    H     -0.023    1.721    -11.532   56
    O     9.721    -2.916    -1.66   57
    H     9.417    -2.736    -2.596   57
    H     9.34    -3.787    -1.348   57
    O     -7.087    -4.974    -5.981   58
    H     -7.388    -4.844    -6.925   58
    H     -6.088    -4.953    -5.943   58
    O     4.858    8.204    6.274   59
    H     4.557    8.898    6.929   59
    H     4.59    7.298    6.6   59
    O     -5.74    4.317    -0.952   60
    H     -4.795    3.992    -0.963   60
    H     -5.777    5.221    -0.525   60
    O     4.091    6.159    -2.019   61
    H     3.956    5.352    -1.444   61
    H     3.21    6.456    -2.388   61
    O     0.259    -1.583    -11.55   62
    H     0.456    -2.193    -10.783   62
    H     1.112    -1.18    -11.882   62
    O     0.167    8.19    -1.584   63
    H     -0.233    8.98    -2.049   63
    H     0.999    7.909    -2.062   63
    O     5.904    5.593    -4.57   64
    H     5.781    4.816    -3.953   64
    H     5.595    6.43    -4.118   64
    O     1.853    2.264    7.653   65
    H     2.031    1.992    8.599   65
    H     0.866    2.341    7.509   65
    O     -3.917    -8.529    3.889   66
    H     -3.102    -8.099    3.5   66
    H     -3.747    -8.76    4.847   66
    O     -5.078    7.76    3.832   67
    H     -4.485    7.317    3.159   67
    H     -5.5    7.064    4.413   67
    O     -5.333    -7.661    -6.49   68
    H     -5.485    -7.46    -7.457   68
    H     -4.51    -7.187    -6.176   68
    O     1.603    9.353    1.706   69
    H     1.306    8.406    1.576   69
    H     2.011    9.452    2.614   69
    O     -8.588    7.768    -2.072   70
    H     -8.096    8.409    -1.483   70
    H     -8.101    6.895    -2.096   70
    O     8.949    -5.62    0.186   71
    H     8.046    -5.232    0.372   71
    H     9.636    -5.141    0.731   71
    O     -2.56    -3.378    -9.638   72
    H     -2.169    -4.295    -9.725   72
    H     -2.903    -3.077    -10.528   72
    O     4.132    5.755    7.273   73
    H     3.367    5.418    7.822   73
    H     4.613    4.982    6.859   73
    O     -8.001    -0.706    -7.091   74
    H     -8.965    -0.793    -7.341   74
    H     -7.795    0.252    -6.889   74
```

(continues on next page)

(continued from previous page)

```
    O     1.985    -9.965    1.717   75
    H     1.164    -10.041    1.152   75
    H     1.726    -9.72    2.651   75
    O     -0.889    -3.751    7.35   76
    H     -1.637    -4.413    7.286   76
    H     -1.14    -2.918    6.857   76
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
```

(continues on next page)

(continued from previous page)

```
O     -3.81    7.752    -5.745    94
H     -3.268    7.416    -6.515    94
H     -3.233    7.809    -4.93    94
O     -2.693    4.73    3.854    95
H     -2.342    3.925    3.376    95
H     -3.179    4.443    4.679    95
O     1.001    -0.574    4.81    96
H     0.553    0.313    4.696    96
H     1.648    -0.72    4.062    96
O     -7.658    3.829    -5.601    97
H     -7.06    3.029    -5.656    97
H     -8.518    3.635    -6.073    97
O     1.26    6.271    8.504    98
H     1.441    6.387    7.527    98
H     0.355    6.637    8.721    98
O     -2.775    -9.109    0.291    99
H     -2.86    -8.211    0.723    99
H     -1.809    -9.316    0.139    99
O     3.256    -1.013    2.987    100
H     3.539    -1.259    2.06    100
H     3.855    -0.293    3.338    100
O     0.21    -0.312    8.228    101
H     1.026    -0.659    7.765    101
H     0.428    0.548    8.688    101
O     5.422    -0.784    -1.655    102
H     5.497    -0.764    -2.652    102
H     4.972    0.05    -1.337    102
O     -1.59    -4.154    -5.957    103
H     -1.997    -4.976    -5.559    103
H     -1.274    -3.551    -5.225    103
O     0.51    -8.72    3.751    104
H     1.317    -8.458    4.28    104
H     -0.042    -9.364    4.281    104
O     -2.594    -6.607    -5.182    105
H     -3.233    -6.683    -4.416    105
H     -1.761    -7.12    -4.974    105
O     -5.857    7.025    -0.001    106
H     -4.915    7.129    0.32    106
H     -6.458    7.622    0.53    106
O     -0.705    -2.368    10.335    107
H     -0.302    -2.039    9.481    107
H     -1.387    -3.071    10.133    107
O     -0.613    6.667    3.919    108
H     -0.003    6.582    4.707    108
H     -1.404    6.068    4.044    108
O     3.935    -9.377    -3.564    109
H     4.784    -8.856    -3.476    109
H     3.704    -9.789    -2.683    109
O     6.435    -3.938    -2.832    110
H     7.169    -3.555    -3.391    110
H     6.591    -3.706    -1.872    110
O     6.896    8.206    0.362    111
H     6.777    8.065    1.345    111
H     6.022    8.469    -0.045    111
O     -0.931    -3.296    -2.501    112
H     -0.842    -2.655    -3.264    112
H     -0.122    -3.232    -1.917    112
```

(continues on next page)

(continued from previous page)

```
    O     -2.26    -0.566    -6.382   113
    H     -3.181    -0.898    -6.176   113
    H     -2.061    -0.717    -7.35   113
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
```

(continues on next page)

(continued from previous page)

```
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
```

(continues on next page)

(continued from previous page)

```
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
```

(continues on next page)

(continued from previous page)

```
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
    H     11.372    -2.508    0.962   170
    O     -3.49    -10.263    -4.969   171
    H     -2.767    -9.841    -5.517   171
    H     -3.819    -9.607    -4.289   171
    O     -1.521    -7.2    2.797   172
    H     -1.357    -6.413    3.392   172
    H     -0.828    -7.899    2.97   172
    O     2.563    -4.792    -5.31   173
    H     1.916    -4.727    -4.55   173
    H     2.546    -5.719    -5.685   173
    O     2.837    -2.181    6.366   174
    H     2.273    -1.594    5.785   174
    H     3.201    -1.646    7.128   174
    O     -4.032    -6.368    -2.157   175
    H     -4.673    -5.671    -2.479   175
    H     -4.118    -6.467    -1.165   175
    O     -3.248    1.69    -4.498   176
    H     -2.674    2.482    -4.291   176
    H     -2.676    0.943    -4.835   176
    O     9.532    2.335    1.199   177
    H     9.808    3.202    0.786   177
    H     8.539    2.233    1.129   177
    O     4.296    6.036    3.312   178
    H     4.884    5.765    4.074   178
    H     3.344    6.057    3.617   178
    O     -4.003    0.618    -2.093   179
    H     -3.993    0.617    -3.093   179
    H     -4.888    0.953    -1.769   179
    O     6.312    5.31    5.167   180
    H     6.936    5.315    4.386   180
    H     6.368    6.187    5.644   180
    O     -6.267    -0.421    5.288   181
    H     -7.045    -1.049    5.313   181
    H     -6.538    0.461    5.671   181
    O     -4.077    0.335    6.986   182
    H     -4.824    -0.155    6.539   182
    H     -3.207    -0.105    6.762   182
    O     -1.592    10.18    -2.288   183
    H     -2.521    10.21    -2.657   183
    H     -1.428    10.996    -1.734   183
    O     -1.179    1.277    4.006   184
    H     -1.662    0.404    3.942   184
    H     -1.541    1.802    4.776   184
    O     -5.857    -7.312    2.548   185
    H     -5.09    -7.722    3.042   185
    H     -6.533    -8.017    2.337   185
    O     8.543    -2.902    -4.126   186
    H     8.959    -3.487    -4.823   186
    H     8.449    -1.973    -4.484   186
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
```

(continues on next page)

(continued from previous page)

```
    O     -0.872    -9.88    -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
```

(continues on next page)

(continued from previous page)

```
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
    H     2.47    4.016    4.736   208
    O     -3.182    3.268    -7.397   209
    H     -2.609    2.52    -7.062   209
    H     -3.481    3.066    -8.33   209
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor
```

Download FQQM_FQFMU_aoresponse.run

```
#!/bin/bash

# QM/FQFMU polarizability calculation

$AMSBIN/ams << eor
system
  Atoms
    C     1.32246     0.46994    -0.44553
    C     0.03851     0.37791     0.32993
    O    -1.13145     0.98673    -0.28822
    C    -1.09292    -0.44743    -0.11660
    H    -0.99733    -1.01009    -1.04597
    H    -1.81522    -0.83104     0.60631
    H     0.10287     0.59921     1.39924
    H     2.07829    -0.18966     0.00080
    H     1.71275     1.49611    -0.42640
    H     1.17353     0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  aoresponse
    Frequencies 0.0 0.5 [eV]
  End
  basis
     type TZP
     core none
  End
  symmetry NOSYM
  title QM/FQFMU molecular polarizability calculation
  xc
    gga PBE
  End
  QMFQ
    FORCEFIELD FQFMU
    Kernel GAUS
    MolCharge 0.00

    AtomType
      Symbol O
      Chi 0.2908429850
      Eta 0.5625181140
      Alpha 2.2187983720
    SubEnd
    AtomType
      Symbol H
      Chi 0.1675711970
      Eta 0.6093265770
```

(continues on next page)

(continued from previous page)

```
      Alpha 1.1906416090
    SubEnd
    Coords
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
    H     -10.69    -5.035    3.321   18
```

(continues on next page)

(continued from previous page)

```
    O     -3.63    -2.187    -1.384    19
    H     -2.767    -2.462    -1.807    19
    H     -3.822    -1.231    -1.607    19
    O     1.942    -6.524    -8.737    20
    H     0.996    -6.22    -8.852    20
    H     2.391    -6.563    -9.63    20
    O     -3.547    2.764    0.052    21
    H     -3.23    1.901    -0.342    21
    H     -2.83    3.455    -0.038    21
    O     1.904    8.539    -7.933    22
    H     2.085    8.395    -6.96    22
    H     0.923    8.447    -8.106    22
    O     5.07    6.46    -6.979    23
    H     5.462    5.945    -6.216    23
    H     5.106    7.438    -6.774    23
    O     1.317    1.858    -4.009    24
    H     0.789    2.105    -4.821    24
    H     1.565    0.891    -4.054    24
    O     -3.466    6.667    1.722    25
    H     -2.758    6.783    1.026    25
    H     -3.233    5.891    2.308    25
    O     9.374    -0.284    -4.021    26
    H     9.226    -0.462    -3.048    26
    H     8.904    0.56    -4.28    26
    O     -1.567    -0.699    -9.148    27
    H     -1.459    -1.69    -9.234    27
    H     -2.068    -0.348    -9.939    27
    O     2.355    -6.756    5.05    28
    H     3.312    -6.948    5.27    28
    H     2.119    -5.837    5.367    28
    O     -11.241    2.669    -0.446    29
    H     -10.442    2.227    -0.039    29
    H     -12.069    2.351    0.017    29
    O     -0.519    2.829    -2.219    30
    H     0.293    2.675    -2.782    30
    H     -0.756    1.984    -1.738    30
    O     2.736    2.528    1.949    31
    H     3.367    2.379    2.711    31
    H     1.894    2.946    2.29    31
    O     -3.662    -3.579    1.114    32
    H     -2.722    -3.33    1.349    32
    H     -3.879    -3.238    0.2    32
    O     -4.354    -0.05    9.809    33
    H     -4.059    -0.99    9.982    33
    H     -4.437    0.098    8.824    33
    O     -8.16    -1.139    0.546    34
    H     -8.636    -1.849    1.066    34
    H     -7.667    -1.558    -0.216    34
    O     -1.843    -1.194    6.421    35
    H     -1.115    -0.876    7.029    35
    H     -1.582    -1.022    5.471    35
    O     -1.16    -3.054    2.421    36
    H     -0.235    -2.945    2.057    36
    H     -1.147    -3.716    3.171    36
    O     -0.447    2.125    -6.198    37
    H     -0.425    2.996    -6.688    37
    H     0.061    1.437    -6.716    37
```

(continues on next page)

(continued from previous page)

```
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
    O     -0.792    1.114    -11.733   56
    H     -0.455    0.182    -11.865   56
    H     -0.023    1.721    -11.532   56
```

(continues on next page)

(continued from previous page)

```
    O    9.721    -2.916    -1.66   57
    H    9.417    -2.736    -2.596   57
    H    9.34    -3.787    -1.348   57
    O    -7.087    -4.974    -5.981   58
    H    -7.388    -4.844    -6.925   58
    H    -6.088    -4.953    -5.943   58
    O    4.858    8.204    6.274   59
    H    4.557    8.898    6.929   59
    H    4.59    7.298    6.6   59
    O    -5.74    4.317    -0.952   60
    H    -4.795    3.992    -0.963   60
    H    -5.777    5.221    -0.525   60
    O    4.091    6.159    -2.019   61
    H    3.956    5.352    -1.444   61
    H    3.21    6.456    -2.388   61
    O    0.259    -1.583    -11.55   62
    H    0.456    -2.193    -10.783   62
    H    1.112    -1.18    -11.882   62
    O    0.167    8.19    -1.584   63
    H    -0.233    8.98    -2.049   63
    H    0.999    7.909    -2.062   63
    O    5.904    5.593    -4.57   64
    H    5.781    4.816    -3.953   64
    H    5.595    6.43    -4.118   64
    O    1.853    2.264    7.653   65
    H    2.031    1.992    8.599   65
    H    0.866    2.341    7.509   65
    O    -3.917    -8.529    3.889   66
    H    -3.102    -8.099    3.5   66
    H    -3.747    -8.76    4.847   66
    O    -5.078    7.76    3.832   67
    H    -4.485    7.317    3.159   67
    H    -5.5    7.064    4.413   67
    O    -5.333    -7.661    -6.49   68
    H    -5.485    -7.46    -7.457   68
    H    -4.51    -7.187    -6.176   68
    O    1.603    9.353    1.706   69
    H    1.306    8.406    1.576   69
    H    2.011    9.452    2.614   69
    O    -8.588    7.768    -2.072   70
    H    -8.096    8.409    -1.483   70
    H    -8.101    6.895    -2.096   70
    O    8.949    -5.62    0.186   71
    H    8.046    -5.232    0.372   71
    H    9.636    -5.141    0.731   71
    O    -2.56    -3.378    -9.638   72
    H    -2.169    -4.295    -9.725   72
    H    -2.903    -3.077    -10.528   72
    O    4.132    5.755    7.273   73
    H    3.367    5.418    7.822   73
    H    4.613    4.982    6.859   73
    O    -8.001    -0.706    -7.091   74
    H    -8.965    -0.793    -7.341   74
    H    -7.795    0.252    -6.889   74
    O    1.985    -9.965    1.717   75
    H    1.164    -10.041    1.152   75
    H    1.726    -9.72    2.651   75
```

(continues on next page)

(continued from previous page)

```
    O     -0.889    -3.751    7.35   76
    H     -1.637    -4.413    7.286   76
    H     -1.14    -2.918    6.857   76
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
    H     -3.233    7.809    -4.93   94
```

(continues on next page)

(continued from previous page)

```
    O     -2.693    4.73    3.854    95
    H     -2.342    3.925    3.376    95
    H     -3.179    4.443    4.679    95
    O     1.001    -0.574    4.81    96
    H     0.553    0.313    4.696    96
    H     1.648    -0.72    4.062    96
    O     -7.658    3.829    -5.601    97
    H     -7.06    3.029    -5.656    97
    H     -8.518    3.635    -6.073    97
    O     1.26    6.271    8.504    98
    H     1.441    6.387    7.527    98
    H     0.355    6.637    8.721    98
    O     -2.775    -9.109    0.291    99
    H     -2.86    -8.211    0.723    99
    H     -1.809    -9.316    0.139    99
    O     3.256    -1.013    2.987    100
    H     3.539    -1.259    2.06    100
    H     3.855    -0.293    3.338    100
    O     0.21    -0.312    8.228    101
    H     1.026    -0.659    7.765    101
    H     0.428    0.548    8.688    101
    O     5.422    -0.784    -1.655    102
    H     5.497    -0.764    -2.652    102
    H     4.972    0.05    -1.337    102
    O     -1.59    -4.154    -5.957    103
    H     -1.997    -4.976    -5.559    103
    H     -1.274    -3.551    -5.225    103
    O     0.51    -8.72    3.751    104
    H     1.317    -8.458    4.28    104
    H     -0.042    -9.364    4.281    104
    O     -2.594    -6.607    -5.182    105
    H     -3.233    -6.683    -4.416    105
    H     -1.761    -7.12    -4.974    105
    O     -5.857    7.025    -0.001    106
    H     -4.915    7.129    0.32    106
    H     -6.458    7.622    0.53    106
    O     -0.705    -2.368    10.335    107
    H     -0.302    -2.039    9.481    107
    H     -1.387    -3.071    10.133    107
    O     -0.613    6.667    3.919    108
    H     -0.003    6.582    4.707    108
    H     -1.404    6.068    4.044    108
    O     3.935    -9.377    -3.564    109
    H     4.784    -8.856    -3.476    109
    H     3.704    -9.789    -2.683    109
    O     6.435    -3.938    -2.832    110
    H     7.169    -3.555    -3.391    110
    H     6.591    -3.706    -1.872    110
    O     6.896    8.206    0.362    111
    H     6.777    8.065    1.345    111
    H     6.022    8.469    -0.045    111
    O     -0.931    -3.296    -2.501    112
    H     -0.842    -2.655    -3.264    112
    H     -0.122    -3.232    -1.917    112
    O     -2.26    -0.566    -6.382    113
    H     -3.181    -0.898    -6.176    113
    H     -2.061    -0.717    -7.35    113
```

(continues on next page)

(continued from previous page)

```
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
```

(continues on next page)

(continued from previous page)

```
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
```

(continues on next page)

(continued from previous page)

```
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
    H     11.372    -2.508    0.962   170
```

(continues on next page)

(continued from previous page)

```
    O     -3.49    -10.263    -4.969   171
    H     -2.767    -9.841    -5.517   171
    H     -3.819    -9.607    -4.289   171
    O     -1.521    -7.2    2.797   172
    H     -1.357    -6.413    3.392   172
    H     -0.828    -7.899    2.97   172
    O     2.563    -4.792    -5.31   173
    H     1.916    -4.727    -4.55   173
    H     2.546    -5.719    -5.685   173
    O     2.837    -2.181    6.366   174
    H     2.273    -1.594    5.785   174
    H     3.201    -1.646    7.128   174
    O     -4.032    -6.368    -2.157   175
    H     -4.673    -5.671    -2.479   175
    H     -4.118    -6.467    -1.165   175
    O     -3.248    1.69    -4.498   176
    H     -2.674    2.482    -4.291   176
    H     -2.676    0.943    -4.835   176
    O     9.532    2.335    1.199   177
    H     9.808    3.202    0.786   177
    H     8.539    2.233    1.129   177
    O     4.296    6.036    3.312   178
    H     4.884    5.765    4.074   178
    H     3.344    6.057    3.617   178
    O     -4.003    0.618    -2.093   179
    H     -3.993    0.617    -3.093   179
    H     -4.888    0.953    -1.769   179
    O     6.312    5.31    5.167   180
    H     6.936    5.315    4.386   180
    H     6.368    6.187    5.644   180
    O     -6.267    -0.421    5.288   181
    H     -7.045    -1.049    5.313   181
    H     -6.538    0.461    5.671   181
    O     -4.077    0.335    6.986   182
    H     -4.824    -0.155    6.539   182
    H     -3.207    -0.105    6.762   182
    O     -1.592    10.18    -2.288   183
    H     -2.521    10.21    -2.657   183
    H     -1.428    10.996    -1.734   183
    O     -1.179    1.277    4.006   184
    H     -1.662    0.404    3.942   184
    H     -1.541    1.802    4.776   184
    O     -5.857    -7.312    2.548   185
    H     -5.09    -7.722    3.042   185
    H     -6.533    -8.017    2.337   185
    O     8.543    -2.902    -4.126   186
    H     8.959    -3.487    -4.823   186
    H     8.449    -1.973    -4.484   186
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
    O     -0.872    -9.88    -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
```

(continues on next page)

(continued from previous page)

```
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
    H     2.47    4.016    4.736   208
```

(continues on next page)

(continued from previous page)

```
    O     -3.182    3.268    -7.397   209
    H     -2.609    2.52    -7.062   209
    H     -3.481    3.066    -8.33   209
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494    211
    H     -10.502    -0.004    1.358    211
    H     -11.916    -0.768    1.646    211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
eor
```

Example: QM/FQ(Fμ): polarizability 2-Methyloxirane (RESPONSE)

Two examples: a QM/FQ calculation and a QM/FQFμ calculation.

`Download FQQM_response.run`

```
#!/bin/bash

# QMFQ polarizability calculation

$AMSBIN/ams << eor
system
  Atoms
    C     1.32246     0.46994    -0.44553
    C     0.03851     0.37791     0.32993
    O    -1.13145     0.98673    -0.28822
    C    -1.09292    -0.44743    -0.11660
    H    -0.99733    -1.01009    -1.04597
    H    -1.81522    -0.83104     0.60631
    H     0.10287     0.59921     1.39924
    H     2.07829    -0.18966     0.00080
    H     1.71275     1.49611    -0.42640
    H     1.17353     0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  basis
     type TZP
     core none
  End
  response
    allcomponents True
    analytic True
    Frequencies 0.0 0.5 [eV]
  End
  symmetry NOSYM
  title QM/FQ molecular polarizability calculation
  xc
    gga PBE
  End

  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00

    AtomType
      Symbol O
      Chi 0.189194
      Eta 0.523700
    SubEnd
    AtomType
      Symbol H
```

(continues on next page)

(continued from previous page)

```
      Chi 0.012767
      Eta 0.537512
    SubEnd
    Coords
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
```

(continues on next page)

(continued from previous page)

```
    H     -10.69    -5.035    3.321   18
    O     -3.63    -2.187    -1.384   19
    H     -2.767    -2.462    -1.807   19
    H     -3.822    -1.231    -1.607   19
    O     1.942    -6.524    -8.737   20
    H     0.996    -6.22    -8.852   20
    H     2.391    -6.563    -9.63   20
    O     -3.547    2.764    0.052   21
    H     -3.23    1.901    -0.342   21
    H     -2.83    3.455    -0.038   21
    O     1.904    8.539    -7.933   22
    H     2.085    8.395    -6.96   22
    H     0.923    8.447    -8.106   22
    O     5.07    6.46    -6.979   23
    H     5.462    5.945    -6.216   23
    H     5.106    7.438    -6.774   23
    O     1.317    1.858    -4.009   24
    H     0.789    2.105    -4.821   24
    H     1.565    0.891    -4.054   24
    O     -3.466    6.667    1.722   25
    H     -2.758    6.783    1.026   25
    H     -3.233    5.891    2.308   25
    O     9.374    -0.284    -4.021   26
    H     9.226    -0.462    -3.048   26
    H     8.904    0.56    -4.28   26
    O     -1.567    -0.699    -9.148   27
    H     -1.459    -1.69    -9.234   27
    H     -2.068    -0.348    -9.939   27
    O     2.355    -6.756    5.05   28
    H     3.312    -6.948    5.27   28
    H     2.119    -5.837    5.367   28
    O     -11.241    2.669    -0.446   29
    H     -10.442    2.227    -0.039   29
    H     -12.069    2.351    0.017   29
    O     -0.519    2.829    -2.219   30
    H     0.293    2.675    -2.782   30
    H     -0.756    1.984    -1.738   30
    O     2.736    2.528    1.949   31
    H     3.367    2.379    2.711   31
    H     1.894    2.946    2.29   31
    O     -3.662    -3.579    1.114   32
    H     -2.722    -3.33    1.349   32
    H     -3.879    -3.238    0.2   32
    O     -4.354    -0.05    9.809   33
    H     -4.059    -0.99    9.982   33
    H     -4.437    0.098    8.824   33
    O     -8.16    -1.139    0.546   34
    H     -8.636    -1.849    1.066   34
    H     -7.667    -1.558    -0.216   34
    O     -1.843    -1.194    6.421   35
    H     -1.115    -0.876    7.029   35
    H     -1.582    -1.022    5.471   35
    O     -1.16    -3.054    2.421   36
    H     -0.235    -2.945    2.057   36
    H     -1.147    -3.716    3.171   36
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
```

(continues on next page)

(continued from previous page)

```
    H     0.061    1.437    -6.716   37
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
    O     -0.792    1.114    -11.733   56
    H     -0.455    0.182    -11.865   56
```

(continues on next page)

(continued from previous page)

```
    H     -0.023    1.721    -11.532   56
    O     9.721    -2.916    -1.66   57
    H     9.417    -2.736    -2.596   57
    H     9.34    -3.787    -1.348   57
    O     -7.087    -4.974    -5.981   58
    H     -7.388    -4.844    -6.925   58
    H     -6.088    -4.953    -5.943   58
    O     4.858    8.204    6.274   59
    H     4.557    8.898    6.929   59
    H     4.59    7.298    6.6   59
    O     -5.74    4.317    -0.952   60
    H     -4.795    3.992    -0.963   60
    H     -5.777    5.221    -0.525   60
    O     4.091    6.159    -2.019   61
    H     3.956    5.352    -1.444   61
    H     3.21    6.456    -2.388   61
    O     0.259    -1.583    -11.55   62
    H     0.456    -2.193    -10.783   62
    H     1.112    -1.18    -11.882   62
    O     0.167    8.19    -1.584   63
    H     -0.233    8.98    -2.049   63
    H     0.999    7.909    -2.062   63
    O     5.904    5.593    -4.57   64
    H     5.781    4.816    -3.953   64
    H     5.595    6.43    -4.118   64
    O     1.853    2.264    7.653   65
    H     2.031    1.992    8.599   65
    H     0.866    2.341    7.509   65
    O     -3.917    -8.529    3.889   66
    H     -3.102    -8.099    3.5   66
    H     -3.747    -8.76    4.847   66
    O     -5.078    7.76    3.832   67
    H     -4.485    7.317    3.159   67
    H     -5.5    7.064    4.413   67
    O     -5.333    -7.661    -6.49   68
    H     -5.485    -7.46    -7.457   68
    H     -4.51    -7.187    -6.176   68
    O     1.603    9.353    1.706   69
    H     1.306    8.406    1.576   69
    H     2.011    9.452    2.614   69
    O     -8.588    7.768    -2.072   70
    H     -8.096    8.409    -1.483   70
    H     -8.101    6.895    -2.096   70
    O     8.949    -5.62    0.186   71
    H     8.046    -5.232    0.372   71
    H     9.636    -5.141    0.731   71
    O     -2.56    -3.378    -9.638   72
    H     -2.169    -4.295    -9.725   72
    H     -2.903    -3.077    -10.528   72
    O     4.132    5.755    7.273   73
    H     3.367    5.418    7.822   73
    H     4.613    4.982    6.859   73
    O     -8.001    -0.706    -7.091   74
    H     -8.965    -0.793    -7.341   74
    H     -7.795    0.252    -6.889   74
    O     1.985    -9.965    1.717   75
    H     1.164    -10.041    1.152   75
```

(continues on next page)

(continued from previous page)

```
    H     1.726    -9.72    2.651   75
    O     -0.889    -3.751    7.35   76
    H     -1.637    -4.413    7.286   76
    H     -1.14    -2.918    6.857   76
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
```

(continues on next page)

(continued from previous page)

```
H     -3.233    7.809    -4.93   94
O     -2.693    4.73    3.854   95
H     -2.342    3.925    3.376   95
H     -3.179    4.443    4.679   95
O     1.001    -0.574    4.81   96
H     0.553    0.313    4.696   96
H     1.648    -0.72    4.062   96
O     -7.658    3.829    -5.601   97
H     -7.06    3.029    -5.656   97
H     -8.518    3.635    -6.073   97
O     1.26    6.271    8.504   98
H     1.441    6.387    7.527   98
H     0.355    6.637    8.721   98
O     -2.775    -9.109    0.291   99
H     -2.86    -8.211    0.723   99
H     -1.809    -9.316    0.139   99
O     3.256    -1.013    2.987   100
H     3.539    -1.259    2.06   100
H     3.855    -0.293    3.338   100
O     0.21    -0.312    8.228   101
H     1.026    -0.659    7.765   101
H     0.428    0.548    8.688   101
O     5.422    -0.784    -1.655   102
H     5.497    -0.764    -2.652   102
H     4.972    0.05    -1.337   102
O     -1.59    -4.154    -5.957   103
H     -1.997    -4.976    -5.559   103
H     -1.274    -3.551    -5.225   103
O     0.51    -8.72    3.751   104
H     1.317    -8.458    4.28   104
H     -0.042    -9.364    4.281   104
O     -2.594    -6.607    -5.182   105
H     -3.233    -6.683    -4.416   105
H     -1.761    -7.12    -4.974   105
O     -5.857    7.025    -0.001   106
H     -4.915    7.129    0.32   106
H     -6.458    7.622    0.53   106
O     -0.705    -2.368    10.335   107
H     -0.302    -2.039    9.481   107
H     -1.387    -3.071    10.133   107
O     -0.613    6.667    3.919   108
H     -0.003    6.582    4.707   108
H     -1.404    6.068    4.044   108
O     3.935    -9.377    -3.564   109
H     4.784    -8.856    -3.476   109
H     3.704    -9.789    -2.683   109
O     6.435    -3.938    -2.832   110
H     7.169    -3.555    -3.391   110
H     6.591    -3.706    -1.872   110
O     6.896    8.206    0.362   111
H     6.777    8.065    1.345   111
H     6.022    8.469    -0.045   111
O     -0.931    -3.296    -2.501   112
H     -0.842    -2.655    -3.264   112
H     -0.122    -3.232    -1.917   112
O     -2.26    -0.566    -6.382   113
H     -3.181    -0.898    -6.176   113
```

(continues on next page)

(continued from previous page)

```
    H     -2.061    -0.717    -7.35   113
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
```

(continues on next page)

(continued from previous page)

```
    H     4.549    -0.31    8.412   132
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
```

(continues on next page)

(continued from previous page)

```
    H     -2.413    -9.373    7.36   151
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
```

(continues on next page)

(continued from previous page)

```
H     11.372    -2.508    0.962   170
O     -3.49    -10.263    -4.969   171
H     -2.767    -9.841    -5.517   171
H     -3.819    -9.607    -4.289   171
O     -1.521    -7.2    2.797   172
H     -1.357    -6.413    3.392   172
H     -0.828    -7.899    2.97   172
O     2.563    -4.792    -5.31   173
H     1.916    -4.727    -4.55   173
H     2.546    -5.719    -5.685   173
O     2.837    -2.181    6.366   174
H     2.273    -1.594    5.785   174
H     3.201    -1.646    7.128   174
O     -4.032    -6.368    -2.157   175
H     -4.673    -5.671    -2.479   175
H     -4.118    -6.467    -1.165   175
O     -3.248    1.69    -4.498   176
H     -2.674    2.482    -4.291   176
H     -2.676    0.943    -4.835   176
O     9.532    2.335    1.199   177
H     9.808    3.202    0.786   177
H     8.539    2.233    1.129   177
O     4.296    6.036    3.312   178
H     4.884    5.765    4.074   178
H     3.344    6.057    3.617   178
O     -4.003    0.618    -2.093   179
H     -3.993    0.617    -3.093   179
H     -4.888    0.953    -1.769   179
O     6.312    5.31    5.167   180
H     6.936    5.315    4.386   180
H     6.368    6.187    5.644   180
O     -6.267    -0.421    5.288   181
H     -7.045    -1.049    5.313   181
H     -6.538    0.461    5.671   181
O     -4.077    0.335    6.986   182
H     -4.824    -0.155    6.539   182
H     -3.207    -0.105    6.762   182
O     -1.592    10.18    -2.288   183
H     -2.521    10.21    -2.657   183
H     -1.428    10.996    -1.734   183
O     -1.179    1.277    4.006   184
H     -1.662    0.404    3.942   184
H     -1.541    1.802    4.776   184
O     -5.857    -7.312    2.548   185
H     -5.09    -7.722    3.042   185
H     -6.533    -8.017    2.337   185
O     8.543    -2.902    -4.126   186
H     8.959    -3.487    -4.823   186
H     8.449    -1.973    -4.484   186
O     -2.288    4.779    9.339   187
H     -2.796    3.953    9.585   187
H     -2.061    4.754    8.366   187
O     -3.934    -3.208    5.916   188
H     -3.188    -2.542    5.928   188
H     -4.429    -3.137    5.05   188
O     -0.872    -9.88    -6.056   189
H     -0.065    -10.416    -5.807   189
```

(continues on next page)

(continued from previous page)

```
    H     -0.715    -8.918    -5.833   189
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
```

(continues on next page)

(continued from previous page)

```
    H     2.47    4.016    4.736   208
    O     -3.182    3.268    -7.397   209
    H     -2.609    2.52    -7.062   209
    H     -3.481    3.066    -8.33   209
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
eor
```

Download FQQM_FQFMU_response.run

```
#!/bin/bash

# QM/FQFMU polarizability calculation

$AMSBIN/ams << eor
system
  Atoms
    C     1.32246     0.46994    -0.44553
    C     0.03851     0.37791     0.32993
    O    -1.13145     0.98673    -0.28822
    C    -1.09292    -0.44743    -0.11660
    H    -0.99733    -1.01009    -1.04597
    H    -1.81522    -0.83104     0.60631
    H     0.10287     0.59921     1.39924
    H     2.07829    -0.18966     0.00080
    H     1.71275     1.49611    -0.42640
    H     1.17353     0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  basis
     type TZP
     core none
  End
  response
    allcomponents True
    analytic True
    Frequencies 0.0 0.5 [eV]
  End
  symmetry NOSYM
  title QM/FQFMU molecular polarizability calculation
  xc
    gga PBE
  End

  QMFQ
    FORCEFIELD FQFMU
    Kernel GAUS
    MolCharge 0.00

    AtomType
      Symbol O
      Chi 0.2908429850
      Eta 0.5625181140
      Alpha 2.2187983720
    SubEnd
    AtomType
      Symbol H
      Chi 0.1675711970
      Eta 0.6093265770
```

(continues on next page)

(continued from previous page)

```
      Alpha 1.1906416090
    SubEnd
    Coords
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
    H     -10.69    -5.035    3.321   18
```

(continues on next page)

(continued from previous page)

```
    O     -3.63    -2.187    -1.384   19
    H     -2.767    -2.462    -1.807   19
    H     -3.822    -1.231    -1.607   19
    O     1.942    -6.524    -8.737   20
    H     0.996    -6.22    -8.852   20
    H     2.391    -6.563    -9.63   20
    O     -3.547    2.764    0.052   21
    H     -3.23    1.901    -0.342   21
    H     -2.83    3.455    -0.038   21
    O     1.904    8.539    -7.933   22
    H     2.085    8.395    -6.96   22
    H     0.923    8.447    -8.106   22
    O     5.07    6.46    -6.979   23
    H     5.462    5.945    -6.216   23
    H     5.106    7.438    -6.774   23
    O     1.317    1.858    -4.009   24
    H     0.789    2.105    -4.821   24
    H     1.565    0.891    -4.054   24
    O     -3.466    6.667    1.722   25
    H     -2.758    6.783    1.026   25
    H     -3.233    5.891    2.308   25
    O     9.374    -0.284    -4.021   26
    H     9.226    -0.462    -3.048   26
    H     8.904    0.56    -4.28   26
    O     -1.567    -0.699    -9.148   27
    H     -1.459    -1.69    -9.234   27
    H     -2.068    -0.348    -9.939   27
    O     2.355    -6.756    5.05   28
    H     3.312    -6.948    5.27   28
    H     2.119    -5.837    5.367   28
    O     -11.241    2.669    -0.446   29
    H     -10.442    2.227    -0.039   29
    H     -12.069    2.351    0.017   29
    O     -0.519    2.829    -2.219   30
    H     0.293    2.675    -2.782   30
    H     -0.756    1.984    -1.738   30
    O     2.736    2.528    1.949   31
    H     3.367    2.379    2.711   31
    H     1.894    2.946    2.29   31
    O     -3.662    -3.579    1.114   32
    H     -2.722    -3.33    1.349   32
    H     -3.879    -3.238    0.2   32
    O     -4.354    -0.05    9.809   33
    H     -4.059    -0.99    9.982   33
    H     -4.437    0.098    8.824   33
    O     -8.16    -1.139    0.546   34
    H     -8.636    -1.849    1.066   34
    H     -7.667    -1.558    -0.216   34
    O     -1.843    -1.194    6.421   35
    H     -1.115    -0.876    7.029   35
    H     -1.582    -1.022    5.471   35
    O     -1.16    -3.054    2.421   36
    H     -0.235    -2.945    2.057   36
    H     -1.147    -3.716    3.171   36
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
    H     0.061    1.437    -6.716   37
```

(continues on next page)

(continued from previous page)

```
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
    O     -0.792    1.114    -11.733   56
    H     -0.455    0.182    -11.865   56
    H     -0.023    1.721    -11.532   56
```

(continues on next page)

(continued from previous page)

```
    O     9.721    -2.916    -1.66   57
    H     9.417    -2.736    -2.596   57
    H     9.34    -3.787    -1.348   57
    O     -7.087    -4.974    -5.981   58
    H     -7.388    -4.844    -6.925   58
    H     -6.088    -4.953    -5.943   58
    O     4.858    8.204    6.274   59
    H     4.557    8.898    6.929   59
    H     4.59    7.298    6.6   59
    O     -5.74    4.317    -0.952   60
    H     -4.795    3.992    -0.963   60
    H     -5.777    5.221    -0.525   60
    O     4.091    6.159    -2.019   61
    H     3.956    5.352    -1.444   61
    H     3.21    6.456    -2.388   61
    O     0.259    -1.583    -11.55   62
    H     0.456    -2.193    -10.783   62
    H     1.112    -1.18    -11.882   62
    O     0.167    8.19    -1.584   63
    H     -0.233    8.98    -2.049   63
    H     0.999    7.909    -2.062   63
    O     5.904    5.593    -4.57   64
    H     5.781    4.816    -3.953   64
    H     5.595    6.43    -4.118   64
    O     1.853    2.264    7.653   65
    H     2.031    1.992    8.599   65
    H     0.866    2.341    7.509   65
    O     -3.917    -8.529    3.889   66
    H     -3.102    -8.099    3.5   66
    H     -3.747    -8.76    4.847   66
    O     -5.078    7.76    3.832   67
    H     -4.485    7.317    3.159   67
    H     -5.5    7.064    4.413   67
    O     -5.333    -7.661    -6.49   68
    H     -5.485    -7.46    -7.457   68
    H     -4.51    -7.187    -6.176   68
    O     1.603    9.353    1.706   69
    H     1.306    8.406    1.576   69
    H     2.011    9.452    2.614   69
    O     -8.588    7.768    -2.072   70
    H     -8.096    8.409    -1.483   70
    H     -8.101    6.895    -2.096   70
    O     8.949    -5.62    0.186   71
    H     8.046    -5.232    0.372   71
    H     9.636    -5.141    0.731   71
    O     -2.56    -3.378    -9.638   72
    H     -2.169    -4.295    -9.725   72
    H     -2.903    -3.077    -10.528   72
    O     4.132    5.755    7.273   73
    H     3.367    5.418    7.822   73
    H     4.613    4.982    6.859   73
    O     -8.001    -0.706    -7.091   74
    H     -8.965    -0.793    -7.341   74
    H     -7.795    0.252    -6.889   74
    O     1.985    -9.965    1.717   75
    H     1.164    -10.041    1.152   75
    H     1.726    -9.72    2.651   75
```

(continues on next page)

(continued from previous page)

```
    O     -0.889    -3.751    7.35   76
    H     -1.637    -4.413    7.286   76
    H     -1.14    -2.918    6.857   76
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
    H     -3.233    7.809    -4.93   94
```

(continues on next page)

(continued from previous page)

```
    O     -2.693    4.73    3.854   95
    H     -2.342    3.925    3.376   95
    H     -3.179    4.443    4.679   95
    O     1.001    -0.574    4.81   96
    H     0.553    0.313    4.696   96
    H     1.648    -0.72    4.062   96
    O     -7.658    3.829    -5.601   97
    H     -7.06    3.029    -5.656   97
    H     -8.518    3.635    -6.073   97
    O     1.26    6.271    8.504   98
    H     1.441    6.387    7.527   98
    H     0.355    6.637    8.721   98
    O     -2.775    -9.109    0.291   99
    H     -2.86    -8.211    0.723   99
    H     -1.809    -9.316    0.139   99
    O     3.256    -1.013    2.987   100
    H     3.539    -1.259    2.06   100
    H     3.855    -0.293    3.338   100
    O     0.21    -0.312    8.228   101
    H     1.026    -0.659    7.765   101
    H     0.428    0.548    8.688   101
    O     5.422    -0.784    -1.655   102
    H     5.497    -0.764    -2.652   102
    H     4.972    0.05    -1.337   102
    O     -1.59    -4.154    -5.957   103
    H     -1.997    -4.976    -5.559   103
    H     -1.274    -3.551    -5.225   103
    O     0.51    -8.72    3.751   104
    H     1.317    -8.458    4.28   104
    H     -0.042    -9.364    4.281   104
    O     -2.594    -6.607    -5.182   105
    H     -3.233    -6.683    -4.416   105
    H     -1.761    -7.12    -4.974   105
    O     -5.857    7.025    -0.001   106
    H     -4.915    7.129    0.32   106
    H     -6.458    7.622    0.53   106
    O     -0.705    -2.368    10.335   107
    H     -0.302    -2.039    9.481   107
    H     -1.387    -3.071    10.133   107
    O     -0.613    6.667    3.919   108
    H     -0.003    6.582    4.707   108
    H     -1.404    6.068    4.044   108
    O     3.935    -9.377    -3.564   109
    H     4.784    -8.856    -3.476   109
    H     3.704    -9.789    -2.683   109
    O     6.435    -3.938    -2.832   110
    H     7.169    -3.555    -3.391   110
    H     6.591    -3.706    -1.872   110
    O     6.896    8.206    0.362   111
    H     6.777    8.065    1.345   111
    H     6.022    8.469    -0.045   111
    O     -0.931    -3.296    -2.501   112
    H     -0.842    -2.655    -3.264   112
    H     -0.122    -3.232    -1.917   112
    O     -2.26    -0.566    -6.382   113
    H     -3.181    -0.898    -6.176   113
    H     -2.061    -0.717    -7.35   113
```

(continues on next page)

(continued from previous page)

```
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
```

(continues on next page)

(continued from previous page)

```
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
```

(continues on next page)

(continued from previous page)

```
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
    H     11.372    -2.508    0.962   170
```

(continues on next page)

(continued from previous page)

```
    O     -3.49    -10.263    -4.969   171
    H     -2.767    -9.841    -5.517   171
    H     -3.819    -9.607    -4.289   171
    O     -1.521    -7.2    2.797   172
    H     -1.357    -6.413    3.392   172
    H     -0.828    -7.899   2.97    172
    O     2.563    -4.792    -5.31   173
    H     1.916    -4.727    -4.55   173
    H     2.546    -5.719    -5.685   173
    O     2.837    -2.181    6.366   174
    H     2.273    -1.594    5.785   174
    H     3.201    -1.646    7.128   174
    O     -4.032    -6.368    -2.157   175
    H     -4.673    -5.671    -2.479   175
    H     -4.118    -6.467    -1.165   175
    O     -3.248    1.69    -4.498   176
    H     -2.674    2.482    -4.291   176
    H     -2.676    0.943    -4.835   176
    O     9.532    2.335    1.199   177
    H     9.808    3.202    0.786   177
    H     8.539    2.233    1.129   177
    O     4.296    6.036    3.312   178
    H     4.884    5.765    4.074   178
    H     3.344    6.057    3.617   178
    O     -4.003    0.618    -2.093   179
    H     -3.993    0.617    -3.093   179
    H     -4.888    0.953    -1.769   179
    O     6.312    5.31    5.167   180
    H     6.936    5.315    4.386   180
    H     6.368    6.187    5.644   180
    O     -6.267    -0.421    5.288   181
    H     -7.045    -1.049    5.313   181
    H     -6.538    0.461    5.671   181
    O     -4.077    0.335    6.986   182
    H     -4.824    -0.155    6.539   182
    H     -3.207    -0.105    6.762   182
    O     -1.592    10.18    -2.288   183
    H     -2.521    10.21    -2.657   183
    H     -1.428    10.996    -1.734   183
    O     -1.179    1.277    4.006   184
    H     -1.662    0.404    3.942   184
    H     -1.541    1.802    4.776   184
    O     -5.857    -7.312    2.548   185
    H     -5.09    -7.722    3.042   185
    H     -6.533    -8.017    2.337   185
    O     8.543    -2.902    -4.126   186
    H     8.959    -3.487    -4.823   186
    H     8.449    -1.973    -4.484   186
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
    O     -0.872    -9.88    -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
```

(continues on next page)

(continued from previous page)

```
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
    H     2.47    4.016    4.736   208
```

(continues on next page)

(continued from previous page)

```
    O     -3.182    3.268    -7.397   209
    H     -2.609    2.52     -7.062   209
    H     -3.481    3.066    -8.33    209
    O     -6.334    1.42     -0.932   210
    H     -7.057    0.73     -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45     7.562   212
    H     -5.324    6.26     7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57    216
    H     10.975    1.438    0.171    216
    H     11.176    1.745    -1.42    216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42     -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
eor
```

Example: QM/FQ(Fμ): excitations 2-Methyloxirane

Two examples: a QM/FQ calculation and a QM/FQFμ calculation.

Download FQQM_td.run

```
#!/bin/bash

# QMFQ excitations calculation

$AMSBIN/ams << eor
system
  Atoms
    C     1.32246     0.46994    -0.44553
    C     0.03851     0.37791     0.32993
    O    -1.13145     0.98673    -0.28822
    C    -1.09292    -0.44743    -0.11660
    H    -0.99733    -1.01009    -1.04597
    H    -1.81522    -0.83104     0.60631
    H     0.10287     0.59921     1.39924
    H     2.07829    -0.18966     0.00080
    H     1.71275     1.49611    -0.42640
    H     1.17353     0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  basis
     type TZP
     core none
  End
  excitations
    lowest 10
    onlysing True
  End
  symmetry NOSYM
  title QM/FQ excitations calculation
  xc
    gga PBE
  End
  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00

    AtomType
      Symbol O
      Chi 0.189194
      Eta 0.523700
    SubEnd
    AtomType
      Symbol H
      Chi 0.012767
      Eta 0.537512
```

(continues on next page)

(continued from previous page)

```
    SubEnd
    Coords
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
    H     -10.69    -5.035    3.321   18
    O     -3.63    -2.187    -1.384   19
```

(continues on next page)

(continued from previous page)

```
    H     -2.767    -2.462    -1.807   19
    H     -3.822    -1.231    -1.607   19
    O     1.942    -6.524    -8.737   20
    H     0.996    -6.22    -8.852   20
    H     2.391    -6.563    -9.63   20
    O     -3.547    2.764    0.052   21
    H     -3.23    1.901    -0.342   21
    H     -2.83    3.455    -0.038   21
    O     1.904    8.539    -7.933   22
    H     2.085    8.395    -6.96   22
    H     0.923    8.447    -8.106   22
    O     5.07    6.46    -6.979   23
    H     5.462    5.945    -6.216   23
    H     5.106    7.438    -6.774   23
    O     1.317    1.858    -4.009   24
    H     0.789    2.105    -4.821   24
    H     1.565    0.891    -4.054   24
    O     -3.466    6.667    1.722   25
    H     -2.758    6.783    1.026   25
    H     -3.233    5.891    2.308   25
    O     9.374    -0.284    -4.021   26
    H     9.226    -0.462    -3.048   26
    H     8.904    0.56    -4.28   26
    O     -1.567    -0.699    -9.148   27
    H     -1.459    -1.69    -9.234   27
    H     -2.068    -0.348    -9.939   27
    O     2.355    -6.756    5.05   28
    H     3.312    -6.948    5.27   28
    H     2.119    -5.837    5.367   28
    O     -11.241    2.669    -0.446   29
    H     -10.442    2.227    -0.039   29
    H     -12.069    2.351    0.017   29
    O     -0.519    2.829    -2.219   30
    H     0.293    2.675    -2.782   30
    H     -0.756    1.984    -1.738   30
    O     2.736    2.528    1.949   31
    H     3.367    2.379    2.711   31
    H     1.894    2.946    2.29   31
    O     -3.662    -3.579    1.114   32
    H     -2.722    -3.33    1.349   32
    H     -3.879    -3.238    0.2   32
    O     -4.354    -0.05    9.809   33
    H     -4.059    -0.99    9.982   33
    H     -4.437    0.098    8.824   33
    O     -8.16    -1.139    0.546   34
    H     -8.636    -1.849    1.066   34
    H     -7.667    -1.558    -0.216   34
    O     -1.843    -1.194    6.421   35
    H     -1.115    -0.876    7.029   35
    H     -1.582    -1.022    5.471   35
    O     -1.16    -3.054    2.421   36
    H     -0.235    -2.945    2.057   36
    H     -1.147    -3.716    3.171   36
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
    H     0.061    1.437    -6.716   37
    O     0.34    -7.527    7.536   38
```

(continues on next page)

(continued from previous page)

```
    H     0.901    -7.234    6.761    38
    H     0.225    -8.52    7.508    38
    O     -1.493    -5.793    -0.627    39
    H     -2.034    -5.067    -1.052    39
    H     -2.041    -6.249    0.074    39
    O     3.987    0.323    -6.433    40
    H     3.903    0.239    -7.426    40
    H     4.943    0.201    -6.168    40
    O     0.987    -6.627    0.39    41
    H     0.661    -7.469    0.819    41
    H     0.249    -6.221    -0.15    41
    O     -9.847    1.488    4.432    42
    H     -10.021    0.672    3.882    42
    H     -10.581    2.151    4.286    42
    O     2.111    10.191    -2.495    43
    H     2.186    10.37    -1.514    43
    H     1.935    11.049    -2.977    43
    O     -4.427    3.296    5.288    44
    H     -4.348    2.855    6.182    44
    H     -4.246    2.626    4.568    44
    O     10.082    3.061    -2.716    45
    H     10.773    3.744    -2.482    45
    H     10.172    2.812    -3.68    45
    O     4.76    -3.022    -4.717    46
    H     5.405    -3.261    -3.991    46
    H     4.172    -3.806    -4.913    46
    O     3.545    -7.379    0.472    47
    H     3.686    -8.159    -0.137    47
    H     2.571    -7.157    0.508    47
    O     7.795    2.559    -1.356    48
    H     7.891    1.812    -0.698    48
    H     8.689    2.776    -1.747    48
    O     3.429    -0.252    -9.178    49
    H     3.495    0.04    -10.133    49
    H     3.756    -1.193    -9.094    49
    O     -3.579    2.402    -10.241    50
    H     -3.406    1.508    -10.654    50
    H     -2.873    3.047    -10.534    50
    O     1.105    -2.91    -1.017    51
    H     0.927    -2.797    -0.04    51
    H     2.018    -3.296    -1.147    51
    O     -0.849    3.983    0.308    52
    H     -0.713    3.325    -0.433    52
    H     -0.973    4.898    -0.077    52
    O     6.657    2.671    4.09    53
    H     7.3    3.358    3.752    53
    H     5.874    3.127    4.513    53
    O     -1.501    6.653    -0.432    54
    H     -1.883    6.036    -1.12    54
    H     -0.862    7.286    -0.869    54
    O     3.395    -4.498    -1.626    55
    H     3.676    -4.53    -0.667    55
    H     4.202    -4.558    -2.213    55
    O     -0.792    1.114    -11.733    56
    H     -0.455    0.182    -11.865    56
    H     -0.023    1.721    -11.532    56
    O     9.721    -2.916    -1.66    57
```

(continues on next page)

(continued from previous page)

```
    H     9.417    -2.736    -2.596   57
    H     9.34    -3.787    -1.348   57
    O     -7.087    -4.974    -5.981   58
    H     -7.388    -4.844    -6.925   58
    H     -6.088    -4.953    -5.943   58
    O     4.858    8.204    6.274   59
    H     4.557    8.898    6.929   59
    H     4.59    7.298    6.6   59
    O     -5.74    4.317    -0.952   60
    H     -4.795    3.992    -0.963   60
    H     -5.777    5.221    -0.525   60
    O     4.091    6.159    -2.019   61
    H     3.956    5.352    -1.444   61
    H     3.21    6.456    -2.388   61
    O     0.259    -1.583    -11.55   62
    H     0.456    -2.193    -10.783   62
    H     1.112    -1.18    -11.882   62
    O     0.167    8.19    -1.584   63
    H     -0.233    8.98    -2.049   63
    H     0.999    7.909    -2.062   63
    O     5.904    5.593    -4.57   64
    H     5.781    4.816    -3.953   64
    H     5.595    6.43    -4.118   64
    O     1.853    2.264    7.653   65
    H     2.031    1.992    8.599   65
    H     0.866    2.341    7.509   65
    O     -3.917    -8.529    3.889   66
    H     -3.102    -8.099    3.5   66
    H     -3.747    -8.76    4.847   66
    O     -5.078    7.76    3.832   67
    H     -4.485    7.317    3.159   67
    H     -5.5    7.064    4.413   67
    O     -5.333    -7.661    -6.49   68
    H     -5.485    -7.46    -7.457   68
    H     -4.51    -7.187    -6.176   68
    O     1.603    9.353    1.706   69
    H     1.306    8.406    1.576   69
    H     2.011    9.452    2.614   69
    O     -8.588    7.768    -2.072   70
    H     -8.096    8.409    -1.483   70
    H     -8.101    6.895    -2.096   70
    O     8.949    -5.62    0.186   71
    H     8.046    -5.232    0.372   71
    H     9.636    -5.141    0.731   71
    O     -2.56    -3.378    -9.638   72
    H     -2.169    -4.295    -9.725   72
    H     -2.903    -3.077    -10.528   72
    O     4.132    5.755    7.273   73
    H     3.367    5.418    7.822   73
    H     4.613    4.982    6.859   73
    O     -8.001    -0.706    -7.091   74
    H     -8.965    -0.793    -7.341   74
    H     -7.795    0.252    -6.889   74
    O     1.985    -9.965    1.717   75
    H     1.164    -10.041    1.152   75
    H     1.726    -9.72    2.651   75
    O     -0.889    -3.751    7.35   76
```

(continues on next page)

(continued from previous page)

```
    H     -1.637    -4.413    7.286   76
    H     -1.14    -2.918    6.857   76
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
    H     -3.233    7.809    -4.93   94
    O     -2.693    4.73    3.854   95
```

(continues on next page)

(continued from previous page)

```
    H     -2.342    3.925    3.376   95
    H     -3.179    4.443    4.679   95
    O     1.001    -0.574    4.81   96
    H     0.553    0.313    4.696   96
    H     1.648    -0.72    4.062   96
    O     -7.658    3.829    -5.601   97
    H     -7.06    3.029    -5.656   97
    H     -8.518    3.635    -6.073   97
    O     1.26    6.271    8.504   98
    H     1.441    6.387    7.527   98
    H     0.355    6.637    8.721   98
    O     -2.775    -9.109    0.291   99
    H     -2.86    -8.211    0.723   99
    H     -1.809    -9.316    0.139   99
    O     3.256    -1.013    2.987   100
    H     3.539    -1.259    2.06   100
    H     3.855    -0.293    3.338   100
    O     0.21    -0.312    8.228   101
    H     1.026    -0.659    7.765   101
    H     0.428    0.548    8.688   101
    O     5.422    -0.784    -1.655   102
    H     5.497    -0.764    -2.652   102
    H     4.972    0.05    -1.337   102
    O     -1.59    -4.154    -5.957   103
    H     -1.997    -4.976    -5.559   103
    H     -1.274    -3.551    -5.225   103
    O     0.51    -8.72    3.751   104
    H     1.317    -8.458    4.28   104
    H     -0.042    -9.364    4.281   104
    O     -2.594    -6.607    -5.182   105
    H     -3.233    -6.683    -4.416   105
    H     -1.761    -7.12    -4.974   105
    O     -5.857    7.025    -0.001   106
    H     -4.915    7.129    0.32   106
    H     -6.458    7.622    0.53   106
    O     -0.705    -2.368    10.335   107
    H     -0.302    -2.039    9.481   107
    H     -1.387    -3.071    10.133   107
    O     -0.613    6.667    3.919   108
    H     -0.003    6.582    4.707   108
    H     -1.404    6.068    4.044   108
    O     3.935    -9.377    -3.564   109
    H     4.784    -8.856    -3.476   109
    H     3.704    -9.789    -2.683   109
    O     6.435    -3.938    -2.832   110
    H     7.169    -3.555    -3.391   110
    H     6.591    -3.706    -1.872   110
    O     6.896    8.206    0.362   111
    H     6.777    8.065    1.345   111
    H     6.022    8.469    -0.045   111
    O     -0.931    -3.296    -2.501   112
    H     -0.842    -2.655    -3.264   112
    H     -0.122    -3.232    -1.917   112
    O     -2.26    -0.566    -6.382   113
    H     -3.181    -0.898    -6.176   113
    H     -2.061    -0.717    -7.35   113
    O     -3.942    -6.403    0.843   114
```

(continues on next page)

(continued from previous page)

```
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
    O     8.133    5.499    3.039   133
```

(continues on next page)

(continued from previous page)

```
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
    O     -5.629    1.786    -5.896   152
```

(continues on next page)

(continued from previous page)

```
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
    H     11.372    -2.508    0.962   170
    O     -3.49    -10.263    -4.969   171
```

(continues on next page)

(continued from previous page)

```
    H     -2.767    -9.841    -5.517   171
    H     -3.819    -9.607    -4.289   171
    O     -1.521    -7.2    2.797   172
    H     -1.357    -6.413    3.392   172
    H     -0.828    -7.899    2.97   172
    O     2.563    -4.792    -5.31   173
    H     1.916    -4.727    -4.55   173
    H     2.546    -5.719    -5.685   173
    O     2.837    -2.181    6.366   174
    H     2.273    -1.594    5.785   174
    H     3.201    -1.646    7.128   174
    O     -4.032    -6.368    -2.157   175
    H     -4.673    -5.671    -2.479   175
    H     -4.118    -6.467    -1.165   175
    O     -3.248    1.69    -4.498   176
    H     -2.674    2.482    -4.291   176
    H     -2.676    0.943    -4.835   176
    O     9.532    2.335    1.199   177
    H     9.808    3.202    0.786   177
    H     8.539    2.233    1.129   177
    O     4.296    6.036    3.312   178
    H     4.884    5.765    4.074   178
    H     3.344    6.057    3.617   178
    O     -4.003    0.618    -2.093   179
    H     -3.993    0.617    -3.093   179
    H     -4.888    0.953    -1.769   179
    O     6.312    5.31    5.167   180
    H     6.936    5.315    4.386   180
    H     6.368    6.187    5.644   180
    O     -6.267    -0.421    5.288   181
    H     -7.045    -1.049    5.313   181
    H     -6.538    0.461    5.671   181
    O     -4.077    0.335    6.986   182
    H     -4.824    -0.155    6.539   182
    H     -3.207    -0.105    6.762   182
    O     -1.592    10.18    -2.288   183
    H     -2.521    10.21    -2.657   183
    H     -1.428    10.996    -1.734   183
    O     -1.179    1.277    4.006   184
    H     -1.662    0.404    3.942   184
    H     -1.541    1.802    4.776   184
    O     -5.857    -7.312    2.548   185
    H     -5.09    -7.722    3.042   185
    H     -6.533    -8.017    2.337   185
    O     8.543    -2.902    -4.126   186
    H     8.959    -3.487    -4.823   186
    H     8.449    -1.973    -4.484   186
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
    O     -0.872    -9.88    -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
    O     -2.461    4.695    -2.278   190
```

(continues on next page)

(continued from previous page)

```
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
    O     6.344    -0.092    7.977   191
    H     6.51    -0.546    7.101   191
    H     6.343    0.899    7.843   191
    O     -5.698    -0.661    2.606   192
    H     -5.841    -0.727    3.593   192
    H     -6.565    -0.816    2.132   192
    O     -4.029    -2.72    10.246   193
    H     -5.009    -2.685    10.442   193
    H     -3.786    -3.634    9.923   193
    O     2.832    -3.188    -7.657   194
    H     2.732    -3.704    -6.806   194
    H     3.768    -3.282    -7.996   194
    O     1.073    -10.115    -2.625   195
    H     0.46    -9.352    -2.419   195
    H     1.445    -10.004    -3.547   195
    O     -6.62    3.126    1.407   196
    H     -5.909    2.655    1.93   196
    H     -6.194    3.705    0.711   196
    O     -0.677    8.512    -5.231   197
    H     -0.896    9.254    -4.597   197
    H     0.11    8.001    -4.887   197
    O     4.12    4.53    0.653   198
    H     4.223    5.225    1.364   198
    H     3.485    3.824    0.967   198
    O     -8.221    -2.456    5.669   199
    H     -8.808    -2.293    6.463   199
    H     -7.852    -3.384    5.712   199
    O     2.898    4.605    -5.775   200
    H     3.521    4.999    -6.45   200
    H     2.968    3.608    -5.798   200
    O     -8.042    -4.503    0.126   201
    H     -7.787    -3.888    -0.62   201
    H     -9.001    -4.768    0.027   201
    O     0.142    -7.397    -5.046   202
    H     0.431    -6.604    -4.511   202
    H     0.919    -7.748    -5.568   202
    O     -5.157    -1.438    -5.812   203
    H     -5.788    -0.913    -6.384   203
    H     -4.908    -2.285    -6.281   203
    O     -1.04    -5.021    4.465   204
    H     -1.767    -5.433    5.015   204
    H     -0.158    -5.409    4.733   204
    O     6.958    4.266    0.911   205
    H     5.988    4.074    0.766   205
    H     7.395    4.46    0.033   205
    O     7.345    0.71    0.51   206
    H     7.532    0.27    1.389   206
    H     6.36    0.714    0.342   206
    O     2.18    -0.941    -4.851   207
    H     2.768    -1.736    -4.7   207
    H     2.46    -0.474    -5.689   207
    O     1.874    4.72    4.35   208
    H     1.127    4.286    3.847   208
    H     2.47    4.016    4.736   208
    O     -3.182    3.268    -7.397   209
```

(continues on next page)

(continued from previous page)

```
    H     -2.609    2.52    -7.062   209
    H     -3.481    3.066    -8.33   209
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
eor
```

`Download FQQM_FQFMU_td.run`

```
#!/bin/bash

# QM/FQFMU excitations calculation

$AMSBIN/ams << eor
system
  Atoms
    C     1.32246     0.46994    -0.44553
    C     0.03851     0.37791     0.32993
    O    -1.13145     0.98673    -0.28822
    C    -1.09292    -0.44743    -0.11660
    H    -0.99733    -1.01009    -1.04597
    H    -1.81522    -0.83104     0.60631
    H     0.10287     0.59921     1.39924
    H     2.07829    -0.18966     0.00080
    H     1.71275     1.49611    -0.42640
    H     1.17353     0.16408    -1.48893
  End
End

task SinglePoint

Engine adf
  Relativity
    Level None
  End
  basis
     type TZP
     core none
  End
  excitations
    lowest 10
    onlysing True
  End
  symmetry NOSYM
  title QM/FQFMU excitations calculation
  xc
    gga PBE
  End
  QMFQ
    FORCEFIELD FQFMU
    Kernel GAUS
    MolCharge 0.00

    AtomType
      Symbol O
      Chi 0.2908429850
      Eta 0.5625181140
      Alpha 2.2187983720
    SubEnd
    AtomType
      Symbol H
      Chi 0.1675711970
      Eta 0.6093265770
      Alpha 1.1906416090
    SubEnd
    Coords
```

(continues on next page)

(continued from previous page)

```
    O     6.179    -4.919    2.699   1
    H     7.008    -4.361    2.695   1
    H     5.687    -4.779    3.558   1
    O     3.341    2.501    5.245   2
    H     2.813    2.455    6.093   2
    H     3.93    1.695    5.175   2
    O     2.692    8.795    -4.87   3
    H     3.592    8.369    -4.781   3
    H     2.444    9.232    -4.005   3
    O     -2.039    5.881    6.772   4
    H     -3.025    6.045    6.742   4
    H     -1.597    6.604    7.301   4
    O     4.733    -3.969    5.277   5
    H     3.944    -3.36    5.362   5
    H     4.456    -4.904    5.497   5
    O     -0.143    3.368    -8.787   6
    H     0.262    2.454    -8.8   6
    H     -0.588    3.548    -9.665   6
    O     0.957    0.844    -8.505   7
    H     1.915    0.563    -8.573   7
    H     0.399    0.276    -9.11   7
    O     -2.971    -5.482    6.927   8
    H     -3.616    -6.141    7.313   8
    H     -3.473    -4.692    6.574   8
    O     3.049    3.883    -9.821   9
    H     2.387    4.418    -9.297   9
    H     2.574    3.376    -10.54   9
    O     6.079    8.112    2.853   10
    H     6.635    7.88    3.651   10
    H     5.245    7.56    2.855   10
    O     3.022    -6.207    8.919   11
    H     2.452    -6.216    9.74   11
    H     2.653    -5.546    8.265   11
    O     1.49    1.433    10.208   12
    H     2.352    1.441    10.715   12
    H     0.73    1.581    10.84   12
    O     -2.472    4.816    -5.151   13
    H     -1.53    5.086    -5.349   13
    H     -2.91    4.492    -5.99   13
    O     -6.138    7.963    -4.272   14
    H     -6.292    7.128    -3.744   14
    H     -5.371    7.824    -4.898   14
    O     6.384    -7.3    0.928   15
    H     5.462    -7.396    0.554   15
    H     6.379    -6.615    1.657   15
    O     8.759    -0.26    -1.401   16
    H     8.949    -1.242    -1.397   16
    H     7.993    -0.066    -0.788   16
    O     0.745    6.829    1.703   17
    H     0.439    6.196    0.992   17
    H     0.186    6.704    2.523   17
    O     -10.018    -4.546    3.877   18
    H     -9.098    -4.855    3.637   18
    H     -10.69    -5.035    3.321   18
    O     -3.63    -2.187    -1.384   19
    H     -2.767    -2.462    -1.807   19
    H     -3.822    -1.231    -1.607   19
```

(continues on next page)

(continued from previous page)

```
    O     1.942    -6.524    -8.737   20
    H     0.996    -6.22    -8.852   20
    H     2.391    -6.563    -9.63   20
    O     -3.547    2.764    0.052   21
    H     -3.23    1.901    -0.342   21
    H     -2.83    3.455    -0.038   21
    O     1.904    8.539    -7.933   22
    H     2.085    8.395    -6.96   22
    H     0.923    8.447    -8.106   22
    O     5.07    6.46    -6.979   23
    H     5.462    5.945    -6.216   23
    H     5.106    7.438    -6.774   23
    O     1.317    1.858    -4.009   24
    H     0.789    2.105    -4.821   24
    H     1.565    0.891    -4.054   24
    O     -3.466    6.667    1.722   25
    H     -2.758    6.783    1.026   25
    H     -3.233    5.891    2.308   25
    O     9.374    -0.284    -4.021   26
    H     9.226    -0.462    -3.048   26
    H     8.904    0.56    -4.28   26
    O     -1.567    -0.699    -9.148   27
    H     -1.459    -1.69    -9.234   27
    H     -2.068    -0.348    -9.939   27
    O     2.355    -6.756    5.05   28
    H     3.312    -6.948    5.27   28
    H     2.119    -5.837    5.367   28
    O     -11.241    2.669    -0.446   29
    H     -10.442    2.227    -0.039   29
    H     -12.069    2.351    0.017   29
    O     -0.519    2.829    -2.219   30
    H     0.293    2.675    -2.782   30
    H     -0.756    1.984    -1.738   30
    O     2.736    2.528    1.949   31
    H     3.367    2.379    2.711   31
    H     1.894    2.946    2.29   31
    O     -3.662    -3.579    1.114   32
    H     -2.722    -3.33    1.349   32
    H     -3.879    -3.238    0.2   32
    O     -4.354    -0.05    9.809   33
    H     -4.059    -0.99    9.982   33
    H     -4.437    0.098    8.824   33
    O     -8.16    -1.139    0.546   34
    H     -8.636    -1.849    1.066   34
    H     -7.667    -1.558    -0.216   34
    O     -1.843    -1.194    6.421   35
    H     -1.115    -0.876    7.029   35
    H     -1.582    -1.022    5.471   35
    O     -1.16    -3.054    2.421   36
    H     -0.235    -2.945    2.057   36
    H     -1.147    -3.716    3.171   36
    O     -0.447    2.125    -6.198   37
    H     -0.425    2.996    -6.688   37
    H     0.061    1.437    -6.716   37
    O     0.34    -7.527    7.536   38
    H     0.901    -7.234    6.761   38
    H     0.225    -8.52    7.508   38
```

(continues on next page)

(continued from previous page)

```
    O     -1.493    -5.793    -0.627   39
    H     -2.034    -5.067    -1.052   39
    H     -2.041    -6.249    0.074   39
    O     3.987    0.323    -6.433   40
    H     3.903    0.239    -7.426   40
    H     4.943    0.201    -6.168   40
    O     0.987    -6.627    0.39   41
    H     0.661    -7.469    0.819   41
    H     0.249    -6.221    -0.15   41
    O     -9.847    1.488    4.432   42
    H     -10.021    0.672    3.882   42
    H     -10.581    2.151    4.286   42
    O     2.111    10.191    -2.495   43
    H     2.186    10.37    -1.514   43
    H     1.935    11.049    -2.977   43
    O     -4.427    3.296    5.288   44
    H     -4.348    2.855    6.182   44
    H     -4.246    2.626    4.568   44
    O     10.082    3.061    -2.716   45
    H     10.773    3.744    -2.482   45
    H     10.172    2.812    -3.68   45
    O     4.76    -3.022    -4.717   46
    H     5.405    -3.261    -3.991   46
    H     4.172    -3.806    -4.913   46
    O     3.545    -7.379    0.472   47
    H     3.686    -8.159    -0.137   47
    H     2.571    -7.157    0.508   47
    O     7.795    2.559    -1.356   48
    H     7.891    1.812    -0.698   48
    H     8.689    2.776    -1.747   48
    O     3.429    -0.252    -9.178   49
    H     3.495    0.04    -10.133   49
    H     3.756    -1.193    -9.094   49
    O     -3.579    2.402    -10.241   50
    H     -3.406    1.508    -10.654   50
    H     -2.873    3.047    -10.534   50
    O     1.105    -2.91    -1.017   51
    H     0.927    -2.797    -0.04   51
    H     2.018    -3.296    -1.147   51
    O     -0.849    3.983    0.308   52
    H     -0.713    3.325    -0.433   52
    H     -0.973    4.898    -0.077   52
    O     6.657    2.671    4.09   53
    H     7.3    3.358    3.752   53
    H     5.874    3.127    4.513   53
    O     -1.501    6.653    -0.432   54
    H     -1.883    6.036    -1.12   54
    H     -0.862    7.286    -0.869   54
    O     3.395    -4.498    -1.626   55
    H     3.676    -4.53    -0.667   55
    H     4.202    -4.558    -2.213   55
    O     -0.792    1.114    -11.733   56
    H     -0.455    0.182    -11.865   56
    H     -0.023    1.721    -11.532   56
    O     9.721    -2.916    -1.66   57
    H     9.417    -2.736    -2.596   57
    H     9.34    -3.787    -1.348   57
```

(continues on next page)

(continued from previous page)

```
O     -7.087    -4.974    -5.981   58
H     -7.388    -4.844    -6.925   58
H     -6.088    -4.953    -5.943   58
O     4.858    8.204    6.274   59
H     4.557    8.898    6.929   59
H     4.59    7.298    6.6   59
O     -5.74    4.317    -0.952   60
H     -4.795    3.992    -0.963   60
H     -5.777    5.221    -0.525   60
O     4.091    6.159    -2.019   61
H     3.956    5.352    -1.444   61
H     3.21    6.456    -2.388   61
O     0.259    -1.583    -11.55   62
H     0.456    -2.193    -10.783   62
H     1.112    -1.18    -11.882   62
O     0.167    8.19    -1.584   63
H     -0.233    8.98    -2.049   63
H     0.999    7.909    -2.062   63
O     5.904    5.593    -4.57   64
H     5.781    4.816    -3.953   64
H     5.595    6.43    -4.118   64
O     1.853    2.264    7.653   65
H     2.031    1.992    8.599   65
H     0.866    2.341    7.509   65
O     -3.917    -8.529    3.889   66
H     -3.102    -8.099    3.5   66
H     -3.747    -8.76    4.847   66
O     -5.078    7.76    3.832   67
H     -4.485    7.317    3.159   67
H     -5.5    7.064    4.413   67
O     -5.333    -7.661    -6.49   68
H     -5.485    -7.46    -7.457   68
H     -4.51    -7.187    -6.176   68
O     1.603    9.353    1.706   69
H     1.306    8.406    1.576   69
H     2.011    9.452    2.614   69
O     -8.588    7.768    -2.072   70
H     -8.096    8.409    -1.483   70
H     -8.101    6.895    -2.096   70
O     8.949    -5.62    0.186   71
H     8.046    -5.232    0.372   71
H     9.636    -5.141    0.731   71
O     -2.56    -3.378    -9.638   72
H     -2.169    -4.295    -9.725   72
H     -2.903    -3.077    -10.528   72
O     4.132    5.755    7.273   73
H     3.367    5.418    7.822   73
H     4.613    4.982    6.859   73
O     -8.001    -0.706    -7.091   74
H     -8.965    -0.793    -7.341   74
H     -7.795    0.252    -6.889   74
O     1.985    -9.965    1.717   75
H     1.164    -10.041    1.152   75
H     1.726    -9.72    2.651   75
O     -0.889    -3.751    7.35   76
H     -1.637    -4.413    7.286   76
H     -1.14    -2.918    6.857   76
```

(continues on next page)

(continued from previous page)

```
    O     6.447    -8.648    -3.384   77
    H     6.956    -8.025    -2.791   77
    H     7.079    -9.101    -4.013   77
    O     -10.087    1.376    -3.258   78
    H     -10.346    1.756    -2.37   78
    H     -10.584    1.853    -3.982   78
    O     -7.721    -6.978    -4.007   79
    H     -7.044    -7.461    -4.562   79
    H     -7.673    -5.998    -4.2   79
    O     3.742    -9.925    -0.53   80
    H     4.013    -10.88    -0.645   80
    H     3.147    -9.841    0.269   80
    O     2.643    11.407    0.185   81
    H     3.532    11.713    0.526   81
    H     2.357    10.591    0.686   81
    O     8.142    -2.725    7.39   82
    H     7.867    -3.625    7.728   82
    H     8.341    -2.124    8.164   82
    O     6.161    3.092    -3.472   83
    H     6.808    3.021    -2.713   83
    H     6.61    2.81    -4.32   83
    O     -0.296    -10.611    0.551   84
    H     0.082    -11.185    -0.175   84
    H     -1.124    -11.036    0.916   84
    O     8.597    -7.332    -1.93   85
    H     9.074    -6.978    -2.734   85
    H     8.663    -6.67    -1.184   85
    O     6.902    -2.876    0.076   86
    H     7.286    -2.931    0.998   86
    H     6.464    -1.986    -0.05   86
    O     -6.753    2.76    8.111   87
    H     -7.086    3.694    8.246   87
    H     -6.953    2.467    7.176   87
    O     -7.552    -4.988    2.842   88
    H     -7.782    -4.906    1.872   88
    H     -7.032    -5.829    2.991   88
    O     -8.182    -7.7    -1.006   89
    H     -7.682    -7.051    -1.581   89
    H     -7.896    -8.632    -1.23   89
    O     8.372    -3.157    2.438   90
    H     8.557    -3.167    3.421   90
    H     9.221    -2.98    1.941   90
    O     -6.198    -1.22    -3.217   91
    H     -6.53    -0.276    -3.211   91
    H     -5.626    -1.367    -4.024   91
    O     4.677    1.242    0.094   92
    H     4.368    1.864    -0.625   92
    H     4.105    1.361    0.906   92
    O     -2.993    -1.195    3.495   93
    H     -3.849    -1.308    2.99   93
    H     -2.282    -1.76    3.075   93
    O     -3.81    7.752    -5.745   94
    H     -3.268    7.416    -6.515   94
    H     -3.233    7.809    -4.93   94
    O     -2.693    4.73    3.854   95
    H     -2.342    3.925    3.376   95
    H     -3.179    4.443    4.679   95
```

(continues on next page)

(continued from previous page)

```
    O     1.001    -0.574    4.81   96
    H     0.553    0.313    4.696   96
    H     1.648    -0.72    4.062   96
    O     -7.658    3.829    -5.601   97
    H     -7.06    3.029    -5.656   97
    H     -8.518    3.635    -6.073   97
    O     1.26    6.271    8.504   98
    H     1.441    6.387    7.527   98
    H     0.355    6.637    8.721   98
    O     -2.775    -9.109    0.291   99
    H     -2.86    -8.211    0.723   99
    H     -1.809    -9.316    0.139   99
    O     3.256    -1.013    2.987   100
    H     3.539    -1.259    2.06   100
    H     3.855    -0.293    3.338   100
    O     0.21    -0.312    8.228   101
    H     1.026    -0.659    7.765   101
    H     0.428    0.548    8.688   101
    O     5.422    -0.784    -1.655   102
    H     5.497    -0.764    -2.652   102
    H     4.972    0.05    -1.337   102
    O     -1.59    -4.154    -5.957   103
    H     -1.997    -4.976    -5.559   103
    H     -1.274    -3.551    -5.225   103
    O     0.51    -8.72    3.751   104
    H     1.317    -8.458    4.28   104
    H     -0.042    -9.364    4.281   104
    O     -2.594    -6.607    -5.182   105
    H     -3.233    -6.683    -4.416   105
    H     -1.761    -7.12    -4.974   105
    O     -5.857    7.025    -0.001   106
    H     -4.915    7.129    0.32   106
    H     -6.458    7.622    0.53   106
    O     -0.705    -2.368    10.335   107
    H     -0.302    -2.039    9.481   107
    H     -1.387    -3.071    10.133   107
    O     -0.613    6.667    3.919   108
    H     -0.003    6.582    4.707   108
    H     -1.404    6.068    4.044   108
    O     3.935    -9.377    -3.564   109
    H     4.784    -8.856    -3.476   109
    H     3.704    -9.789    -2.683   109
    O     6.435    -3.938    -2.832   110
    H     7.169    -3.555    -3.391   110
    H     6.591    -3.706    -1.872   110
    O     6.896    8.206    0.362   111
    H     6.777    8.065    1.345   111
    H     6.022    8.469    -0.045   111
    O     -0.931    -3.296    -2.501   112
    H     -0.842    -2.655    -3.264   112
    H     -0.122    -3.232    -1.917   112
    O     -2.26    -0.566    -6.382   113
    H     -3.181    -0.898    -6.176   113
    H     -2.061    -0.717    -7.35   113
    O     -3.942    -6.403    0.843   114
    H     -4.632    -6.735    1.485   114
    H     -3.718    -5.452    1.057   114
```

(continues on next page)

(continued from previous page)

```
    O     -6.161    -3.718    7.927   115
    H     -6.312    -3.364    8.85   115
    H     -5.542    -3.107    7.433   115
    O     -3.529    2.117    11.116   116
    H     -2.591    2.344    11.376   116
    H     -3.524    1.334    10.494   116
    O     11.264    0.732    2.673   117
    H     10.509    1.161    2.179   117
    H     11.435    1.232    3.522   117
    O     5.878    4.725    -8.717   118
    H     5.363    5.296    -8.078   118
    H     5.285    4.457    -9.477   118
    O     6.383    -1.828    5.578   119
    H     5.775    -2.616    5.484   119
    H     7.277    -2.134    5.904   119
    O     5.055    -6.955    6.321   120
    H     4.585    -7.29    7.138   120
    H     5.653    -7.672    5.963   120
    O     -2.944    -5.135    9.603   121
    H     -2.164    -5.688    9.895   121
    H     -3.022    -5.167    8.607   121
    O     -7.242    2.071    5.296   122
    H     -8.153    1.824    4.966   122
    H     -6.948    2.921    4.859   122
    O     -4.801    -10.888    0.386   123
    H     -4.064    -10.22    0.494   123
    H     -5.563    -10.466    -0.105   123
    O     1.498    -2.946    1.749   124
    H     2.0    -2.206    2.198   124
    H     2.01    -3.8    1.842   124
    O     2.527    9.263    4.264   125
    H     1.982    9.895    4.815   125
    H     3.379    9.052    4.743   125
    O     7.852    1.576    -5.351   126
    H     7.338    0.737    -5.531   126
    H     8.162    1.969    -6.217   126
    O     -5.735    4.508    -7.708   127
    H     -6.255    4.132    -6.941   127
    H     -4.806    4.137    -7.694   127
    O     1.852    5.922    -3.635   128
    H     1.113    5.297    -3.387   128
    H     2.305    5.592    -4.463   128
    O     -4.456    1.801    2.702   129
    H     -3.992    2.104    1.869   129
    H     -4.646    0.821    2.64   129
    O     0.803    -5.527    -3.252   130
    H     1.341    -5.551    -2.41   130
    H     0.008    -4.934    -3.122   130
    O     -7.448    4.67    3.716   131
    H     -7.111    4.066    2.994   131
    H     -8.163    5.263    3.348   131
    O     3.596    -0.597    8.511   132
    H     3.424    -0.862    9.46   132
    H     4.549    -0.31    8.412   132
    O     8.133    5.499    3.039   133
    H     8.974    5.936    2.719   133
    H     7.642    5.11    2.26   133
```

(continues on next page)

(continued from previous page)

```
    O     -1.186    3.025    6.358   134
    H     -1.453    3.989    6.371   134
    H     -1.716    2.523    7.041   134
    O     1.391    6.827    5.842   135
    H     1.717    6.033    5.329   135
    H     1.576    7.659    5.32   135
    O     8.48    0.178    3.175   136
    H     8.689    -0.233    4.062   136
    H     8.429    1.172    3.272   136
    O     5.491    -2.645    -8.951   137
    H     6.099    -3.205    -8.389   137
    H     5.372    -3.076    -9.846   137
    O     5.858    -6.18    -7.286   138
    H     5.479    -6.48    -8.161   138
    H     5.775    -6.917    -6.615   138
    O     4.819    -9.959    3.19   139
    H     5.303    -10.736    2.788   139
    H     4.151    -9.607    2.534   139
    O     -10.22    -2.106    -2.896   140
    H     -11.165    -1.943    -3.179   140
    H     -9.63    -2.132    -3.703   140
    O     9.523    0.622    5.574   141
    H     10.389    0.614    6.075   141
    H     8.774    0.824    6.205   141
    O     3.768    2.327    -2.459   142
    H     2.96    2.318    -3.048   142
    H     4.573    2.567    -3.002   142
    O     9.057    -3.727    5.016   143
    H     8.826    -3.519    5.967   143
    H     9.288    -4.696    4.934   143
    O     -3.865    -9.077    -2.573   144
    H     -3.642    -9.53    -1.71   144
    H     -3.999    -8.099    -2.412   144
    O     7.735    6.089    -1.411   145
    H     7.706    6.786    -0.695   145
    H     6.871    6.092    -1.914   145
    O     2.451    -7.59    -6.329   146
    H     2.186    -7.189    -7.206   146
    H     3.252    -8.175    -6.459   146
    O     10.049    4.728    0.155   147
    H     9.866    5.617    0.576   147
    H     10.223    4.852    -0.822   147
    O     -6.754    6.196    5.779   148
    H     -7.052    5.609    5.027   148
    H     -7.543    6.682    6.156   148
    O     0.795    -3.52    -9.602   149
    H     0.443    -4.455    -9.558   149
    H     1.511    -3.401    -8.915   149
    O     -7.039    1.511    -3.64   150
    H     -6.551    1.817    -2.823   150
    H     -8.024    1.548    -3.473   150
    O     -3.164    -9.103    6.758   151
    H     -3.773    -8.477    7.245   151
    H     -2.413    -9.373    7.36   151
    O     -5.629    1.786    -5.896   152
    H     -4.658    1.779    -5.658   152
    H     -6.169    1.508    -5.102   152
```

(continues on next page)

(continued from previous page)

```
    O     7.288    2.496    7.375   153
    H     7.76    3.141    7.975   153
    H     7.083    2.941    6.503   153
    O     -0.563    -1.687    -4.735   154
    H     0.383    -1.395    -4.876   154
    H     -1.178    -1.082    -5.241   154
    O     0.097    5.872    -7.772   155
    H     -0.007    4.886    -7.9   155
    H     -0.289    6.353    -8.559   155
    O     0.263    3.329    2.77   156
    H     -0.195    2.506    3.107   156
    H     -0.167    3.625    1.917   156
    O     -5.112    -3.742    3.386   157
    H     -6.023    -4.152    3.352   157
    H     -4.724    -3.716    2.465   157
    O     5.173    0.414    4.591   158
    H     5.914    0.938    4.169   158
    H     5.536    -0.445    4.951   158
    O     -0.54    9.346    4.364   159
    H     -0.519    8.387    4.082   159
    H     -1.136    9.862    3.749   159
    O     -7.739    -2.707    -1.775   160
    H     -7.119    -2.238    -2.405   160
    H     -8.675    -2.642    -2.119   160
    O     4.069    -1.93    0.595   161
    H     4.054    -2.91    0.792   161
    H     4.477    -1.777    -0.306   161
    O     7.679    -4.202    -7.817   162
    H     8.547    -4.383    -7.354   162
    H     6.957    -4.751    -7.397   162
    O     -4.112    -3.518    -7.265   163
    H     -3.62    -3.431    -8.131   163
    H     -3.457    -3.538    -6.51   163
    O     -4.079    2.718    8.002   164
    H     -4.036    1.739    7.799   164
    H     -5.026    2.979    8.185   164
    O     -0.788    -5.992    -8.417   165
    H     -0.874    -5.596    -7.502   165
    H     -1.656    -6.413    -8.68   165
    O     -0.642    -8.0    -2.36   166
    H     -1.093    -7.251    -1.876   166
    H     -0.617    -7.798    -3.339   166
    O     -5.431    -4.01    -2.659   167
    H     -6.327    -3.586    -2.795   167
    H     -4.768    -3.311    -2.392   167
    O     1.622    -4.606    7.267   168
    H     0.644    -4.448    7.405   168
    H     2.063    -3.751    6.996   168
    O     -9.426    -1.829    3.261   169
    H     -8.578    -1.754    3.786   169
    H     -9.786    -2.759    3.34   169
    O     10.397    -2.728    0.938   170
    H     9.987    -2.336    0.114   170
    H     11.372    -2.508    0.962   170
    O     -3.49    -10.263    -4.969   171
    H     -2.767    -9.841    -5.517   171
    H     -3.819    -9.607    -4.289   171
```

(continues on next page)

(continued from previous page)

```
    O     -1.521    -7.2    2.797   172
    H     -1.357    -6.413    3.392   172
    H     -0.828    -7.899    2.97   172
    O     2.563    -4.792    -5.31   173
    H     1.916    -4.727    -4.55   173
    H     2.546    -5.719    -5.685   173
    O     2.837    -2.181    6.366   174
    H     2.273    -1.594    5.785   174
    H     3.201    -1.646    7.128   174
    O     -4.032    -6.368    -2.157   175
    H     -4.673    -5.671    -2.479   175
    H     -4.118    -6.467    -1.165   175
    O     -3.248    1.69    -4.498   176
    H     -2.674    2.482    -4.291   176
    H     -2.676    0.943    -4.835   176
    O     9.532    2.335    1.199   177
    H     9.808    3.202    0.786   177
    H     8.539    2.233    1.129   177
    O     4.296    6.036    3.312   178
    H     4.884    5.765    4.074   178
    H     3.344    6.057    3.617   178
    O     -4.003    0.618    -2.093   179
    H     -3.993    0.617    -3.093   179
    H     -4.888    0.953    -1.769   179
    O     6.312    5.31    5.167   180
    H     6.936    5.315    4.386   180
    H     6.368    6.187    5.644   180
    O     -6.267    -0.421    5.288   181
    H     -7.045    -1.049    5.313   181
    H     -6.538    0.461    5.671   181
    O     -4.077    0.335    6.986   182
    H     -4.824    -0.155    6.539   182
    H     -3.207    -0.105    6.762   182
    O     -1.592    10.18    -2.288   183
    H     -2.521    10.21    -2.657   183
    H     -1.428    10.996    -1.734   183
    O     -1.179    1.277    4.006   184
    H     -1.662    0.404    3.942   184
    H     -1.541    1.802    4.776   184
    O     -5.857    -7.312    2.548   185
    H     -5.09    -7.722    3.042   185
    H     -6.533    -8.017    2.337   185
    O     8.543    -2.902    -4.126   186
    H     8.959    -3.487    -4.823   186
    H     8.449    -1.973    -4.484   186
    O     -2.288    4.779    9.339   187
    H     -2.796    3.953    9.585   187
    H     -2.061    4.754    8.366   187
    O     -3.934    -3.208    5.916   188
    H     -3.188    -2.542    5.928   188
    H     -4.429    -3.137    5.05   188
    O     -0.872    -9.88    -6.056   189
    H     -0.065    -10.416    -5.807   189
    H     -0.715    -8.918    -5.833   189
    O     -2.461    4.695    -2.278   190
    H     -2.779    4.946    -3.192   190
    H     -1.851    3.906    -2.342   190
```

(continues on next page)

(continued from previous page)

```
    O     6.344    -0.092    7.977    191
    H     6.51    -0.546    7.101    191
    H     6.343    0.899    7.843    191
    O     -5.698    -0.661    2.606    192
    H     -5.841    -0.727    3.593    192
    H     -6.565    -0.816    2.132    192
    O     -4.029    -2.72    10.246    193
    H     -5.009    -2.685    10.442    193
    H     -3.786    -3.634    9.923    193
    O     2.832    -3.188    -7.657    194
    H     2.732    -3.704    -6.806    194
    H     3.768    -3.282    -7.996    194
    O     1.073    -10.115    -2.625    195
    H     0.46    -9.352    -2.419    195
    H     1.445    -10.004    -3.547    195
    O     -6.62    3.126    1.407    196
    H     -5.909    2.655    1.93    196
    H     -6.194    3.705    0.711    196
    O     -0.677    8.512    -5.231    197
    H     -0.896    9.254    -4.597    197
    H     0.11    8.001    -4.887    197
    O     4.12    4.53    0.653    198
    H     4.223    5.225    1.364    198
    H     3.485    3.824    0.967    198
    O     -8.221    -2.456    5.669    199
    H     -8.808    -2.293    6.463    199
    H     -7.852    -3.384    5.712    199
    O     2.898    4.605    -5.775    200
    H     3.521    4.999    -6.45    200
    H     2.968    3.608    -5.798    200
    O     -8.042    -4.503    0.126    201
    H     -7.787    -3.888    -0.62    201
    H     -9.001    -4.768    0.027    201
    O     0.142    -7.397    -5.046    202
    H     0.431    -6.604    -4.511    202
    H     0.919    -7.748    -5.568    202
    O     -5.157    -1.438    -5.812    203
    H     -5.788    -0.913    -6.384    203
    H     -4.908    -2.285    -6.281    203
    O     -1.04    -5.021    4.465    204
    H     -1.767    -5.433    5.015    204
    H     -0.158    -5.409    4.733    204
    O     6.958    4.266    0.911    205
    H     5.988    4.074    0.766    205
    H     7.395    4.46    0.033    205
    O     7.345    0.71    0.51    206
    H     7.532    0.27    1.389    206
    H     6.36    0.714    0.342    206
    O     2.18    -0.941    -4.851    207
    H     2.768    -1.736    -4.7    207
    H     2.46    -0.474    -5.689    207
    O     1.874    4.72    4.35    208
    H     1.127    4.286    3.847    208
    H     2.47    4.016    4.736    208
    O     -3.182    3.268    -7.397    209
    H     -2.609    2.52    -7.062    209
    H     -3.481    3.066    -8.33    209
```

(continues on next page)

(continued from previous page)

```
    O     -6.334    1.42    -0.932   210
    H     -7.057    0.73    -0.933   210
    H     -6.418    1.991    -0.115   210
    O     -11.485    0.121    1.494   211
    H     -10.502    -0.004    1.358   211
    H     -11.916    -0.768    1.646   211
    O     -4.428    6.45    7.562   212
    H     -5.324    6.26    7.16   212
    H     -4.448    6.234    8.538   212
    O     -7.198    8.279    2.233   213
    H     -6.522    8.005    2.917   213
    H     -7.442    9.239    2.372   213
    O     -0.931    10.111    0.639   214
    H     -0.955    9.372    -0.034   214
    H     -0.074    10.065    1.154   214
    O     6.652    -0.975    -5.911   215
    H     6.955    -1.129    -6.852   215
    H     6.039    -1.715    -5.634   215
    O     11.643    1.502    -0.57   216
    H     10.975    1.438    0.171   216
    H     11.176    1.745    -1.42   216
    O     -8.267    -2.766    -5.204   217
    H     -8.194    -1.919    -5.73   217
    H     -7.565    -3.41    -5.509   217
    O     4.001    -4.795    1.063   218
    H     4.818    -4.764    1.639   218
    H     3.835    -5.737    0.771   218
    O     -3.518    -0.073    -11.143   219
    H     -4.375    -0.493    -11.442   219
    H     -2.889    -0.009    -11.918   219
    O     5.278    8.33    -4.17   220
    H     5.812    8.57    -3.36   220
    H     5.813    8.525    -4.992   220
    O     -7.165    5.42    -3.118   221
    H     -6.784    4.835    -2.402   221
    H     -7.286    4.886    -3.955   221
    O     4.501    8.636    -0.932   222
    H     4.421    7.683    -1.224   222
    H     3.623    9.097    -1.055   222
    O     -1.648    -10.25    4.458   223
    H     -2.159    -10.03    3.627   223
    H     -2.211    -10.044    5.259   223
    O     -1.212    7.616    8.694   224
    H     -0.792    8.473    8.398   224
    H     -2.05    7.814    9.203   224
    O     -9.049    1.489    0.827   225
    H     -8.587    2.255    1.272   225
    H     -8.522    0.652    0.972   225
    SubEnd
  END
EndEngine
eor
```

Example: QM/FDE/FQ: excitations of acrolein in water

Three examples: a QM/FDE/FQ calculation with the frozen-layer response modeled with FQ, one without the response, and one with no explicit FQ layer.

Download FQQM_FDE_FDERESP.run

```
#!/bin/bash


# This example shows how to perform an excited-state calculation
# of an organic molecule in aqueous solution using the QM/FDE/FQ model
# In this example the response of the FDE layer is also included
# by means of the FQ method

# The calculation starts with the solute embedded within an FQ environment

AMS_JOBNAME="acrolein" "$AMSBIN/ams" <<EOF
System
  Atoms
    C    26.163000   27.087000   27.854000
    O    25.922000   26.598000   28.918000
    H    25.425000   27.661000   27.251000
    C    27.541000   26.980000   27.309000
    C    27.781000   27.369000   26.027000
    H    28.266000   26.451000   27.925000
    H    28.823000   27.287000   25.727000
    H    27.066000   27.797000   25.330000
  End
End

Task SinglePoint

Engine ADF
  Title Acrolein PBE/DZ single point
  Symmetrytolerance 1e-2
  Relativity
    Level None
  End
  Basis
    Type DZ
    Core None
  End
  XC
    GGA PBE
  End
  ExactDensity
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit

  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00
```

(continues on next page)

(continued from previous page)

```
   AtomType
     Symbol O
     Chi 0.189194
     Eta 0.523700
   SubEnd
   AtomType
     Symbol H
     Chi 0.012767
     Eta 0.537512
   SubEnd
   Coords
     O    28.826000   25.183000   30.762000   1
     H    28.995000   26.089000   31.149000   1
     H    28.144000   24.706000   31.315000   1
     O    27.763000   31.250000   26.749000   2
     H    27.469000   30.950000   25.842000   2
     H    26.974000   31.279000   27.363000   2
     O    24.563000   24.510000   25.661000   3
     H    25.293000   24.179000   25.064000   3
     H    23.710000   24.040000   25.436000   3
     O    24.219000   23.640000   28.272000   4
     H    23.531000   22.942000   28.076000   4
     H    24.474000   24.103000   27.423000   4
     O    22.979000   24.696000   30.549000   5
     H    23.726000   24.648000   29.886000   5
     H    23.222000   24.170000   31.364000   5
     O    22.709000   27.494000   30.324000   6
     H    23.077000   26.584000   30.512000   6
     H    22.782000   27.689000   29.346000   6
     O    29.009000   27.562000   32.091000   7
     H    28.267000   27.944000   31.541000   7
     H    29.871000   28.003000   31.840000   7
     O    26.815000   24.369000   24.266000   8
     H    27.032000   25.081000   23.598000   8
     H    27.472000   24.407000   25.018000   8
     O    26.727000   23.089000   28.975000   9
     H    26.980000   23.432000   29.879000   9
     H    25.761000   23.284000   28.803000   9
     O    25.883000   31.425000   28.603000   10
     H    25.156000   31.260000   29.269000   10
     H    26.624000   31.936000   29.038000   10
     O    29.317000   29.573000   27.978000   11
     H    30.130000   29.270000   27.481000   11
     H    28.782000   30.188000   27.398000   11
     O    24.110000   29.703000   24.837000   12
     H    23.685000   28.874000   25.201000   12
     H    23.551000   30.064000   24.091000   12
     O    26.708000   30.327000   24.312000   13
     H    25.716000   30.396000   24.419000   13
     H    26.920000   29.651000   23.606000   13
     O    30.565000   24.816000   28.707000   14
     H    30.237000   24.502000   27.816000   14
     H    29.793000   24.898000   29.337000   14
     O    28.442000   23.780000   26.550000   15
     H    27.844000   23.801000   27.351000   15
     H    28.454000   22.857000   26.166000   15
   SubEnd
```

(continues on next page)

(continued from previous page)

```
  End
EndEngine
EOF

mv acrolein.results/adf.rkf t21.acrolein

# The QM water layer is then simulated

#H2O B3LYP/DZP
AMS_JOBNAME=H2O "$AMSBIN/ams" <<EOF
System
  Atoms
    O    23.219000   27.243000   25.752000
    H    22.533000   27.083000   25.042000
    H    23.738000   26.403000   25.909000
    O    27.812000   28.937000   30.049000
    H    28.442000   28.831000   29.280000
    H    26.915000   28.573000   29.797000
    O    23.304000   28.960000   27.797000
    H    23.864000   29.646000   27.332000
    H    23.228000   28.146000   27.220000
  End
End

Task SinglePoint

Engine ADF
  Title H2O PBE/DZ single point
  Symmetrytolerance 1e-2
  Relativity
    Level None
  End
  Basis
    Type DZ
    Core None
  End
  XC
    GGA PBE
  End
  ExactDensity
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit

  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00
    AtomType
      Symbol O
      Chi 0.189194
      Eta 0.523700
    SubEnd
    AtomType
      Symbol H
```

(continues on next page)

(continued from previous page)

```
      Chi 0.012767
      Eta 0.537512
    SubEnd
    Coords
      O    28.826000   25.183000   30.762000   1
      H    28.995000   26.089000   31.149000   1
      H    28.144000   24.706000   31.315000   1
      O    27.763000   31.250000   26.749000   2
      H    27.469000   30.950000   25.842000   2
      H    26.974000   31.279000   27.363000   2
      O    24.563000   24.510000   25.661000   3
      H    25.293000   24.179000   25.064000   3
      H    23.710000   24.040000   25.436000   3
      O    24.219000   23.640000   28.272000   4
      H    23.531000   22.942000   28.076000   4
      H    24.474000   24.103000   27.423000   4
      O    22.979000   24.696000   30.549000   5
      H    23.726000   24.648000   29.886000   5
      H    23.222000   24.170000   31.364000   5
      O    22.709000   27.494000   30.324000   6
      H    23.077000   26.584000   30.512000   6
      H    22.782000   27.689000   29.346000   6
      O    29.009000   27.562000   32.091000   7
      H    28.267000   27.944000   31.541000   7
      H    29.871000   28.003000   31.840000   7
      O    26.815000   24.369000   24.266000   8
      H    27.032000   25.081000   23.598000   8
      H    27.472000   24.407000   25.018000   8
      O    26.727000   23.089000   28.975000   9
      H    26.980000   23.432000   29.879000   9
      H    25.761000   23.284000   28.803000   9
      O    25.883000   31.425000   28.603000   10
      H    25.156000   31.260000   29.269000   10
      H    26.624000   31.936000   29.038000   10
      O    29.317000   29.573000   27.978000   11
      H    30.130000   29.270000   27.481000   11
      H    28.782000   30.188000   27.398000   11
      O    24.110000   29.703000   24.837000   12
      H    23.685000   28.874000   25.201000   12
      H    23.551000   30.064000   24.091000   12
      O    26.708000   30.327000   24.312000   13
      H    25.716000   30.396000   24.419000   13
      H    26.920000   29.651000   23.606000   13
      O    30.565000   24.816000   28.707000   14
      H    30.237000   24.502000   27.816000   14
      H    29.793000   24.898000   29.337000   14
      O    28.442000   23.780000   26.550000   15
      H    27.844000   23.801000   27.351000   15
      H    28.454000   22.857000   26.166000   15
    SubEnd
  End
EndEngine
EOF

mv H2O.results/adf.rkf t21.H2O

# Finally, the QM/FDE/FQ calculation is performed
```

(continues on next page)

(continued from previous page)

```
# The response of the FDE layer is activated with the FDERESP keyword

AMS_JOBNAME=FDE "$AMSBIN/ams" <<EOF
System
 Atoms
    C    26.163000   27.087000   27.854000   adf.f=frag1
    O    25.922000   26.598000   28.918000   adf.f=frag1
    H    25.425000   27.661000   27.251000   adf.f=frag1
    C    27.541000   26.980000   27.309000   adf.f=frag1
    C    27.781000   27.369000   26.027000   adf.f=frag1
    H    28.266000   26.451000   27.925000   adf.f=frag1
    H    28.823000   27.287000   25.727000   adf.f=frag1
    H    27.066000   27.797000   25.330000   adf.f=frag1
    O    23.219000   27.243000   25.752000   adf.f=frag2 imol=1
    H    22.533000   27.083000   25.042000   adf.f=frag2 imol=1
    H    23.738000   26.403000   25.909000   adf.f=frag2 imol=1
    O    27.812000   28.937000   30.049000   adf.f=frag2 imol=2
    H    28.442000   28.831000   29.280000   adf.f=frag2 imol=2
    H    26.915000   28.573000   29.797000   adf.f=frag2 imol=2
    O    23.304000   28.960000   27.797000   adf.f=frag2 imol=3
    H    23.864000   29.646000   27.332000   adf.f=frag2 imol=3
    H    23.228000   28.146000   27.220000   adf.f=frag2 imol=3
  End
End

Task SinglePoint

Engine ADF
  Title Acrolein-H2O PBE/PW91K/DZ  FDE single point with interaction energy
  SymmetryTolerance 1e-2
  STOFit
  Relativity
    Level None
  End
  XC
    GGA PBE
  End
  ExactDensity
  FDE
    PW91K
    GGAPOTXFD PBEx
    GGAPOTCFD PBEc
    FULLGRID
    ENERGY
  End
  Fragments
   frag1  t21.acrolein
   frag2  t21.H2O    type=FDE
  End
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit
  Excitations
    LOWEST 4
```

(continues on next page)

(continued from previous page)

```
  ONLYSING True
End

QMFQ
  FORCEFIELD FQ
  MolCharge 0.00
  FDERESP
  AtomType
    Symbol O
    Chi 0.189194
    Eta 0.523700
  SubEnd
  AtomType
    Symbol H
    Chi 0.012767
    Eta 0.537512
  SubEnd
  Coords
    O    28.826000   25.183000   30.762000   1
    H    28.995000   26.089000   31.149000   1
    H    28.144000   24.706000   31.315000   1
    O    27.763000   31.250000   26.749000   2
    H    27.469000   30.950000   25.842000   2
    H    26.974000   31.279000   27.363000   2
    O    24.563000   24.510000   25.661000   3
    H    25.293000   24.179000   25.064000   3
    H    23.710000   24.040000   25.436000   3
    O    24.219000   23.640000   28.272000   4
    H    23.531000   22.942000   28.076000   4
    H    24.474000   24.103000   27.423000   4
    O    22.979000   24.696000   30.549000   5
    H    23.726000   24.648000   29.886000   5
    H    23.222000   24.170000   31.364000   5
    O    22.709000   27.494000   30.324000   6
    H    23.077000   26.584000   30.512000   6
    H    22.782000   27.689000   29.346000   6
    O    29.009000   27.562000   32.091000   7
    H    28.267000   27.944000   31.541000   7
    H    29.871000   28.003000   31.840000   7
    O    26.815000   24.369000   24.266000   8
    H    27.032000   25.081000   23.598000   8
    H    27.472000   24.407000   25.018000   8
    O    26.727000   23.089000   28.975000   9
    H    26.980000   23.432000   29.879000   9
    H    25.761000   23.284000   28.803000   9
    O    25.883000   31.425000   28.603000   10
    H    25.156000   31.260000   29.269000   10
    H    26.624000   31.936000   29.038000   10
    O    29.317000   29.573000   27.978000   11
    H    30.130000   29.270000   27.481000   11
    H    28.782000   30.188000   27.398000   11
    O    24.110000   29.703000   24.837000   12
    H    23.685000   28.874000   25.201000   12
    H    23.551000   30.064000   24.091000   12
    O    26.708000   30.327000   24.312000   13
    H    25.716000   30.396000   24.419000   13
    H    26.920000   29.651000   23.606000   13
```

(continues on next page)

(continued from previous page)

```
      O    30.565000   24.816000   28.707000   14
      H    30.237000   24.502000   27.816000   14
      H    29.793000   24.898000   29.337000   14
      O    28.442000   23.780000   26.550000   15
      H    27.844000   23.801000   27.351000   15
      H    28.454000   22.857000   26.166000   15
    SubEnd
  End
EndEngine
EOF
```

Download FQQM_FDE_noFDERESP.run

```
#!/bin/bash


# This example shows how to perform an excited-state calculation
# of an organic molecule in aqueous solution using the QM/FDE/FQ model
# In this example the response of the FDE layer is not included

# The calculation starts with the solute embedded within an FQ environment

AMS_JOBNAME="acrolein" "$AMSBIN/ams" <<EOF
System
  Atoms
    C    26.163000   27.087000   27.854000
    O    25.922000   26.598000   28.918000
    H    25.425000   27.661000   27.251000
    C    27.541000   26.980000   27.309000
    C    27.781000   27.369000   26.027000
    H    28.266000   26.451000   27.925000
    H    28.823000   27.287000   25.727000
    H    27.066000   27.797000   25.330000
  End
End

Task SinglePoint

Engine ADF
  Title Acrolein PBE/DZ single point
  Symmetrytolerance 1e-2
  Relativity
    Level None
  End
  Basis
    Type DZ
    Core None
  End
  XC
    GGA PBE
  End
  ExactDensity
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
```

(continues on next page)

(continued from previous page)

```
NoSymFit

QMFQ
  FORCEFIELD FQ
  MolCharge 0.00

  AtomType
    Symbol O
    Chi 0.189194
    Eta 0.523700
  SubEnd
  AtomType
    Symbol H
    Chi 0.012767
    Eta 0.537512
  SubEnd
  Coords
    O    28.826000   25.183000   30.762000   1
    H    28.995000   26.089000   31.149000   1
    H    28.144000   24.706000   31.315000   1
    O    27.763000   31.250000   26.749000   2
    H    27.469000   30.950000   25.842000   2
    H    26.974000   31.279000   27.363000   2
    O    24.563000   24.510000   25.661000   3
    H    25.293000   24.179000   25.064000   3
    H    23.710000   24.040000   25.436000   3
    O    24.219000   23.640000   28.272000   4
    H    23.531000   22.942000   28.076000   4
    H    24.474000   24.103000   27.423000   4
    O    22.979000   24.696000   30.549000   5
    H    23.726000   24.648000   29.886000   5
    H    23.222000   24.170000   31.364000   5
    O    22.709000   27.494000   30.324000   6
    H    23.077000   26.584000   30.512000   6
    H    22.782000   27.689000   29.346000   6
    O    29.009000   27.562000   32.091000   7
    H    28.267000   27.944000   31.541000   7
    H    29.871000   28.003000   31.840000   7
    O    26.815000   24.369000   24.266000   8
    H    27.032000   25.081000   23.598000   8
    H    27.472000   24.407000   25.018000   8
    O    26.727000   23.089000   28.975000   9
    H    26.980000   23.432000   29.879000   9
    H    25.761000   23.284000   28.803000   9
    O    25.883000   31.425000   28.603000   10
    H    25.156000   31.260000   29.269000   10
    H    26.624000   31.936000   29.038000   10
    O    29.317000   29.573000   27.978000   11
    H    30.130000   29.270000   27.481000   11
    H    28.782000   30.188000   27.398000   11
    O    24.110000   29.703000   24.837000   12
    H    23.685000   28.874000   25.201000   12
    H    23.551000   30.064000   24.091000   12
    O    26.708000   30.327000   24.312000   13
    H    25.716000   30.396000   24.419000   13
    H    26.920000   29.651000   23.606000   13
    O    30.565000   24.816000   28.707000   14
```

(continues on next page)

(continued from previous page)

```
      H    30.237000   24.502000   27.816000   14
      H    29.793000   24.898000   29.337000   14
      O    28.442000   23.780000   26.550000   15
      H    27.844000   23.801000   27.351000   15
      H    28.454000   22.857000   26.166000   15
    SubEnd
  End
EndEngine
EOF

mv acrolein.results/adf.rkf t21.acrolein

# The QM water layer is then simulated

#H2O B3LYP/DZP
AMS_JOBNAME=H2O "$AMSBIN/ams" <<EOF
System
  Atoms
    O    23.219000   27.243000   25.752000
    H    22.533000   27.083000   25.042000
    H    23.738000   26.403000   25.909000
    O    27.812000   28.937000   30.049000
    H    28.442000   28.831000   29.280000
    H    26.915000   28.573000   29.797000
    O    23.304000   28.960000   27.797000
    H    23.864000   29.646000   27.332000
    H    23.228000   28.146000   27.220000
  End
End

Task SinglePoint

Engine ADF
  Title H2O PBE/DZ single point
  Symmetrytolerance 1e-2
  Relativity
    Level None
  End
  Basis
    Type DZ
    Core None
  End
  XC
    GGA PBE
  End
  ExactDensity
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit

  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00
    AtomType
```

(continues on next page)

(continued from previous page)

```
      Symbol O
      Chi 0.189194
      Eta 0.523700
    SubEnd
    AtomType
      Symbol H
      Chi 0.012767
      Eta 0.537512
    SubEnd
    Coords
      O    28.826000   25.183000   30.762000   1
      H    28.995000   26.089000   31.149000   1
      H    28.144000   24.706000   31.315000   1
      O    27.763000   31.250000   26.749000   2
      H    27.469000   30.950000   25.842000   2
      H    26.974000   31.279000   27.363000   2
      O    24.563000   24.510000   25.661000   3
      H    25.293000   24.179000   25.064000   3
      H    23.710000   24.040000   25.436000   3
      O    24.219000   23.640000   28.272000   4
      H    23.531000   22.942000   28.076000   4
      H    24.474000   24.103000   27.423000   4
      O    22.979000   24.696000   30.549000   5
      H    23.726000   24.648000   29.886000   5
      H    23.222000   24.170000   31.364000   5
      O    22.709000   27.494000   30.324000   6
      H    23.077000   26.584000   30.512000   6
      H    22.782000   27.689000   29.346000   6
      O    29.009000   27.562000   32.091000   7
      H    28.267000   27.944000   31.541000   7
      H    29.871000   28.003000   31.840000   7
      O    26.815000   24.369000   24.266000   8
      H    27.032000   25.081000   23.598000   8
      H    27.472000   24.407000   25.018000   8
      O    26.727000   23.089000   28.975000   9
      H    26.980000   23.432000   29.879000   9
      H    25.761000   23.284000   28.803000   9
      O    25.883000   31.425000   28.603000   10
      H    25.156000   31.260000   29.269000   10
      H    26.624000   31.936000   29.038000   10
      O    29.317000   29.573000   27.978000   11
      H    30.130000   29.270000   27.481000   11
      H    28.782000   30.188000   27.398000   11
      O    24.110000   29.703000   24.837000   12
      H    23.685000   28.874000   25.201000   12
      H    23.551000   30.064000   24.091000   12
      O    26.708000   30.327000   24.312000   13
      H    25.716000   30.396000   24.419000   13
      H    26.920000   29.651000   23.606000   13
      O    30.565000   24.816000   28.707000   14
      H    30.237000   24.502000   27.816000   14
      H    29.793000   24.898000   29.337000   14
      O    28.442000   23.780000   26.550000   15
      H    27.844000   23.801000   27.351000   15
      H    28.454000   22.857000   26.166000   15
    SubEnd
  End
```

(continues on next page)

(continued from previous page)

```
EndEngine
EOF

mv H2O.results/adf.rkf t21.H2O

# Finally, the QM/FDE/FQ calculation is performed

AMS_JOBNAME=FDE "$AMSBIN/ams" <<EOF
System
 Atoms
    C    26.163000   27.087000   27.854000   adf.f=frag1
    O    25.922000   26.598000   28.918000   adf.f=frag1
    H    25.425000   27.661000   27.251000   adf.f=frag1
    C    27.541000   26.980000   27.309000   adf.f=frag1
    C    27.781000   27.369000   26.027000   adf.f=frag1
    H    28.266000   26.451000   27.925000   adf.f=frag1
    H    28.823000   27.287000   25.727000   adf.f=frag1
    H    27.066000   27.797000   25.330000   adf.f=frag1
    O    23.219000   27.243000   25.752000   adf.f=frag2 imol=1
    H    22.533000   27.083000   25.042000   adf.f=frag2 imol=1
    H    23.738000   26.403000   25.909000   adf.f=frag2 imol=1
    O    27.812000   28.937000   30.049000   adf.f=frag2 imol=2
    H    28.442000   28.831000   29.280000   adf.f=frag2 imol=2
    H    26.915000   28.573000   29.797000   adf.f=frag2 imol=2
    O    23.304000   28.960000   27.797000   adf.f=frag2 imol=3
    H    23.864000   29.646000   27.332000   adf.f=frag2 imol=3
    H    23.228000   28.146000   27.220000   adf.f=frag2 imol=3
  End
End

Task SinglePoint

Engine ADF
  Title Acrolein-H2O PBE/PW91K/DZ  FDE single point with interaction energy
  SymmetryTolerance 1e-2
  STOFit
  Relativity
    Level None
  End
  XC
    GGA PBE
  End
  ExactDensity
  FDE
    PW91K
    GGAPOTXFD PBEx
    GGAPOTCFD PBEc
    FULLGRID
    ENERGY
  End
  Fragments
   frag1  t21.acrolein
   frag2  t21.H2O    type=FDE
  End
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
```

(continues on next page)

(continued from previous page)

```
End
NoPrint BAS FUNCTIONS
NoSymFit
Excitations
  LOWEST 4
  ONLYSING True
End

QMFQ
  FORCEFIELD FQ
  MolCharge 0.00
  AtomType
    Symbol O
    Chi 0.189194
    Eta 0.523700
  SubEnd
  AtomType
    Symbol H
    Chi 0.012767
    Eta 0.537512
  SubEnd
  Coords
    O    28.826000   25.183000   30.762000   1
    H    28.995000   26.089000   31.149000   1
    H    28.144000   24.706000   31.315000   1
    O    27.763000   31.250000   26.749000   2
    H    27.469000   30.950000   25.842000   2
    H    26.974000   31.279000   27.363000   2
    O    24.563000   24.510000   25.661000   3
    H    25.293000   24.179000   25.064000   3
    H    23.710000   24.040000   25.436000   3
    O    24.219000   23.640000   28.272000   4
    H    23.531000   22.942000   28.076000   4
    H    24.474000   24.103000   27.423000   4
    O    22.979000   24.696000   30.549000   5
    H    23.726000   24.648000   29.886000   5
    H    23.222000   24.170000   31.364000   5
    O    22.709000   27.494000   30.324000   6
    H    23.077000   26.584000   30.512000   6
    H    22.782000   27.689000   29.346000   6
    O    29.009000   27.562000   32.091000   7
    H    28.267000   27.944000   31.541000   7
    H    29.871000   28.003000   31.840000   7
    O    26.815000   24.369000   24.266000   8
    H    27.032000   25.081000   23.598000   8
    H    27.472000   24.407000   25.018000   8
    O    26.727000   23.089000   28.975000   9
    H    26.980000   23.432000   29.879000   9
    H    25.761000   23.284000   28.803000   9
    O    25.883000   31.425000   28.603000   10
    H    25.156000   31.260000   29.269000   10
    H    26.624000   31.936000   29.038000   10
    O    29.317000   29.573000   27.978000   11
    H    30.130000   29.270000   27.481000   11
    H    28.782000   30.188000   27.398000   11
    O    24.110000   29.703000   24.837000   12
    H    23.685000   28.874000   25.201000   12
```

(continues on next page)

(continued from previous page)

```
      H    23.551000   30.064000   24.091000   12
      O    26.708000   30.327000   24.312000   13
      H    25.716000   30.396000   24.419000   13
      H    26.920000   29.651000   23.606000   13
      O    30.565000   24.816000   28.707000   14
      H    30.237000   24.502000   27.816000   14
      H    29.793000   24.898000   29.337000   14
      O    28.442000   23.780000   26.550000   15
      H    27.844000   23.801000   27.351000   15
      H    28.454000   22.857000   26.166000   15
    SubEnd
  End
EndEngine
EOF
```

Download FQQM_FDE_noFQ.run

```
#!/bin/bash

# In this example, the FDE method is used to simulate the absorption
# spectrum of acrolein in water.
# Normally, the FDE method neglects to include the response of the frozen
# layer, however this assumption can lead to large systematic errors
# One way to include the response of the environment in a very cost-efficient
# way is to resort to the FQ method, as shown below

AMS_JOBNAME="acrolein" "$AMSBIN/ams" <<EOF
System
  Atoms
    C    26.163000   27.087000   27.854000
    O    25.922000   26.598000   28.918000
    H    25.425000   27.661000   27.251000
    C    27.541000   26.980000   27.309000
    C    27.781000   27.369000   26.027000
    H    28.266000   26.451000   27.925000
    H    28.823000   27.287000   25.727000
    H    27.066000   27.797000   25.330000
  End
End

Task SinglePoint

Engine ADF
  Title Acrolein PBE/DZ single point
  Symmetrytolerance 1e-2
  Relativity
    Level None
  End
  Basis
    Type DZ
    Core None
  End
  XC
    GGA PBE
  End
  ExactDensity
  EPrint
```

(continues on next page)

(continued from previous page)

```
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit
EndEngine
EOF

mv acrolein.results/adf.rkf t21.acrolein

#H2O B3LYP/DZP
AMS_JOBNAME=H2O "$AMSBIN/ams" <<EOF
System
  Atoms
    O    23.219000   27.243000   25.752000
    H    22.533000   27.083000   25.042000
    H    23.738000   26.403000   25.909000
    O    27.812000   28.937000   30.049000
    H    28.442000   28.831000   29.280000
    H    26.915000   28.573000   29.797000
    O    23.304000   28.960000   27.797000
    H    23.864000   29.646000   27.332000
    H    23.228000   28.146000   27.220000
  End
End

Task SinglePoint

Engine ADF
  Title H2O PBE/DZ single point
  Symmetrytolerance 1e-2
  Relativity
    Level None
  End
  Basis
    Type DZ
    Core None
  End
  XC
    GGA PBE
  End
  ExactDensity
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit

EndEngine
EOF

mv H2O.results/adf.rkf t21.H2O

# Having converged the two QM layers the FDE calculation can be carried out
# The response of the frozen layer is modelled using the FQ method, note the use
# of the FDERESP keyword in the QMFQ block
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=FDE "$AMSBIN/ams" <<EOF
System
 Atoms
    C    26.163000   27.087000   27.854000   adf.f=frag1
    O    25.922000   26.598000   28.918000   adf.f=frag1
    H    25.425000   27.661000   27.251000   adf.f=frag1
    C    27.541000   26.980000   27.309000   adf.f=frag1
    C    27.781000   27.369000   26.027000   adf.f=frag1
    H    28.266000   26.451000   27.925000   adf.f=frag1
    H    28.823000   27.287000   25.727000   adf.f=frag1
    H    27.066000   27.797000   25.330000   adf.f=frag1
    O    23.219000   27.243000   25.752000   adf.f=frag2 imol=1
    H    22.533000   27.083000   25.042000   adf.f=frag2 imol=1
    H    23.738000   26.403000   25.909000   adf.f=frag2 imol=1
    O    27.812000   28.937000   30.049000   adf.f=frag2 imol=2
    H    28.442000   28.831000   29.280000   adf.f=frag2 imol=2
    H    26.915000   28.573000   29.797000   adf.f=frag2 imol=2
    O    23.304000   28.960000   27.797000   adf.f=frag2 imol=3
    H    23.864000   29.646000   27.332000   adf.f=frag2 imol=3
    H    23.228000   28.146000   27.220000   adf.f=frag2 imol=3
  End
End

Task SinglePoint

Engine ADF
  Title Acrolein-H2O PBE/PW91K/DZ  FDE single point with interaction energy
  SymmetryTolerance 1e-2
  STOFit
  Relativity
    Level None
  End
  XC
    GGA PBE
  End
  ExactDensity
  FDE
    PW91K
    GGAPOTXFD PBEx
    GGAPOTCFD PBEc
    FULLGRID
    ENERGY
  End
  Fragments
   frag1  t21.acrolein
   frag2  t21.H2O    type=FDE
  End
  EPrint
    SFO NOEIG NOOVL NOORBPOP
    SCF NOPOP
  End
  NoPrint BAS FUNCTIONS
  NoSymFit
  Excitations
    LOWEST 4
    ONLYSING True
```

(continues on next page)

(continued from previous page)

```
  End

  QMFQ
    FORCEFIELD FQ
    MolCharge 0.00
    FDERESP
    AtomType
      Symbol O
      Chi 0.189194
      Eta 0.523700
    SubEnd
    AtomType
      Symbol H
      Chi 0.012767
      Eta 0.537512
    SubEnd
  End
EndEngine
EOF
```

14.3 Structure and Reactivity

14.3.1 Geometry Optimizations

Example: Geometry Optimization: H2O

Download GO_H2O.run

```
#!/bin/sh

"$AMSBIN/ams" <<eor

Task GeometryOptimization

System
    Atoms
        O 0.0 0.0 0.0
        H 1.0 0.0 0.0
        H 0.0 0.0 1.0
    End
End

Properties
   NormalModes Yes
End

Engine ADF
    Symmetry NoSym
    basis
        Type DZP
        CreateOutput Yes
    End
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor
```

Example: Restraint Geometry Optimization: H2O

Download GO_restraint.run

```
#! /bin/sh

# The restraint does not have to be satisfied at the start of the geometry
# optimization. An extra force is added to restrain the bond length, angle, or
# dihedral angle to a certain value.

#  == angle restraint ==

AMS_JOBNAME=Angle $AMSBIN/ams <<eor
System
  atoms
     O          0.001356    0.000999    0.000000
     H          0.994442   -0.037855    0.000000
     H         -0.298554    0.948531    0.000000
  end
  Symmetrize Yes
end

Task GeometryOptimization

Restraints
  Angle 3 1 2 125.0 10.0
End

Engine ADF
  title WATER geometry optimization with angle restraint
  basis
    type DZP
  end
EndEngine
eor

echo "Angle after optimization:"
$AMSBIN/amsreport Angle.results/adf.rkf -r angle#3#1#2


# == bond length restraint ==


AMS_JOBNAME=Bond $AMSBIN/ams <<eor
System
  atoms
     O          0.001356    0.000999    0.000000
     H          0.994442   -0.037855    0.000000
     H         -0.298554    0.948531    0.000000
  end
  Symmetrize Yes
end
```

(continues on next page)

(continued from previous page)

```
Task GeometryOptimization

Restraints
   Distance  1 2 1.03 10.0
   Distance  1 3 1.03 10.0
end

Engine ADF
  title WATER Geometry Optimization with bond length restraint
  basis
    type DZP
  end
EndEngine
eor

echo "Bond-dinstances after optimization:"
$AMSBIN/amsreport Bond.results/adf.rkf -r distance#1#2
$AMSBIN/amsreport Bond.results/adf.rkf -r distance#1#3


# == Restraining dihedral ==


AMS_JOBNAME=Dihedral $AMSBIN/ams <<eor
System
  atoms
     C        -0.004115   -0.000021    0.000023
     C         1.535711    0.000022    0.000008
     H        -0.399693    1.027812   -0.000082
     H        -0.399745   -0.513934    0.890139
     H        -0.399612   -0.513952   -0.890156
     H         1.931188    0.514066    0.890140
     H         1.931432    0.513819   -0.890121
     H         1.931281   -1.027824    0.000244
  end
end

Task GeometryOptimization

Restraints
   Dihedral 6 2 1 3 20.00 10.0
end

Engine ADF
  title Restraining dihedral of ethane
  basis
    type DZP
  end
  Symmetry NoSym
EndEngine
eor

echo "Dihedral after optimization:"
$AMSBIN/amsreport Dihedral.results/adf.rkf -r dihedral#6#2#1#3
```

Example: Constraint Geometry Optimization: H2O

Download GO_constraints.run

```
#! /bin/sh

# The key Constraints can only be used to enforce constraint in a geometry
# optimization. The Constraints do not have to be satisfied at the start
# of the geometry optimization.


# == Example for angle constraint for water ==


AMS_JOBNAME=angle $AMSBIN/ams <<eor

System
    Atoms
        O          0.001356    0.000999    0.000000
        H          0.994442   -0.037855    0.000000
        H         -0.298554    0.948531    0.000000
    End
End

Task GeometryOptimization

GeometryOptimization
   Convergence Step=1.0e-3
End

Constraints
  Angle 3 1 2 125.0
End

Engine ADF
    Basis
        Type DZP
    End
EndEngine
eor

echo "Angle after optimization:"
$AMSBIN/amsreport angle.results/adf.rkf -r angle#3#1#2


# == Example for fixed-coodinates constraint for water ==


AMS_JOBNAME=fixed_coordinates $AMSBIN/ams <<eor

System
    Atoms
        O          0.001356    0.000999    0.000000
        H          0.994442   -0.037855    0.000000
        H         -0.298554    0.948531    0.000000
    End
End

```

(continues on next page)

(continued from previous page)

```
Task GeometryOptimization

GeometryOptimization
   Convergence Step=1.0e-3
End

Constraints
  Coordinate 1 x 0.0
  Coordinate 1 y 0.0
  Coordinate 2 x 1.0
  Coordinate 2 y 0.0
End

Engine ADF
    symmetry NoSym
    Basis
     Type DZP
    End
    LinearScaling
        Overlap_Int 98
    End
EndEngine
eor

echo "Angle after optimization:"
$AMSBIN/amsreport fixed_coordinates.results/adf.rkf -r angle#3#1#2


# == Example for bond length constraint for water ==


AMS_JOBNAME=bond_length $AMSBIN/ams <<eor
System
    Atoms
        O         0.001356    0.000999    0.000000
        H         0.994442   -0.037855    0.000000
        H        -0.298554    0.948531    0.000000
    End
End

Task GeometryOptimization

GeometryOptimization
   Convergence Step=1.0e-3
End

Constraints
  Distance  1 2 1.03
  Distance  1 3 1.03
End

Engine ADF
    Symmetry NoSym
    Basis
     Type DZP
    End
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

echo "Bonds distance after optimization:"
$AMSBIN/amsreport bond_length.results/adf.rkf -r distance#1#2
$AMSBIN/amsreport bond_length.results/adf.rkf -r distance#1#3


# == Example for dihedral angle constraint for ethane ==


AMS_JOBNAME=dihedral $AMSBIN/ams <<eor

Task GeometryOptimization

GeometryOptimization
   Convergence Step=1.0e-3
End

System
    Atoms
      C        -0.004115   -0.000021    0.000023
      C         1.535711    0.000022    0.000008
      H        -0.399693    1.027812   -0.000082
      H        -0.399745   -0.513934    0.890139
      H        -0.399612   -0.513952   -0.890156
      H         1.931188    0.514066    0.890140
      H         1.931432    0.513819   -0.890121
      H         1.931281   -1.027824    0.000244
    End
End

Constraints
  Dihedral 6 2 1 3  20.00
End

Engine ADF
    Symmetry NoSym
    Basis
      type DZP
    End
EndEngine
eor

echo "Dihedral after optimization:"
$AMSBIN/amsreport dihedral.results/adf.rkf -r dihedral#6#2#1#3


# == Example for Block constraint (with a dihedral constraint) for ethane ==


AMS_JOBNAME=block_and_dihedral $AMSBIN/ams <<eor
System
    Atoms
        C   -0.004115   -0.000021    0.000023 region=b1
        C    1.535711    0.000022    0.000008 region=b2
        H   -0.399693    1.027812   -0.000082 region=b1
        H   -0.399745   -0.513934    0.890139 region=b1
```

(continues on next page)

(continued from previous page)

```
        H   -0.399612   -0.513952   -0.890156 region=b1
        H    1.931188    0.514066    0.890140 region=b2
        H    1.931432    0.513819   -0.890121 region=b2
        H    1.931281   -1.027824    0.000244 region=b2
    End
End

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1.0e-4 Step=1.0e-3
End

Constraints
  Dihedral  6 2 1 3  20.00
  Block b1
  Block b2
End

Engine ADF
    Symmetry NoSym
    Basis
      type DZP
    End
EndEngine
eor


echo "Dihedral after optimization:"
$AMSBIN/amsreport block_and_dihedral.results/adf.rkf -r dihedral#6#2#1#3
echo "Bonds distance after optimization:"
$AMSBIN/amsreport block_and_dihedral.results/adf.rkf -r distance#1#3
$AMSBIN/amsreport block_and_dihedral.results/adf.rkf -r distance#2#6
```

Example: Initial Hessian

Download `InitHess.run`

```
#! /bin/sh

# In this example the Hessian from a DFTB and a MOPAC calculation is used as a
# starting guess for the Hessian in an ADF geometry optimization of Isobutane.
# This example consist of 4 calculations:
# - MOPAC frequency calculation
# - DFTB frequency calculation
# - ADF geometry optimization with initial MOPAC Hessian
# - ADF geometry optimization with initial DFTB Hessian


# ======================
# Generate MOPAC Hessian
# ======================


AMS_JOBNAME=mopac_hessian $AMSBIN/ams << eor
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Properties
    NormalModes Yes
End

System
    Symmetrize
    Atoms
        C      -3.01417223      -2.81892970       0.27443080
        C      -2.32560254      -1.48669854      -0.04099291
        C      -0.81033566      -1.60274167       0.15652441
        H      -2.65505660      -3.61825489      -0.40916512
        H      -4.11459952      -2.71726364       0.16084942
        H      -2.80179295      -3.12071121       1.32226630
        H      -2.71211792      -0.71788031       0.66394035
        C      -2.65045192      -1.03659727      -1.46944155
        H      -0.32255475      -0.62449394      -0.04145480
        H      -0.38172118      -2.36329425      -0.53112669
        H      -0.58228356      -1.89634208       1.20336328
        H      -2.17532684      -0.05436369      -1.67852390
        H      -3.74843413      -0.92249496      -1.59450083
        H      -2.28021380      -1.77910197      -2.20888365
    End
End

Engine Mopac
EndEngine
eor


# =======================
# Generate a dftb Hessian
# =======================


AMS_JOBNAME=dftb_hessian $AMSBIN/ams <<eor

Task SinglePoint

Properties
   NormalModes True
End

System
    Symmetrize
    Atoms
     C      -3.014172230000      -2.818929700000       0.274430800000
     C      -2.325602540000      -1.486698540000      -0.040992910000
     C      -0.810335660000      -1.602741670000       0.156524410000
     H      -2.655056600000      -3.618254890000      -0.409165120000
     H      -4.114599520000      -2.717263640000       0.160849420000
     H      -2.801792950000      -3.120711210000       1.322266300000
     H      -2.712117920000      -0.717880310000       0.663940350000
     C      -2.650451920000      -1.036597270000      -1.469441550000
     H      -0.322554750000      -0.624493940000      -0.041454800000
```

(continues on next page)

(continued from previous page)

```
     H      -0.381721180000      -2.363294250000      -0.531126690000
     H      -0.582283560000      -1.896342080000       1.203363280000
     H      -2.175326840000      -0.054363690000      -1.678523900000
     H      -3.748434130000      -0.922494960000      -1.594500830000
     H      -2.280213800000      -1.779101970000      -2.208883650000
    End

    Charge 0
End

Engine DFTB
    Model SCC-DFTB
    ResourcesDir DFTB.org/mio-1-1
    SCC
    End
EndEngine
eor

# Use the MOPAC Hessian generated earlier for ADF optimization

AMS_JOBNAME=adf_mopachessian "$AMSBIN/ams" <<eor
LoadSystem
   File mopac_hessian.results/ams.rkf
End
Task GeometryOptimization
GeometryOptimization
   InitialHessian
      Type FromFile
      File mopac_hessian.results/mopac.rkf
   End
End
Engine ADF
EndEngine
eor

# Use the DFTB Hessian generated earlier for ADF optimization

AMS_JOBNAME=adf_dftbhessian "$AMSBIN/ams" <<eor
LoadSystem
   File dftb_hessian.results/ams.rkf
End
Task GeometryOptimization
GeometryOptimization
   InitialHessian
      Type FromFile
      File dftb_hessian.results/dftb.rkf
   End
End
Engine ADF
EndEngine
eor
```

Example: Geometry optimization with an external electric field or point charges: LiF

Download GO_LiF_Efield.run

```
#! /bin/sh


# In the first example a geometry optimization is performed with an external
# homogeneous electric field. In the second example a geometry optimization is
# performed with an external point charges

# Note that SYMMETRY NOSYM should be used. In case of point charges it is
# important to use the QPNEAR subkeyword of the BECKEGRID key with a large
# enough value that would include some of the point charges.


AMS_JOBNAME=Efield $AMSBIN/ams <<eor
System
  Atoms
     F    0.000000     0.800000     0.000000
     Li   0.000000    -0.800000     0.000000
  End
  ElectrostaticEmbedding
    ElectricField 0.0 0.0 0.01 [a.u.]
  End
End

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1E-5 Step=1E-3
End

Engine ADF
  Title LiF Geometry Optimization in the presence of electric field
  Basis
    type DZP
  End
  Symmetry NoSym
EndEngine
eor


AMS_JOBNAME=PointCharges $AMSBIN/ams <<eor
System
  Atoms
     F    0.000000     0.800000     0.000000
     Li   0.000000    -0.800000     0.000000
  End
  ElectrostaticEmbedding
    MultipolePotential
      Coordinates
         0.0 0.0  5.3  0.5
         0.0 0.0 -5.3 -0.5
      End
    End
  End
End
```

(continues on next page)

(continued from previous page)

```
Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1E-5 Step=1E-3
End

RigidMotions
  AllowTranslations None
End

Engine ADF
  Title LiF Geometry Optimization in the presence of point charges
  BeckeGrid
    QPnear 20
  End
  Basis
    type DZP
  End
  Symmetry NoSym
EndEngine
eor
```

14.3.2 Transition States, Linear Transits, Intrinsic Reaction Coordinates

Example: LT, Frequencies, TS, and IRC: HCN

Download HCN.run

```
#! /bin/sh

# Linear Transit (PES scan)

AMS_JOBNAME=PES_Scan "$AMSBIN/ams" <<eor

Task PESScan
System
    Atoms
        C   0.00000000   0.00000000  -1.16013597
        N   0.00000000   0.00000000   0.00000000
        H   0.00000000   0.00000000   0.99666657
    End
End

PESScan
    CalcPropertiesAtPESPoints Yes
    ScanCoordinate
        nPoints 10
        Angle 3 2 1 180.0 0
    End
End

Engine ADF
    Basis
```

(continues on next page)

(continued from previous page)

```
        Type DZP
    End
EndEngine
eor

# Transition state search starting from highest point in PES scan (i.e. PESPoint(7))

AMS_JOBNAME=TS_search "$AMSBIN/ams" <<eor

Task TransitionStateSearch

LoadSystem
  File PES_Scan.results/PESPoint(7).rkf
End

GeometryOptimization
  InitialHessian
    Type Calculate
  End
  Convergence Gradients=0.00001
End

Properties
  NormalModes Yes
End

Engine ADF
    Basis
        Type DZP
    End
EndEngine
eor

# IRC from TS

AMS_JOBNAME=IRC "$AMSBIN/ams" <<eor

Task IRC

LoadSystem
  File TS_search.results/adf.rkf
End

Engine ADF
    Basis
        Type DZP
    End
EndEngine
eor
```

Example: TS search using partial Hessian: C2H6 internal rotation

Download TS_C2H6.run

```
#! /bin/sh

# Frequently when searching for a transition state, one needs an accurate second
# derivatives matrix, a Hessian. An exact Hessian may be obtained analytically
# but this may be very expensive for large molecules. In such cases it may be
# beneficial to calculate Hessian matrix elements only for atoms directly
# involved in the reaction for which a transition state is sought for. The rest
# of the Hessian can then be approximated using a cheaper method.

# In this example, a saddle point of the ethane internal rotation around C-C
# bond is found. In principle, only hydrogen atoms contribute to the normal mode
# we are interested in. Therefore we calculate a partial Hessian matrix
# including hydrogen atoms only. For this purpose, the SelectedRegionForHessian
# key is used.
# In the calculation symmetry will be used.

AMS_JOBNAME=Partial_Hessian $AMSBIN/ams <<eor
System
  Symmetrize
  atoms
     C       0.000000000000       0.000000000000       0.767685465031
     C       0.000000000000       0.000000000000      -0.767685465031
     H       0.964354016767       0.347635559279       1.177128271450 region=hess
     H      -0.181115782790      -1.008972856410       1.177128271450 region=hess
     H      -0.783238233981       0.661337297125       1.177128271450 region=hess
     H      -0.500471876676       0.894626767091      -1.177128271450 region=hess
     H      -0.524533568868      -0.880734742626      -1.177128271450 region=hess
     H       1.025005445540      -0.013892024465      -1.177128271450 region=hess
  end
end

Task SinglePoint

Properties
  NormalModes Yes
  SelectedRegionForHessian hess
End

Engine ADF
  title Ethane transition state search using partial Hessian
  beckegrid
    quality good
  end
  basis
    core Large
    type DZ
    CreateOutput Yes
  end
EndEngine
eor

# After the Hessian is calculated, the resulting TAPE21 file is used for a
# subsequent transition state search:
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=TS $AMSBIN/ams <<eor
LoadSystem
   File Partial_Hessian.results/ams.rkf
End

Task TransitionStateSearch

GeometryOptimization
   InitialHessian
      Type FromFile
      File Partial_Hessian.results/adf.rkf
   End
End

EngineRestart Partial_Hessian.results/adf.rkf

Engine ADF
  title Ethane transition state search using partial Hessian
  beckegrid
    quality good
  end
  basis
    core Large
    type DZ
  end
EndEngine

eor

# Important note: care should be taken to specify correct mode in the
# TransitionState keyword. Because a significant part of the Hessian will not be
# calculated exactly, it is possible that it will have more than one negative
# eigenvalue, in which case the one we are interested in may not be the first
# one. In such a case, one needs to specify the correct mode number in the
# TransitionState keyword.
```

Example: Relativistic ZORA TS search: CH4 + HgCl2 <==> CH3HgCl + HCl

Download `TS_CH4_HgCl2.run`

```
#! /bin/sh


# Compute the approximate initial hessian using DFTB:

AMS_JOBNAME=InitialHessian $AMSBIN/ams <<eor

System
  atoms
     C         0.049484    0.042994    0.000000
     H        -0.068980    0.638928   -0.915972
     H        -0.068980    0.638928    0.915972
     H        -0.841513   -0.626342    0.000000
     H         0.555494   -1.148227    0.000000
     Hg        2.303289   -0.007233    0.000000
```

(continues on next page)

(continued from previous page)

```
     Cl        4.429752    0.776056    0.000000
     Cl        1.342057   -2.676083    0.000000
  end
  Symmetrize Yes
end

Task SinglePoint

Properties
  NormalModes Yes
End

Engine DFTB
  Model GFN1-xTB
EndEngine
eor

# A ZORA scalar relativistic Transition State calculation using the
# initial hessian computed by DFTB

AMS_JOBNAME=TS $AMSBIN/ams <<eor

LoadSystem
  File InitialHessian.results/dftb.rkf
End

EngineDebugging
  NeverQuiet True
End

Task TransitionStateSearch

GeometryOptimization
  Convergence
    Energy 1.25e-6
  End
  InitialHessian
    Type FromFile
    File InitialHessian.results/dftb.rkf
  End
End

Properties
  PESPointCharacter Yes
End

Engine ADF
  Title Transition State: CH4 + HgCl2 <==> CH3HgCl + HCl
  basis
    Type TZP
    Core Small
    PerAtomType Symbol=Cl File=TZP/Cl.2p
  end
  relativity
    level scalar
    formalism ZORA
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine

eor
```

Example: TS reaction coordinate: F- + CH3Cl

Download TSRC_SN2.run

```
#! /bin/sh


# With the ReactionCoordinate block key one can specify a reaction coordinate along␣
↪which the
# transition state is sought for. This feature is especially useful when an
# accurate Hessian is not available.

# This example tries to find the TS for the S_N2 reaction of F^- + CH_3Cl <==> CH_3F␣
↪+ Cl^-

$AMSBIN/ams <<eor
System
  atoms
     C              0.000000     0.000000     0.000000
     H             -0.530807     0.919384693  0.012892
     H             -0.530807    -0.919384693  0.012892
     H              1.061614     0.000000     0.012892
     Cl             0.000000     0.000000    -2.124300
     F              0.000000     0.000000     2.019100
  end
  charge -1
end

Task TransitionStateSearch

TransitionStateSearch
  ReactionCoordinate
     distance 1 5  1.0
     distance 1 6 -1.0
  end
End

Properties
  PESPointCharacter Yes
End

Engine ADF
  title Transition state search for Sn2 reaction of F- + CH3Cl
  beckegrid
    quality good
  end
  eprint
    sfo noeig noovl
  end
  basis
    core NONE
```

(continues on next page)

(continued from previous page)

```
    type TZ2P
  end
  scf
    converge 1.0e-6 1.0e-6
    diis
      ok 0.01
    end
    iterations 99
  end
  xc
    gga OPBE
    lda VWN
  end
EndEngine
eor
```

Example: (non-)Linear Transit: H2O

Download Transit_H2O.run

```
#! /bin/sh

# AMS is capable of performing linear transits
# Here multiple coordinate descriptors are combined.
# This combines the individual coordinates into one compound coordinate,
# i.e. all coordinates will transit together through their respective ranges.

$AMSBIN/ams <<eor
System
  atoms
     O              0.000000     0.000000     0.000000
     H              0.000000    -0.689440    -0.578509
     H              0.000000     0.689440    -0.578509
  end
end

Task PESSCAN

PESScan
   ScanCoordinate
      nPoints 4
      distance 1 2 0.8 1.5
      angle 2 1 3 100.0 120.0
   End
End

GeometryOptimization
   Convergence
      gradients 0.0001
   End
End

Engine ADF
  Basis
    Type DZP
```

(continues on next page)

(continued from previous page)

```
  End
EndEngine
eor
```

14.3.3 Total energy, Multiplet States, S2, Localized hole, CEBE

Example: Total Energy calculation: H2O

Download Energy_H2O.run

```
#! /bin/sh


# If the TOTALENERGY is included the total energy will be calculated.

# This example performs single point runs for H2 O with PBE/DZP with frozen
# cores and all-electron and B3LYP/DZP with all-electron and HARTREEFOCK/DZP
# with all-electron The tests run in C(2v) symmetry. Integration quality is good
# which should give total energies accurate at least up to 10^-4 atomic units.
# The key EXACTDENSITY is used for higher accuracy of the results.

# Note that only energy difference comparisons are meaningful. These are the
# only energies that play a role in chemistry of course, and for this one does
# not need total energies.


# ----------------------------
# H2O PBE/DZP with frozen core
# ----------------------------

AMS_JOBNAME=FrozenCore $AMSBIN/ams <<eor
System
  atoms
     O        0.00000        0.00000        0.00000
     H        0.00000       -0.68944       -0.57851
     H        0.00000        0.68944       -0.57851
  end
end

Task SinglePoint

Engine ADF
  title H2O PBE/DZP (frozen core) single point calculation
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core Small
    type DZP
  end
  noprint BAS FUNCTIONS
  nototen
```

(continues on next page)

(continued from previous page)

```
  numericalquality Good
  totalenergy
  xc
    gga PBE
  end
  Relativity
    Level None
  End
EndEngine
eor


# ------------------------
# H2O PBE/DZP all-electron
# ------------------------

AMS_JOBNAME=AllElectron $AMSBIN/ams <<eor
System
  atoms
     O        0.00000        0.00000        0.00000
     H        0.00000       -0.68944       -0.57851
     H        0.00000        0.68944       -0.57851
  end
end

Task SinglePoint

Engine ADF
  title H2O PBE/DZP (all-electron) single point calculation
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZP
  end
  noprint BAS FUNCTIONS
  nototen
  numericalquality Good
  totalenergy
  xc
    gga PBE
  end
  Relativity
    Level None
  End
EndEngine
eor

# --------------------------
# H2O B3LYP/DZP all-electron
# --------------------------

AMS_JOBNAME=B3LYP $AMSBIN/ams <<eor
System
```

(continues on next page)

(continued from previous page)

```
  atoms
     O        0.00000        0.00000        0.00000
     H        0.00000       -0.68944       -0.57851
     H        0.00000        0.68944       -0.57851
  end
end

Task SinglePoint

Engine ADF
  title H2O B3LYP/DZP (all-electron) single point calculation
  beckegrid
    quality good
  end
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZP
  end
  noprint BAS FUNCTIONS
  nototen
  totalenergy
  xc
    hybrid B3LYP
  end
  zlmfit
    quality good
  end
  Relativity
    Level None
  End
EndEngine
eor

# --------------------------------
# H2O HARTREEFOCK/DZP all-electron
# --------------------------------

AMS_JOBNAME=HF $AMSBIN/ams <<eor
System
  atoms
     O        0.00000        0.00000        0.00000
     H        0.00000       -0.68944       -0.57851
     H        0.00000        0.68944       -0.57851
  end
end

Task SinglePoint

Engine ADF
  title H2O HARTREEFOCK/DZP (all-electron) single point calculation
  beckegrid
    quality good
```

(continues on next page)

(continued from previous page)

```
  end
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  exactdensity
  basis
    core None
    type DZP
  end
  noprint BAS FUNCTIONS
  nototen
  totalenergy
  xc
    hartreefock
  end
  zlmfit
    quality good
  end
  Relativity
    Level None
  End
EndEngine
eor
```

Example: Multiplet States: [Cr(NH3)6]3+

Download `SD_CrNH3_6.run`

```
#! /bin/sh


# The computation of multiplet states corresponding to an open-shell system can
# be carried out with ADF by first computing the 'Average-of-Configuration'
# (aoc) state, where all orbitals in the open shell are degenerate and equally
# occupied. This computation is spin-restricted and serves as a fragment file
# for the multiplet run, where then different occupation numbers are assigned to
# the various orbitals in the open shell. The corresponding energies are
# computed in the field of the aoc, which is achieved by not iterating the self-
# consistency equations to convergence but only computing the orbitals in the
# initial field.

# Since ADF requires that all symmetry-partners in an irreducible representation
# (irrep) have equal occupations, the multiplet calculation, where such orbitals
# are not equally occupied, must be carried out in a formally lower point group
# symmetry. The point group to select and the appropriate occupation numbers to
# apply must be worked out by the user 'on paper' in advance.
# See the discussion of Multiplet energies in ADF manual.

# The script starts with the 'creation' of the required basic atoms, N, H, Cr
# using a fair basis set quality.

# The next step is the computation of the ammonia fragment NH3. This is not a
# crucial step here: the multiplet state computation can equally well be carried
# out by not using any intermediate compound fragments. However, it illustrates
```

(continues on next page)

(continued from previous page)

```
# once more how a bigger molecule can be built up from smaller, but not trivial
# fragments.

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/Cr.2p
mv TAPE12 t12.rel

AMS_JOBNAME=Cr $AMSBIN/ams <<eor
System
  Atoms
    Cr 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create Cr file=$AMSRESOURCES/ADF/ZORA/TZP/Cr.2p
  corepotentials t12.rel
     Cr 1
  end
EndEngine
eor


AMS_JOBNAME=NH3 $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N    0               0                    0
     H   -0.95522523      0                    0.3711068
     H    0.477612615   -0.8272493155158333   0.3711068
     H    0.477612615    0.8272493155158333   0.3711068
  end
end

Task SinglePoint

Engine ADF
  title AMMONIA
  basis
    core Small
    type TZP
    CreateOutput No
  end
  noprint sfo,frag,functions
  symmetry C(3V)
EndEngine
eor

# The input of the atomic coordinates does not use expressions. In this case to
# enforce exact symmetry relations 14-digit input values are required.
# The symmetrization key is used to enforce the symmetry to machine precision,
# and to put the molecule in the right orientation for the symmetry.
# The symmetry specification is redundant: the program would
# also find it by itself.

# == Average-of-Configuration ==

# The next step is to compute the reference state, with respect to which we will
```

(continues on next page)

(continued from previous page)

```
# later compute the multiplet states. The reference state is the so-called
# 'Average-of-configuration' (aoc) state. The result file (TAPE21) of this
# calculation will be used as a fragment file.

AMS_JOBNAME=AOC $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     Cr      0.00000000     0.00000000     0.00000000
     N       0.00000000     1.71464325     1.21243632  adf.f=NH3|1
     H       0.00000000     1.46615142     2.20663314  adf.f=NH3|1
     H      -0.82724932     2.29340040     1.03672570  adf.f=NH3|1
     H       0.82724932     2.29340040     1.03672570  adf.f=NH3|1
     N      -1.48492461    -0.85732162     1.21243632  adf.f=NH3|2
     H      -1.26972438    -0.73307571     2.20663314  adf.f=NH3|2
     H      -1.57251835    -1.86311912     1.03672570  adf.f=NH3|2
     H      -2.39976767    -0.43028128     1.03672570  adf.f=NH3|2
     N       1.48492461    -0.85732162     1.21243632  adf.f=NH3|3
     H       1.26972438    -0.73307571     2.20663314  adf.f=NH3|3
     H       2.39976767    -0.43028128     1.03672570  adf.f=NH3|3
     H       1.57251835    -1.86311912     1.03672570  adf.f=NH3|3
     N       0.00000000    -1.71464325    -1.21243632  adf.f=NH3|4
     H       0.00000000    -1.46615142    -2.20663314  adf.f=NH3|4
     H       0.82724932    -2.29340040    -1.03672570  adf.f=NH3|4
     H      -0.82724932    -2.29340040    -1.03672570  adf.f=NH3|4
     N       1.48492461     0.85732162    -1.21243632  adf.f=NH3|5
     H       1.26972438     0.73307571    -2.20663314  adf.f=NH3|5
     H       1.57251835     1.86311912    -1.03672570  adf.f=NH3|5
     H       2.39976767     0.43028128    -1.03672570  adf.f=NH3|5
     N      -1.48492461     0.85732162    -1.21243632  adf.f=NH3|6
     H      -1.26972438     0.73307571    -2.20663314  adf.f=NH3|6
     H      -2.39976767     0.43028128    -1.03672570  adf.f=NH3|6
     H      -1.57251835     1.86311912    -1.03672570  adf.f=NH3|6
  end
end

Task SinglePoint

Engine ADF
  title Cr(NH3)6 : Average-of-Configuration run
  comment
     using NH3-fragments
  end
  fragments
     Cr  Cr.results/adf.rkf
     NH3 NH3.results/adf.rkf
  end
  Basis CreateOutput=No
  irrepoccupations
     A1.G 8.75
     A2.G 2
     E1.G 16 1.5 0.75
     A1.U 2
     A2.U 8
     E1.U 20
  end
  scf
```

(continues on next page)

(continued from previous page)

```
    iterations 25
    mixing 0.15
  end
  symmetry D(3d)
EndEngine

eor


# Occupation numbers are specified, to make certain what the reference state is
# that we will start from in the subsequent calculations. The result file TAPE21
# is saved to serve as fragment file in the subsequent calculations.

# == One-determinant states ==

# Now, we proceed with the multiplet calculations. In the example they are
# combined in one single run, but they could also be evaluated in separate runs.
# For each calculation it is required to:

# a) Use the aoc TAPE21 file as fragment file
# b) Choose which molecular orbitals in the open shell to occupy: select the
#    appropriate point group symmetry and the UnRestricted key if necessary and
#    specify the occupation numbers, using the irreducible representations of the
#    selected point group.

# The results are one-determinant calculations, which must then, later, be
# combined analytically to obtain the required multiplet energy values.

AMS_JOBNAME=SD $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     Cr       0.00000000      0.00000000      0.00000000   adf.f=CrA6
     N        0.00000000      1.71464325      1.21243632   adf.f=CrA6
     H        0.00000000      1.46615142      2.20663314   adf.f=CrA6
     H       -0.82724932      2.29340040      1.03672570   adf.f=CrA6
     H        0.82724932      2.29340040      1.03672570   adf.f=CrA6
     N       -1.48492461     -0.85732162      1.21243632   adf.f=CrA6
     H       -1.26972438     -0.73307571      2.20663314   adf.f=CrA6
     H       -1.57251835     -1.86311912      1.03672570   adf.f=CrA6
     H       -2.39976767     -0.43028128      1.03672570   adf.f=CrA6
     N        1.48492461     -0.85732162      1.21243632   adf.f=CrA6
     H        1.26972438     -0.73307571      2.20663314   adf.f=CrA6
     H        2.39976767     -0.43028128      1.03672570   adf.f=CrA6
     H        1.57251835     -1.86311912      1.03672570   adf.f=CrA6
     N        0.00000000     -1.71464325     -1.21243632   adf.f=CrA6
     H        0.00000000     -1.46615142     -2.20663314   adf.f=CrA6
     H        0.82724932     -2.29340040     -1.03672570   adf.f=CrA6
     H       -0.82724932     -2.29340040     -1.03672570   adf.f=CrA6
     N        1.48492461      0.85732162     -1.21243632   adf.f=CrA6
     H        1.26972438      0.73307571     -2.20663314   adf.f=CrA6
     H        1.57251835      1.86311912     -1.03672570   adf.f=CrA6
     H        2.39976767      0.43028128     -1.03672570   adf.f=CrA6
     N       -1.48492461      0.85732162     -1.21243632   adf.f=CrA6
     H       -1.26972438      0.73307571     -2.20663314   adf.f=CrA6
     H       -2.39976767      0.43028128     -1.03672570   adf.f=CrA6
     H       -1.57251835      1.86311912     -1.03672570   adf.f=CrA6
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  title Cr(NH3)6 : SlaterDeterminants run
  fragments
     CrA6  AOC.results/adf.rkf
  end
  noprint frag
  scf
    iterations 0
  end
  slaterdeterminants
     Check AOC
     A1.g   4 0.375         // 4 0.375
     A2.g   1               // 1
     E1.g:1 4 0.375 0.1875  // 4 0.375  0.1875
     E1.g:2 4 0.375 0.1875  // 4 0.375  0.1875
     A1.u   1//1
     A2.u   4//4
     E1.u:1 5//5
     E1.u:2 5//5
     SUBEND
     State1
     A1.g   4 1             // 4 1
     A2.g   1               // 1
     E1.g:1 4 0     0       // 4 0      1
     E1.g:2 4 0     0       // 4 0      0
     A1.u   1//1
     A2.u   4//4
     E1.u:1 5//5
     E1.u:2 5//5
     SUBEND
     State2
     A1.g   4 1             // 4 1
     A2.g   1               // 1
     E1.g:1 4 0     0       // 4 1      0
     E1.g:2 4 0     0       // 4 0      0
     A1.u   1//1
     A2.u   4//4
     E1.u:1 5//5
     E1.u:2 5//5
     SUBEND
     State3
     A1.g   4 1             // 4 1
     A2.g   1               // 1
     E1.g:1 4 0     1       // 4 0      0
     E1.g:2 4 0     0       // 4 0      0
     A1.u   1//1
     A2.u   4//4
     E1.u:1 5//5
     E1.u:2 5//5
     SUBEND
  end
  symmetry C(I)
```

(continues on next page)

(continued from previous page)

```
  unrestricted
EndEngine
eor


# The SlaterDeterminants block may contain any number of sub blocks, each
# starting with an (arbitrary) title record, followed by a set of occupation
# numbers and closed by a SubEnd record. Each such subkey block specifies a
# single one-determinant-state calculation. All occupation numbers must
# reference the irreps of the specified point group symmetry, C(I) in the
# example, and must be just a reassignment of the electrons that are equally
# distributed over the corresponding degenerate irreps in the reference aoc
# calculation.

# The so-obtained energies of the one-determinant states can now be combined to
# calculate the desired multiplet energies. See the ADF manual.

# Note carefully that in the calculation of the SingleDeterminants, the scf
# procedure is prevented to cycle to convergence by setting the subkey
# Iterations to zero in the SCF data block.
```

Example: Calculation of S2: CuH+

Download `CuH+_S-squared.run`

```
#! /bin/sh


# Example calculates expectation value of S^2 (<S^2>) of CuH+ in various
# symmetries, using unrestricted density functional theory.

AMS_JOBNAME=first $AMSBIN/ams <<eor
System
  atoms
     Cu    0.0    0.0 0.0
     H     1.463 0.0 0.0
  end
  charge 1.0
end

Task SinglePoint

Engine ADF
  title calculate expectation value of S-squared
  basis
    core Small
    type TZP
  end
  spinpolarization 1.0
  symmetry Nosym
  unrestricted Yes
EndEngine

eor
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=second $AMSBIN/ams <<eor
System
  atoms
     Cu   0.0 0.0 0.0
     H    0.0 0.0 1.463
  end
  charge 1.0
end

Task SinglePoint

Engine ADF
  title calculate expectation value of S-squared
  basis
    core Small
    type TZP
  end
  spinpolarization 1.0
  symmetry C(LIN)
  unrestricted Yes
EndEngine

eor


AMS_JOBNAME=third $AMSBIN/ams <<eor
System
  atoms
     Cu   0.0   0.0 0.0
     H    1.463 0.0 0.0
  end
  charge 1.0
end

Task SinglePoint

Engine ADF
  title calculate expectation value of S-squared
  basis
    core Small
    type TZP
  end
  spinpolarization 1.0
  symmetry C(S)
  unrestricted Yes
EndEngine

eor


AMS_JOBNAME=fourth $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     Cu   0.0   0.0 0.0
     H    1.463 0.0 0.0
```

(continues on next page)

(continued from previous page)

```
  end
  charge 1.0
end

Task SinglePoint

Engine ADF
  title calculate expectation value of S-squared
  basis
    core Small
    type TZP
  end
  spinpolarization -1.0
  unrestricted Yes
EndEngine

eor
```

Example: Localized Hole: N2+

Download ModStPot_N2+.run

```
#! /bin/sh


# This calculation illustrates:

# - How to specify the net total charge on a molecule
# - How to enforce breaking the symmetry that is present in the start-up
#   situation, in this case to localize a hole in the electron density on one of
#   the two equivalent atoms.
# - How to prevent the scf from oscillating back and forth between the two
#   equivalent situations or from even restoring the unwanted symmetry


$AMSBIN/ams <<eor
System
  Atoms
     N 0 0  -2.0
     N 0 0   2.0
  End
  Charge 1
End

Task SinglePoint

Engine ADF
  Title N2+ hole localization
  IrrepOccupations
     sigma 3 // 1 0 1
     pi    2 // 2
  End

  ModifyStartPotential
     N/1    0.5   0.5
```

(continues on next page)

(continued from previous page)

```
     N/2       4   1
  End

  Basis
    Type DZP
    Core Small
    CreateOutput Yes
  End

  SCF
    DIIS
      ok 0.5
    End
  End
  Unrestricted Yes
  SpinPolarization 1

  Relativity
    Level None
  End

  Symmetry C(lin)
EndEngine
eor

# The purpose of this run is to compute the N2+ ion, with the hole localized on
# one of the atoms. In a very small system like N2+ this is a tricky thing to
# do. The program has a tendency towards the symmetric solution, with the hole
# delocalized. A few trial runs, just putting a net +1 charge into the system,
# will reveal that clearly.

# To achieve the desired situation we apply the key modifystartpotential to
# break the symmetry of the initial potential. A potential is generated as if
# the electronic cloud in the second N fragment is spin-polarized in a ratio 4:1
# (this precise value is not very relevant), which achieves that initially a
# non-symmetric solution is obtained. The symmetry must be specified, lest the
# program determine and use the higher symmetry from the nuclear frame. This
# would prevent any symmetry breaking altogether.

# Next, in order to prevent that the system relaxes to the symmetric situation,
# we apply the keeporbitals option of the occupations key. This fixes the
# occupied orbitals in the sense that in each scf cycle the program will try to
# keep the electrons in orbitals that resemble the previously occupied orbitals
# as much as possible.

# The key modifystartpotential here demonstrated has a more relevant and less
# unstable application in larger systems. See the User's Guide for references.
```

Example: Broken spin-symmetry: Fe4S4

Download Fe4S4_BrokenSymm.run

```
#! /bin/sh


# This calculation shows a spin-flip restart feature that allows to exchange
# alpha and beta fit coefficients for selected atoms upon restart. First the
# high spin configuration with 8 more alpha-electrons than beta-electrons is
# calculated (S_z=4). Next the broken spin-symmetry configuration is calculated
# (S_z=0), using the subkey spinflip in the restart key. In this case the spin
# will be flipped for iron atoms 1 and 2. Note that the used symmetry C(2v) is
# lower than the actual symmetry T(d), such that iron atoms 1 and 2 can have a
# different spin than iron atoms 3 and 4.


AMS_JOBNAME=HighSpin $AMSBIN/ams <<eor
System
  atoms
     Fe      -0.000000000000      -1.256142548900       0.888226914500
     Fe       0.000000000000       1.256142548900       0.888226914500
     Fe      -1.256142548900       0.000000000000      -0.888226914500
     Fe       1.256142548900      -0.000000000000      -0.888226914500
     S       -1.845393493800       0.000000000000       1.304890253400
     S        1.845393493800      -0.000000000000       1.304890253400
     S       -0.000000000000      -1.845393493800      -1.304890253400
     S        0.000000000000       1.845393493800      -1.304890253400
  End
End

Task SinglePoint

Engine ADF
  Title Fe4S4 High-spin configuration

  Unrestricted Yes
  SpinPolarization 8.0

  symmetry C(2v)

  XC
    GGA OPBE
  End

  ! Omit printing of SFOs, populations, etc. since they may change per computer
  ! In real-life calculations you may want to have them
  NoPrint SFO SCF FIT NUMINT MDC DensityAtNucleus
EndEngine
eor


AMS_JOBNAME=LowSpin $AMSBIN/ams <<eor
System
  atoms
     Fe      -0.000000000000      -1.256142548900       0.888226914500
     Fe       0.000000000000       1.256142548900       0.888226914500
     Fe      -1.256142548900       0.000000000000      -0.888226914500
```

(continues on next page)

(continued from previous page)

```
     Fe       1.256142548900      -0.000000000000      -0.888226914500
     S       -1.845393493800       0.000000000000       1.304890253400
     S        1.845393493800      -0.000000000000       1.304890253400
     S       -0.000000000000      -1.845393493800      -1.304890253400
     S        0.000000000000       1.845393493800      -1.304890253400
  End
End

Task SinglePoint

EngineRestart HighSpin.results/adf.rkf

Engine ADF
  Title Fe4S4 low-spin configuration

  Restart
    SpinFlip 1 2
  End

  Unrestricted Yes
  SpinPolarization 0.0

  symmetry C(2v)

  XC
    GGA OPBE
  End
EndEngine
eor
```

Example: Core-electron binding energies (CEBE): NNO

Download CEBE_NNO.run

```
#!/bin/sh


# ADF is well suited for calculating Core Electron Binding Energies (CEBEs). In
# this example it is shown how one can differentiate between the 1s CEBEs of the
# two non-equivalent nitrogen atoms in N2 O, using a delta-SCF technique. It
# starts with a regular calculation that has the purpose of preparing a
# reference adf.rkf file for the NNO molecule, which will later be useful in the
# energy analysis. The result file is saved to t21.NNO.

# The same GGA functional is specified throughout the run. The amount of output
# is reduced by using some print keys.

# The prepare the nitrogen atom with a core hole (restricted) will be used as a
# fragment later. This enables selection of where the core hole should be.

# prepare the Nitrogen core hole restricted to be used as fragment


AMS_JOBNAME=N_ch $AMSBIN/ams <<eor
System
```

(continues on next page)

(continued from previous page)

```
  atoms
     N   0.0    0.0     0.0
  end
end

Task SinglePoint

Engine ADF
  title N atom core hole
  irrepoccupations
     s    1 2
     p    3
  end
  basis
    type TZ2P
    core None
    CreateOutput Yes
  end
  numericalquality good
  xc
    gga pw86x pw91c
  end
EndEngine

eor


# Now perform the restricted ground state molecule for analysis later. The
# adf.rkf result file is saved.
# Remark: In the core hole calculation the 2 N atoms will have different fragments.
# Therefore also in this calculation the 2 N atoms are treated with 2 different␣
↪fragments,
# which is done here by using 2 different region: 'one' and 'two'


AMS_JOBNAME=NNO $AMSBIN/ams <<eor
System
  atoms
     N   0.0    0.0    -1.1284  region=one
     N   0.0    0.0     0.0     region=two
     O   0.0    0.0     1.1841
  end
end

Task SinglePoint

Engine ADF
  title NNO
  basis
    type TZ2P
    core None
  end
  noprint sfofragpop fragsfo
  numericalquality good
  xc
    gga pw86x pw91c
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine

eor


# Next follow two sets of almost identical calculations in which a 1s electron
# is removed from one or the other N atom (please note that the deepest s level
# is associated with the 1s of the oxygen atom). The molecular NNO result file
# is used as fragment. An unrestricted calculation is done and a positive charge
# is specified. The final result file for the molecule with the core hole is
# saved. Then another calculation is done to conveniently obtain the energy with
# respect to the normal molecule. This is repeated for a core hole on the other
# N atom.


AMS_JOBNAME=NNO_unr1 $AMSBIN/ams <<eor
System
  atoms
     N   0.0    0.0    -1.1284  region=one  adf.f=N_ch
     N   0.0    0.0     0.0     region=two
     O   0.0    0.0     1.1841
  end
  charge 1
end

Task SinglePoint

Engine ADF
  title NNO unrestricted core hole
  fragments
     N_ch N_ch.results/adf.rkf
  end
  irrepoccupations
     sigma  1 1 1 4 // 1 0 1 4
     pi     4       // 4
  end
  basis
    type TZ2P
    core None
  end
  noprint sfofragpop fragsfo
  numericalquality good
  spinpolarization 1
  unrestricted Yes
  xc
    gga pw86x pw91c
  end
EndEngine

eor


# In the second calculation the result file of one of the unrestricted NNO
# calculations is used as restart file, which ensures that the hole stays at its
# place, because the starting density is already correct. The result file
# t21.NNO for the normal NNO calculation is specified as fragment to serve as an
# energy reference. The final Bonding Energy printed by ADF indicates what the
```

(continues on next page)

(continued from previous page)

```
# CEBE is. However, please check Chong, D.P. Accurate DFT Calculation of Core-
# Electron Binding Energies in Reviews in Modern Quantum Chemistry, A
# Celebration of the Contributions of R.G. Parr, edited by K.D. Sen (World
# Scientific Publishing Co., Singapore), 1106-1139 (2002) for more detailed
# information on Core-Electron Binding Energies. This reference also contain
# information on empirical corrections that may have to be made on the final
# numbers.


AMS_JOBNAME=NNO_unr1_1 $AMSBIN/ams <<eor
System
  atoms
     N   0.0    0.0    -1.1284  region=one  adf.f=NNO
     N   0.0    0.0     0.0     region=two  adf.f=NNO
     O   0.0    0.0     1.1841              adf.f=NNO
  end
  charge 1
end

Task SinglePoint

EngineRestart NNO_unr1.results/adf.rkf

Engine ADF
  title NNO unr. core hole
  fragments
     NNO NNO.results/adf.rkf
  end
  irrepoccupations
     sigma  1 1 1 4 // 1 0 1 4
     pi     4       // 4
  end
  noprint sfofragpop fragsfo
  numericalquality good
  spinpolarization 1
  unrestricted Yes
  xc
    gga pw86x pw91c
  end
EndEngine

eor


# core hole nr. 2

AMS_JOBNAME=NNO_unr2 $AMSBIN/ams <<eor
System
  atoms
     N   0.0    0.0    -1.1284  region=one
     N   0.0    0.0     0.0     region=two  adf.f=N_ch
     O   0.0    0.0     1.1841
  end
  charge 1
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  title NNO unrestricted core hole
  fragments
     N_ch N_ch.results/adf.rkf
  end
  irrepoccupations
     sigma  1 1 1 4 // 1 0 1 4
     pi     4       // 4
  end
  basis
    type TZ2P
    core None
  end
  noprint sfofragpop fragsfo
  numericalquality good
  spinpolarization 1
  unrestricted Yes
  xc
    gga pw86x pw91c
  end
EndEngine

eor

# analysis for core hole 2

AMS_JOBNAME=unr3 $AMSBIN/ams <<eor
System
  atoms
     N   0.0    0.0    -1.1284  region=one  adf.f=NNO
     N   0.0    0.0     0.0     region=two  adf.f=NNO
     O   0.0    0.0     1.1841              adf.f=NNO
  end
  charge 1
end

Task SinglePoint

EngineRestart NNO_unr2.results/adf.rkf

Engine ADF
  title NNO unr. core hole
  fragments
     NNO NNO.results/adf.rkf
  end
  irrepoccupations
     sigma  1 1 1 4 // 1 0 1 4
     pi     4       // 4
  end
  noprint sfofragpop fragsfo
  numericalquality good
  spinpolarization 1
  unrestricted Yes
  xc
    gga pw86x pw91c
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine

eor

# Similarly, one could easily have prepared an oxygen with a core hole and
# determined the CEBE of the oxygen 1s atom.
```

Example: CEBE 1s Iodine in HI

Download CEBE_HI.run

```
#! /bin/sh

# Spin-orbit coupled relativistic ZORA, X2C, and RA-X2C
# are compared in this example for the calculation of
# Core Electron Binding Energies (CEBEs).
# In this case the CEBE of the 1s of Iodine in the molecule HI.
# For such calculations X2C or RA-X2C is recommended.
# Note, for more accurate calculations one should increase the basis set.
# In this example the LDA functional is used.

# Results of this example will show that for X2C the CEBE is around 33.211 keV
# and for RA-X2C the CEBE is around 33.202 keV.
# If one increases the basis set (especially in the core region), for example, with a␣
↪QZ4P basis set,
# the results obtained with X2C and RA-X2C will be much closer.

# In case of ZORA, without the scaled ZORA energy correction
# the CEBE in this case is around 34.515 keV, much larger than the results with X2C␣
↪or RA-X2C.
# If one includes the scaled ZORA energy correction (can be found in the output)
# the CEBE in case of ZORA is around 33.238 keV.

# NOTE: The scaled ZORA energy correction should only be used to compare two␣
↪calculations
# in which the only difference in the calculation is the electron configuration.
# Then the difference in energy of this term should be added to the difference in␣
↪energy of
# the two electron configurations.
# This term should not be used otherwise.
# In practice it is useful only for core excitation energy calculations.

for f in "ZORA" "X2C" "RA-X2C"
do

AMS_JOBNAME=HI_$f $AMSBIN/ams <<eor
System
  atoms
     H 0.0 0.0 0.000
     I 0.0 0.0 1.609
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  basis
    core None
    type DZ
  end
  numericalquality good
  relativity
    level spin-orbit
    formalism $f
  end
  nuclearmodel gaussian
EndEngine
eor

AMS_JOBNAME=HI_core_$f $AMSBIN/ams <<eor
System
  atoms
     H 0.0 0.0 0.000
     I 0.0 0.0 1.609
  end
  charge 1
end

Task SinglePoint

Engine ADF
  basis
    core None
    type DZ
  end
  numericalquality good
  relativity
    level spin-orbit
    formalism $f
    SpinOrbitMagnetization NonCollinear
  end
  unrestricted
  symmetry nosym
  irrepoccupations
    A1/2 1 0 52
  end
  nuclearmodel gaussian
EndEngine
eor
done
```

Example: Constrained DFT: H2O+ … H2O

Download WaterDimer.run

```
#!/bin/sh


# Example of Constrained Density Functional Theory (CDFT) for the charged Water
# dimer, in which the charge is constrained to be on one of the waters. In this
# case there is one constraint, with a total charge of 1.0 on the atoms 1-3.

$AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000
     H        -0.9358409558        0.2646136961      0.0000000000
     H        -0.0304663436       -0.9828924420      0.0000000000
     O         0.0000000000       -2.9053396088      0.0000000000
     H        -0.4092227596       -3.3374838250     -0.7701260000
     H        -0.4092227596       -3.3374838250      0.7701260000
  end
  charge 1
end

Task SinglePoint

Engine ADF
  title water dimer h2o+ ... h2o

  cdft
    constraints 1.0
    natomsperset 3
    nconstraints 1
    theatoms 1 2 3
  end

  basis
    core none
    type TZP
  end

  unrestricted
  spinpolarization 1

  symmetry NOSYM

  xc
    gga PW91
  end

  noprint BAS FUNCTIONS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
EndEngine

eor
```

Example: ROKS: O2

Download O2_ROKS.run

```
#!/bin/sh

# Example restricted open shell Kohn-Sham (ROKS, ROSCF) for O2.

# The method implemented in ADF needs integer occupation numbers,
# and a positive spin polarization.
# In this case one can use symmetry, since the calculated occupation numbers
# will be integer, also for the orbitals in the degenerate irreps.
# Orbitals in degenerate irreps for a given spin need to be either fully occupied or␣
↪zero.
# For example, one of the orbitals in the PI.g irrep will have
# 2 occupied alpha electrons amd 0 beta electrons.

"$AMSBIN/ams" << eor
Task SinglePoint
System
    Atoms
        O   0.000 0.000 -0.604
        O   0.000 0.000  0.604
    End
End

Engine ADF
    Basis
        Type TZ2P
        Core None
    End
    Occupations IntegerAufbau
    Unrestricted Yes
    SpinPolarization 2
    Scf
        ROSCF
        End
    End
    NumericalQuality Good
    XC
        Hybrid B3LYP
    End
EndEngine
eor
```

Example: ROKS: SF

Download SF_ROKS.run

```
#!/bin/sh

# Example restricted open shell Kohn-Sham (ROKS, ROSCF) for SF.

# The method implemented in ADF needs integer occupation numbers,
# and a positive spin polarization.
# In this case one can not use full C(lin) symmetry, since one of
```

(continues on next page)

(continued from previous page)

```
# the orbitals in one of the degenerate representation might not
# be fully occupied or zero.
# If one would use C(lin) one of the orbitals in the degenerate
# irrep Pi would have 1 beta electron.
# This looks like an integer number, however, 1 beta electron in Pi
# effectively would lead to a fractional 0.5 electron in Pi:x and
# 0.5 electron in Pi:y beta orbital.
# Since this method needs integer occupations, a lower symmetry is
# used, in this case C(2v).
# Note that in this case one needs to use the key Occupations IntegerAufbau
# or explicit integer occupation numbers with the key IrrepOccupations,
# otherwise one could still end up with fractional occupation numbers.

"$AMSBIN/ams" << eor
Task SinglePoint
System
    Atoms
        S  0.0000 0.0000 0.0000
        F  0.0000 0.0000 1.6000
    End
End

Engine ADF
    Basis
        Type TZ2P
        Core None
    End
    SpinPolarization 1
    XC
        Hybrid B3LYP
    End
    Scf
        ROSCF
        End
    End
    symmetry C(2v)
    Unrestricted Yes
    Occupations IntegerAufbau
    NumericalQuality Good
    IrrepOccupations
       A1 7 // 7
       B1 3 // 3
       B2 3 // 2
    End
EndEngine
eor
```

14.4 Spectroscopic Properties

14.4.1 IR Frequencies, (resonance) Raman, VROA, VCD

Example: Numerical Frequencies: NH3

Download Freq_NH3.run

```
#! /bin/sh

# Summary:
# - Frequencies with symmetric displacements
# - Frequencies with Cartesian displacements
# - Isotope effects in the frequencies

# == Frequencies with symmetric displacements ==

# Computation of frequencies by symmetric displacements. The assumed equilibrium
# input structure should be given in Cartesian coordinates.

# The symmetry is determined automatically by the program as C(3v), from the
# input coordinates. During the calculation first symmetric atomic displacements
# are constructed. The number of such displacements in each irreducible
# representation corresponds to the number of frequencies with the corresponding
# symmetry. All displaced geometries within one representation have the same
# symmetry, which enables us to use it to speed up the computation
# significantly.

AMS_JOBNAME=NH3_symm $AMSBIN/ams <<eor
System
  atoms
     N  0.0000    0.0000    0.0000
     H  0.4729    0.8190    0.3821
     H -0.9457    0.0000    0.3821
     H  0.4729   -0.8190    0.3821
  end
  Symmetrize Yes
end

Task SinglePoint

Properties
  NormalModes Yes
End

NormalModes
  Displacements Symmetric
End

Thermo
  Temperatures 300:400:10
End

Engine ADF
  title NH3 frequencies in symmetric displacements
  BeckeGrid
```

(continues on next page)

(continued from previous page)

```
    quality good
  End
  Basis
    type TZP
    core Small
  End
EndEngine
eor


# == Frequencies with Cartesian displacements ==

# Computation of frequencies by Cartesian displacements. The assumed equilibrium
# input structure is given in internal coordinates.

AMS_JOBNAME=NH3_cartesian $AMSBIN/ams <<eor
System
  atoms
     N  0.0000    0.0000    0.0000
     H  0.4729    0.8190    0.3821
     H -0.9457    0.0000    0.3821
     H  0.4729   -0.8190    0.3821
  end
  Symmetrize Yes
end

Task SinglePoint

Properties
  NormalModes Yes
End

NormalModes
  Hessian Numerical
  Displacements Cartesian
End

Thermo
  Temperatures 300:400:10
End

Engine ADF
  title NH3 frequencies
  BeckeGrid
    quality good
  End
  Basis
    type TZP
    core Small
  End
EndEngine
eor


# The symmetry is determined automatically by the program as C(3v), from the
# input coordinates. In a Frequencies calculation the symmetry (specified on
# input or computed internally) is used for analysis and in some cases to speed
```

(continues on next page)

(continued from previous page)

```
# up the calculation.

# The equilibrium coordinate values are supplied as identifiers that are
# associated with values in the define block.

# The key thermo addresses the thermodynamical analysis (only available in a
# Frequencies calculation, otherwise ignored). The specification Temperatures␣
↪300:400:10
# means that the thermodynamic properties are printed for the temperature range
# 300-400K using 10 equidistant steps and for a pressure of 1.0 atmosphere
# (default).

# Frequencies calculations suffer easily from numerical inaccuracies. Therefore,
# the default numerical integration precision in a Frequencies calculation is
# higher than in an ordinary single-point or minimization run.


# == Isotope effects in the frequencies ==

# Calculate a different isotope of H, in this case deuterium.

AMS_JOBNAME=Isotope $AMSBIN/ams <<eor
System
  atoms
     N   0.0000    0.0000    0.0000
     H   0.4729    0.8190    0.3821 mass=2.014
     H  -0.9457    0.0000    0.3821 mass=2.014
     H   0.4729   -0.8190    0.3821 mass=2.014
  end
  Symmetrize Yes
end

Task SinglePoint

Properties
  NormalModes Yes
End

NormalModes
  Hessian Numerical
  Displacements Cartesian
End

Thermo
  Temperatures 300:400:10
End

Engine ADF
  title NH3 frequencies
  BeckeGrid
    quality good
  End
  Basis
    type TZP
    core Small
  End
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor
```

Example: Numerical Frequencies, spin-orbit coupled ZORA: UF6

Download Freq_UF6.run

```
#!/bin/sh

# Optimization and frequencies for scalar relativistic ZORA

AMS_JOBNAME=SR_GO $AMSBIN/ams <<eor
System
  atoms
     U  0.00000  0.00000  0.00000
     F  2.00000  0.00000  0.00000
     F -2.00000  0.00000  0.00000
     F  0.00000  2.00000  0.00000
     F  0.00000 -2.00000  0.00000
     F  0.00000  0.00000  2.00000
     F  0.00000  0.00000 -2.00000
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1E-4
End

Engine ADF
  title UF6 geometry optimization + freq: scalar ZORA
  beckegrid quality=good
  basis type=TZP
  relativity
    level scalar
    formalism ZORA
  end
EndEngine
eor


AMS_JOBNAME=SR_Freq $AMSBIN/ams <<eor
LoadSystem
  File SR_GO.results/adf.rkf
End

Task SinglePoint

Properties
  NormalModes Yes
End

LoadEngine SR_GO.results/adf.rkf
eor
```

(continues on next page)

(continued from previous page)

```
# # Optimization and frequencies for spinorbit ZORA

AMS_JOBNAME=SO_GO $AMSBIN/ams <<eor
System
  atoms
     U  0.00000  0.00000  0.00000
     F  2.00000  0.00000  0.00000
     F -2.00000  0.00000  0.00000
     F  0.00000  2.00000  0.00000
     F  0.00000 -2.00000  0.00000
     F  0.00000  0.00000  2.00000
     F  0.00000  0.00000 -2.00000
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence Gradients=1E-4
End

Engine ADF
  title UF6 geometry optimization + freq: spinorbit ZORA
  beckegrid quality=good
  basis type=TZP
  relativity
    level spin-orbit
    formalism ZORA
  end
EndEngine
eor


AMS_JOBNAME=SO_Freq $AMSBIN/ams <<eor
LoadSystem
  File SO_GO.results/adf.rkf
End

Task SinglePoint

Properties
  NormalModes Yes
End

NormalModes
  Displacements Symmetric
End

LoadEngine SO_GO.results/adf.rkf
eor
```

Example: Numerical Frequencies, accurate Hartree-Fock: H2O

Download H2O_HF_freq.run

```
#!/bin/sh

# Example shows a Hartree-Fock frequency calculation with an accurate basis set.
# Note that numerical issues may be present in ADF with large basis sets icw
# Hartree-Fock or (range-separated) (meta-)hybrids.
# An RIHartreeFock  VeryGood fit set is recommended in those cases.

# First a geometry optimization is performed.


AMS_JOBNAME=GO $AMSBIN/ams <<eor
System
  atoms
     O          0.000000    0.000000   -0.007124
     H          0.000000    0.751933    0.556531
     H          0.000000   -0.751933    0.556531
  end
end

Task GeometryOptimization

Engine ADF
  title accurate HF geometry optimization with large QZ4P basis set
  basis
    type QZ4P
    core None
  end
  numericalquality Good
  rihartreefock
    quality verygood
  end
  xc
    hartreefock
  end
EndEngine
eor


# Next the frequency calculation is done.


AMS_JOBNAME=Freq $AMSBIN/ams <<eor

LoadSystem
  File GO.results/adf.rkf
End

Task SinglePoint

Properties
  NormalModes Yes
End

Engine ADF
```

(continues on next page)

(continued from previous page)

```
  title accurate HF frequency calculation with large QZ4P basis set
  basis
    type QZ4P
    core None
  end
  numericalquality Good
  rihartreefock
    quality verygood
  end
  xc
    hartreefock
  end
EndEngine
eor
```

Example: Analytic Frequencies: CN

Download CN_SecDeriv.run

```
#! /bin/sh


# Calculation of normal modes is requested by specifying in the AMS input:
# Properties
#   NormalModes Yes
# End
# ADF will compute the Hessian analytically if possible.
# If the Hessian cannot be computed analytically, numerical differentiation
# will automatically be used.

# A good quality is specified for the numerical Becke integration to be sure of
# reliable results. In general, it seems advisable to use high accuracy for
# heavy nuclei at the moment, whereas default integration accuracy is usually
# sufficient for light atoms. The precision of the fit may be improved with the
# ZlmFit block keyword.


$AMSBIN/ams <<eor
System
  Symmetrize
  atoms
     N  -1.3  0.0   0.0
     C   0.0  0.0   0.0
  end
  charge -1
end

Task SinglePoint

Properties
  NormalModes Yes
End

Engine ADF
  title CN
```

(continues on next page)

(continued from previous page)

```
  beckegrid
    quality good
  end
  basis
    type DZ
    core None
    CreateOutput Yes
  end
  xc
    lda Xonly
  end
EndEngine

eor
```

Example: Analytic Frequencies: CH4

Download CH4_SecDeriv.run

```
#! /bin/sh


# In this example we perform a geometry optimization immediately followed by
# calculation of frequencies. This is done by setting the task to
# GeometryOptimization and by requesting the normal modes in Properties block

# Note: when using this feature, one might improve the precision if one uses
# NumericalQuality Good. In order to save time this is not done in this example.

$AMSBIN/ams <<eor
System
  atoms
     C  0.0                 0.0                 0.0
     H  0.6264250420707439  0.6264250420707439  0.6264250420707439
     H  0.6264250420707439 -0.6264250420707439 -0.6264250420707439
     H -0.6264250420707439  0.6264250420707439 -0.6264250420707439
     H -0.6264250420707439 -0.6264250420707439  0.6264250420707439
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    Gradients 1E-4
  End
End

Properties
  NormalModes Yes
End

Engine ADF
  basis
    type TZP
```

(continues on next page)

(continued from previous page)

```
    core None
  end
EndEngine
eor
```

Example: Analytic Frequencies: HI

Download HI_SecDer_ZORA.run

```
#!/bin/sh


# Analytical frequencies of HI using the scalar ZORA relativistic approximation.
# A good NumericalQuality is used.


$AMSBIN/ams <<eor
System
  atoms
     I 0.0 0.0 0.0
     H 0.0 0.0 1.6090
  end
end

Task SinglePoint

Properties
  NormalModes Yes
End

Engine ADF
  title HI scalar, ZORA,
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  xc
    lda Xonly
  end
EndEngine
eor
```

Example: Mobile Block Hessian (MBH): Ethanol

Download MBH_Ethanol.run

```
#! /bin/sh


# A frequency calculation is performed using the mobile block Hessian (MBH)
# method. The coordinates in the ATOMS section can be the partially optimized
# coordinates (or the fully optimized coordinates would work too).


AMS_JOBNAME=Ethanol_MBH $AMSBIN/ams <<eor
System
  atoms
     C      -0.029587   -0.006554    0.008124
     H      -0.087498   -0.025163    1.109913
     H       1.027473   -0.056237   -0.302751
     H      -0.565305   -0.891154   -0.376242
     C      -0.694908    1.238909   -0.501807
     H      -0.670258    1.265092   -1.608847
     O      -2.069894    1.175059   -0.017251
     H      -0.182335    2.138977   -0.109315
     H      -2.586972    1.972802   -0.317216
  end
end

Task SinglePoint
Properties
    NormalModes True
End
NormalModes
    Displacements Block
    BlockDisplacements
        BlockAtoms 1 2 3 4
    End
End


Engine ADF
  title ethanol: second derivatives with MBH approach. CH3 is treated as a rigid block
  numericalquality good
  scf
    converge 1.0e-8
  end
  symmetry nosym
EndEngine
eor


# For comparison in this example also a calculation is performed without any
# restrictions.


AMS_JOBNAME=Ethanol_full $AMSBIN/ams <<eor
System
  atoms
     C      -0.029587   -0.006554    0.008124
```

(continues on next page)

(continued from previous page)

```
     H      -0.087498   -0.025163    1.109913
     H       1.027473   -0.056237   -0.302751
     H      -0.565305   -0.891154   -0.376242
     C      -0.694908    1.238909   -0.501807
     H      -0.670258    1.265092   -1.608847
     O      -2.069894    1.175059   -0.017251
     H      -0.182335    2.138977   -0.109315
     H      -2.586972    1.972802   -0.317216
  end
end

Task SinglePoint
Properties
    NormalModes True
End

Engine ADF
  title ethanol: complete vibrational spectrum, compare with MBH above
  numericalquality good
EndEngine
eor
```

Example: Mobile Block Hessian: CH4

Download MBH_CH4.run

```
#! /bin/sh

# A frequency calculation is performed using the mobile block Hessian
# (MBH)method. The coordinates in the ATOMS section should be the partially
# optimized coordinates (or the fully optimized coordinates would work too).

# Such geometry optimization will not be discussed here any further. The next
# input for ADF shows how to perform a frequency calculation with MBH.


$AMSBIN/ams <<eor
System
  atoms
     C         0.000000    0.000000    0.000000
     H         0.634671    0.634671    0.634671
     H        -0.634671   -0.634671    0.634671
     H        -0.634671    0.634671   -0.634671
     H         0.634671   -0.634671   -0.634671
  end
end

Task SinglePoint
Properties
    NormalModes True
End

NormalModes
  Displacements Block
  BlockDisplacements
```

(continues on next page)

(continued from previous page)

```
    BlockAtoms 1 2 3 4
    RadialDisplacement 0.001
  End
End

Engine ADF
  title Methane
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  numericalquality verygood
  symmetry nosym
EndEngine
eor

# The quality of the frequencies/modes depends largely on the block choice. Best
# results are obtained when grouping atoms in a block if those atoms are known
# to form rather rigid structures. For instance, grouping the 11 atoms of
# benzene side group into a block, will usually result in representative
# frequencies. In this example the block choice is only illustrative for the
# methodology.
```

Example: Raman: NH3

Download Freq_NH3_RAMAN.run

```
#! /bin/sh

# Summary:
# - Analytical frequencies with subsequent calculation of Raman intensities in a
#   range
# - Numerical frequencies including all Raman intensities


# == Raman Intensities for Selected Frequencies ==

# The VibrationalAnalysis task (type ModeScanning) can be used to calculate Raman
# intensities for a range of frequencies only.
# The Raman intensities are calculated by numerical differentiation of the
# polarizability tensor. Only frequencies frequencies withing the interval
# that are known to be Raman-active will be included.


AMS_JOBNAME=NH3 $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N       0.0000    0.0000    0.0000
     H       0.4729    0.8190    0.3821
     H      -0.9457    0.0000    0.3821
     H       0.4729   -0.8190    0.3821
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint
Properties
  NormalModes True
End
thermo
  temperatures 300:400:10
end
Engine ADF
  title NH3 frequencies
  beckegrid
    quality good
  end
  basis
    core Small
    type TZP
  end
EndEngine
eor

AMS_JOBNAME=RamanRange1 $AMSBIN/ams <<eor
LoadSystem
  File NH3.results/ams.rkf
End
Task VibrationalAnalysis
VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile NH3.results/adf.rkf
      ModeSelect
         FreqRange 0.0 2000.0
      End
   End
End
Properties
   Raman True
End
thermo
  temperatures 300:400:10
end

EngineRestart NH3.results/adf.rkf

Engine ADF
  title NH3 Raman intensities in the range 0-2000 cm-1
  beckegrid
    quality good
  end
  basis
    core Small
    type TZP
  end
  responseformalism response
EndEngine
eor

AMS_JOBNAME=RamanRange2 $AMSBIN/ams <<eor
LoadSystem
```

(continues on next page)

(continued from previous page)

```
  File NH3.results/ams.rkf
End

Task VibrationalAnalysis

VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile NH3.results/adf.rkf
      ModeSelect
         FreqRange 2000.0 5000.0
      End
   End
End

Properties
   Raman True
End

EngineRestart NH3.results/adf.rkf

Engine ADF
  title NH3 Raman intensities in the range 2000-5000 cm-1
  beckegrid
    quality good
  end
  basis
    core Small
    type TZP
  end
  responseformalism response
EndEngine
eor

# == Raman Intensities for All Frequencies ==

# Raman scattering intensities and depolarization ratios for all molecular
# vibrations at a certain laser frequency can be calculated in a single run.

# In this example the static Raman scattering is calculated (omega = 0). This
# type of calculation is very similar to an IR intensity calculation. In fact,
# all IR output is automatically generated as well. At all distorted geometries
# the dipole polarizability tensor is calculated. This is very time-consuming
# and is only feasible for small molecules.


AMS_JOBNAME=Raman $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N       0.0000    0.0000    0.0000
     H       0.4729    0.8190    0.3821
     H      -0.9457    0.0000    0.3821
     H       0.4729   -0.8190    0.3821
  end
end
Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Properties
   NormalModes True
   Raman True
End
thermo
  temperatures 300:400:10
end

Engine ADF
  title NH3 frequencies with Raman intensities
  beckegrid
    quality good
  end
  basis
    core Small
    type TZP
  end
  response
    raman
  end
EndEngine
eor
```

Example: Raman: HI

Download HI_Raman.run

```
#!/bin/sh

# Example shows a calculation of the Raman spectrum of HI
# with a scalar relativistic ZORA and a spin-orbit coupled ZORA Hamiltonian.
# Note that used basis set is too small to get accurate results.

# == scalar relativistic ZORA ==

# 3 calculations are performed.
# First a scalar relativistic geometry optimization is performed
# and analytical frequencies are calculated. This calculation produces a restart file,
# which is used in the second and third alculation as input for the Raman calculation.
# The FreqRange keyword is used to calculate Raman intensities for a range of␣
↪frequencies only.
# In the second calculation RESPONSE is used and in the third calculation AORESPONSE.

AMS_JOBNAME=SR $AMSBIN/ams <<eor
System
  atoms
     H  0.0000 0.0000 0.0000
     I  0.0000 0.0000 1.6092
  end
end

Task GeometryOptimization
Properties
  NormalModes Yes
End
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  basis
    core None
    type DZ
  end
  noprint SFO
  relativity
    level scalar
    formalism ZORA
  end
EndEngine
eor

AMS_JOBNAME=SR_response $AMSBIN/ams <<eor
Task VibrationalAnalysis

VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile SR.results/adf.rkf
      ModeSelect
         FreqRange 0.0 4000.0
      End
   End
End

Properties
   Raman True
End
Raman
   IncidentFrequency 2.0 [eV]
End

LoadSystem
   File SR.results/ams.rkf
End

EngineRestart SR.results/adf.rkf

Engine ADF
  responseformalism response
  basis
    core None
    type DZ
  end
  noprint SFO
  relativity
    level scalar
    formalism ZORA
  end
EndEngine
eor

AMS_JOBNAME=SR_aoresponse $AMSBIN/ams <<eor
Task VibrationalAnalysis
```

(continues on next page)

(continued from previous page)

```
VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile SR.results/adf.rkf
      ModeSelect
         FreqRange 0.0 4000.0
      End
   End
End

Properties
   Raman True
End
Raman
   IncidentFrequency 2.0 [eV]
End

LoadSystem
   File SR.results/ams.rkf
End

EngineRestart SR.results/adf.rkf

Engine ADF
  responseformalism aoresponse
  basis
    core None
    type DZ
  end
  noprint SFO
  relativity
    level scalar
    formalism ZORA
  end
EndEngine
eor

# == Relativistic Spinorbit ZORA ==

# 4 calculations are performed.
# First a spin-orbit coupled relativistic geometry optimization is performed.
# This calculation produces a restart file which is used in the second calculation
# to calculate numerical frequencies. This calculation produces a restart file
# which is used in the third and fourth calculation as input for the Raman␣
↪calculation.
# The FreqRange keyword is used to calculate Raman intensities for a range of␣
↪frequencies only.
# In the third calculation RESPONSE is used and in the fourth calculation AORESPONSE.
# Symmetry NOSYM is needed for AORESPONSE icw spin-orbit coupling.

AMS_JOBNAME=SO $AMSBIN/ams <<eor
System
  atoms
     H  0.0000 0.0000 0.0000
     I  0.0000 0.0000 1.6092
  end
end
```

(continues on next page)

(continued from previous page)

```
Task GeometryOptimization
Properties
  NormalModes Yes
End

Engine ADF
  basis
    core None
    type DZ
  end
  noprint SFO
  relativity
    level spin-orbit
    formalism ZORA
  end
EndEngine
eor

AMS_JOBNAME=SO_response $AMSBIN/ams <<eor
Task VibrationalAnalysis

VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile SO.results/adf.rkf
      ModeSelect
         FreqRange 0.0 4000.0
      End
   End
End

Properties
   Raman True
End
Raman
   IncidentFrequency 2.0 [eV]
End

LoadSystem
   File SO.results/ams.rkf
End

EngineRestart SO.results/adf.rkf

Engine ADF
  responseformalism response
  basis
    core None
    type DZ
  end
  noprint SFO
  relativity
    level spin-orbit
    formalism ZORA
  end
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

AMS_JOBNAME=SO_aoresponse $AMSBIN/ams <<eor
Task VibrationalAnalysis

VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile SO.results/adf.rkf
      ModeSelect
         FreqRange 0.0 4000.0
      End
   End
End

Properties
   Raman True
End
Raman
   IncidentFrequency 2.0 [eV]
End

LoadSystem
   File SO.results/ams.rkf
End

EngineRestart SO.results/adf.rkf

Engine ADF
  responseformalism aoresponse
  basis
    core None
    type DZ
  end
  noprint SFO
  relativity
    level spin-orbit
    formalism ZORA
  end
  symmetry nosym
EndEngine
eor
```

Example: Resonance Raman, excited state finite lifetime: HF

Download HF_ResonanceRaman.run

```
#! /bin/sh

# Example shows a calculation of the Resonance Raman spectrum (RRS) of HF. In
# this example the RRS is calculated from the geometrical derivatives of the
# frequency-dependent polarizability, including a finite lifetime.

# In the AMS input one then needs to request 'Raman' in the 'Properties' block
# and specify 'IncidentFrequency' and 'LifeTime' in the Raman Block
```

(continues on next page)

(continued from previous page)

```
# Note that used basis set is too small to get accurate results.

$AMSBIN/ams <<eor
System
  atoms
     H 0.0000 0.0000 0.0000
     F 0.0000 0.0000 0.9170
  end
end

Task SinglePoint
Properties
   Raman True
End
NormalModes
  Displacements Symmetric
  SymmetricDisplacements
     Type InfraredAndRaman
  End
End
Raman
  IncidentFrequency 0.52362 [Hartree]
  LifeTime 0.0034
  FreqRange 0.0 5000.0
End

Engine ADF
  title HF ao-raman
  basis
    Type DZP
    Core None
    CreateOutput Yes
  end
  noprint SFO
  numericalquality Good
  Relativity
    Level None
  End
EndEngine
eor
```

Example: Vibrational Raman optical activity (VROA): H2O2

`Download VROA.run`

```
#! /bin/sh

# Example showing a calculation of normal Vibrational Raman Optical Activity (VROA).

# For accuracy reasons 'NumericalQuality Good' and EXACTDENSITY are used.

$AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     O        -0.750254   -0.034490    0.015133
     O         0.750254    0.034490    0.015133
     H        -0.943532    0.744006    0.580040
     H         0.943532   -0.744006    0.580040
  end
end

Task SinglePoint

Properties
  VROA True
End

Raman
  IncidentFrequency 2.41 [eV]
End

NumericalDifferentiation
   Parallel nGroups=1
End

Engine ADF
  title VROA TEST H2O2

  numericalquality Good

  exactdensity

  basis
    type TZP
    core None
  end

  xc
    gga BLYP
  end
EndEngine

eor

# Note that used basis set is too small to get accurate results. Better is to
# use larger basis sets for such a small molecule, like one of the even
# tempered basis sets (for example Type ET/ET-QZ3P-1DIFFUSE), or use augmented
# basis sets (for example Type AUG/ATZ2P).
```

Example: Resonance VROA: H2O2

Download VROA_RESO.run

```
#! /bin/sh

# Example shows a calculation of resonance vibrational Raman optical activity
# (resonance VROA).

# For accuracy reasons 'NumericalQuality Good' and EXACTDENSITY are used.
```

(continues on next page)

(continued from previous page)

```
$AMSBIN/ams <<eor
System
  atoms
     O        -0.750254   -0.034490    0.015133
     O         0.750254    0.034490    0.015133
     H        -0.943532    0.744006    0.580040
     H         0.943532   -0.744006    0.580040
  End
End

UseSymmetry NO

Task SinglePoint

Properties
  VROA True
End

Raman
  IncidentFrequency 5.15462 [eV]
  Lifetime 0.0037
End

Engine ADF
  Title VROA-RESO TEST H2O2

  NumericalQuality Good

  ExactDensity

  Basis
    Core None
    Type TZP
  End

  SCF
    Converge 1.0e-10
  End

  XC
    GGA BLYP
  End
EndEngine

eor

# Note that used basis set is too small to get accurate results. Better is to
# use larger basis sets for such a small molecule, like one of the even
# tempered basis sets (for example Type ET/ET-QZ3P-1DIFFUSE), or use augmented
# basis sets (for example Type AUG/ATZ2P).
```

Example: Raman and VROA for approximate modes

*Download DFTB_modes_ADF_Raman_VROA.run <../../../examples/adf/DFTB_modes_ADF_Raman_VROA/DFTB_modes_ADF_Raman_*

```
#! /bin/sh

# Example to calculate vibrational modes with an approximate method,
# in this case DFTB (using GFN1-xTB)
# Next scan selected modes with ADF to calculate (resonance) Raman and VROA.

AMS_JOBNAME=dftb $AMSBIN/ams <<eor
System
  atoms
     O        -0.750254   -0.034490    0.015133
     O         0.750254    0.034490    0.015133
     H        -0.943532    0.744006    0.580040
     H         0.943532   -0.744006    0.580040
  end
end
Task SinglePoint
Properties
  NormalModes True
End
Engine DFTB
   Model GFN1-xTB
EndEngine
eor

AMS_JOBNAME=offresonance $AMSBIN/ams <<eor
LoadSystem
   file dftb.results/ams.rkf
End
Task VibrationalAnalysis
VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile dftb.results/dftb.rkf
      ModeSelect
         FreqRange 0.0 1000.0
      End
   End
End
Properties
  Raman True
  VROA True
End
Raman
  IncidentFrequency 2.41 [eV]
End
Engine ADF
  basis
    type TZP
    core None
  end
  xc
    gga BLYP
  end
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

AMS_JOBNAME=resonance $AMSBIN/ams <<eor
LoadSystem
   file dftb.results/ams.rkf
End
Task VibrationalAnalysis
VibrationalAnalysis
   Type ModeScanning
   NormalModes
      ModeFile dftb.results/dftb.rkf
      ModeSelect
         FreqRange 0.0 1000.0
      End
   End
End
Properties
  Raman True
  VROA True
End
Raman
  IncidentFrequency 5.15462 [eV]
  Lifetime 0.0037
End
Engine ADF
  basis
    type TZP
    core None
  end
  xc
    gga BLYP
  end
EndEngine
eor
```

Example: Vibrational Circular Dichroism (VCD): NHDT

Download VCD_COG_NHDT.run

```
#! /bin/sh


# Analytical frequencies with subsequent calculation of vibrational circular
# dichroism (VCD)

# Recommended is use to use high accuracy for the geometry optimization which
# one needs to do before the frequency calculation. This simple example is an
# NHDT molecule, which is NH3 where one hydrogen atom is replaced with deuterium
# and another with tritium.

$AMSBIN/ams <<eor
System
  atoms
     N       0.000000    0.000000    0.010272
     H      -0.471582   -0.816803    0.407861
```

(continues on next page)

(continued from previous page)

```
     H       0.943163    0.000000    0.407861 mass=2.014101778
     H      -0.471582    0.816803    0.407861 mass=3.01604927
  end
end

Task SinglePoint

Properties
  VCD Yes
End

Engine ADF
  basis
    core None
    type TZP
  end
  xc
    gga BP86
  end
  beckegrid
    quality verygood
  end
EndEngine

eor
```

Example: unrestricted VCD: CHFClBr

Download VCD_CHFClBr_DZP.run

```
#! /bin/sh

# Analytical frequencies with subsequent calculation of unrestricted
# vibrational circular dichroism (VCD)

# This test is using a small basis set; use of a large basis set for
# chemical application is recommended

$AMSBIN/ams <<eor

Task SinglePoint

System
   Atoms
       C        -1.194957    0.619864    0.018398
       Br        0.712857    0.534545    0.755317
       Cl       -0.705615   -0.120328   -1.557332
       F        -1.636918    1.848290   -0.093005
       H        -1.832068   -0.042181    0.608806
   End
   Charge 1
End

Properties
   VCD Yes
```

(continues on next page)

(continued from previous page)

```
End

Engine ADF
   Basis
       Type DZP
       Core None
   End
   XC
     GGA BLYP
   end
   Unrestricted Yes
   SpinPolarization 1
   LinearScaling
     Overlap_Int 98
   End
EndEngine
eor
```

14.4.2 Excitation energies: UV/Vis spectra, X-ray absorption, CD, MCD

Example: Excitation energies and polarizability: Au2

Download Au2_Resp.run

```
#! /bin/sh


# A calculation of response properties of the Au2 dimer, with ZORA relativistic
# corrections

# In the response module infinite symmetries cannot be handled (see the User's
# Guide), so we specify a lower subgroup in the input file, here D(8h).

# In this sample run the LB94 potential is used. The implementation implies that
# the XC potential is evaluated from the exact charge density, rather than the
# cheaper and faster fitted density (as is the case for other XC functionals).
# This means that the computation times are longer. In a small molecule like Au2
# this hardly shows, but in larger molecules the differences may be more
# significant. Note that the LB94 is a model potential, thus the calculated bond
# energy has not so much meaning. If the BASIS key is specified and the model
# potential LB94 is used, in the create run of the atoms the BP86 functional is used.

# Excitation energies are computed, in principle the lowest 10 in each irrep of
# the symmetry (see the User's Guide for more details).

$AMSBIN/ams <<eor
System
  atoms
     Au   0.0 0.0 1.236
     Au   0.0 0.0 -1.236
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  title Au2, Response Properties
  excitations
    lowest 10
    tolerance 1e-10
  end
  basis
    PerAtomType Symbol=Au File=ZORA/DZ/Au.4d
  end
  response
    allcomponents
  end
  symmetry D(8h)
  xc
    gga LB94
  end
EndEngine

eor
```

Example: Excitation energies open shell molecule: CN

Download CN_unr_exci.run

```
#!/bin/sh


# Calculation of the excitation energies of the open shell molecule CN

# In this example, the lowest 20 excitation energies of CN are calculated in a
# spin-unrestricted TDDFT calculation. In the MO -> MO transitions part for the
# excitations of the output file, the spin of each molecular orbitals are also
# specified to help assign the spin state of the excited states. The transitions
# are always from alpha spin-orbital to alpha spin-orbital or from beta spin-
# orbital to beta spin-orbital.

AMS_JOBNAME=exci $AMSBIN/ams <<eor
System
  atoms
     C  0.0000  0.0000  0.0000
     N  0.0000  0.0000  1.1718
  end
end

Task SinglePoint

Engine ADF
  title excitation energies of CN
  excitations
    lowest 20
    nto
    sfoanalysis
  end
  basis
```

(continues on next page)

(continued from previous page)

```
    type AUG/ADZP
  end
  spinpolarization 1
  unrestricted
EndEngine

eor


# Next the same example for CN is given with the Tamm-Dancoff approximation
# (TDA) approximation (including TDA in the input). Due to this approximation
# the calculated excitation energies will not be exactly the same as in the
# first example.

AMS_JOBNAME=exci_tda $AMSBIN/ams <<eor
System
  atoms
     C  0.0000  0.0000  0.0000
     N  0.0000  0.0000  1.1718
  end
end

Task SinglePoint

Engine ADF
  title excitation energies (TDA) of CN
  excitations
    lowest 20
    nto
    sfoanalysis
  end
  basis
    type AUG/ADZP
  end
  spinpolarization 1
  tda
  unrestricted
EndEngine

eor


# Note that the basis set is still far from complete, the ATZ2P is better. For
# accuracy reasons one may need to increase the numerical quality. If one uses
# the augmented basis sets for accuracy reasons one could use NumericalQuality
# Good.

# The third calculation is the calculation of spin-flip excitation energies for
# CN. Again these energies will not be exactly the same as in the first example.
# For open-shell molecules, spin-flip transition can result in transition to the
# ground state with a different S_z value, while the symmetry of the transition
# density is A1 (Sigma+ for linear molecules). The excitation energy of this
# transition should be zero and this can be used to test the reliability of
# spin-flip TDDFT. Indeed the calculation of the spin-flip excitation energies
# of CN shows one value which is close to zero and has a transition density of
# Sigma+ symmetry.
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=exci_tda_spinflip $AMSBIN/ams <<eor
System
  atoms
     C  .0000  .0000  .0000
     N  .0000  .0000 1.1718
  end
end

Task SinglePoint

Engine ADF
  title spin-flip excitation energies (TDA) of CN
  excitations
    lowest 20
    nto
    sfoanalysis
  end
  basis
    type AUG/ADZP
  end
  sftddft
  spinpolarization 1
  tda
  unrestricted
EndEngine

eor
```

Example: Spin-flip excitation energies: SiH2

Download `SiH2_spinflip.run`

```
#!/bin/sh


# Calculation of the spin-flip excitation energies of the open shell molecule SiH2

AMS_JOBNAME=SpinFlip $AMSBIN/ams <<eor
System
  atoms
     Si 0.0  0.0                0.0
     H  0.0 -1.095664024353027 -1.045576810836792
     H  0.0  1.095664024353027 -1.045576810836792
  end
end

Task SinglePoint

Engine ADF
  title spin-flip excitation energies of SiH2
  excitations
    lowest 20
  end
  forcealda
```

(continues on next page)

(continued from previous page)

```
  basis
    type TZ2P
  end
  sftddft
  spinpolarization 2
  tda
  unrestricted
EndEngine
eor

# In this example, the lowest 20 spin-flip excitation energies of SiH2 are
# calculated in a spin-unrestricted TDDFT calculation.

# In this case an excited state is used as reference, which means that there can
# also be a negative excitation energy, which is indeed the case. The electron
# configuration used in the SCF is (a1)^1 (b1)^1, with Sz=1, thus a ^3B_1
# state, which is an excited state. The ^1A_1 state with electron configuration
# (a1)^2 is lower in energy, and is the ground state.

# There is also an excited 1A1 state with electron configuration (b1)^2. The
# transition from the ground 1A1 state to the excited 1A1 state is an excitation
# from the electron configuration (a1)^2 to (b1)^2. This transition is actually
# a double excitation, which means that some double excitations can be reached
# using spin-flip TDDFT with carefully selected reference states.

# In the MO -> MO transitions part for the excitations of the output file, the
# spin of each molecular orbitals are also specified to help assign the spin
# state of the excited states. Note that in these spin-flip calculations the
# transitions are always from alpha spin-orbital to beta or from beta spin-
# orbital to alpha spin-orbital.

# For open-shell molecules, spin-flip transition can result in transition to the
# ground state with a different Sz value, while the symmetry of the transition
# density is A1. The excitation energy of this transition should be zero and
# this can be used to test the reliability of spin-flip TDDFT. Indeed the
# calculation of the spin-flip excitation energies of SiH2 shows one value which
# is close to zero and has a transition density of A1 symmetry.

# The 1A1 state with electron configuration (a1)^2 can also be used in the
# calculation of the excitation energies. This is a closed shell configuration,
# in which case we do not need the spin-flip method.


AMS_JOBNAME=Excitation $AMSBIN/ams <<eor
System
  atoms
     Si 0.0  0.0                0.0
     H  0.0 -1.095664024353027 -1.045576810836792
     H  0.0  1.095664024353027 -1.045576810836792
  end
end

Task SinglePoint

Engine ADF
  title excitation energies of SiH2
  excitations
```

(continues on next page)

(continued from previous page)

```
    lowest 20
  end
  basis
    type TZ2P
  end
EndEngine

eor

# The transition from the ground ^1A_1 state to the excited ^1A_1 state, which
# is an excitation from the electron configuration (a1)^2 to (b1)^2, can not be
# reached in this calculation, since it has mainly double excitation character.
# Of course, other excited ^1A_1 states can be reached.
```

Example: TDHF excitation energies: N2

Download N2_TDHF.run

```
#! /bin/sh


# Calculation of the excitation energies of N2 using time-dependent Hartree-Fock
# (TDHF). It also shows the possibility to use the Tamm-Dancoff approximation
# (TDA). This examples consists of 4 calculations:

# TDHF non-relativistic and spin-orbit coupled
# TDA non-relativistic and spin-orbit coupled (CIS)


# == TDHF non-relativistic ==


AMS_JOBNAME=TDHF_NR $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N 0 0 0
     N 0 0 1.0977
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality Good
  end
  excitations
    lowest 5
  end
  basis
    core None
    type DZ
  end
  xc
```

(continues on next page)

(continued from previous page)

```
    hartreefock
  end
  Relativity
    Level None
  End
EndEngine
eor



# == TDHF spin-orbit coupled ==


AMS_JOBNAME=TDHF_SO $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N 0 0 0
     N 0 0 1.0977
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality Good
  end
  excitations
    lowest 20
  end
  basis
    core None
    type DZ
  end
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    hartreefock
  end
EndEngine
eor



# == TDA non-relativistic ==


AMS_JOBNAME=TDA_NR $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N 0 0 0
     N 0 0 1.0977
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  beckegrid
    quality Good
  end
  excitations
    lowest 5
  end
  basis
    core None
    type DZ
  end
  tda
  xc
    hartreefock
  end
  Relativity
    Level None
  End
EndEngine
eor



# == TDA spin-orbit coupled ==


AMS_JOBNAME=TDA_SO $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N 0 0 0
     N 0 0 1.0977
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality Good
  end
  excitations
    lowest 20
  end
  basis
    core None
    type DZ
  end
  relativity
    level spin-orbit
    formalism ZORA
  end
  tda
```

(continues on next page)

(continued from previous page)

```
  xc
    hartreefock
  end
EndEngine
eor
```

Example: excitation energies CAM-B3LYP: Pyridine

Download Pyridine_CAM-B3LYP.run

```
#! /bin/sh


# Calculation of the excitation energies of Pyridine using the range-separated
# functional CAM-B3LYP. To use this range-separated functional CAM-B3LYP LibXC is␣
↪needed.
# Note that CAM-B3LYP is not the same as CAMY-B3LYP.
# For the LibXC range separated functionals, like CAM-B3LYP, the kernel consists
# of range separated ALDA plus the kernel of the range separated exact exchange part.

$AMSBIN/ams <<eor
System
  atoms
     N       0.00000000       0.00000000       1.07112000
     C       0.00000000       0.00000000       3.88763000
     C       0.00000000      -1.14432000       1.77593000
     C       0.00000000       1.14432000       1.77593000
     C       0.00000000       1.19947000       3.17299000
     C       0.00000000      -1.19947000       3.17299000
     H       0.00000000      -2.16297000       3.68508000
     H       0.00000000      -2.06734000       1.18970000
     H       0.00000000       0.00000000       4.97897000
     H       0.00000000       2.16297000       3.68508000
     H       0.00000000       2.06734000       1.18970000
  end
end

Task SinglePoint

Engine ADF
  excitations
    onlysing
  end
  basis
    type DZP
    core None
  end
  xc
    libxc CAM-B3LYP
  end
EndEngine

eor
```

Example: CAMY-B3LYP excitation energies: H2O

Download CAMY-B3LYP_TDDFT_H2O.run

```
#! /bin/sh


# Example for the range separated functional CAMY-B3LYP for the calculation of
# excitation energies of water. Note that CAMY-B3LYP is not the same as CAM-
# B3LYP.


$AMSBIN/ams <<eor
   Task SinglePoint
   System
      ATOMS
         O     0.000000    0.000000    0.000000
         H     0.000000    0.500000   -0.6
         H     0.000000   -0.500000   -0.6
      END
   END
   Engine ADF
      TITLE  H2O excitations with CAMY-B3LYP functional
      NOPRINT BAS FIT FUNCTIONS SFO
      XC
         HYBRID CAMY-B3LYP
         xcfun
         RANGESEP GAMMA=0.34
      END
      BeckeGrid
         Quality good
      End
      EXCITATIONS
         LOWEST 10
         ONLYSING
      END
      basis
         Type DZP
      END
   EndEngine
eor
```

Example: Full XC kernel in excitation energy calculation: H2O+

Download XCFUN_FULLKERNEL_TDDFT_H2O.run

```
#! /bin/sh

# Example of using the full XC kernel in the calculation of excitation energies,
# in this case for the BP86 functional. Note that SYMMETRY NOSYM is required and
# XCFUN should be added to the XC block key. In this case spin-flip excitations
# are calculated.

$AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     O         0.000000    0.000000    0.000631
     H         0.000000    0.771769   -0.605006
     H         0.000000   -0.771769   -0.605006
  end
  charge 1
end

Task SinglePoint

Engine ADF
  title H2O+ spin-flip TDDFT with BP and the non-ALDA kernel

  basis
    type DZP
  end

  xc
    gga Becke Perdew
    xcfun
  end

  beckegrid
    quality good
  end

  excitations
    fullkernel
    lowest 5
  end

  sftddft
  tda

  unrestricted
  spinpolarization 1

  symmetry NOSYM

  noprint BAS FIT FUNCTIONS SFO
EndEngine

eor
```

Example: Use of xcfun in excitation energy calculations: H2O

Download XCFUN_TDDFT_H2O.run

```
#! /bin/sh

# Example for the use of the xcfun library for a calculation of excitation
# energies, in this case for the BP86 functional. Note that for closed shell
# molecules only singlet-singlet can be calculated with xcfun, therefore the
# key 'onlysing' in the 'excitations' block the  is used.

$AMSBIN/ams <<eor
```

(continues on next page)

(continued from previous page)

```
System
  atoms
     O         0.000000    0.000000    0.000631
     H         0.000000    0.771769   -0.605006
     H         0.000000   -0.771769   -0.605006
  end
end

Task SinglePoint

Engine ADF
  title H2O TDDFT with XCFUN
  beckegrid
    quality good
  end
  excitations
    lowest 5
    onlysing
  end
  basis
    type DZP
  end
  noprint BAS FIT FUNCTIONS SFO
  xc
    gga Becke Perdew
    xcfun
  end
EndEngine

eor
```

Example: Core excitation energies: TiCl4

Download TiCl4_CoreExci.run

```
#! /bin/sh


# Calculation of the 2p Ti and 2p Cl core excitation energies of TiCl4

# The state selective method (key SELECTEXCITATION) can be used to calculate
# core excitation energies. The use of the key SELECTEXCITATION is similar as
# the use of the key MODIFYEXCITATION. However, the key SELECTEXCITATION can not
# be used in combination with the key MODIFYEXCITATION. In the state selective
# method (key SELECTEXCITATION) the one-electron excited state configuration
# space remains complete, whereas it is reduced in case the scheme with the
# MODIFYEXCITATION key.

AMS_JOBNAME=SOT $AMSBIN/ams <<eor
System
  atoms [Bohr]
     Ti 0.       0.         0.
     Cl    2.36754     2.36754     2.36754
     Cl   -2.36754    -2.36754     2.36754
     Cl    2.36754    -2.36754    -2.36754
```

(continues on next page)

(continued from previous page)

```
     Cl   -2.36754     2.36754    -2.36754
  end
end

Task SinglePoint

Engine ADF
  title TiCl4 TD-DFT spinorbit relativistic 2p Cl core excitations single orbital␣
↪transitions
  beckegrid
    quality good
  end
  eprint
    eigval 1000 1000
  end
  excitations
    allowed
    kfwrite 0
    lowest 40
    singleorbtrans
  end
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  relativity
    level spin-orbit
    formalism ZORA
  end
  selectexcitation
    useoccrange -20.0 -16.0
    usescaledzora
  end
  symmetry T(D)
  xc
    gga LB94
  end
EndEngine

eor

# == An example with the key MODIFYEXCITATION ==

# In this example, the 12 lowest singlet-singlet excitation energies of T2
# symmetry are calculated, the dipole allowed excitations. This can also be
# achieved using the ALLOWED subkey in the key Excitation.

# In this example only excitations from the 2t2 -orbital are included (see the
# key MODIFYEXCITATION), an almost pure 2p core orbital of titanium. The orbital
# energies of the uninteresting other occupied orbitals are artificially changed
# to a large negative value (-1d6 hartree).


AMS_JOBNAME=T2 $AMSBIN/ams <<eor
System
  atoms [Bohr]
```

(continues on next page)

(continued from previous page)

```
     Ti 0.       0.        0.
     Cl    2.36754     2.36754     2.36754
     Cl   -2.36754    -2.36754     2.36754
     Cl    2.36754    -2.36754    -2.36754
     Cl   -2.36754     2.36754    -2.36754
  end
end

Task SinglePoint

Engine ADF
  title TiCl4 TD-DFT scalar relativistic 2p Ti core excitations
  beckegrid
    quality good
  end
  eprint
    eigval 1000 1000
  end
  excitations
    davidson
       T2 12
    end
    onlysing
  end
  modifyexcitation
    useoccupied
       T2 2
    end
    usescaledzora
  end
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry T(D)
  xc
    gga LB94
  end
EndEngine

eor

# In this example the 2p Cl core excitation energies of TiCl4 are calculated.

# In this example only excitations from occupied orbitals are considered which
# have orbital energies between -8 and -6 hartree, namely the 5a1, 1e, 1t1,
# 4t2, and 5t2 orbitals, which are almost pure 2p core orbitals of chlorine.
# The orbital energies of the uninteresting other occupied orbitals are again
# artificially changed to a large negative value (-1d6 hartree).

AMS_JOBNAME=OccRange $AMSBIN/ams <<eor
System
```

(continues on next page)

(continued from previous page)

```
  atoms [Bohr]
     Ti 0.        0.         0.
     Cl    2.36754     2.36754     2.36754
     Cl   -2.36754    -2.36754     2.36754
     Cl    2.36754    -2.36754    -2.36754
     Cl   -2.36754     2.36754    -2.36754
  end
end

Task SinglePoint

Engine ADF
  title TiCl4 TD-DFT scalar relativistic 2p Cl core excitations
  beckegrid
    quality good
  end
  eprint
    eigval 1000 1000
  end
  excitations
    allowed
    lowest 50
  end
  modifyexcitation
    useoccrange -8.0 -6.0
    usescaledzora
  end
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry T(D)
  xc
    gga LB94
  end
EndEngine

eor


# Another possibility is the use of the subkey OccVirtRange:

AMS_JOBNAME=OccVirt $AMSBIN/ams <<eor
System
  atoms [Bohr]
     Ti 0.        0.         0.
     Cl    2.36754     2.36754     2.36754
     Cl   -2.36754    -2.36754     2.36754
     Cl    2.36754    -2.36754    -2.36754
     Cl   -2.36754     2.36754    -2.36754
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title TiCl4 TD-DFT scalar relativistic 2p Cl core excitations
  beckegrid
    quality good
  end
  eprint
    eigval 1000 1000
  end
  excitations
    allowed
    lowest 50
  end
  modifyexcitation
    useoccvirtrange 7.0 100.0
    usescaledzora
  end
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry T(D)
  xc
    gga LB94
  end
EndEngine

eor


AMS_JOBNAME=SelectT2 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     Ti 0.        0.         0.
     Cl    2.36754     2.36754     2.36754
     Cl   -2.36754    -2.36754     2.36754
     Cl    2.36754    -2.36754    -2.36754
     Cl   -2.36754     2.36754    -2.36754
  end
end

Task SinglePoint

Engine ADF
  title TiCl4 TD-DFT scalar relativistic 2p Ti core excitations
  beckegrid
    quality good
  end
  eprint
    eigval 1000 1000
```

(continues on next page)

(continued from previous page)

```
  end
  excitations
    davidson
       T2 12
    end
    onlysing
  end
  basis
    core None
    type DZ
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  selectexcitation
    useoccupied
       T2 2
    end
    usescaledzora
  end
  symmetry T(D)
  xc
    gga LB94
  end
EndEngine

eor

AMS_JOBNAME=SelectOccVir $AMSBIN/ams <<eor
System
  atoms [Bohr]
     Ti 0.        0.         0.
     Cl    2.36754     2.36754     2.36754
     Cl   -2.36754    -2.36754     2.36754
     Cl    2.36754    -2.36754    -2.36754
     Cl   -2.36754     2.36754    -2.36754
  end
end

Task SinglePoint

Engine ADF
  title TiCl4 TD-DFT scalar relativistic 2p Cl core excitations
  beckegrid
    quality good
  end
  eprint
    eigval 1000 1000
  end
  excitations
    allowed
    lowest 25
  end
  basis
    core None
```

(continues on next page)

(continued from previous page)

```
    type DZ
    CreateOutput Yes
  end
  relativity
    level scalar
    formalism ZORA
  end
  selectexcitation
    useoccvirtrange 7.0 100.0
    usescaledzora
  end
  symmetry T(D)
  xc
    gga LB94
  end
EndEngine

eor
```

Example: X-Ray Absorption and Emission Quadrupole Oscillator strengths at the Cl K-edge: TiCl4

Download `TiCl4_XASXES.run`

```
#! /bin/sh


# Calculation of the 1s Cl core excitation energies of TiCl4. Both X-Ray
# Emission (XES keyword) and X-Ray absorption (XAS keyword in EXCITATION key)
# are calculated. The individual (zeroth and higher order) oscillator strength
# components to the total oscillator strengths will be printed
# (ALLXESQUADRUPOLE, ALLXASQUADRUPOLE).

# In order to calculate a 1s-orbital that is localized on one of the Cl atoms, a
# frozen core approach is used on the other Cl atoms, such that there is only
# one Cl 1s-orbital, which will remain localized in the molecule.

# In order to get information of the population analysis on all core orbitals, a
# large negative value was included as argument for the orbpoper subkey of the
# block key EPRINT. This orbital analysis shows that the 2 A1 orbital is indeed
# the 1s Cl orbital, see subkey corehole of block key XES, and subkey
# useoccupied of block key MODIFYEXCITATION.

AMS_JOBNAME=XASXES $AMSBIN/ams <<eor
System
  symmetrize
  symmetry C(3V)
  atoms
     Ti             0.0000    0.0000    0.0000
     Cl             0.0000    0.0000    2.2930
     Cl.1           1.0809   -1.8722   -0.7643
     Cl.1          -2.1618    0.0000   -0.7643
     Cl.1           1.0809    1.8722   -0.7643
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title TiCl4 X-Ray Absorption and Emission Quadrupole Oscillator strengths at the Cl␣
↪K-edge
  eprint
    eigval 1000 1000
    orbpoper -10000 100
  end
  excitations
    allxasquadrupole
    davidson
    end
    lowest 10
    onlysing
    xas
  end
  modifyexcitation
    useoccupied
       A1 2
    end
  end
  basis
    PerAtomType Symbol=Cl.1 File=$AMSRESOURCES/ADF/TZP/Cl.2p
    core None
    type TZP
  end
  relativity
    level scalar
    formalism ZORA
  end
  scf
    iterations 200
  end
  symmetry C(3V)
  xc
    gga Becke Perdew
  end
  xes
    allxesquadrupole
    corehole A1 2
  end
EndEngine

eor


# In the second example, to test gauge dependency, the molecule is shifted 100
# Angstrom in the z-direction.

AMS_JOBNAME=Gauge $AMSBIN/ams <<eor
System
  symmetrize
  symmetry C(3V)
  atoms
     Ti              0.0000    0.0000    100.0000
     Cl              0.0000    0.0000    102.2930
```

(continues on next page)

(continued from previous page)

```
     Cl.1            1.0809   -1.8722     99.2357
     Cl.1           -2.1618    0.0000     99.2357
     Cl.1            1.0809    1.8722     99.2357
  end
end

Task SinglePoint

EngineRestart XASXES.results/adf.rkf

Engine ADF
  title TiCl4 X-Ray Absorption and Emission Quadrupole Oscillator strengths at the Cl␣
↪K-edge shifted 100Ang in z-direction
  eprint
    eigval 1000 1000
    orbpoper -10000 100
  end
  excitations
    allxasquadrupole
    davidson
    end
    lowest 10
    onlysing
    xas
  end
  modifyexcitation
    useoccupied
       A1 2
    end
  end
  basis
    PerAtomType Symbol=Cl.1 File=$AMSRESOURCES/ADF/TZP/Cl.2p
    core None
    type TZP
  end
  relativity
    level scalar
    formalism ZORA
  end
  scf
    iterations 200
  end
  symmetry C(3V)
  xc
    gga Becke Perdew
  end
  xes
    allxesquadrupole
    corehole A1 2
  end
EndEngine

eor
```

Example: (Core) Excitation energies including spin-orbit coupling: Ne

Download Ne_exciso.run

```
#! /bin/sh


# Calculation of the (core) excitation energies of Ne including spin-orbit
# coupling.

# The state selective method (key SELECTEXCITATION) can be used to reduce the
# computational costs of, for example, core excitation energies. In this scheme
# a guess vector for the orbital transition has to be provided. It should be
# used in combination with the davidson method to calculate excitation energies.
# An overlap criterion is used to follow the wanted eigenvector. This key can
# also be used in case of spin-orbit coupling. The use of the key
# SELECTEXCITATION is similar as the use of the key MODIFYEXCITATION. However,
# the key SELECTEXCITATION can not be used in combination with the key
# MODIFYEXCITATION. In the state selective method (key SELECTEXCITATION) the
# one-electron excited state configuration space remains complete, whereas it is
# reduced in case the scheme with the MODIFYEXCITATION key.


AMS_JOBNAME=Frag $AMSBIN/ams <<eor
System
  atoms
     Ne .0000  .0000  0.0000
  end
end

Task SinglePoint

Engine ADF
  title Ne
  excitations
    lowest 8
  end
  basis
    type QZ4P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  symmetry d(8h)
  xc
    model SAOP
  end
EndEngine
eor


AMS_JOBNAME=Excitations $AMSBIN/ams <<eor
System
  atoms
     Ne .0000  .0000  0.0000  adf.f=Frag
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  title Ne spin-orbit
  excitations
    alsorestricted
    lowest 8
  end
  fragments
     Frag Frag.results/adf.rkf
  end
  numericalquality Good
  relativity
    level spin-orbit
    formalism ZORA
  end
  stcontrib
  symmetry d(8h)
  xc
    model SAOP
  end
EndEngine
eor


# The difference between the core excitation calculation and the standard
# excitation is the extra subkey MODIFYEXCITATION or SELECTEXCITATION in the
# core excitation calculation (in italic).

# ADF can not handle ATOM and linear symmetries in excitation calculations.
# Therefore a subsymmetry is used, in this case symmetry d(8h).

# A relatively large QZ4P basis set is used, which is still insufficient for
# excitations to Rydberg-like orbitals, one needs more diffuse functions.

# The key STCONTRIB is used, which will give a composition of the spin-orbit
# coupled excitation in terms of singlet-singlet and singlet-triplet scalar
# relativistic excitations. In order to use the key STCONTRIB the scalar
# relativistic fragment should be the complete molecule.

# In this case the key MODIFYEXCITATION or SELECTEXCITATION takes care that only
# excitations from the occupied 1s-orbital (spinor) are included. In symmetry
# d(8H) the 1s-orbital (spinor) is of A1.g (E1/2.g) symmetry.
```

Download Ne_CoreExci.run

```
#! /bin/sh


AMS_JOBNAME=Fragmod $AMSBIN/ams <<eor
System
  atoms
     Ne .0000  .0000  0.0000
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  title Ne
  excitations
    lowest 8
  end
  modifyexcitation
    useoccupied
       A1.g 1
    end
    usescaledzora
  end
  basis
    type QZ4P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  symmetry d(8h)
  xc
    model SAOP
  end
EndEngine
eor


AMS_JOBNAME=Excitationsmod $AMSBIN/ams <<eor
System
  atoms
     Ne .0000  .0000  0.0000  adf.f=Frag
  end
end

Task SinglePoint

Engine ADF
  title Ne spin-orbit
  excitations
    alsorestricted
    lowest 11
  end
  fragments
     Frag Fragmod.results/adf.rkf
  end
  modifyexcitation
    useoccupied
       E1/2.g 1
    end
    usescaledzora
  end
  numericalquality Good
```

(continues on next page)

(continued from previous page)

```
  relativity
    level spin-orbit
    formalism ZORA
  end
  stcontrib
  symmetry d(8h)
  xc
    model SAOP
  end
EndEngine
eor


AMS_JOBNAME=Fragsel $AMSBIN/ams <<eor
System
  atoms
     Ne .0000  .0000  0.0000
  end
end

Task SinglePoint

Engine ADF
  title Ne
  excitations
    lowest 8
  end
  basis
    type QZ4P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  selectexcitation
    useoccupied
       A1.g 1
    end
    usescaledzora
  end
  symmetry d(8h)
  xc
    model SAOP
  end
EndEngine
eor


AMS_JOBNAME=Excitationssel $AMSBIN/ams <<eor
System
  atoms
     Ne .0000  .0000  0.0000  adf.f=Frag
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title Ne spin-orbit
  excitations
    alsorestricted
    lowest 11
  end
  fragments
     Frag Fragsel.results/adf.rkf
  end
  numericalquality Good
  relativity
    level spin-orbit
    formalism ZORA
  end
  selectexcitation
    useoccupied
       E1/2.g 1
    end
    usescaledzora
  end
  stcontrib
  symmetry d(8h)
  xc
    model SAOP
  end
EndEngine
eor
```

Example: Excitation energies perturbative spin-orbit coupling: AgI

Download AgI_asoexcit.run

```
#! /bin/sh


# Calculation of the excitation energies of AgI including spin-orbit coupling in
# a perturbative way.

# ADF can not handle ATOM and linear symmetries in excitation calculations.
# In case of linear symmetries ADF will automatically use a
# a subsymmetry, in this case symmetry C(7v).
# In the excitation part of the output the symmetry labels are those of C(LIN) in␣
↪this case.

# For such a small molecule, a relatively small TZ2P basis set is used, which is
# not sufficient for excitations to Rydberg-like orbitals, one needs more
# diffuse functions. For larger molecules a TZ2P basis set is in most cases
# sufficient. In that case one might even consider smaller basis sets.

# The key SOPERT is included in scalar relativistic ZORA calculations of
# excitation energies. First scalar relativistic TDDFT calculations are
# performed to determine the lowest 60 singlet-singlet and singlet-triplet
# excited states and the spin-orbit coupling operator is applied to these
```

(continues on next page)

(continued from previous page)

```
# single-group excited states to obtain the excitation energies with spin-orbit
# coupling effects included.


$AMSBIN/ams <<eor
System
  atoms
     Ag 0.0000  0.0000  2.5446
     I  0.0000  0.0000  0.0000
  end
end

Task SinglePoint

Engine ADF
  title [AgI]
  excitations
    lowest 60
  end
  basis
    type TZ2P
    core None
    CreateOutput Yes
  end
  numericalquality Good
  sopert
  end
EndEngine

eor
```

Example: Excitation energies including spin-orbit coupling for open shell: PbF

Download PbF_excisou.run

```
#!/bin/sh


# Application of a spin-orbit coupled relativistic excitation energy calculation
# for an open-shell system. Note that the used method used is highly
# approximate. In this example the effect of spin-orbit coupling is large, in
# which case this method can be of use.

# First a scalar relativistic calculation is performed with symmetry. The
# resulting scalar relativistic adf.rkf file is used as fragment in the spin-orbit
# coupled calculation. Using a scalar relativistic fragment is not necessary,
# but makes the identification of the spin-orbit coupled results easier.

AMS_JOBNAME=PbF_SR $AMSBIN/ams <<eor
System
  atoms
     Pb    0     0    0
     F     0     0    2.06
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  basis
    core None
    type TZ2P
  end
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga bp86
  end
EndEngine
eor


# Next the spin-orbit coupled excitation unrestricted calculation is performed.
# The Tamm-Dancoff approximation (TDA) is needed and symmetry NOSYM should be
# used. Best is to use the noncollinear approximation.


AMS_JOBNAME=PbF_SO $AMSBIN/ams <<eor
System
  atoms
     Pb   0    0   0    adf.f=PbF
     F    0    0   2.06 adf.f=PbF
  end
end

Task SinglePoint

Engine ADF
  excitations
    lowest 14
  end
  fragments
     PbF PbF_SR.results/adf.rkf
  end
  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization noncollinear
  end
  symmetry nosym
  tda
  unrestricted
  xc
    gga bp86
  end
EndEngine
eor
```

Example: Circular Dichroism (CD) spectrum: DMO

Download DMO_CD.run

```
#!/bin/sh


# If the subkey CDSPECTRUM is included in the key EXCITATIONS, the rotatory
# strength is calculated for the calculated excitations, in order to calculate
# the CD (Circular Dichroism) spectrum. Only useful for chiral molecules.

# With the VELOCITY keyword also the dipole-velocity representation of the
# rotatory strength is calculated.

# Note: results will be physically meaningless due to small basis set. purpose
# of this job is to provide a test case for the CD implementation

# Do not use less strict convergence criteria than default, better to use
# tighter criteria. The approximations in the evaluation of the integrals one
# makes with the linear scaling techniques are effectively switched off by
# setting LINEARSCALING 100 (recommended to use this).


$AMSBIN/ams <<eor
System
  atoms
    O  0.0                 0.0                   1.141417145729065
    C -0.5970248579978943 -0.4282023310661316   -0.09432400763034821
    C  0.5970248579978943  0.4282023310661316   -0.09432400763034821
    H -0.4430356621742249 -1.487679243087769    -0.302562028169632
    H  0.4430356621742249  1.487679243087769    -0.302562028169632
    C -1.97874128818512    0.09404171258211136  -0.3866550326347351
    H -2.723244667053223  -0.4288762211799622    0.2206400185823441
    H -2.043423652648926   1.15996515750885     -0.1578925102949142
    H -2.236041307449341  -0.05517048016190529  -1.439977645874023
    C  1.97874128818512   -0.09404171258211136  -0.3866550326347351
    H  2.236041307449341   0.05517048016190529  -1.439977645874023
    H  2.723244667053223   0.4288762211799622    0.2206400185823441
    H  2.043423652648926  -1.15996515750885     -0.1578925102949142
  end
end

Task SinglePoint

Engine ADF
  title dimethyloxirane excitations + CD
  comment
     results will be physically  meaningless due to small basis set.
     purpose of this job is to provide a test case for the CD implementation
  end
  beckegrid
    quality good
  end
  excitations
    cdspectrum
    lowest 10
    onlysing
    velocity
```

(continues on next page)

(continued from previous page)

```
  end
  linearscaling 100
  end
  basis
    core Small
    type DZP
  end
  scf
    converge 1e-7
  end
  xc
    gga becke perdew
  end
EndEngine
eor
```

Example: CD spectrum with spin-orbit coupling: C2H3I

Download C2H3I_SOCD.run

```
#!/bin/sh

# Example for the calculation of an electronic CD spectrum including spin-orbit␣
↪coupling
# for the molecule Iodoethene.
# The lowest 3 excited states have high triplet character, but due to spin-orbit␣
↪coupling
# have non-negligible singlet character, such that the electronic CD spectrum is non-
↪zero.
# Phosphorescence dissymmetry factors are calculated.

$AMSBIN/ams <<eor
System
  atoms
     C     -0.016815      0.033201     -0.680171
     I     -0.322076      1.796278     -1.883624
     C     -0.019887      0.012368      0.646966
     H     -0.185572     -0.930351      1.163416
     H      0.141293     -0.844352     -1.293414
     H      0.134807      0.889756      1.262747
  end
end

Task SinglePoint

engine ADF
  numericalquality Good
  excitations
    cdspectrum
    fullkernel
    lowest 4
  end
  basis
    PerAtomType Symbol=C File=ZORA/DZP/C
    PerAtomType Symbol=H File=ZORA/DZ/H
```

(continues on next page)

(continued from previous page)

```
    PerAtomType Symbol=I File=ZORA/TZP/I
  end
  relativity
    level spin-orbit
  end
  title iodoethene-socd
  xc
    LibXC BP86
  end
endengine
eor
```

Example: CD spectrum, hybrid functional: Twisted ethene

Download Twist_Ethene_TDDFT.run

```
#!/bin/sh


# If the subkey CDSPECTRUM is included in the key EXCITATIONS, the rotatory
# strength is calculated for the calculated excitations, in order to calculate
# the CD (Circular Dichroism) spectrum. Only useful for chiral molecules. A
# hybrid functional is used in this case.

# With the VELOCITY keyword also the dipole-velocity representation of the
# rotatory strength is calculated.

# Note: results will be physically meaningless due to small basis set. purpose
# of this job is to provide a test case for the CD implementation

# Do not use less strict convergence criteria than default, better to use
# tighter criteria. The approximations in the evaluation of the integrals one
# makes with the linear scaling techniques are effectively switched off by
# setting LINEARSCALING 99 (recommended to use this).

$AMSBIN/ams <<eor
System
  symmetrize
  atoms
     C 0.000000 0.000000 0.000000
     C -0.000000 -0.000000 1.350716
     H 0.565703 -0.745428 -0.540179
     H -0.565703 0.745428 -0.540179
     H 0.035837 -0.935093 1.890895
     H -0.035837 0.935093 1.890895
  end
end

Task SinglePoint

Engine ADF
  title twisted ethene
  comment
     purpose of this job is to provide a test case for the CD implementation
     with hybrid functionals, and to test the "velocity" keyword
```

(continues on next page)

(continued from previous page)

```
  end
  excitations
    cdspectrum
    lowest 20
    onlysing
    velocity
  end
  linearscaling 99
  end
  basis
    core Small
    type TZ2P
  end
  numericalquality good
  rihartreefock
    fitsetquality Normal
  end
  scf
    converge 1.e-8
  end
  xc
    hybrid PBE0
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: MCD: H2O

Download H2O_MCD.run

```
#! /bin/sh


# Example for the calculation of magnetic circular dichroism (MCD). If the
# subkey MCD is included in the key EXCITATIONS the MCD parameters of the
# calculated excitations are calculated (A and B terms). The keys RELATIVISTIC
# ZORA and SOMCD are required for a calculation of temperature-dependent C
# terms. The key ALLPOINTS is required for an MCD calculation (if the molecule
# has symmetry).


$AMSBIN/ams <<eor
System
  atoms
    O             0.0000    0.0000    0.0000
    H             0.0000   -0.7071    0.7071
    H             0.0000    0.7071    0.7071
  end
  charge 1
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title water MCD
  allpoints
  excitations
    lowest 20
    mcd NMCDTERM=5
    onlysing
  end
  basis
    type DZP
  end
  relativity
    level scalar
    formalism ZORA
  end
  somcd
  spinpolarization 1
  unrestricted
EndEngine
eor
```

Example: MCD including zero-field splitting: H2O

Download H2O_MCD_ZFS.run

```
#! /bin/sh


# Example for the calculation of magnetic circular dichroism (MCD) including
# zero-field splitting (ZFS). If the subkey MCD is included in the key
# EXCITATIONS the MCD parameters of the calculated excitations are calculated (A
# and B terms). The keys RELATIVISTIC ZORA and SOMCD are required for a
# calculation of temperature-dependent C terms. The key ALLPOINTS is required
# for an MCD calculation (if the molecule has symmetry). For zero-field
# splitting (ZFS) the electron spin S<=1.


$AMSBIN/ams <<eor
System
  atoms
    O            0.0000    0.0000    0.0000
    H            0.0000   -0.7071    0.7071
    H            0.0000    0.7071    0.7071
  end
  charge 2
end

Task SinglePoint

Engine ADF
  title water MCD with zero-field splitting
  allpoints
  excitations
```

(continues on next page)

(continued from previous page)

```
    lowest 20
    mcd NMCDTERM=5 NODIRECT NTEMP=2 TMIN=1.0 TMAX=300.0 NBFIELD=2 BMIN=1.0 BMAX=8.0
    onlysing
  end
  basis
    type DZP
  end
  relativity
    level scalar
    formalism ZORA
  end
  somcd
  spinpolarization 2
  unrestricted
  zfs
EndEngine
eor
```

Example: CV(n)-DFT excitation energies: Formamide

Download Formamide_CVnDFT.run

```
#!/bin/sh


# Calculation of the excitation energies of Formamide using the CV(n)-DFT
# method. Due to bugs in older versions it is important to use ADF2016.105 or
# later.

# Example of CV(infinity)-DFT where the molecular orbitals for the particular
# excitation are optimized (relaxed), which is called the R-CV(infinity)-DFT method.
# In this case, the transition U-vector is frozen, while the orbitals are
# relaxed. ALLPOINTS should be used in case of a symmetric molecule.
# EXACTDENSITY should be used for precision reasons. Note that the irrep labels
# for C(s) symmetry that are used in the excitation code (A' and A'') differ
# from those that are used during the SCF (AA and AAA). The example contains of
# 2 parts. In the first example singlet-singlet excitations are calculated using
# the subkeyword ONLYSING in the EXCITATIONS block key. In the second example
# singlet-triplet excitations are calculated replacing the subkeyword ONLYSING
# with the subkeyword ONLYTRIP in the EXCITATIONS block key.


AMS_JOBNAME=SS $AMSBIN/ams <<eor
System
  atoms
     H -0.927427 -0.600301 0.000000
     H  1.070498 -1.782390 0.000000
     H  2.024514 -0.325050 0.000000
     C  0.000000  0.000000 0.000000
     O  0.000000  1.225060 0.000000
     N  1.119392 -0.775069 0.000000
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  allpoints
  cvndft
    iteration 10
    r_cv_dft &
      damporbrelax 0.2
      relaxalpha 1
      relaxbeta 1
    end
    tolerance 0.001
  end
  exactdensity
  excitations
    analytical
    davidson
       A' 2
       A'' 2
    end
    lowest 4
    onlysing
    tolerance 1e-5
  end
  basis
    core None
    fittype ZORA/QZ4P
    type DZ
  end
  numericalquality good
  tda
  title Formamide
  xc
    hybrid B3LYP
  end
EndEngine
eor



AMS_JOBNAME=ST $AMSBIN/ams <<eor
System
  atoms
     H -0.927427 -0.600301 0.000000
     H  1.070498 -1.782390 0.000000
     H  2.024514 -0.325050 0.000000
     C  0.000000  0.000000 0.000000
     O  0.000000  1.225060 0.000000
     N  1.119392 -0.775069 0.000000
  end
end

Task SinglePoint

Engine ADF
  allpoints
  cvndft
    iteration 10
```

(continues on next page)

(continued from previous page)

```
    r_cv_dft &
      damporbrelax 0.2
      relaxalpha 1
      relaxbeta 1
    end
    tolerance 0.001
  end
  exactdensity
  excitations
    analytical
    davidson
       A' 2
       A'' 2
    end
    lowest 4
    onlytrip
    tolerance 1e-5
  end
  basis
    core None
    fittype ZORA/QZ4P
    type DZ
  end
  numericalquality good
  tda
  title Formamide
  xc
    hybrid B3LYP
  end
EndEngine
eor
```

Example: HDA excitation energies: NH3

Download NH3_HDA.run

```
#!/bin/sh

# Calculation of the excitation energies of NH3 using the
# Hybrid Diagonal Approximation (HDA).
# Relevant in case hybrid XC functionals are used.
# For speed reasons one should set a (not so large) value for HDA_CutOff.

AMS_JOBNAME=NH3 $AMSBIN/ams << eor
System
  symmetrize
  atoms
     N   0.000000    0.000000    0.000000
     H   0.956305    0.000000    0.292372
     H  -0.478152    0.828184    0.292372
     H  -0.478152   -0.828184    0.292372
  end
end

Task SinglePoint
```

(continues on next page)

(continued from previous page)

```
Engine ADF
  Basis
    Type TZP
  End

  XC
    Hybrid B3LYP
  End

  Excitations
    Allowed
    lowest 21
    HDA
    HDA_CutOff 40.0 [eV]
  End
EndEngine
eor
```

Example: HDA spin-orbit coupled excitation energies: H2O

Download H2O_SO_HDA.run

```
#!/bin/sh

# Calculation of the spin-orbit (SO) coupled excitation energies of H2O using the
# Hybrid Diagonal Approximation (HDA).
# Relevant in case hybrid XC functionals are used.
# For HDA icw SO symmetry NOSYM is required.
# For speed reasons one should not use a too large value for HDA_CutOff.

AMS_JOBNAME=H2O_SO $AMSBIN/ams << eor
System
  atoms
     O   0.000000    0.000000    0.000000
     H   0.000000    0.759062   -0.587729
     H   0.000000   -0.759062   -0.587729
  end
end

Task SinglePoint

Engine ADF
  Basis
   Type TZP
  End

  Symmetry NOSYM

  relativity
    level spin-orbit
    formalism ZORA
  end

  XC
```

(continues on next page)

(continued from previous page)

```
    Hybrid B3LYP
  End

  Excitations
    lowest 50
    HDA
    HDA_CutOff 40.0
  END
EndEngine
eor
```

Example: TD-DFT+TB excitation energies: beta-Carotene

Download TD-DFTB_betacarotene.run

```
#!/bin/sh


# Calculation of the excitation energies of beta-Carotene using the TD-DFT+TB
# method. This method is meant for GGA or LDA functionals. In this example PBE
# is used during the SCF. In the the calculation of the excitation energies the
# TD-DFTB+TB method is used in which the required integrals are approximated in
# the same way as in a TD-DFTB calculation. Symmetry NOSYM is required.

# The key MODIFYEXCITATION allows to remove single orbital transitions based on
# certain criteria. For example, for a large calculation, one can combine
# several of the filtering techniques with the key MODIFYEXCITATION. If one uses
# the next input, only the lowest 10000 single orbital transitions are used
# (which have the lowest orbital energy differences). Out of these 10000 the
# single orbital transitions with an oscillator strength smaller than 0.001 are
# removed, and single orbital transitions are removed that have an energy
# difference larger than 1.0 Hartree. In the space of single-orbital transition
# that are left the lowest 1000 eigenvectors are calculated.

#    ModifyExcitation
#     UseOccVirtNumbers 1 10000
#     OscStrength 0.001
#     UseOccVirtRange 0.0 1.0
#    End
#    Excitations
#     TDDFTB
#     OnlySing
#     Lowest 1000
#    End

# In the actual example only single orbital transitions with an oscillator
# strength smaller than 0.001 are removed, which one might do if one is
# interested only in excitation energies with a high oscillator strength.


$AMSBIN/ams <<eor
System
  atoms
     C      -1.67096000       1.41980000      -1.15887000
     C      -0.38686000       2.25210000      -1.41391000
```

(continues on next page)

(continued from previous page)

```
     C      -1.74087000       0.49471000      -0.15347000
     C      -2.78739000       1.72912000      -2.05465000
     C       0.64868000       2.05301000      -0.28395000
     C       0.23104000       1.84552000      -2.77135000
     C      -0.70560000       3.76543000      -1.46182000
     C      -0.54745000       0.11313000       0.69574000
     C      -3.00351000      -0.21803000       0.26657000
     C      -3.76926000       0.89738000      -2.51350000
     C       0.78914000       0.59489000       0.13981000
     C      -4.86625000       1.29159000      -3.36539000
     C      -5.83916000       0.35729000      -3.66241000
     C      -4.93554000       2.71910000      -3.84917000
     C      -7.05227000       0.56749000      -4.37867000
     C      -8.05019000      -0.37384000      -4.49171000
     C      -9.34557000      -0.15733000      -5.06205000
     C     -10.30909000      -1.14399000      -4.91514000
     C      -9.64286000       1.17054000      -5.71730000
     C     -11.69180000      -1.05048000      -5.21459000
     C     -12.61955000      -2.01468000      -4.87148000
     C     -14.02165000      -1.83518000      -4.98190000
     C     -15.01934000      -2.67078000      -4.50243000
     C     -16.37196000      -2.20132000      -4.51598000
     C     -14.71829000      -4.01459000      -3.88235000
     C     -17.44392000      -2.81539000      -3.90884000
     C     -18.74098000      -2.23505000      -3.81068000
     C     -19.81891000      -2.70482000      -3.08600000
     C     -21.01079000      -1.89278000      -3.01886000
     C     -19.77834000      -4.00300000      -2.31809000
     C     -22.11699000      -2.17469000      -2.26831000
     C     -23.34229000      -1.38813000      -2.11260000
     C     -24.62364000      -2.22792000      -1.86832000
     C     -23.37421000      -0.02091000      -2.14864000
     C     -25.81771000      -1.33236000      -1.46707000
     C     -24.97031000      -3.02505000      -3.14664000
     C     -24.41619000      -3.23491000      -0.71197000
     C     -24.66206000       0.77221000      -2.09033000
     C     -22.15090000       0.86196000      -2.20093000
     C     -25.91751000      -0.06612000      -2.31089000
     H      -2.80910000       2.75370000      -2.43250000
     H       1.61418000       2.46808000      -0.61595000
     H       0.33480000       2.64516000       0.59383000
     H      -0.50118000       1.95999000      -3.58430000
     H       1.10267000       2.47962000      -3.00160000
     H       0.55846000       0.79551000      -2.76878000
     H      -1.27083000       4.08204000      -0.57268000
     H       0.23706000       4.33391000      -1.48690000
     H      -1.27659000       4.05053000      -2.35573000
     H      -0.69913000       0.51646000       1.71643000
     H      -0.54631000      -0.98424000       0.82135000
     H      -3.05490000      -1.23815000      -0.15105000
     H      -3.01240000      -0.33500000       1.36235000
     H      -3.91214000       0.31400000      -0.03789000
     H      -3.75416000      -0.15646000      -2.22797000
     H       1.07957000      -0.02480000      -0.72369000
     H       1.58320000       0.48035000       0.89321000
     H      -5.69282000      -0.64720000      -3.25046000
     H      -4.00625000       2.99962000      -4.36844000
```

(continues on next page)

(continued from previous page)

```
     H      -5.76674000       2.88684000      -4.54145000
     H      -5.05424000       3.41602000      -3.00441000
     H      -7.22923000       1.55534000      -4.80743000
     H      -7.87841000      -1.35648000      -4.03966000
     H      -9.98472000      -2.07170000      -4.43134000
     H      -8.86494000       1.42343000      -6.45273000
     H     -10.60514000       1.16993000      -6.23979000
     H      -9.66280000       1.98327000      -4.97344000
     H     -12.06891000      -0.12920000      -5.66646000
     H     -12.24343000      -2.93190000      -4.41065000
     H     -14.35090000      -0.88585000      -5.41809000
     H     -16.54169000      -1.22776000      -4.98799000
     H     -15.37918000      -4.79019000      -4.29660000
     H     -13.68496000      -4.33124000      -4.05827000
     H     -14.88466000      -3.99412000      -2.79324000
     H     -17.27327000      -3.77423000      -3.41639000
     H     -18.87740000      -1.27545000      -4.32124000
     H     -20.99304000      -0.98523000      -3.62576000
     H     -19.85755000      -3.82238000      -1.23433000
     H     -20.62535000      -4.64779000      -2.59868000
     H     -18.85754000      -4.56711000      -2.49752000
     H     -22.10976000      -3.13142000      -1.74142000
     H     -26.74148000      -1.93003000      -1.53138000
     H     -25.70276000      -1.04055000      -0.40816000
     H     -24.12173000      -3.65146000      -3.45913000
     H     -25.83556000      -3.68339000      -2.96560000
     H     -25.21348000      -2.35763000      -3.98638000
     H     -24.03490000      -2.73308000       0.18974000
     H     -25.38111000      -3.70177000      -0.46026000
     H     -23.72534000      -4.04703000      -0.97564000
     H     -24.71649000       1.28228000      -1.10831000
     H     -24.60164000       1.58888000      -2.83152000
     H     -21.95198000       1.22417000      -3.22414000
     H     -22.31941000       1.76042000      -1.58543000
     H     -21.24680000       0.35543000      -1.84399000
     H     -26.00957000      -0.32720000      -3.37736000
     H     -26.81690000       0.51083000      -2.04725000
  end
end

Task SinglePoint

Engine ADF
  excitations
    onlysing
    td-dftb
  end
  modifyexcitation
    oscstrength 0.001
  end
  symmetry NOSYM
  xc
    gga PBE
  end
EndEngine

eor
```

Example: sTDA excitation energies: Adenine

Download Adenine_sTDA.run

```
#! /bin/sh


# Calculation of the excitation energies of Adenine using the simplified Tamm-
# Dancoff (sTDA) method by Grimme. This method is meant for hybrid functionals
# or range-separated functionals during the SCF. In this example the hybrid PBE0
# is used during the SCF. In the the calculation of the excitation energies the
# sTDA method is used in which the required integrals are approximated. For
# hybrid functionals ADF will automatically set the parameters that are needed
# in this method. For range-separated functional one needs to set the parameters
# manually. Symmetry NOSYM is required. A TZP or TZ2P basis set is recommended
# for this method.


$AMSBIN/ams <<eor
System
  atoms
     N         1.966533   -0.556678    0.000000
     C         1.351210   -1.750556    0.000000
     N         0.039008   -2.019043    0.000000
     C        -0.679938   -0.889685    0.000000
     C        -0.192782    0.421840    0.000000
     C         1.210298    0.557343    0.000000
     N        -2.051805   -0.746568    0.000000
     C        -2.311388    0.608600    0.000000
     N        -1.225604    1.347025    0.000000
     N         1.828584    1.760602    0.000000
     H         2.012099   -2.612788    0.000000
     H        -2.725918   -1.499744    0.000000
     H        -3.323166    0.988113    0.000000
     H         1.288936    2.610953    0.000000
     H         2.835883    1.795025    0.000000
  end
end

Task SinglePoint

Engine ADF
  excitations
    onlysing
    stda
  end
  basis
    type DZ
  end
  symmetry nosym
  xc
    hybrid PBE0
  end
EndEngine

eor
```

Example: sTDDFT excitation energies: Adenine

Download Adenine_sTDDFT.run

```
#! /bin/sh


# Calculation of the excitation energies of Adenine using the simplified time-
# dependent DFT (sTDDFT) method by Grimme. This method is meant for hybrid
# functionals or range-separated functionals during the SCF. In this example the
# hybrid PBE0 is used during the SCF. In the the calculation of the excitation
# energies the sTDDFT method is used in which the required integrals are
# approximated. For hybrid functionals ADF will automatically set the parameters
# that are needed in this method. For range-separated functional one needs to
# set the parameters manually. Symmetry NOSYM is required A TZP or TZ2P basis
# set is recommended for this method.


$AMSBIN/ams <<eor
System
  atoms
     N         1.966533   -0.556678    0.000000
     C         1.351210   -1.750556    0.000000
     N         0.039008   -2.019043    0.000000
     C        -0.679938   -0.889685    0.000000
     C        -0.192782    0.421840    0.000000
     C         1.210298    0.557343    0.000000
     N        -2.051805   -0.746568    0.000000
     C        -2.311388    0.608600    0.000000
     N        -1.225604    1.347025    0.000000
     N         1.828584    1.760602    0.000000
     H         2.012099   -2.612788    0.000000
     H        -2.725918   -1.499744    0.000000
     H        -3.323166    0.988113    0.000000
     H         1.288936    2.610953    0.000000
     H         2.835883    1.795025    0.000000
  end
end

Task SinglePoint

Engine ADF
  excitations
    onlysing
    stddft
  end
  basis
    type DZ
  end
  symmetry nosym
  xc
    hybrid pbe0
  end
EndEngine

eor
```

Example: sTDA excitation energies RS functional: Bimane

Download Bimane_wB97X.run

```
#! /bin/sh


# Calculation of the excitation energies of Bimane using the simplified Tamm-
# Dancoff (sTDA) method by Grimme, for a range-separated function. This method
# is meant for hybrid functionals or range-separated functionals during the SCF.
# In this example the range-separated functional wB97X is used during the SCF.
# In the the calculation of the excitation energies the sTDA method is used
# in which the required integrals are approximated. For range-separated
# functional one needs to set the parameters manually. Note that the alpha and
# beta parameters for sTDA for range separated functionals in the Risthaus paper
# (dx.doi.org/10.1039/C3CP54517B) are mixed up. A TZP or TZ2P basis set is
# recommended for this method.
# To use the wB97X range-separated functional LibXC is needed. Symmetry NOSYM is
# required. The sTDA parameters should be set with the subkeys GRIMMEAEX,
# GRIMMEALPHA, and GRIMMEBETA of the key MODIFYEXCITATION.


$AMSBIN/ams <<eor

  Task SinglePoint

  System
    Atoms
     C      1.331698   -1.879605   -0.108603
     C      2.132097   -0.654088   -0.003948
     C      1.302873    0.426430   -0.104903
     C     -1.300787    0.433951   -0.101824
     C     -2.136017   -0.641707    0.001280
     C     -1.342993   -1.871870   -0.104659
     N     -0.000534   -0.021248   -0.324354
     N     -0.004568   -1.403598   -0.331622
     O     -1.679176   -3.039253   -0.043168
     O      1.661467   -3.048960   -0.049039
     C     -1.615093    1.891323   -0.058117
     C     -3.618641   -0.678771    0.170347
     C      1.626055    1.881942   -0.064754
     C      3.614962   -0.700043    0.160744
     H     -4.049414    0.325596    0.200326
     H     -3.887430   -1.197656    1.097229
     H     -4.086348   -1.233771   -0.650635
     H     -1.173280    2.374317    0.820379
     H     -2.696603    2.026980   -0.008719
     H     -1.243008    2.407629   -0.949030
     H      2.707594    2.010775   -0.000483
     H      1.174301    2.373283    0.803861
     H      1.270709    2.395192   -0.964373
     H      4.079173   -1.241294   -0.671464
     H      3.883147   -1.237421    1.077108
     H      4.049818    0.301869    0.208419
    End
  End

  Engine ADF
```

(continues on next page)

(continued from previous page)

```
    Basis
      Type DZ
      Core None
    End
    Symmetry nosym
    Excitations
      STDA
      OnlySing
    end
    XC
      LibXC wB97X
    End
    ModifyExcitation
      GRIMMEPERTC
      GRIMMEAEX   0.56
      GRIMMEALPHA 4.51
      GRIMMEBETA  8.00
    end
  EndEngine
eor
```

Example: sTDA excitation energies wB97: TCNE-Benzene

Download TCNE-Benzene_wB97.run

```
#! /bin/sh


# Calculation of the excitation energies of the complex TCNE-Benzene using the
# simplified Tamm-Dancoff (sTDA) method by Grimme, for a range-separated
# function. This method is meant for hybrid functionals or range-separated
# functionals during the SCF. In this example the range-separated functional
# wB97 is used during the SCF. In the the calculation of the excitation
# energies the sTDA method is used in which the required integrals are
# approximated. For range-separated functional one needs to set the parameters
# manually. Note that the alpha and beta parameters for sTDA for range separated
# functionals in the Risthaus paper (dx.doi.org/10.1039/C3CP54517B) are mixed
# up. A TZP or TZ2P basis set is recommended for this method.

# For the excitation energies of the complex TCNE-Benzene, charge
# transfer (CT) states are important. Symmetry NOSYM is required.
# The subkey DESCRIPTORS is added to the key EXCITATIONS such that CT descriptors
# are calculated. The the atomic distance criterion used for the calculation of
# CT_AT was set to 3.0 Angstrom.
# The sTDA parameters should be set with the subkeys GRIMMEAEX, GRIMMEALPHA, and
# GRIMMEBETA of the key MODIFYEXCITATION.

$AMSBIN/ams <<eor
System
  atoms
     C     -2.274205   -1.398789   -0.000000
     C     -2.274206    1.398791   -0.000000
     C     -2.274506   -0.699551    1.213150
     C     -2.274505   -0.699551   -1.213150
     C     -2.274506    0.699551    1.213150
```

(continues on next page)

(continued from previous page)

```
     C     -2.274505    0.699551   -1.213150
     H     -2.278385   -2.491323   -0.000000
     H     -2.278386    2.491321   -0.000000
     C      1.368996   -0.686302    0.000000
     C      1.368996    0.686301    0.000000
     C      1.379968   -1.434052   -1.220900
     C      1.379968   -1.434052    1.220895
     C      1.379968    1.434051   -1.220900
     C      1.379967    1.434051    1.220895
     N      1.400955   -2.054479   -2.204820
     N      1.400955   -2.054479    2.204824
     N      1.400955    2.054481   -2.204820
     N      1.400954    2.054481    2.204824
     H     -2.276046   -1.246672    2.158733
     H     -2.276045   -1.246672   -2.158730
     H     -2.276045    1.246671   -2.158730
     H     -2.276046    1.246671    2.158733
  end
end

Task SinglePoint

Engine ADF
  excitations
    descriptors
    descriptors_ct_at_rab 3.0
    onlysing
    stda
  end
  Basis
    Type DZ
    Core None
    CreateOutput yes
  End
  modifyexcitation
    grimmeaex 0.61
    grimmealpha 4.41
    grimmebeta 8.00
    grimmepertc
  end
  symmetry nosym
  xc
    libxc wB97
  end
EndEngine

eor
```

14.4.3 Excited state (geometry) optimizations

Example: Excited state geometry optimization: N2

`Download EGO_N2.run`

```
#!/bin/sh

# Example for a singlet excited state geometry optimization for N_2.

# Needed for such excited state optimizations are the key EXCITATIONS (to
# calculate excitation energies), and the key EXCITEDGO (to select for which
# excitation a geometry optimization should be performed).
# In this case a hybrid functional B3LYP is used.


$AMSBIN/ams <<eor
System
  atoms
     N          0.0         0.0       -0.7
     N          0.0         0.0        0.7
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    gradients 0.0001
  End
End

Engine ADF
  title N2 Excited state geometry
  allpoints
  beckegrid
    quality good
  end
  excitations
    lowest 10
    onlysing
  end
  excitedgo
    output 2
    state S-.u 1
  end
  basis
    type DZ
    core NONE
    CreateOutput Yes
  end
  xc
    hybrid B3LYP
  end
EndEngine
eor
```

`Download EGO_N2_EIGENF.run`

```
#!/bin/sh

# Example for a singlet excited state geometry optimization with eigenvector
# following (subkeyword EIGENFOLLOW of key EXCITEDGO), GGA functional used.


$AMSBIN/ams <<eor
System
  atoms
     N           0.0          0.0       -0.55
     N           0.0          0.0        0.55
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    gradients 0.0001
  End
End

Engine ADF
  title N2 Eigenvector follow. Lowest state at the starting geometry is not the␣
↪lowest at min.
  beckegrid
    quality good
  end
  excitations
    lowest 10
    onlysing
  end
  excitedgo
    eigenfollow
    output 2
    state A 1
  end
  basis
    core NONE
    type DZ
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
EndEngine
eor

echo "StartGeometry"
$AMSBIN/amsreport ams.results/ams.rkf Geometry-Angstrom
echo "EndGeometry"
```

Example: Excited state geometry optimization with a constraint: CH2O

Download EGO_CH2O_trip_constr.run

```
#!/bin/sh

# Example for an excited triplet state geometry optimization with a constraint
# included.

# Needed for such excited state optimizations are the key EXCITATIONS (to
# calculate excitation energies), and the key EXCITEDGO (to select for which
# excitation a geometry optimization should be performed).
# Note that one has to select the state with the irrep name A'', and not with AAA.


$AMSBIN/ams <<eor
System
  atoms
     C  0.0  0.0  0.0
     O  1.2  0.0  0.0
     H -0.5  0.5  0.9
     H -0.5  0.4 -0.9
  end
end

Task GeometryOptimization

Constraints
  Distance 1 2 1.2
end

Engine ADF
  excitations
    davidson
    end
    lowest 5
    onlytrip
  end
  excitedgo
    output 2
    state A 1
    triplet
  end
  numericalquality good
  basis
    core NONE
    type DZ
  end
  symmetry nosym
EndEngine
eor
```

Example: Spin-flip excited state geometry optimization: CH2

Download EGO_CH2_sf.run

```
#!/bin/sh

# Example for a spin-flip excited state geometry optimization with a triplet
# reference, and a frequency calculation afterwards.

# Needed for such excited state optimizations are the key EXCITATIONS (to
# calculate excitation energies), and the key EXCITEDGO (to select for which
# excitation a geometry optimization should be performed).
# In this case spin-flip excitations are calculated.

AMS_JOBNAME=CH2 $AMSBIN/ams <<eor
System
  atoms
     C          0.000000    0.000000   0.0
     H          0.7         0.0        0.7
     H         -0.7         0.0        0.7
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    energy    0.0001
    gradients 0.0001
  End
  MaxIterations 50
End

Engine ADF
  title CH2 Excited state geometry with triplet reference and spin-flip excitation
  excitations
    lowest 10
  end
  excitedgo
    output 2
    state B2 1
  end
  forcealda
  basis
    type DZP
    core NONE
  end
  scf
    converge 1.0e-9
  end
  sftddft
  unrestricted
  spinpolarization 2
  symmetry C(2V)
  tda
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
# Next the frequencies are calculated of the excited state. LoadSystem is used to
# pick up the optimized excited state geometry of the previous calculation.
# Note that in a numerical frequencies calculation symmetry, is turned off except to␣
↪reduce the
# number of points calculated. Thus irrespective of the specified point group
# symmetry the symmetry label A of SYMMETRY NOSYM should be used to select the
# excited state. Care should be taken to ensure that the correct state is chosen
# in this frequencies calculation as the excited state number can change when
# the point group is changed. In this case instead of 'B2 1' one needs to select
# 'A 2'.



AMS_JOBNAME=ES_FREQUENCIES $AMSBIN/ams <<eor
LoadSystem
  File CH2.results/adf.rkf
End

Task SinglePoint
Properties
  NormalModes Yes
End

Engine ADF
  title CH2 Excited state frequencies with triplet reference and spin-flip excitation
  excitations
    lowest 10
  end
  symmetry NOSYM
  excitedgo
    output 2
    state A 2
  end
  forcealda
  basis
    core NONE
    type DZP
  end
  scf
    converge 1.0e-9
  end
  sftddft
  unrestricted
  spinpolarization 2
  tda
EndEngine
eor
```

Example: Numerical Frequencies of an excited state: PH2

Download EGO_PH2.run

```
#!/bin/sh

# Example for an excited state geometry optimization and frequency calculation.

# Needed for such excited state optimizations are the key EXCITATIONS (to
# calculate excitation energies), and the key EXCITEDGO (to select for which
# excitation a geometry optimization should be performed).

# The ground state and excited state are openshell.


AMS_JOBNAME=PH2 $AMSBIN/ams <<eor
System
  atoms
     P          0.000000    0.000000   0.0
     H          0.7         0.0        0.7
     H         -0.7         0.0        0.7
  end
end

Task GeometryOptimization

GeometryOptimization
  Convergence
    energy    0.0001
    gradients 0.0001
  End
  MaxIterations 50
End

Engine ADF
  title PH2 Excited state geometry
  beckegrid
    quality good
  end
  exactdensity
  excitations
    lowest 10
    onlysing
  end
  excitedgo
    output 1
    state B2 1
  end
  basis
    core NONE
    type DZ
  end
  scf
    converge 1.0e-9
  end
  unrestricted
  spinpolarization 1
  xc
```

(continues on next page)

(continued from previous page)

```
    gga BP86
  end
EndEngine
eor


# Next the frequencies are calculated of the excited state. LoadSystem is used to
# pick up the optimized excited state geometry of the previous calculation. Note that␣
↪in a
# numerical frequencies calculation symmetry is turned off except to reduce the
# number of points calculated. Thus irrespective of the specified point group
# symmetry the symmetry label A of SYMMETRY NOSYM should be used to select the
# excited state. Care should be taken to ensure that the correct state is chosen
# in this frequencies calculation as the excited state number can change when
# the point group is changed. In this case instead of 'B2 1' one needs to select
# 'A 1'. Accurate SCF convergence parameters are used.


AMS_JOBNAME=PH2_Freq $AMSBIN/ams <<eor
LoadSystem
  File PH2.results/adf.rkf
End

Task SinglePoint
Properties
  NormalModes Yes
End

Engine ADF
  title PH2 Excited state frequencies
  beckegrid
    quality good
  end
  exactdensity
  excitations
    lowest 10
    onlysing
  end
  symmetry nosym
  excitedgo
    cpks
      eps 0.000001
    end
    output 2
    state A 1
  end
  basis
    type DZ
    core NONE
  end
  scf
    converge 1.0e-9
  end
  spinpolarization 1
  unrestricted
  xc
    gga BP86
```

(continues on next page)

(continued from previous page)

```
  end
EndEngine
eor
```

Example: TD-DFT+TB: Singlet excited state geometry optimization

Download EGO_TD-DFTB.run

```
#!/bin/sh


# Excited state geometry optimization using TD-DFT+TB, for the molecule m-Cresol.
# The lowest singlet excited state S1 is optimized.
# This method is meant for GGA or LDA functionals. In this example PBE
# is used during the SCF. In the the calculation of the excitation energies the
# TD-DFTB+TB method is used in which the required integrals are approximated in
# the same way as in a TD-DFTB calculation. Symmetry NOSYM is required.
# For excited state gradients all electron basis sets are required.

$AMSBIN/ams <<eor
System
  atoms
     C    -1.885     0.511    -0.236
     C    -1.213    -0.724    -0.202
     C     0.201    -0.869    -0.029
     C     1.002     0.345    -0.018
     C     0.289     1.570     0.081
     C    -1.111     1.718     0.028
     C     0.871    -2.198     0.048
     O     0.982     2.797     0.335
     H    -2.954     0.574    -0.397
     H    -1.791    -1.644    -0.345
     H     2.054     0.328    -0.314
     H    -1.578     2.668     0.285
     H     1.958     2.608     0.558
     H     1.664    -2.297    -0.731
     H     0.152    -3.015    -0.093
     H     1.370    -2.341     1.021
  end
end

Task GeometryOptimization

Engine ADF
  excitations
    onlysing
    td-dftb
  end
  basis
    core none
  end
  symmetry NOSYM
  xc
    gga PBE
  end
```

(continues on next page)

(continued from previous page)

```
  excitedGO
    singlet
    state A 1
  end
EndEngine
eor
```

Example: TD-DFT+TB: Triplet excited state geometry optimization

Download EGO_TD-DFTB_triplet.run

```
#!/bin/sh


# Excited state geometry optimization using TD-DFT+TB, for the molecule p-
↪Benzoquinone.
# The triplet excited state T3 is optimized.
# This method is meant for GGA or LDA functionals. In this example PBE
# is used during the SCF. In the the calculation of the excitation energies the
# TD-DFTB+TB method is used in which the required integrals are approximated in
# the same way as in a TD-DFTB calculation. Symmetry NOSYM is required.
# For excited state gradients all electron basis sets are required.

# Note that for symmetric systems, one could perturb the coordinates
# to break the symmetry of the system. This can be done, for example,
# by including PerturbCoordinates 0.05 in the System block.
# In this example this is not done, which in practice means that the
# molecule remains symmetric.

$AMSBIN/ams <<eor
System
  atoms
        C    0.00000000    -0.67337000     1.26921000
        C    0.00000000     0.67337000     1.26921000
        C    0.00000000     1.44457000     0.00000000
        C    0.00000000     0.67337000    -1.26921000
        C    0.00000000    -0.67337000    -1.26921000
        C    0.00000000    -1.44457000     0.00000000
        O    0.00000000     2.67997000     0.00000000
        O    0.00000000    -2.67997000     0.00000000
        H    0.00000000     1.26436000    -2.18667000
        H    0.00000000    -1.26436000    -2.18667000
        H    0.00000000     1.26436000     2.18667000
        H    0.00000000    -1.26436000     2.18667000
  end
end

Task GeometryOptimization

Engine ADF
  excitations
    onlytrip
    td-dftb
  end
  basis
```

(continues on next page)

(continued from previous page)

```
    core none
  end
  symmetry NOSYM
  xc
    gga PBE
  end
  excitedGO
    triplet
    state A 3
  end
EndEngine
eor
```

14.4.4 Vibrationally resolved (electronic) spectra

Example: Franck-Condon Factors: NO2

Download FranckCondon_NO2.run

```
#!/bin/sh

# As an example of a Franck-Condon calculation, lets look at the transition of
# NO2 to NO2 - . NO2 is a small molecule with only three vibrational modes.
# Putting an extra electron on the molecule will cause a big displacement,
# resulting in large electron-phonon couplings.

# In general, the larger the molecule, the smaller the displacement and hence
# the electron-phonon couplings and Franck-Condon factors. Moreover, larger
# molecules have more vibrational modes, meaning that the already smaller
# displacement will generally be smeared out over more modes, resulting in an
# additional decrease in electron-phonon couplings. This is fortunate, since the
# number of Franck-Condon factors increases factorially with the number of
# vibrational modes, making it prohibitively expensive to take more than a few
# vibrational quanta into account for most molecules.

# In order to calculate the Franck-Condon factors for Nitrite and Nitrogen
# dioxide, the equilibrium positions of the nuclei and the vibrational modes
# have to be obtained:


AMS_JOBNAME=NO2 $AMSBIN/ams <<eor
System
  atoms
     N         0.000000    0.000000   -0.016179
     O         0.000000    1.098646   -0.492918
     O         0.000000   -1.098646   -0.492918
  end
end

Task GeometryOptimization
GeometryOptimization
  Convergence
    Gradients 1.0e-5
  End
```

(continues on next page)

(continued from previous page)

```
End
Properties
    NormalModes True
End

Engine ADF
  basis
    core NONE
    type DZP
  end
  spinpolarization 1
  title Nitrogen dioxide
  unrestricted
  xc
    lda SCF VWN
  end
EndEngine
eor

# We are using an already optimized geometry for the second calculation but in a
# real experiment one should run geometry optimization first

AMS_JOBNAME=NO2_minus $AMSBIN/ams <<eor
System
  atoms
     N         0.000000    0.000000    0.126041
     O         0.000000    1.070642   -0.555172
     O         0.000000   -1.070642   -0.555172
  end
  charge -1.0
end

Task SinglePoint
Properties
    NormalModes True
End

Engine ADF
  basis
    core NONE
    type DZP
  end
  title Nitrite
  xc
    lda SCF VWN
  end
EndEngine

eor


# This runscript produces two adf.rkf files containing the frequencies and the
# normal modes for both charge states. Lets first look at the ground state to
# ground state overlap:


$AMSBIN/fcf <<eor
```

(continues on next page)

(continued from previous page)

```
  STATE1 NO2.results/adf.rkf
  STATE2 NO2_minus.results/adf.rkf
  TRANSLATE
  ROTATE
eor

rm TAPE61 logfile
```

Example: Vibronic-Structure Tracking: Naphthalene

Download Naphthalene_VST.run

```
#!/bin/sh


# Calculation of a vibrationally resolved electronically excited
# state spectrum with the VG-FC vibronic-structure tracking (VST)
# method of the molecule naphthalene.
# Important is to be at the ground state geometry of naphthalene.
# Note that a geometry optimization of naphthalene is skipped here,
# but we do use optimized coordinates.

# The lowest singlet-singlet excitation of naphthalene and
# its excited state gradient is calculated at the optimized
# ground state geometry of naphthalene.

# In the single point calclation the 'Gradients' are requested in the properties block
# In addition one needs to include the EXCITATIONS key and the EXCITEDGO key such␣
↪that the
# excited state nuclear gradients are calculated.

AMS_JOBNAME=ES $AMSBIN/ams <<eor
System
  atoms
     C 0.0  0.705203949620  2.423171793837
     C 0.0  1.398484082024  1.236799665234
     C 0.0 -1.398484082024  1.236799665234
     C 0.0  0.715180917267  0.0
     C 0.0 -0.705203949620  2.423171793837
     C 0.0 -0.715180917267  0.0
     C 0.0  1.398484082024 -1.236799665234
     C 0.0  0.705203949620 -2.423171793837
     C 0.0 -0.705203949620 -2.423171793837
     C 0.0 -1.398484082024 -1.236799665234
     H 0.0  2.493594473281  1.232860022766
     H 0.0  1.244970418595  3.374117886966
     H 0.0 -1.244970418595  3.374117886966
     H 0.0 -2.493594473281  1.232860022766
     H 0.0 -2.493594473281 -1.232860022766
     H 0.0 -1.244970418595 -3.374117886966
     H 0.0  1.244970418595 -3.374117886966
     H 0.0  2.493594473281 -1.232860022766
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Properties
    Gradients Yes
End

Engine ADF
  excitations
    lowest 1
    nto
    onlysing
  end
  excitedgo
    allgradients
  end
  basis
    type DZ
    core None
  end
EndEngine
eor

# The calculation of the vibrationally resolved spectrum of the lowest
# singlet excited state of naphthalene is calculated with the VG-FC VST method.
# The excited state gradient is read from the file ES.results/adf.rkf.

AMS_JOBNAME=VibronicStructure "$AMSBIN/ams" <<eor
    Task VibrationalAnalysis
    System
        Atoms
            C 0.0  0.705203949620  2.423171793837
            C 0.0  1.398484082024  1.236799665234
            C 0.0 -1.398484082024  1.236799665234
            C 0.0  0.715180917267  0.0
            C 0.0 -0.705203949620  2.423171793837
            C 0.0 -0.715180917267  0.0
            C 0.0  1.398484082024 -1.236799665234
            C 0.0  0.705203949620 -2.423171793837
            C 0.0 -0.705203949620 -2.423171793837
            C 0.0 -1.398484082024 -1.236799665234
            H 0.0  2.493594473281  1.232860022766
            H 0.0  1.244970418595  3.374117886966
            H 0.0 -1.244970418595  3.374117886966
            H 0.0 -2.493594473281  1.232860022766
            H 0.0 -2.493594473281 -1.232860022766
            H 0.0 -1.244970418595 -3.374117886966
            H 0.0  1.244970418595 -3.374117886966
            H 0.0  2.493594473281 -1.232860022766
        End
    End
    VibrationalAnalysis
        Type VibronicStructureTracking
        AbsorptionSpectrum
            LineWidth 50.0
            AbsorptionRange -200.0 4000.0
        End
        ExcitationSettings
```

(continues on next page)

(continued from previous page)

```
            ExcitationFile ES.results/adf.rkf
            Singlet
                B2.u 1
            End
        End
    End
    Engine ADF
        basis
            Type DZ
            Core None
        End
    EndEngine
eor
```

Example: Vibronic Density of States

Download DOS_FCF.run

```
#!/bin/bash

# The simplified PLAMS script calculates the density of states (DOS)
# for the transfer between two systems, in this case NO2 radical and NO2 anion
# We start from pre-optimized geometries to save time, though normally the
# geometry optimization should come first, then we calculate the frequencies
# for both systems and finally the absorption and emission FCF spectra
# and finally the DOS is calculated from the FCF spectra using the FCFDOS utility
# implemented in PLAMS

cp $AMSHOME/examples/adf/DOS_FCF/no2_1.xyz no2_1.xyz
cp $AMSHOME/examples/adf/DOS_FCF/no2_2.xyz no2_2.xyz
cp $AMSHOME/examples/adf/DOS_FCF/DOS_FCF.py .

$AMSBIN/plams DOS_FCF.py

# Every time you run a PLAMS script, a uniquely named working directory is
# created (plams.*****). This folder will contain one subdirectory per job. Each
# job directory contains the job's input and results files. In this case there
# will be 1 directory for the ADF calculation of the atomic charges that are
# used for DRF. Next there is a directory which calculates the full system.
```

14.4.5 (Hyper-)Polarizabilities, dispersion coefficients, ORD, magnetizabilities, Verdet constants

Example: Polarizabilities including spin-orbit coupling: AgI

Download AgI_SO_Pol.run

```
#! /bin/sh


# Example shows an frequency-dependent ZORA calculation of complex linear
# response of AgI including spin-orbit coupling.
```

(continues on next page)

(continued from previous page)

```
# The polarizability is calculated using the AORESPONSE key, in which spin-orbit
# coupling is taken into account. In this case a spin-restricted calculation is
# required, but, unlike the rest of AORESPONSE, also NOSYM. In this example
# lifetime effects are included.


"$AMSBIN/ams" <<eor

  Task SinglePoint

  System
    ATOMS
      Ag   0.0 0.0  1.2723
      I    0.0 0.0 -1.2723
    END
  End

  Engine ADF
    TITLE AgI,SO-ZORA
    basis
      type TZ2P
      core None
    end
    symmetry nosym
    allpoints
    BeckeGrid
     quality good
    End
      linearscaling 9.0
    End
    xc
      lda VWN
    end
    relativity
      level spin-orbit
    end
    aoresponse
      scf
      frequencies 0.085 [Hartree]
      lifetime 0.007
      ALDA
      EL_DIPOLE_EL_DIPOLE
    end
  EndEngine
eor
```

Example: damped first hyperpolarizability: LiH

Download LiH_DampedBeta.run

```
#! /bin/sh


# If the subkey lifetime and BETA or QUADRATIC is included in the key
# AORESPONSE, the damped (frequency dependent) first hyperpolarizability is
# calculated. This test example consists of two calculations calculations: one
# with BETA and the other with QUADRATIC.

# The subkey EOPE is used, which means the electro-optical Pockels effect
# beta(-omega;omega,0). This example can easily be modified to calculate the
# static beta(0;0,0), the optical rectification beta(0;omega,-omega), the second
# harmonic generation beta(-2omega;omega,omega), or the general case
# beta(-(omega1+omega2);omega1,omega2).

# Note: results will be physically meaningless due to small basis set. Purpose
# of this job is to provide a test case for the first hyperpolarizability
# implementation

# In the first example the first hyperpolarizability is calculated with the
# subkey BETA, for EOPE, the electro-optical Pockels effect
# beta(-omega;omega,0).

###########################
# Damped EOPE, 2n+1
###########################

AMS_JOBNAME=EOPE_beta $AMSBIN/ams <<eor
System
  atoms
     Li    0.00000000    0.00000000    3.49467000
     H     0.00000000    0.00000000    1.89402300
  end
end

Task SinglePoint

Engine ADF
  title Damped EOPE of LiH, 2n+1
  allpoints
  aoresponse
    alda
    beta
    eope
    frequencies 0.1000 0.0000 [Hartree]
    lifetime 0.0034
    scf iterations 50
  end
  basis
    core None
    type SZ
  end
  numericalquality good
  symmetry nosym
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor


# In the second example the first hyperpolarizability is calculated with the
# subkey QUADRATIC, again for EOPE, the electro-optical Pockels effect
# beta(-omega;omega,0).

################################
# Damped EOPE, Quadratic
################################

AMS_JOBNAME=EOPE_quadratic $AMSBIN/ams <<eor
System
  atoms
     Li    0.00000000    0.00000000    3.49467000
     H     0.00000000    0.00000000    1.89402300
  end
end

Task SinglePoint

Engine ADF
  title Damped EOPE of LiH, Quadratic
  allpoints
  aoresponse
    alda
    eope
    frequencies 0.1000 0.0000 [Hartree]
    lifetime 0.0034
    quadratic
    scf iterations 50
  end
  basis
    core None
    type SZ
  end
  numericalquality good
  symmetry nosym
EndEngine
eor


# For the static case beta(0;0,0) use the subkey STATIC
#   aoresponse
#     ...
#     frequencies 0.0000 0.0000 [Hartree]
#     static
#   end

# For optical rectification beta(0;omega,-omega) use the subkey OPTICALR.
#   aoresponse
#     ...
#     frequencies 0.1000 -0.1000 [Hartree]
#     opticalr
#   end

# For the second harmonic generation beta(-2omega;omega,omega) use the subkey SHG
```

(continues on next page)

(continued from previous page)

```
#   aoresponse
#    ...
#    frequencies 0.1000 0.1000 [Hartree]
#    shg
#   end

# Or in the general case for beta(-(omega1+omega2);omega1,omega2) choose two
# input frequencies omega1 and omega2
#   aoresponse
#    ...
#    frequencies omega1 omega2 [Hartree]
#   end
```

Example: damped second hyperpolarizability: LiH

Download LiH_DampedGamma.run

```
#! /bin/sh


# If the subkey lifetime and GAMMA or CUBIC is included in the key AORESPONSE,
# the damped (frequency dependent) second hyperpolarizability is calculated.
# This test example consists of three calculations calculations: two with GAMMA
# and one with CUBIC.

# The the first two examples subkey EFIOR is used, which means the electric
# field induced optical rectification gamma(0;omega,-omega,0). In the third
# example two-photon absorption (TPA) cross sections are calculated, the reduced
# form of gamma(-omega;omega,omega,-omega). Examples can easily be modified to
# calculate the static case gamma(0;0,0,0), the optical Kerr effect
# gamma(-omega;omega,0,0), the intensity dependent refractive index
# gamma(-omega;omega,omega,-omega), the electric field induced second harmonic
# generation gamma(-2omega;omega,omega,0), the third harmonic generation
# gamma(-3omega;omega,omega,omega) use the subkey THG, or the general case
# gamma(-(omega1+omega2+omega3);omega1,omega2,omega3).

# Note: results will be physically meaningless due to small basis set. Purpose
# of this job is to provide a test case for the second hyperpolarizability
# implementation

# In the first example the second hyperpolarizability is calculated with the
# subkey GAMMA, for EFIOR, the electric field induced optical rectification
# gamma(0;omega,-omega,0).


####################
# Damped EFIOR, 2n+1
####################

AMS_JOBNAME=EFIOR_gamma $AMSBIN/ams <<eor
System
  atoms
     Li    0.00000000    0.00000000    3.49467000
     H     0.00000000    0.00000000    1.89402300
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  title Damped EFIOR of LiH, 2n+1
  allpoints
  aoresponse
    alda
    efior
    frequencies 0.1000 -0.1000 0.0000 [Hartree]
    gamma
    lifetime 0.0034
    scf iterations 50
  end
  basis
    core None
    type SZ
  end
  numericalquality good
  symmetry nosym
EndEngine
eor


# In the second example the second hyperpolarizability is calculated with the
# subkey CUBIC, again for EFIOR, the electric field induced optical
# rectification gamma(0;omega,-omega,0).

#####################
# Damped EFIOR, Cubic
#####################

AMS_JOBNAME=EFOR_cubic $AMSBIN/ams <<eor
System
  atoms
     Li    0.00000000    0.00000000    3.49467000
     H     0.00000000    0.00000000    1.89402300
  end
end

Task SinglePoint

Engine ADF
  title Damped EFIOR of LiH, Cubic
  allpoints
  aoresponse
    alda
    cubic
    efior
    frequencies 0.1000 -0.1000 0.0000 [Hartree]
    lifetime 0.0034
    scf iterations 50
  end
  basis
    core None
    type SZ
```

(continues on next page)

(continued from previous page)

```
  end
  numericalquality good
  symmetry nosym
EndEngine
eor


# In the third example the subkeyword TPA is be used to calculate the gammagamma
# corresponding to the two photon absorption process (i.e., the reduced form of
# gamma(-omega;omega,omega,-omega)), however, TPA can ONLY be used with keyword
# GAMMA.

##################
# Damped TPA, 2n+1
##################

AMS_JOBNAME=TPA_gamma $AMSBIN/ams <<eor
System
  atoms
     Li    0.00000000    0.00000000    3.49467000
     H     0.00000000    0.00000000    1.89402300
  end
end

Task SinglePoint

Engine ADF
  title Damped TPA of LiH, 2n+1
  allpoints
  aoresponse
    alda
    frequencies 0.1000 0.1000 -0.1000 [Hartree]
    gamma
    lifetime 0.0034
    scf iterations 50
    tpa
  end
  basis
    core None
    type SZ
  end
  numericalquality good
  symmetry nosym
EndEngine
eor


# For the static case gamma(0;0,0,0) use the subkey STATIC
#   aoresponse
#    ...
#    frequencies 0.0000 0.0000 0.0000 [Hartree]
#    static
#   end

# For the optical Kerr effect gamma(-omega;omega,0,0) use the subkey OKE
#   aoresponse
#    ...
```

(continues on next page)

(continued from previous page)

```
#    frequencies 0.1000 0.0000 0.0000 [Hartree]
#    oke
#   end

# For the intensity dependent refractive index gamma(-omega;omega,omega,-omega)
# use the subkey IDRI
#   aoresponse
#    ...
#    frequencies 0.1000 0.1000 -0.1000 [Hartree]
#    idri
#   end

# For the electric field induced second harmonic generation
# gamma(-2omega;omega,omega,0) use the subkey EFISHG
#   aoresponse
#    ...
#    frequencies 0.1000 0.1000 0.0000 [Hartree]
#    efishg
#   end

# For the third harmonic generation gamma(-3omega;omega,omega,omega) use the
# subkey THG
#   aoresponse
#    ...
#    frequencies 0.1000 0.1000 0.1000 [Hartree]
#    thg
#   end

# Or in the general case for gamma(-(omega1+omega2+omega3);omega1,omega2,omega3)
# choose three input frequencies omega1, omega2, and omega3
#   aoresponse
#    ...
#    frequencies omega1 omega2 omega3 [Hartree]
#   end
```

Example: Verdet constants: H2O

Download H2O_Verdet.run

```
#! /bin/sh


# Specify the subkey MAGOPTROT in the AORESPONSE key to calculate the Verdet
# constant.


$AMSBIN/ams <<eor
System
  atoms
     O           0.000000    0.134692    0.000000
     H           0.869763   -0.538741    0.000000
     H          -0.869763   -0.538794    0.000000
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  allpoints
  aoresponse
    alda
    frequencies 0.088558 [Hartree]
    giao
    magoptrot
    scf converge 1d-6 iterations 25
  end
  basis
    core None
    type TZP
  end
  numericalquality Good
  symmetry nosym
  xc
    gga revPBE
    lda vwn
  end
  Relativity
    Level None
  End
EndEngine
eor
```

Example: Dispersion Coefficients: HF

Download Disper_HF.run

```
#! /bin/sh


# General dispersion coefficients (beyond de dipole-dipole C6 interaction
# coefficient) are computed with the auxiliary program DISPER. It uses two
# output files from previous ADF Response calculations. In the example, the two
# ADF runs are one and the same and the relevant TENSOR output file is used
# twice.

$AMSBIN/ams <<eor
System
  atoms
     H  0 0  -0.8708056087
     F  0 0   0.04619439132
  end
end

Task SinglePoint

Engine ADF
  title Van der Waals coefficients HF
  basis
    core Small
    type DZP
```

(continues on next page)

(continued from previous page)

```
  end
  response
    allcomponents
    alltensor
    maxwaals 8
    vanderwaals 7
  end
  symmetry C(8v)
EndEngine

eor

# Polarizabilities are computed at 7 (imaginary) frequencies between 0 and
# infinity. The program determines internally the actual frequency values in
# this range to use. The user only specifies the number of them, thereby
# determining the precision of, in fact, a numerical integration over the zero-
# infinity frequency range. A value of 7 is rather low.

# MaxWaals determines that not only the C6 but also C7 and C8 coefficients are
# computed. A value higher than 8 would not be recommended, because the
# available basis sets would be inadequate for higher coefficients.

# In DISPER calculations the preparatory Response calculation must use the
# AllTensor and AllComponents subkeys.

# The calculation produces a file TENSOR. The subsequent DISPER run uses two
# such files. In this example, both are taken from the same ADF run, copying the
# TENSOR file to, respectively, tensorA and tensorB. These names are prescribed
# for a DISPER calculation.

cp ams.results/TENSOR tensorA
cp ams.results/TENSOR tensorB

$AMSBIN/disper -n1 <<eor
eor

# The DISPER program needs no other input than just the files tensorA and
# tensorB, which must both be present as local files. Note the '-n1' flag: this
# enforces that a single-node (non-parallel) run is performed. The current
# implementation does not support parallelization of DISPER, because the kid
# processes may not have the (local to the master!) files tensorA and tensorB.
```

Example: Optical Rotation Dispersion (ORD): DMO

Download `DMO_ORD.run`

```
#!/bin/sh


# If the subkey OPTICALROTATION is included in the key RESPONSE, the (frequency
# dependent) optical rotation is calculated.

# Note: results will be physically meaningless due to small basis set. purpose
# of this job is to provide a test case for the ORD implementation

```

(continues on next page)

(continued from previous page)

```
# Do not use less strict convergence criteria than default, better to use
# tighter criteria. The approximations in the evaluation of the integrals one
# makes with the linear scaling techniques are effectively switched off by
# setting LINEARSCALING 100 (recommended to use this).


$AMSBIN/ams <<eor
System
  atoms
    O  0.0                 0.0                     1.141417145729065
    C -0.5970248579978943 -0.4282023310661316   -0.09432400763034821
    C  0.5970248579978943  0.4282023310661316   -0.09432400763034821
    H -0.4430356621742249 -1.487679243087769    -0.302562028169632
    H  0.4430356621742249  1.487679243087769    -0.302562028169632
    C -1.97874128818512    0.09404171258211136  -0.3866550326347351
    H -2.723244667053223  -0.4288762211799622    0.2206400185823441
    H -2.043423652648926   1.15996515750885     -0.1578925102949142
    H -2.236041307449341  -0.05517048016190529  -1.439977645874023
    C  1.97874128818512   -0.09404171258211136  -0.3866550326347351
    H  2.236041307449341   0.05517048016190529  -1.439977645874023
    H  2.723244667053223   0.4288762211799622    0.2206400185823441
    H  2.043423652648926  -1.15996515750885     -0.1578925102949142
  end
end

Task SinglePoint

Engine ADF
  title dimethyloxirane, ORD
  comment
     results will be physically meaningless due to small basis set.
     purpose of this job is to provide a test case for the ORD implementation
  end
  beckegrid
    quality good
  end
  linearscaling 100
  end
  basis
    core Small
    type DZP
  end
  response
    allcomponents
    opticalrotation
  end
  xc
    gga becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: ORD, lifetime effects (key AORESPONSE): DMO

Download DMO_ORD_aoresponse.run

```
#!/bin/sh


# If the subkey OPTICALROTATION is included in the key AORESPONSE, the
# (frequency dependent) optical rotation is calculated. In this example lifetime
# effects are included. This test example consists of two ORD calculations: one
# with and one without the velocity gauge.

# Note: results will be physically meaningless due to small basis set. purpose
# of this job is to provide a test case for the ORD implementation


AMS_JOBNAME=opticalrotation $AMSBIN/ams <<eor
System
  atoms
    O  0.0                   0.0                   1.141417145729065
    C -0.5970248579978943 -0.4282023310661316   -0.09432400763034821
    C  0.5970248579978943  0.4282023310661316   -0.09432400763034821
    H -0.4430356621742249 -1.487679243087769    -0.302562028169632
    H  0.4430356621742249  1.487679243087769    -0.302562028169632
    C -1.97874128818512    0.09404171258211136  -0.3866550326347351
    H -2.723244667053223  -0.4288762211799622    0.2206400185823441
    H -2.043423652648926   1.15996515750885     -0.1578925102949142
    H -2.236041307449341  -0.05517048016190529  -1.439977645874023
    C  1.97874128818512   -0.09404171258211136  -0.3866550326347351
    H  2.236041307449341   0.05517048016190529  -1.439977645874023
    H  2.723244667053223   0.4288762211799622    0.2206400185823441
    H  2.043423652648926  -1.15996515750885     -0.1578925102949142
  end
end

Task SinglePoint

Engine ADF
  title dimethyloxirane, ORD
  comment
     results will be physically meaningless due to small basis set.
     You may also want to increase the NumericalQuality to "Good".
     purpose of this job is to provide a test case for the ORD implementation
  end
  allpoints
  aoresponse
    alda
    frequencies 2.103923139 [eV]
    lifetime 0.007
    opticalrotation
    scf iter 20
  end
  basis
    core Small
    type DZP
  end
  numericalquality Normal
  xc
```

(continues on next page)

(continued from previous page)

```
    gga becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor


# In the second example the subkey OPTICALROTATION of the key AORESPONSE is
# changed into VELOCITYORD:


AMS_JOBNAME=velocityord $AMSBIN/ams <<eor
System
  atoms
    O  0.0                 0.0                    1.141417145729065
    C -0.5970248579978943 -0.4282023310661316   -0.09432400763034821
    C  0.5970248579978943  0.4282023310661316   -0.09432400763034821
    H -0.4430356621742249 -1.487679243087769    -0.302562028169632
    H  0.4430356621742249  1.487679243087769    -0.302562028169632
    C -1.97874128818512    0.09404171258211136  -0.3866550326347351
    H -2.723244667053223  -0.4288762211799622    0.2206400185823441
    H -2.043423652648926   1.15996515750885     -0.1578925102949142
    H -2.236041307449341  -0.05517048016190529  -1.439977645874023
    C  1.97874128818512   -0.09404171258211136  -0.3866550326347351
    H  2.236041307449341   0.05517048016190529  -1.439977645874023
    H  2.723244667053223   0.4288762211799622    0.2206400185823441
    H  2.043423652648926  -1.15996515750885     -0.1578925102949142
  end
end

Task SinglePoint

Engine ADF
  title dimethyloxirane, ORD with velocity gauge
  comment
     results will be physically  meaningless due to small basis set.
     You may also want to increase the NumericalQuality to "Good".
     purpose of this job is to provide a test case for the ORD implementation
  end
  allpoints
  aoresponse
    alda
    frequencies 2.103923139 [eV]
    lifetime 0.007
    scf iter 20
    velocityord
  end
  basis
    core Small
    type DZP
  end
  numericalquality Normal
  xc
    gga becke perdew
```

(continues on next page)

(continued from previous page)

```
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: Polarizability: first order perturbed density

Download ALPHA_DENSITY.run

```
#! /bin/sh


$AMSBIN/ams <<eor
System
  atoms
     O     0.00000000   -0.06692444    0.00000000
     H     0.76766392    0.53123258    0.00000000
     H    -0.76766392    0.53123258    0.00000000
  end
end

Task SinglePoint

Engine ADF
  aoresponse
    alda
    alpha
    frequencies 2.2489 [eV]
    lifetime 0.0037
    scf iter 200
  end
  beckegrid
    quality verygood
  end
  basis
    core None
    type TZP
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity Level=None
EndEngine
eor

$AMSBIN/densf << eor
adffile ams.results/adf.rkf
TAPE16file ams.results/TAPE16

Grid Inline
-4.0   -3.0   -3.0
```

(continues on next page)

(continued from previous page)

```
-4.0   -3.0   -2.2
-4.0   -3.0   -1.4
-4.0   -3.0   -0.6
-4.0   -3.0   0.2
-4.0   -3.0   1.0
-4.0   -3.0   1.8
-4.0   -3.0   2.6
-4.0   -2.2   -3.0
-4.0   -2.2   -2.2
-4.0   -2.2   -1.4
-4.0   -2.2   -0.6
-4.0   -2.2   0.2
-4.0   -2.2   1.0
-4.0   -2.2   1.8
-4.0   -2.2   2.6
-4.0   -1.4   -3.0
-4.0   -1.4   -2.2
-4.0   -1.4   -1.4
-4.0   -1.4   -0.6
-4.0   -1.4   0.2
-4.0   -1.4   1.0
-4.0   -1.4   1.8
-4.0   -1.4   2.6
-4.0   -0.6   -3.0
-4.0   -0.6   -2.2
-4.0   -0.6   -1.4
-4.0   -0.6   -0.6
-4.0   -0.6   0.2
-4.0   -0.6   1.0
-4.0   -0.6   1.8
-4.0   -0.6   2.6
-4.0   0.2   -3.0
-4.0   0.2   -2.2
-4.0   0.2   -1.4
-4.0   0.2   -0.6
-4.0   0.2   0.2
-4.0   0.2   1.0
-4.0   0.2   1.8
-4.0   0.2   2.6
-4.0   1.0   -3.0
-4.0   1.0   -2.2
-4.0   1.0   -1.4
-4.0   1.0   -0.6
-4.0   1.0   0.2
-4.0   1.0   1.0
-4.0   1.0   1.8
-4.0   1.0   2.6
-4.0   1.8   -3.0
-4.0   1.8   -2.2
-4.0   1.8   -1.4
-4.0   1.8   -0.6
-4.0   1.8   0.2
-4.0   1.8   1.0
-4.0   1.8   1.8
-4.0   1.8   2.6
-4.0   2.6   -3.0
-4.0   2.6   -2.2
```

(continues on next page)

(continued from previous page)

```
-4.0   2.6   -1.4
-4.0   2.6   -0.6
-4.0   2.6   0.2
-4.0   2.6   1.0
-4.0   2.6   1.8
-4.0   2.6   2.6
-4.0   3.4   -3.0
-4.0   3.4   -2.2
-4.0   3.4   -1.4
-4.0   3.4   -0.6
-4.0   3.4   0.2
-4.0   3.4   1.0
-4.0   3.4   1.8
-4.0   3.4   2.6
-3.2   -3.0   -3.0
-3.2   -3.0   -2.2
-3.2   -3.0   -1.4
-3.2   -3.0   -0.6
-3.2   -3.0   0.2
-3.2   -3.0   1.0
-3.2   -3.0   1.8
-3.2   -3.0   2.6
-3.2   -2.2   -3.0
-3.2   -2.2   -2.2
-3.2   -2.2   -1.4
-3.2   -2.2   -0.6
-3.2   -2.2   0.2
-3.2   -2.2   1.0
-3.2   -2.2   1.8
-3.2   -2.2   2.6
-3.2   -1.4   -3.0
-3.2   -1.4   -2.2
-3.2   -1.4   -1.4
-3.2   -1.4   -0.6
-3.2   -1.4   0.2
-3.2   -1.4   1.0
-3.2   -1.4   1.8
-3.2   -1.4   2.6
-3.2   -0.6   -3.0
-3.2   -0.6   -2.2
-3.2   -0.6   -1.4
-3.2   -0.6   -0.6
-3.2   -0.6   0.2
-3.2   -0.6   1.0
-3.2   -0.6   1.8
-3.2   -0.6   2.6
-3.2   0.2   -3.0
-3.2   0.2   -2.2
-3.2   0.2   -1.4
-3.2   0.2   -0.6
-3.2   0.2   0.2
-3.2   0.2   1.0
-3.2   0.2   1.8
-3.2   0.2   2.6
-3.2   1.0   -3.0
-3.2   1.0   -2.2
-3.2   1.0   -1.4
```

(continues on next page)

(continued from previous page)

```
-3.2   1.0   -0.6
-3.2   1.0   0.2
-3.2   1.0   1.0
-3.2   1.0   1.8
-3.2   1.0   2.6
-3.2   1.8   -3.0
-3.2   1.8   -2.2
-3.2   1.8   -1.4
-3.2   1.8   -0.6
-3.2   1.8   0.2
-3.2   1.8   1.0
-3.2   1.8   1.8
-3.2   1.8   2.6
-3.2   2.6   -3.0
-3.2   2.6   -2.2
-3.2   2.6   -1.4
-3.2   2.6   -0.6
-3.2   2.6   0.2
-3.2   2.6   1.0
-3.2   2.6   1.8
-3.2   2.6   2.6
-3.2   3.4   -3.0
-3.2   3.4   -2.2
-3.2   3.4   -1.4
-3.2   3.4   -0.6
-3.2   3.4   0.2
-3.2   3.4   1.0
-3.2   3.4   1.8
-3.2   3.4   2.6
-2.4   -3.0   -3.0
-2.4   -3.0   -2.2
-2.4   -3.0   -1.4
-2.4   -3.0   -0.6
-2.4   -3.0   0.2
-2.4   -3.0   1.0
-2.4   -3.0   1.8
-2.4   -3.0   2.6
-2.4   -2.2   -3.0
-2.4   -2.2   -2.2
-2.4   -2.2   -1.4
-2.4   -2.2   -0.6
-2.4   -2.2   0.2
-2.4   -2.2   1.0
-2.4   -2.2   1.8
-2.4   -2.2   2.6
-2.4   -1.4   -3.0
-2.4   -1.4   -2.2
-2.4   -1.4   -1.4
-2.4   -1.4   -0.6
-2.4   -1.4   0.2
-2.4   -1.4   1.0
-2.4   -1.4   1.8
-2.4   -1.4   2.6
-2.4   -0.6   -3.0
-2.4   -0.6   -2.2
-2.4   -0.6   -1.4
-2.4   -0.6   -0.6
```

(continues on next page)

(continued from previous page)

```
-2.4   -0.6   0.2
-2.4   -0.6   1.0
-2.4   -0.6   1.8
-2.4   -0.6   2.6
-2.4   0.2   -3.0
-2.4   0.2   -2.2
-2.4   0.2   -1.4
-2.4   0.2   -0.6
-2.4   0.2   0.2
-2.4   0.2   1.0
-2.4   0.2   1.8
-2.4   0.2   2.6
-2.4   1.0   -3.0
-2.4   1.0   -2.2
-2.4   1.0   -1.4
-2.4   1.0   -0.6
-2.4   1.0   0.2
-2.4   1.0   1.0
-2.4   1.0   1.8
-2.4   1.0   2.6
-2.4   1.8   -3.0
-2.4   1.8   -2.2
-2.4   1.8   -1.4
-2.4   1.8   -0.6
-2.4   1.8   0.2
-2.4   1.8   1.0
-2.4   1.8   1.8
-2.4   1.8   2.6
-2.4   2.6   -3.0
-2.4   2.6   -2.2
-2.4   2.6   -1.4
-2.4   2.6   -0.6
-2.4   2.6   0.2
-2.4   2.6   1.0
-2.4   2.6   1.8
-2.4   2.6   2.6
-2.4   3.4   -3.0
-2.4   3.4   -2.2
-2.4   3.4   -1.4
-2.4   3.4   -0.6
-2.4   3.4   0.2
-2.4   3.4   1.0
-2.4   3.4   1.8
-2.4   3.4   2.6
-1.6   -3.0   -3.0
-1.6   -3.0   -2.2
-1.6   -3.0   -1.4
-1.6   -3.0   -0.6
-1.6   -3.0   0.2
-1.6   -3.0   1.0
-1.6   -3.0   1.8
-1.6   -3.0   2.6
-1.6   -2.2   -3.0
-1.6   -2.2   -2.2
-1.6   -2.2   -1.4
-1.6   -2.2   -0.6
-1.6   -2.2   0.2
```

(continues on next page)

(continued from previous page)

```
-1.6   -2.2   1.0
-1.6   -2.2   1.8
-1.6   -2.2   2.6
-1.6   -1.4   -3.0
-1.6   -1.4   -2.2
-1.6   -1.4   -1.4
-1.6   -1.4   -0.6
-1.6   -1.4   0.2
-1.6   -1.4   1.0
-1.6   -1.4   1.8
-1.6   -1.4   2.6
-1.6   -0.6   -3.0
-1.6   -0.6   -2.2
-1.6   -0.6   -1.4
-1.6   -0.6   -0.6
-1.6   -0.6   0.2
-1.6   -0.6   1.0
-1.6   -0.6   1.8
-1.6   -0.6   2.6
-1.6   0.2   -3.0
-1.6   0.2   -2.2
-1.6   0.2   -1.4
-1.6   0.2   -0.6
-1.6   0.2   0.2
-1.6   0.2   1.0
-1.6   0.2   1.8
-1.6   0.2   2.6
-1.6   1.0   -3.0
-1.6   1.0   -2.2
-1.6   1.0   -1.4
-1.6   1.0   -0.6
-1.6   1.0   0.2
-1.6   1.0   1.0
-1.6   1.0   1.8
-1.6   1.0   2.6
-1.6   1.8   -3.0
-1.6   1.8   -2.2
-1.6   1.8   -1.4
-1.6   1.8   -0.6
-1.6   1.8   0.2
-1.6   1.8   1.0
-1.6   1.8   1.8
-1.6   1.8   2.6
-1.6   2.6   -3.0
-1.6   2.6   -2.2
-1.6   2.6   -1.4
-1.6   2.6   -0.6
-1.6   2.6   0.2
-1.6   2.6   1.0
-1.6   2.6   1.8
-1.6   2.6   2.6
-1.6   3.4   -3.0
-1.6   3.4   -2.2
-1.6   3.4   -1.4
-1.6   3.4   -0.6
-1.6   3.4   0.2
-1.6   3.4   1.0
```

(continues on next page)

(continued from previous page)

```
-1.6   3.4   1.8
-1.6   3.4   2.6
-0.8   -3.0   -3.0
-0.8   -3.0   -2.2
-0.8   -3.0   -1.4
-0.8   -3.0   -0.6
-0.8   -3.0   0.2
-0.8   -3.0   1.0
-0.8   -3.0   1.8
-0.8   -3.0   2.6
-0.8   -2.2   -3.0
-0.8   -2.2   -2.2
-0.8   -2.2   -1.4
-0.8   -2.2   -0.6
-0.8   -2.2   0.2
-0.8   -2.2   1.0
-0.8   -2.2   1.8
-0.8   -2.2   2.6
-0.8   -1.4   -3.0
-0.8   -1.4   -2.2
-0.8   -1.4   -1.4
-0.8   -1.4   -0.6
-0.8   -1.4   0.2
-0.8   -1.4   1.0
-0.8   -1.4   1.8
-0.8   -1.4   2.6
-0.8   -0.6   -3.0
-0.8   -0.6   -2.2
-0.8   -0.6   -1.4
-0.8   -0.6   -0.6
-0.8   -0.6   0.2
-0.8   -0.6   1.0
-0.8   -0.6   1.8
-0.8   -0.6   2.6
-0.8   0.2   -3.0
-0.8   0.2   -2.2
-0.8   0.2   -1.4
-0.8   0.2   -0.6
-0.8   0.2   0.2
-0.8   0.2   1.0
-0.8   0.2   1.8
-0.8   0.2   2.6
-0.8   1.0   -3.0
-0.8   1.0   -2.2
-0.8   1.0   -1.4
-0.8   1.0   -0.6
-0.8   1.0   0.2
-0.8   1.0   1.0
-0.8   1.0   1.8
-0.8   1.0   2.6
-0.8   1.8   -3.0
-0.8   1.8   -2.2
-0.8   1.8   -1.4
-0.8   1.8   -0.6
-0.8   1.8   0.2
-0.8   1.8   1.0
-0.8   1.8   1.8
```

(continues on next page)

(continued from previous page)

```
-0.8   1.8   2.6
-0.8   2.6   -3.0
-0.8   2.6   -2.2
-0.8   2.6   -1.4
-0.8   2.6   -0.6
-0.8   2.6   0.2
-0.8   2.6   1.0
-0.8   2.6   1.8
-0.8   2.6   2.6
-0.8   3.4   -3.0
-0.8   3.4   -2.2
-0.8   3.4   -1.4
-0.8   3.4   -0.6
-0.8   3.4   0.2
-0.8   3.4   1.0
-0.8   3.4   1.8
-0.8   3.4   2.6
0.0   -3.0   -3.0
0.0   -3.0   -2.2
0.0   -3.0   -1.4
0.0   -3.0   -0.6
0.0   -3.0   0.2
0.0   -3.0   1.0
0.0   -3.0   1.8
0.0   -3.0   2.6
0.0   -2.2   -3.0
0.0   -2.2   -2.2
0.0   -2.2   -1.4
0.0   -2.2   -0.6
0.0   -2.2   0.2
0.0   -2.2   1.0
0.0   -2.2   1.8
0.0   -2.2   2.6
0.0   -1.4   -3.0
0.0   -1.4   -2.2
0.0   -1.4   -1.4
0.0   -1.4   -0.6
0.0   -1.4   0.2
0.0   -1.4   1.0
0.0   -1.4   1.8
0.0   -1.4   2.6
0.0   -0.6   -3.0
0.0   -0.6   -2.2
0.0   -0.6   -1.4
0.0   -0.6   -0.6
0.0   -0.6   0.2
0.0   -0.6   1.0
0.0   -0.6   1.8
0.0   -0.6   2.6
0.0   0.2   -3.0
0.0   0.2   -2.2
0.0   0.2   -1.4
0.0   0.2   -0.6
0.0   0.2   0.2
0.0   0.2   1.0
0.0   0.2   1.8
0.0   0.2   2.6
```

(continues on next page)

(continued from previous page)

```
0.0   1.0   -3.0
0.0   1.0   -2.2
0.0   1.0   -1.4
0.0   1.0   -0.6
0.0   1.0   0.2
0.0   1.0   1.0
0.0   1.0   1.8
0.0   1.0   2.6
0.0   1.8   -3.0
0.0   1.8   -2.2
0.0   1.8   -1.4
0.0   1.8   -0.6
0.0   1.8   0.2
0.0   1.8   1.0
0.0   1.8   1.8
0.0   1.8   2.6
0.0   2.6   -3.0
0.0   2.6   -2.2
0.0   2.6   -1.4
0.0   2.6   -0.6
0.0   2.6   0.2
0.0   2.6   1.0
0.0   2.6   1.8
0.0   2.6   2.6
0.0   3.4   -3.0
0.0   3.4   -2.2
0.0   3.4   -1.4
0.0   3.4   -0.6
0.0   3.4   0.2
0.0   3.4   1.0
0.0   3.4   1.8
0.0   3.4   2.6
0.8   -3.0   -3.0
0.8   -3.0   -2.2
0.8   -3.0   -1.4
0.8   -3.0   -0.6
0.8   -3.0   0.2
0.8   -3.0   1.0
0.8   -3.0   1.8
0.8   -3.0   2.6
0.8   -2.2   -3.0
0.8   -2.2   -2.2
0.8   -2.2   -1.4
0.8   -2.2   -0.6
0.8   -2.2   0.2
0.8   -2.2   1.0
0.8   -2.2   1.8
0.8   -2.2   2.6
0.8   -1.4   -3.0
0.8   -1.4   -2.2
0.8   -1.4   -1.4
0.8   -1.4   -0.6
0.8   -1.4   0.2
0.8   -1.4   1.0
0.8   -1.4   1.8
0.8   -1.4   2.6
0.8   -0.6   -3.0
```

(continues on next page)

(continued from previous page)

```
0.8   -0.6   -2.2
0.8   -0.6   -1.4
0.8   -0.6   -0.6
0.8   -0.6   0.2
0.8   -0.6   1.0
0.8   -0.6   1.8
0.8   -0.6   2.6
0.8   0.2   -3.0
0.8   0.2   -2.2
0.8   0.2   -1.4
0.8   0.2   -0.6
0.8   0.2   0.2
0.8   0.2   1.0
0.8   0.2   1.8
0.8   0.2   2.6
0.8   1.0   -3.0
0.8   1.0   -2.2
0.8   1.0   -1.4
0.8   1.0   -0.6
0.8   1.0   0.2
0.8   1.0   1.0
0.8   1.0   1.8
0.8   1.0   2.6
0.8   1.8   -3.0
0.8   1.8   -2.2
0.8   1.8   -1.4
0.8   1.8   -0.6
0.8   1.8   0.2
0.8   1.8   1.0
0.8   1.8   1.8
0.8   1.8   2.6
0.8   2.6   -3.0
0.8   2.6   -2.2
0.8   2.6   -1.4
0.8   2.6   -0.6
0.8   2.6   0.2
0.8   2.6   1.0
0.8   2.6   1.8
0.8   2.6   2.6
0.8   3.4   -3.0
0.8   3.4   -2.2
0.8   3.4   -1.4
0.8   3.4   -0.6
0.8   3.4   0.2
0.8   3.4   1.0
0.8   3.4   1.8
0.8   3.4   2.6
1.6   -3.0   -3.0
1.6   -3.0   -2.2
1.6   -3.0   -1.4
1.6   -3.0   -0.6
1.6   -3.0   0.2
1.6   -3.0   1.0
1.6   -3.0   1.8
1.6   -3.0   2.6
1.6   -2.2   -3.0
1.6   -2.2   -2.2
```

(continues on next page)

(continued from previous page)

```
1.6   -2.2   -1.4
1.6   -2.2   -0.6
1.6   -2.2   0.2
1.6   -2.2   1.0
1.6   -2.2   1.8
1.6   -2.2   2.6
1.6   -1.4   -3.0
1.6   -1.4   -2.2
1.6   -1.4   -1.4
1.6   -1.4   -0.6
1.6   -1.4   0.2
1.6   -1.4   1.0
1.6   -1.4   1.8
1.6   -1.4   2.6
1.6   -0.6   -3.0
1.6   -0.6   -2.2
1.6   -0.6   -1.4
1.6   -0.6   -0.6
1.6   -0.6   0.2
1.6   -0.6   1.0
1.6   -0.6   1.8
1.6   -0.6   2.6
1.6   0.2   -3.0
1.6   0.2   -2.2
1.6   0.2   -1.4
1.6   0.2   -0.6
1.6   0.2   0.2
1.6   0.2   1.0
1.6   0.2   1.8
1.6   0.2   2.6
1.6   1.0   -3.0
1.6   1.0   -2.2
1.6   1.0   -1.4
1.6   1.0   -0.6
1.6   1.0   0.2
1.6   1.0   1.0
1.6   1.0   1.8
1.6   1.0   2.6
1.6   1.8   -3.0
1.6   1.8   -2.2
1.6   1.8   -1.4
1.6   1.8   -0.6
1.6   1.8   0.2
1.6   1.8   1.0
1.6   1.8   1.8
1.6   1.8   2.6
1.6   2.6   -3.0
1.6   2.6   -2.2
1.6   2.6   -1.4
1.6   2.6   -0.6
1.6   2.6   0.2
1.6   2.6   1.0
1.6   2.6   1.8
1.6   2.6   2.6
1.6   3.4   -3.0
1.6   3.4   -2.2
1.6   3.4   -1.4
```

(continues on next page)

(continued from previous page)

```
1.6   3.4   -0.6
1.6   3.4   0.2
1.6   3.4   1.0
1.6   3.4   1.8
1.6   3.4   2.6
2.4   -3.0   -3.0
2.4   -3.0   -2.2
2.4   -3.0   -1.4
2.4   -3.0   -0.6
2.4   -3.0   0.2
2.4   -3.0   1.0
2.4   -3.0   1.8
2.4   -3.0   2.6
2.4   -2.2   -3.0
2.4   -2.2   -2.2
2.4   -2.2   -1.4
2.4   -2.2   -0.6
2.4   -2.2   0.2
2.4   -2.2   1.0
2.4   -2.2   1.8
2.4   -2.2   2.6
2.4   -1.4   -3.0
2.4   -1.4   -2.2
2.4   -1.4   -1.4
2.4   -1.4   -0.6
2.4   -1.4   0.2
2.4   -1.4   1.0
2.4   -1.4   1.8
2.4   -1.4   2.6
2.4   -0.6   -3.0
2.4   -0.6   -2.2
2.4   -0.6   -1.4
2.4   -0.6   -0.6
2.4   -0.6   0.2
2.4   -0.6   1.0
2.4   -0.6   1.8
2.4   -0.6   2.6
2.4   0.2   -3.0
2.4   0.2   -2.2
2.4   0.2   -1.4
2.4   0.2   -0.6
2.4   0.2   0.2
2.4   0.2   1.0
2.4   0.2   1.8
2.4   0.2   2.6
2.4   1.0   -3.0
2.4   1.0   -2.2
2.4   1.0   -1.4
2.4   1.0   -0.6
2.4   1.0   0.2
2.4   1.0   1.0
2.4   1.0   1.8
2.4   1.0   2.6
2.4   1.8   -3.0
2.4   1.8   -2.2
2.4   1.8   -1.4
2.4   1.8   -0.6
```

(continues on next page)

(continued from previous page)

```
2.4   1.8   0.2
2.4   1.8   1.0
2.4   1.8   1.8
2.4   1.8   2.6
2.4   2.6   -3.0
2.4   2.6   -2.2
2.4   2.6   -1.4
2.4   2.6   -0.6
2.4   2.6   0.2
2.4   2.6   1.0
2.4   2.6   1.8
2.4   2.6   2.6
2.4   3.4   -3.0
2.4   3.4   -2.2
2.4   3.4   -1.4
2.4   3.4   -0.6
2.4   3.4   0.2
2.4   3.4   1.0
2.4   3.4   1.8
2.4   3.4   2.6
3.2   -3.0   -3.0
3.2   -3.0   -2.2
3.2   -3.0   -1.4
3.2   -3.0   -0.6
3.2   -3.0   0.2
3.2   -3.0   1.0
3.2   -3.0   1.8
3.2   -3.0   2.6
3.2   -2.2   -3.0
3.2   -2.2   -2.2
3.2   -2.2   -1.4
3.2   -2.2   -0.6
3.2   -2.2   0.2
3.2   -2.2   1.0
3.2   -2.2   1.8
3.2   -2.2   2.6
3.2   -1.4   -3.0
3.2   -1.4   -2.2
3.2   -1.4   -1.4
3.2   -1.4   -0.6
3.2   -1.4   0.2
3.2   -1.4   1.0
3.2   -1.4   1.8
3.2   -1.4   2.6
3.2   -0.6   -3.0
3.2   -0.6   -2.2
3.2   -0.6   -1.4
3.2   -0.6   -0.6
3.2   -0.6   0.2
3.2   -0.6   1.0
3.2   -0.6   1.8
3.2   -0.6   2.6
3.2   0.2   -3.0
3.2   0.2   -2.2
3.2   0.2   -1.4
3.2   0.2   -0.6
3.2   0.2   0.2
```

(continues on next page)

(continued from previous page)

```
3.2   0.2   1.0
3.2   0.2   1.8
3.2   0.2   2.6
3.2   1.0   -3.0
3.2   1.0   -2.2
3.2   1.0   -1.4
3.2   1.0   -0.6
3.2   1.0   0.2
3.2   1.0   1.0
3.2   1.0   1.8
3.2   1.0   2.6
3.2   1.8   -3.0
3.2   1.8   -2.2
3.2   1.8   -1.4
3.2   1.8   -0.6
3.2   1.8   0.2
3.2   1.8   1.0
3.2   1.8   1.8
3.2   1.8   2.6
3.2   2.6   -3.0
3.2   2.6   -2.2
3.2   2.6   -1.4
3.2   2.6   -0.6
3.2   2.6   0.2
3.2   2.6   1.0
3.2   2.6   1.8
3.2   2.6   2.6
3.2   3.4   -3.0
3.2   3.4   -2.2
3.2   3.4   -1.4
3.2   3.4   -0.6
3.2   3.4   0.2
3.2   3.4   1.0
3.2   3.4   1.8
3.2   3.4   2.6
4.0   -3.0   -3.0
4.0   -3.0   -2.2
4.0   -3.0   -1.4
4.0   -3.0   -0.6
4.0   -3.0   0.2
4.0   -3.0   1.0
4.0   -3.0   1.8
4.0   -3.0   2.6
4.0   -2.2   -3.0
4.0   -2.2   -2.2
4.0   -2.2   -1.4
4.0   -2.2   -0.6
4.0   -2.2   0.2
4.0   -2.2   1.0
4.0   -2.2   1.8
4.0   -2.2   2.6
4.0   -1.4   -3.0
4.0   -1.4   -2.2
4.0   -1.4   -1.4
4.0   -1.4   -0.6
4.0   -1.4   0.2
4.0   -1.4   1.0
```

(continues on next page)

(continued from previous page)

```
4.0   -1.4   1.8
4.0   -1.4   2.6
4.0   -0.6   -3.0
4.0   -0.6   -2.2
4.0   -0.6   -1.4
4.0   -0.6   -0.6
4.0   -0.6   0.2
4.0   -0.6   1.0
4.0   -0.6   1.8
4.0   -0.6   2.6
4.0   0.2   -3.0
4.0   0.2   -2.2
4.0   0.2   -1.4
4.0   0.2   -0.6
4.0   0.2   0.2
4.0   0.2   1.0
4.0   0.2   1.8
4.0   0.2   2.6
4.0   1.0   -3.0
4.0   1.0   -2.2
4.0   1.0   -1.4
4.0   1.0   -0.6
4.0   1.0   0.2
4.0   1.0   1.0
4.0   1.0   1.8
4.0   1.0   2.6
4.0   1.8   -3.0
4.0   1.8   -2.2
4.0   1.8   -1.4
4.0   1.8   -0.6
4.0   1.8   0.2
4.0   1.8   1.0
4.0   1.8   1.8
4.0   1.8   2.6
4.0   2.6   -3.0
4.0   2.6   -2.2
4.0   2.6   -1.4
4.0   2.6   -0.6
4.0   2.6   0.2
4.0   2.6   1.0
4.0   2.6   1.8
4.0   2.6   2.6
4.0   3.4   -3.0
4.0   3.4   -2.2
4.0   3.4   -1.4
4.0   3.4   -0.6
4.0   3.4   0.2
4.0   3.4   1.0
4.0   3.4   1.8
4.0   3.4   2.6
End
Density scf
aoresponse alpha lifetime imag
eor
```

Example: Hyperpolarizabilities of He and H2

Download Hyperpol.run

```
#! /bin/sh


# This sample illustrates the computation of (hyper) polarizability tensors for
# the He atom and the H2 molecule.

# The symmetry is specified, because the Response module in ADF cannot yet
# handle the infinite symmetries ATOM, C(lin), D(lin).


$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/He
mv TAPE12 t12.rel

AMS_JOBNAME=He8 $AMSBIN/ams <<eor
System
  Atoms
     He 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create He  $AMSRESOURCES/ADF/Special/Vdiff/He.2s2p2d2f
  corepotentials t12.rel
     He 1
  end
EndEngine
eor

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/H
mv TAPE12 t12.rel

AMS_JOBNAME=H7 $AMSBIN/ams <<eor
System
  Atoms
     H 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create H  $AMSRESOURCES/ADF/Special/Vdiff/H.2p
  corepotentials t12.rel
     H 1
  end
EndEngine
eor

AMS_JOBNAME=H2 $AMSBIN/ams <<eor
System
  atoms
     H 0 0 -0.37305
     H 0 0  0.37305
  end
  ElectrostaticEmbedding
    ElectricField 0 0 0.001 [a.u.]
```

(continues on next page)

(continued from previous page)

```
  End
end

Task SinglePoint

Engine ADF
  title expt geometrie H2(VII),VWN
  fragments
     H H7.results/adf.rkf
  end
  noprint sfo,frag,functions
  numericalquality Good
  response
    allcomponents
    dynahyp
    hyperpol 0.03 [Hartree]
  end
  symmetry C(8v)
EndEngine
eor


# The Response data block specifies (AllComponents) that not only the (default)
# zz-dipole polarizability is to be computed, but the complete tensor. The
# subkey HyperPol instructs the program to compute hyper*polarizabilities and
# not only polarizabilities. The DynaHyp subkey implies that the *frequency-
# dependent (hyper)polarizability is calculated. In that case the main laser
# frequency has to be specified, in hartree units, after the HyperPol subkey.

# Only the first hyperpolarizability has been implemented in ADF. Some
# information on second hyperpolarizabilities can be obtained from the
# calculation of the first one in a finite field (ElectricField).

# In similar fashion the frequency-dependent hyperpolarizability is computed for
# He, but only the zzz-component because now the AllComponents subkey is
# omitted.


AMS_JOBNAME=He $AMSBIN/ams <<eor
System
  atoms
     He 0 0 0
  end
  ElectrostaticEmbedding
    ElectricField 0 0 0.001 [a.u.]
  End
end

Task SinglePoint

Engine ADF
  title hyperpolarizability He with the LB94 potential
  fragments
     He He8.results/adf.rkf
  end
  noprint sfo,frag,functions
  numericalquality Good
```

(continues on next page)

(continued from previous page)

```
  response
    dynahyp
    hyperpol 0.07 [Hartree]
  end
  scf
    accelerationmethod LISTi
    converge 1e-8
  end
  symmetry C(8v)
  xc
    gga LB94
  end
EndEngine
eor
```

Example: Damped Verdet constants: Propene

Download DampedVerdet.run

```
#!/bin/sh


# Specify the subkey MAGOPTROT in the AORESPONSE key to calculate the Verdet
# constant. Here it is specified together with the LIFETIME key, such that the
# real and imaginary part of the damped Verdet constant will be calculated.


$AMSBIN/ams <<eor
System
  atoms
     C   0.867000     1.441800     3.000000
     C   0.849400     2.777300     3.000000
     C   2.115500     0.591200     3.000000
     H  -0.088300     0.909000     3.000000
     H  -0.085900     3.336500     3.000000
     H   1.772400     3.363200     3.000000
     H   2.737100     0.793300     2.115200
     H   1.876900    -0.479100     3.000000
     H   2.737100     0.793300     3.884800
  end
end

Task SinglePoint

Engine ADF
  title Propene
  allpoints
  aoresponse
    alda
    frequencies 0.2 [Hartree]
    giao
    lifetime 0.007
    magoptrot
    scf converge 1d-5 iterations 25
  end
```

(continues on next page)

(continued from previous page)

```
  beckegrid
    quality good
  end
  basis
    core None
    type DZP
  end
  noprint sfo
  symmetry nosym
  xc
    model SAOP
  end
EndEngine

eor
```

Example: Static magnetizability: H2O

Download H2O_magnet.run

```
#! /bin/sh


# Basic example for a magnetizability calculation.

# One should set iterations=0 for STATIC magnetizability. If one does not use
# SYMMETRY NOSYM, one should set use ALLPOINTS for correct results in
# AORESPONSE.


AMS_JOBNAME=Static $AMSBIN/ams <<eor
System
  atoms
     O 0.0  0.0                 0.0
     H 0.0 -0.7569504380226135 -0.5858823657035828
     H 0.0  0.7569504380226135 -0.5858823657035828
  end
end

Task SinglePoint

Engine ADF
  allpoints
  aoresponse
    alda
    magneticpert
    scf iterations 0  converge 1e-3
  end
  linearscaling 100
  end
  basis
    core None
    type DZP
  end
  xc
```

(continues on next page)

(continued from previous page)

```
    gga revPBE
    lda
  end
EndEngine
eor

AMS_JOBNAME=Frequency $AMSBIN/ams <<eor
System
  atoms
     O 0.0  0.0                  0.0
     H 0.0 -0.7569504380226135 -0.5858823657035828
     H 0.0  0.7569504380226135 -0.5858823657035828
  end
end

Task SinglePoint

Engine ADF
  allpoints
  aoresponse
    alda
    frequencies 0.182253 0.197012 [Hartree]
    magneticpert
    scf iterations 30  converge 1e-3
  end
  basis
    core None
    type DZP
  end
  linearscaling 100
  end
  xc
    gga revPBE
    lda
  end
EndEngine
eor
```

Example: Dynamic magnetizability: H2O

Download H2O_TD_magnet.run

```
#!/bin/sh


# Example for time-dependent magnetizability with GIAOs (Gauge including atomic
# orbitals).


$AMSBIN/ams <<eor
System
  atoms
     O           0.000000    0.134692    0.000000
     H           0.869763   -0.538741    0.000000
     H          -0.869763   -0.538794    0.000000
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  title H2O, TD magnetizability
  allpoints
  aoresponse
    alda
    fitaoderiv
    frequencies 2.103923139 [eV]
    giao
    magneticpert
    scf conv 1d-6 iter 25
  end
  comment
     purpose of this job is to calculate time-dependent magnetizability
     with GIAOs
  end
  linearscaling 100
  end
  basis
    core None
    type TZP
  end
  numericalquality good
  symmetry nosym
  xc
    gga revPBE
    lda vwn
  end
EndEngine
eor
```

Example: Time-dependent current-density-functional theory: C2H4:

Download C2H4_TDCDFT.run

```
#!/bin/sh


# Calculation of excitation energies and response properties of C2 H4 , with the
# VK functional, thus using time-dependent current-density-functional theory.


AMS_JOBNAME=C2H4_ET-pVQZ $AMSBIN/ams <<eor
  Task SinglePoint
  System
    Atoms
       C 0.000000    0.000000    0.666318
       C 0.000000    0.000000   -0.666318
       H 0.000000    0.928431    1.239388
       H 0.000000   -0.928431    1.239388
       H 0.000000    0.928431   -1.239388
```

(continues on next page)

(continued from previous page)

```
      H 0.000000   -0.928431   -1.239388
    END
  END
  Engine ADF
    title C2H4 excitation energy calculation with the VK functional
    basis
      Type ET/ET-pVQZ
    END
    EXCITATIONS
    END
    CURRENTRESPONSE
    END
  EndEngine
eor


AMS_JOBNAME=C2H4_TZ2P $AMSBIN/ams <<eor
  Task SinglePoint
  System
    Atoms
      C 0.000000    0.000000    0.666318
      C 0.000000    0.000000   -0.666318
      H 0.000000    0.928431    1.239388
      H 0.000000   -0.928431    1.239388
      H 0.000000    0.928431   -1.239388
      H 0.000000   -0.928431   -1.239388
    END
  End
  Engine ADF
    title C2H4 response calculation with the VK functional
    basis
      Type TZ2P
    END
    RESPONSE
    ALLCOMPONENTS
    END
    CURRENTRESPONSE
    END
  EndEngine
eor
```

Example: Damped complex polarizabilities with POLTDDFT: Au10

Download Au10_POLTDDFT.run

```
#!/bin/sh

# Calculation of damped complex polarizabilities of Au10
# with POLTDDFT.
# It is very important to use basis sets with specially made
# auxiliary fit sets available in $AMSHOME/atomicdata/ADF/POLTDDFT
#
# Symmetry irreps are chosen which for which the electronic transitions
# are dipole allowed.
# The analysis option is used, which will give an analysis
```

(continues on next page)

(continued from previous page)

```
# of the absorption and CD spectrum in terms of single orbital transitions.
# If the regionsforanalysis option is used, the absorption and CD spectrum
# will be split in region_i -> region_j terms.

$AMSBIN/ams <<eor
System
  atoms
     Au      -1.44000000       0.00000000       2.88000000 region=Au4
     Au       1.44000000      -0.00000000      -2.88000000 region=Au4
     Au      -1.44000000       0.00000000      -2.88000000 region=Au4
     Au       1.44000000      -0.00000000       2.88000000 region=Au4
     Au      -0.00000000      -2.03647000      -1.44000000 region=Au6
     Au       0.00000000       2.03647000      -1.44000000 region=Au6
     Au       1.44000000      -0.00000000       0.00000000 region=Au6
     Au      -0.00000000      -2.03647000       1.44000000 region=Au6
     Au      -1.44000000       0.00000000      -0.00000000 region=Au6
     Au       0.00000000       2.03647000       1.44000000 region=Au6
  end
end

Task SinglePoint

Engine ADF
  title Au10 DZ GGA LB94
  basis
    Type POLTDDFT/DZ
  end
  poltddft
    irrep
       B1.u
       B2.u
       B3.u
    end
    analysis
    regionsforanalysis Au4 Au6
  end
  xc
    gga LB94
  end
EndEngine
eor
```

Example: POLTDDFT with hybrid functional: NH3

Download NH3_POLTDDFT.run

```
#!/bin/sh

# Calculation of damped complex polarizabilities of NH3 with POLTDDFT.
# A hybrid XC functionals is used.
# For hybrid functionals POLTDDFT wil use HDA (hybrid diagonal approximation).
# For speed reasons one should not use a too large value for CutOff.
# To further speed up HDA one can use the subkey HDA_fitted, as is
# done in the second example.
# It is very important to use basis sets with specially made
```

(continues on next page)

(continued from previous page)

```
# auxiliary fit sets available in $AMSHOME/atomicdata/ADF/POLTDDFT

AMS_JOBNAME=NH3 $AMSBIN/ams << eor
System
  symmetrize
  atoms
     N   0.000000    0.000000    0.000000
     H   0.956305    0.000000    0.292372
     H  -0.478152    0.828184    0.292372
     H  -0.478152   -0.828184    0.292372
  end
end

Task SinglePoint

Engine ADF
  Basis
    Type POLTDDFT/TZP
  End

  XC
    Hybrid B3LYP
  End

  POLTDDFT
    KGrid 40.
    NGrid 1600
    FreqRange 0.02 20.02
    NFreq 1000
    Lifetime 0.075
    Cutoff 20.
  END
EndEngine
eor

AMS_JOBNAME=NH3_HDA_fited $AMSBIN/ams << eor
System
  symmetrize
  atoms
     N   0.000000    0.000000    0.000000
     H   0.956305    0.000000    0.292372
     H  -0.478152    0.828184    0.292372
     H  -0.478152   -0.828184    0.292372
  end
end

Task SinglePoint

Engine ADF
  Basis
    Type POLTDDFT/TZP
  End

  XC
    Hybrid B3LYP
  End
```

(continues on next page)

(continued from previous page)

```
  POLTDDFT
    KGrid 40.
    NGrid 1600
    FreqRange 0.02 20.02
    NFreq 1000
    Lifetime 0.075
    Cutoff 20.
    HDA_fitted
  END
EndEngine
eor
```

14.4.6 Ligand Field DFT (LFDFT)

Example: Ligand Field DFT: Co 2+

Download `Co_LFDFT.run`

```
#!/bin/sh

# Application of the Ligand Field DFT approach for a Co^2+ d^7 electron
# configuration. Here Co is surrounded by 6 Water molecules.

# First an average of configuration calculation (AOC) is performed, where 7
# electrons are equally distributed over the 5 orbitals that have the most
# dominant Co 3d character. Depending on the electron configuration this might
# be a non-trivial task. Symmetry NOSYM should be specified.


SCM_LFDFT="$AMSHOME/examples/adf/Co_LFDFT/LFDFT"
export SCM_LFDFT


$AMSBIN/ams <<eor
System
  atoms
     Co        0.000000    0.000000    0.000000
     O         2.113495    0.000000    0.000000
     H         2.687264   -0.787336    0.000000
     H         2.687264    0.787336    0.000000
     O         0.000000   -2.016255    0.000000
     H         0.000000   -2.587300    0.791160
     H         0.000000   -2.587300   -0.791160
     O        -2.113495    0.000000    0.000000
     H        -2.687264   -0.787336    0.000000
     H        -2.687264    0.787336    0.000000
     O         0.000000    2.016255    0.000000
     H         0.000000    2.587300    0.791160
     H         0.000000    2.587300   -0.791160
     O         0.000000    0.000000    2.124849
     H         0.785128    0.000000    2.701737
     H        -0.785128    0.000000    2.701737
     O         0.000000    0.000000   -2.124849
     H         0.785128    0.000000   -2.701737
```

(continues on next page)

(continued from previous page)

```
     H         -0.785128    0.000000   -2.701737
  end
  charge 2
end

Task SinglePoint

Engine ADF
  irrepoccupations
     A 56 1.4 1.4 1.4 1.4 1.4
  end
  basis
    core small
    type TZP
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry nosym
  xc
    gga bp86
  end
EndEngine
eor


# When the AOC calculation is ready, you need to make sure that indeed the
# partially occupied orbitals are dominantly d orbitals. In the ADF output you
# can find the character of the MOs in the list of all MOs, ordered by energy,
# with the most significant SFO gross populations.

# Next two LFDFT calculations are performed, first without spin-orbit coupling
# (soc 0), next including spin-orbit coupling (soc 1). In this case there is 1
# shell, and the nlval for 3d is '3 2'. The MO indices should be the
# fractionally occupied levels of the AOC calculation (in this case 29 30 31 32
# 33).


$AMSBIN/lfdft <<eor
  adffile ams.results/adf.rkf
  nshell 1
  nlval1 3 2
  MOind1 29 30 31 32 33
  soc    0
eor

$AMSBIN/lfdft <<eor
  adffile ams.results/adf.rkf
  nshell 1
  nlval1 3 2
  MOind1 29 30 31 32 33
  soc    1
eor
```

Example: Ligand Field DFT: f-d transitions in Pr 3+

`Download Pr_LFDFT.run`

```
#!/bin/sh


# Application of the Ligand Field DFT approach for the atomic Pr 4f^2 -> Pr 4f^1
# 5d^1 transition.

# First an average of configuration calculation (AOC) is performed for the 4f^2
# electron configuration, where 2 electrons are equally distributed over the 7
# orbitals that have the most dominant Pr 4f character. Depending on the
# electron configuration this might be a non-trivial task.

SCM_LFDFT="$AMSHOME/examples/adf/Pr_LFDFT/LFDFT"
export SCM_LFDFT

AMS_JOBNAME=f2 $AMSBIN/ams <<eor
System
  atoms
     Pr        0.000000    0.000000    0.000000
  end
end

Task SinglePoint

Engine ADF
  irrepoccupations
     A 54 0.2857143 0.2857143 0.2857143 0.2857143 0.2857143 0.2857143 0.2857143
  end
  basis
    core none
    type ZORA/TZ2P+
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  symmetry nosym
EndEngine
eor

# Next the an LFDFT calculation is performed including spin-orbit coupling (soc
# 1). In this case there is 1 shell, and the nlval for 4f is '4 3'. The MO
# indices should be the fractionally occupied levels of the AOC calculation (28
# 29 30 31 32 33 34). Results are added on adf.rkf.


$AMSBIN/lfdft <<eor
 adffile f2.results/adf.rkf
 nshel 1
 nlval 4 3
 MOind 28 29 30 31 32 33 34
 soc 1
eor
```

(continues on next page)

(continued from previous page)

```
rm logfile

# Next an average of configuration calculation (AOC) is performed for the 4f^1
# 5d^1 electron configuration, where 1 electron is equally distributed over the
# 7 orbitals that have the most dominant Pr 4f character, and 1 electron is
# equally distributed over the 7 orbitals that have the most dominant Pr 5d
# character. Depending on the electron configuration this might be a non-trivial
# task.

AMS_JOBNAME=f1d1 $AMSBIN/ams <<eor
System
  atoms
     Pr         0.000000    0.000000    0.000000
  end
end

Task SinglePoint

Engine ADF
  irrepoccupations
     A 54 0.1428571 0.1428571 0.1428571 0.1428571 0.1428571 0.1428571 0.1428571 0.2 0.
↪2 0.2 0.2 0.2
  end
  basis
    core none
    type ZORA/TZ2P+
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  symmetry nosym
EndEngine
eor

# Next the an LFDFT calculation is performed including spin-orbit coupling for
# both 4f and 5d (soc 1 1). In this case there are 2 shells, and the nlval for
# 4f is '4 3', and for 5d '5 2'. The MO indices should be the fractionally
# occupied levels of the AOC calculation for 4f (28 29 30 31 32 33 34), and for
# 5d (35 36 37 38 39). Results are added on adf.rkf.

$AMSBIN/lfdft <<eor
 adffile f1d1.results/adf.rkf
 nshel 2
 nlval1 4 3
 nlval2 5 2
 MOind1 28 29 30 31 32 33 34
 MOind2 35 36 37 38 39
 soc 1 1
eor

rm logfile

# Next excitation energies are calculated for exciting from the ground state of
# the Pr 4f^2 -> Pr 4f^1 5d^1. Input for lfdft_tdm are the 2 .t21 files that were
# calculated previously. Results of the excitation energies are added on the
```

(continues on next page)

(continued from previous page)

```
# adf.rkf file belonging to state2.

$AMSBIN/lfdft_tdm <<eor
 state1 f2.results/adf.rkf
 state2 f1d1.results/adf.rkf
eor
```

Example: LFDFT: g-tensor of Co(acacen)

Download Co_LFDFT_gtensor.run

```
#!/bin/sh

# Application of the Ligand Field DFT approach for the calculation
# of the g-tensor. The g-tensor is only calculated for Kramer doublet states.
# This example calculates the g-tensor of the ground and excited doublet states
# of Co(acacen) with a Co d^7 electron configuration.

# First an average of configuration calculation (AOC) is performed, where 7
# electrons are equally distributed over the 5 orbitals that have the most
# dominant Co 3d character. Depending on the electron configuration this might
# be a non-trivial task. Here the Keeporbitals=0 and Freeze functionality is used,
# such that ADF will on successive SCF cycles assign electrons to the MOs
# that maximally resemble - in spatial form - those that were occupied in 0th
# (in fact 1st) cycle. Note that the orbitals that have the most
# dominant Co 3d character in the 1st cycle are orbitals 71, 72, 73, 74, and 75,
# whereas at the end of the SCF they are 71, 72, 73, 74, and 77.
#
# Symmetry NOSYM should be specified.


SCM_LFDFT="$AMSHOME/examples/adf/Co_LFDFT/LFDFT"
export SCM_LFDFT

$AMSBIN/ams <<eor
System
  Atoms
    Co      0.000000    0.000000    0.000000
    N       1.327385    1.278147    0.000000
    N       1.327385   -1.278147    0.000000
    O      -1.395357    1.224920    0.000000
    O      -1.395357   -1.224920    0.000000
    C       2.704031    0.764453    0.000000
    C       2.704031   -0.764453    0.000000
    C      -1.309408    2.515386    0.000000
    C      -1.309408   -2.515386    0.000000
    C       1.166666    2.611094    0.000000
    C       1.166666   -2.611094    0.000000
    C      -0.103243   -3.216502    0.000000
    C      -0.103243    3.216502    0.000000
    H       3.246281   -1.139682    0.884137
    H       3.246281    1.139682    0.884137
    H       3.246281    1.139682   -0.884137
    H       3.246281   -1.139682   -0.884137
    C      -2.626185    3.259046    0.000000
```

(continues on next page)

(continued from previous page)

```
   C     -2.626185   -3.259046    0.000000
   C      2.374732    3.518898    0.000000
   C      2.374732   -3.518898    0.000000
   H     -3.453721   -2.540862    0.000000
   H     -3.453721    2.540862    0.000000
   H     -0.145492    4.305797    0.000000
   H     -0.145492   -4.305797    0.000000
   H      3.008094    3.350507    0.884680
   H      3.008094   -3.350507    0.884680
   H      3.008094    3.350507   -0.884680
   H      3.008094   -3.350507   -0.884680
   H     -2.710536    3.904463    0.886944
   H     -2.710536   -3.904463    0.886944
   H     -2.710536    3.904463   -0.886944
   H     -2.710536   -3.904463   -0.886944
   H      2.061821   -4.568536    0.000000
   H      2.061821    4.568536    0.000000
  End
  Charge 0
End

task SinglePoint

Engine adf
  Symmetry NOSYM
  IrrepOccupations
   A 140 1.4 1.4 1.4 1.4 1.4
  End
  Occupations Keeporbitals=0 Freeze
  basis
     Core None
     Type DZ
     PerAtomType Symbol=Co File=ZORA/TZP/Co
  End
  XC
    GGA PBE
  End
EndEngine
eor

# When the AOC calculation is ready, you need to make sure that indeed the
# partially occupied orbitals are dominantly d orbitals. In the ADF output you
# can find the character of the MOs in the list of all MOs, ordered by energy,
# with the most significant SFO gross populations.

# First the LFDFT calculation is performed without spin-orbit coupling (soc 0),
# in which pure spin states are calculated, doublets and quartets in this case.
# Next the LFDFT calculation is performed including spin-orbit coupling (soc 1),
# which is needed for the g-tensor calculation. In this case there is 1
# shell, and the nlval for 3d is '3 2'. The MO indices should be the
# fractionally occupied levels of the AOC calculation (in this case 71 72 73 74 77).

# One should be careful when interpreting the g-tensor for 2 Kramer doublets
# that are close in energy. In the effective Hamiltonian used to interpret ESR
# experiments, an effective spin=3/2 might be used.

$AMSBIN/lfdft << eor
```

(continues on next page)

(continued from previous page)

```
 adffile ams.results/adf.rkf
 nshel 1
 nlval 3 2
 MOind 71 72 73 74 77
 soc 0.0
 DegeneracyThreshold 1.0E-5
eor

$AMSBIN/lfdft << eor
 adffile ams.results/adf.rkf
 nshel 1
 nlval 3 2
 MOind 71 72 73 74 77
 soc 1.0
 DegeneracyThreshold 1.0E-5
eor
```

Example: LFDFT: g-tensor of Cu(H2O)6

Download Cu_LFDFT_gtensor.run

```
#!/bin/sh

# Application of the Ligand Field DFT approach for the calculation
# of the g-tensor. The g-tensor is only calculated for doublet states.
# This example calculates the g-tensor of the ground and excited doublet states
# of [Cu(H2O)6]^2+ with a Cu^2+ d^9 electron configuration.

# First an average of configuration calculation (AOC) is performed, where 9
# electrons are equally distributed over the 5 orbitals that have the most
# dominant Cu 3d character. Depending on the electron configuration this might
# be a non-trivial task. Symmetry NOSYM should be specified.


SCM_LFDFT="$AMSHOME/examples/adf/Cu_LFDFT_gtensor/LFDFT"
export SCM_LFDFT

$AMSBIN/ams <<eor
System
  Atoms
   Cu        0.000000    0.000000    0.000000
   O         0.000000    0.000000    2.325873
   O         0.000000    0.000000   -2.325873
   O        -2.004009    0.000000    0.000000
   O         2.004009    0.000000    0.000000
   O         0.000000   -2.004009    0.000000
   O         0.000000    2.004009    0.000000
   H         0.000000    0.773171   -2.924278
   H         0.000000   -0.773171   -2.924278
   H        -0.794410    2.570047    0.000000
   H         0.794410    2.570047    0.000000
   H        -2.570047    0.000000   -0.794410
   H        -2.570047    0.000000    0.794410
   H         0.794410   -2.570047    0.000000
   H        -0.794410   -2.570047    0.000000
```

(continues on next page)

(continued from previous page)

```
   H         2.570047    0.000000   -0.794410
   H         2.570047    0.000000    0.794410
   H         0.000000   -0.773171    2.924278
   H         0.000000    0.773171    2.924278
  End
  Charge 2
End

task SinglePoint

Engine adf
  Symmetry NOSYM
  IrrepOccupations
    A 78 1.8 1.8 1.8 1.8 1.8
  End
  basis
     Core None
     Type TZP
  End
  XC
    GGA PBE
  End
EndEngine
eor

# When the AOC calculation is ready, you need to make sure that indeed the
# partially occupied orbitals are dominantly d orbitals. In the ADF output you
# can find the character of the MOs in the list of all MOs, ordered by energy,
# with the most significant SFO gross populations.

# Next the LFDFT calculation is performed including spin-orbit coupling (soc 1),
# which is needed for the g-tensor calculation. In this case there is 1
# shell, and the nlval for 3d is '3 2'. The MO indices should be the
# fractionally occupied levels of the AOC calculation (in this case 40 41 42 43 44).

$AMSBIN/lfdft << eor
 adffile ams.results/adf.rkf
 nshel 1
 nlval 3 2
 MOind 40 41 42 43 44
 soc 1.0
 DegeneracyThreshold 1.0E-5
eor
```

Example: LFDFT: XMCD 3d-4f transitions in Er 3+

Download Er_LFDFT_MCD.run

```
#!/bin/sh

# Application of the Ligand Field DFT approach for the calculation
# of the X-ray Magnetic Circular Dichroism (XMCD) for lanthanide ion and complexes.
# This example calculates the XMCD spectra of the Er3+ ion
# for an Er 4f^11 (ground state) -> Er 3d^9 4f^12 transition, in which case a 3d core
# electron is promoted to the 4f shell.
```

(continues on next page)

(continued from previous page)

```
# First an average of configuration calculation (AOC) is performed, where 11
# electrons are equally distributed over the 7 orbitals that have the most
# dominant Er 4f character. Depending on the electron configuration this might
# be a non-trivial task.
# Symmetry NOSYM should be specified.

SCM_LFDFT="$AMSHOME/examples/adf/Er_LFDFT_MCD/LFDFT"
export SCM_LFDFT

AMS_JOBNAME=GS $AMSBIN/ams <<eor
System
  Atoms
     Er      0.0000000000      0.0000000000      0.0000000000
  End
  Charge +3
End

task SinglePoint

Engine adf
  Unrestricted No
  Symmetry NOSYM
  IrrepOccupations
   A 54 1.571428571428571 1.571428571428571 1.571428571428571 1.571428571428571 1.
↪571428571428571 1.571428571428571 1.571428571428571
  End
  basis
     Core None
     Type TZP
  End
  XC
    GGA PBE
  End
EndEngine
eor

# When the AOC calculation is ready, you need to make sure that indeed the
# partially occupied orbitals are dominantly f orbitals. In the ADF output you
# can find the character of the MOs in the list of all MOs, ordered by energy,
# with the most significant SFO gross populations.

# Next the LFDFT calculation is performed including a magnetic field of 1 Tesla
# in the z-direction (Bfield 0 0 1),
# which is needed for the XMCD calculation. In this case there is 1
# shell, and the nlval for 4f is '4 3'. The MO indices should be the
# fractionally occupied levels of the AOC calculation (in this case 28 29 30 31 32 33␣
↪34).

$AMSBIN/lfdft << eor
 adffile GS.results/adf.rkf
 nshel 1
 nlval 4 3
 MOind 28 29 30 31 32 33 34
 soc 1.0
 Bfield 0.0 0.0 1.0
 DegeneracyThreshold 1.0E-8
```

(continues on next page)

(continued from previous page)

```
eor

# Then, we need to calculate the electronic structure corresponding to the Er
# core electron excitation. In this case, we promote 1 electron from 3d orbitals
# to the 4f. Therefore, an average of configuration calculation (AOC) is
# performed, where 9 electrons are equally distributed over the 5 orbitals
# that have the most dominant Er 3d character; and 11+1 electrons are
# equally distributed over the 7 orbitals that have the most dominant Er 4f
# character. Symmetry NOSYM should be specified.

AMS_JOBNAME=ES $AMSBIN/ams <<eor
System
  Atoms
     Er      0.0000000000      0.0000000000      0.0000000000
  End
  Charge +3
End

task SinglePoint

Engine adf
  Unrestricted No
  Symmetry NOSYM
  IrrepOccupations
   A 18 1.8 1.8 1.8 1.8 1.8 26 1.714285714285714 1.714285714285714 1.714285714285714␣
↪1.714285714285714 1.714285714285714 1.714285714285714 1.714285714285714
  End
  basis
     Core None
     Type TZP
  End
  XC
    GGA PBE
  End
EndEngine
eor

# When the AOC calculation is ready, you need to make sure that indeed the
# partially occupied orbitals are dominantly d and f orbitals. In the ADF output you
# can find the character of the MOs in the list of all MOs, ordered by energy,
# with the most significant SFO gross populations.

# Next the LFDFT calculation is performed, again including a magnetic field of 1␣
↪Tesla
# in the z-direction (Bfield 0 0 1),
# which is needed for the XMCD calculation. In this case there are 2
# shells, and the nlval for 3d is '3 2' and 4f is '4 3'. The MO indices should be the
# fractionally occupied levels of the AOC calculation
# (in this case 10, 11, 12, 13, and 14 for the 3d; and 28, 29, 30, 31, 32, 33, and 34␣
↪for 4f).

$AMSBIN/lfdft << eor
 adffile ES.results/adf.rkf
 nshel 2
 nlval1 3 2
 nlval2 4 3
 MOind1 10 11 12 13 14
```

(continues on next page)

(continued from previous page)

```
 MOind2 28 29 30 31 32 33 34
 soc 1.0 1.0
 Bfield 0.0 0.0 1.0
 DegeneracyThreshold 1.0E-08
eor

# Finally, we use lfdft_tdm to calculate the oscillator strengths for the
# Er 4f^11 (ground state) -> 3d^9 4f^12 transitions.
# Input for lfdft_tdm are the 2 adf.rkf files that were
# calculated previously. Results of the excitation energies are added on the
# adf.rkf file belonging to state2.

$AMSBIN/lfdft_tdm << eor
 STATE1 GS.results/adf.rkf
 STATE2 ES.results/adf.rkf
eor
```

14.4.7 NMR chemical shifts and spin-spin coupling constants

Example: NMR Chemical Shifts: HBr

Download HBr.run

```
#! /bin/sh


# Computation of the NMR chemical shifts for HBr.


# == Non-relativistic ==


AMS_JOBNAME=NR $AMSBIN/ams <<eor
System
  atoms
     H   .0000  .0000  .0000
     Br  .0000  .0000 1.4140
  end
end

Task SinglePoint

Engine ADF
  basis CreateOutput=Yes
  save TAPE10
  title HBr non-relativistic
  xc
    gga Becke Perdew
  end
  zlmfit
    quality good
  end
  Relativity
    Level None
```

(continues on next page)

(continued from previous page)

```
  End
EndEngine
eor

# The TAPE21 result file and TAPE10 of ADF must be present under those names for
# the NMR calculation
# The NMR program uses mostly only one input (block) key NMR. The subkeys
# specify what output is produced (OUT) and for which Nuclei the NMR data are
# computed and printed (NUC). See the ADF manual.

$AMSBIN/nmr <<eor
tape10file NR.results/TAPE10
adffile NR.results/adf.rkf
NMR
  Out  TENS
  Nuc  1  2
End
eor

rm logfile TAPE15


# == Scalar relativistic ZORA (default) ==


AMS_JOBNAME=SR $AMSBIN/ams <<eor
System
  atoms
     H   .0000  .0000  .0000
     Br  .0000  .0000 1.4140
  end
end

Task SinglePoint

Engine ADF
  basis CreateOutput=Yes

  save TAPE10
  title HBr scalar relativistic ZORA
  xc
    gga Becke Perdew
  end
  zlmfit
    quality good
  end
EndEngine
eor

$AMSBIN/nmr <<eor
tape10file SR.results/TAPE10
adffile SR.results/adf.rkf
NMR
  Out  TENS
  Nuc  1  2
End
eor
```

(continues on next page)

(continued from previous page)

```
rm logfile TAPE15


# == Relativistic spin-orbit ZORA ==


# Using Spin-Orbit implies that NOSYM symmetry must be used in the ADF
# calculation: the NMR program cannot handle symmetry calculations in
# combination with spin-orbit terms and will stop with an error message if you
# try to do so.


AMS_JOBNAME=SO $AMSBIN/ams <<eor
System
  atoms
     H   .0000  .0000  .0000
     Br  .0000  .0000 1.4140
  end
end

Task SinglePoint

Engine ADF
  basis CreateOutput=Yes
  relativity
    level spin-orbit
    formalism ZORA
  end
  save TAPE10
  symmetry NoSYM
  title HBr relativistic spinorbit ZORA
  xc
    gga Becke Perdew
  end
  zlmfit
    quality good
  end
EndEngine
eor


$AMSBIN/nmr <<eor
 tape10file SO.results/TAPE10
 adffile SO.results/adf.rkf
 NMR
   U1K BEST
   OUT TENS
   NUC  1 2
 End
eor

rm logfile TAPE15
```

Example: NMR Chemical Shifts: HgMeBr

Three examples: a non-relativistic, a ZORA scalar relativistic, and a ZORA Spin-Orbit coupled one.

Download HgMeBr_pnr.run

```
#!/bin/sh


# NMR data are computed for the 1st and 3rd nucleus only. The UIK subkey is used
# to indicate that certain terms are to be included in the 'U-matrix', which
# goes into the first-order change of the MO's due to the applied magnetic
# field. See the documentation for more information.

# The 'BEST' specification means the best (recommended) options for each
# relativistic option are included for this sub key. In a non-relativistic run
# it has no meaning. In a spin-orbit run it would include the ZORA Spin-Orbit
# terms for a ZORA calculation.

$AMSBIN/ams <<eor
System
  atoms
     Hg  0.0                 0.0                  0.0
     Br  0.0                 0.0                 -2.406
     C   0.0                 0.0                  2.074
     H   1.0289885545161663  0.0                  2.4628220089896287
     H  -0.5144942772580832  0.8911302284144289  2.4628220089896287
     H  -0.5144942772580832 -0.8911302284144289  2.4628220089896287
  end
end

Task SinglePoint

Engine ADF
  title HgMeBr
  basis
    Core Small
    Type DZ
    PerAtomType Symbol=C Core=None
  end
  noprint SFO, Frag, Functions
  numericalquality Good
  save TAPE10
  xc
    gga PW91
  end
  Relativity
    Level None
  End
EndEngine
eor

$AMSBIN/nmr << eor
tape10file ams.results/TAPE10
adffile ams.results/adf.rkf
NMR
  OUT ISO
  U1K BEST
```

(continues on next page)

(continued from previous page)

```
  NUC 1 3
END
eor
```

Download HgMeBr_zsc.run

```
#!/bin/sh


$AMSBIN/ams <<eor
System
  atoms
     Hg  0.0                 0.0                 0.0
     Br  0.0                 0.0                -2.406
     C   0.0                 0.0                 2.074
     H   1.0289885545161663  0.0                 2.4628220089896287
     H  -0.5144942772580832  0.8911302284144289  2.4628220089896287
     H  -0.5144942772580832 -0.8911302284144289  2.4628220089896287
  end
end

Task SinglePoint

Engine ADF
  title HgMeBr, scalar, ZORA
  basis
    core None
    type DZ
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  save TAPE10
  xc
    gga PW91
  end
EndEngine
eor

$AMSBIN/nmr  <<eor
tape10file ams.results/TAPE10
adffile ams.results/adf.rkf
NMR
  OUT ISO
  U1K BEST
  NUC 1 3
END
eor
```

Download HgMeBr_zso.run

```
#!/bin/sh


$AMSBIN/ams <<eor
```

(continues on next page)

(continued from previous page)

```
System
  atoms
     Hg  0.0                 0.0                 0.0
     Br  0.0                 0.0                -2.406
     C   0.0                 0.0                 2.074
     H   1.0289885545161663  0.0                 2.4628220089896287
     H  -0.5144942772580832  0.8911302284144289  2.4628220089896287
     H  -0.5144942772580832 -0.8911302284144289  2.4628220089896287
  end
end

Task SinglePoint

Engine ADF
  title HgMeBr, spinorbit, ZORA
  basis
    core None
    type DZ
  end
  numericalquality Good
  relativity
    level spin-orbit
    formalism ZORA
  end
  save TAPE10
  symmetry NOSYM
  xc
    gga PW91
  end
EndEngine
eor

$AMSBIN/nmr  <<eor
tape10file ams.results/TAPE10
adffile ams.results/adf.rkf
NMR
  OUT ISO
  U1K BEST
  NUC 1 3
END
eor
mv TAPE21 HgMeBr.t21
```

Example: NMR Chemical Shifts, SAOP potential: CH4

Download CH4_SAOP.run

```
#! /bin/sh


# Computation of the NMR chemical shifts for CH4, with the model potential
# SAOP.

# Important: use SAVE TAPE10. This is necessary for SAOP, since the nmr program
# does not know about SAOP or other model potentials. On TAPE10 the SCF
```

(continues on next page)

(continued from previous page)

```
# potential is written, which is read in by the nmr program.

# Note: For SAOP one needs an all-electron basis set (i.e. Core None)


$AMSBIN/ams <<eor
System
  atoms
     C  0                   0                   0
     H  0.6264250420707439  0.6264250420707439  0.6264250420707439
     H  0.6264250420707439 -0.6264250420707439 -0.6264250420707439
     H -0.6264250420707439  0.6264250420707439 -0.6264250420707439
     H -0.6264250420707439 -0.6264250420707439  0.6264250420707439
  end
end

Task SinglePoint

Engine ADF
  basis
    type TZ2P
    core None
  end
  save TAPE10
  xc
    model saop
  end
EndEngine
eor

$AMSBIN/nmr <<eor
  adffile ams.results/adf.rkf
  tape10file ams.results/TAPE10
  NMR
    Out  TENS
    Nuc  1  2
  End
eor
```

Example: NMR Nucleus-independent chemical shifts (NICS): PF3

Download `NMR_NICS.run`

```
#!/bin/sh


# The NMR program enables the calculation of so-called nucleus-independent
# chemical shifts (NICS). More details are available in the Properties Programs
# User's Guide.

# In the ADF run, the ElectrostaticEmbedding key is used to define points charges␣
↪with zero
# charge. The GHOSTS key in the nmr program then basically copies this block.
# For the interpretation of the results we refer to the literature.

```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=NICS $AMSBIN/ams <<eor
System
  atoms
     P         1.000000    0.000000    0.000000
     F         1.813256    1.425667    0.000000
     F         1.813256   -0.712834    1.234664
     F         1.813256   -0.712834   -1.234664
  end
  ElectrostaticEmbedding
    MultipolePotential
      Coordinates
         3.0 4.0 5.0 0.0
         1.0 2.0 3.0 0.0
      end
    end
  end
end

Task SinglePoint

Engine ADF
  title PF3 - scf
  basis
    core Large
    type TZP
    CreateOutput Yes
  end
  numericalquality Good
  save TAPE10
  symmetry NOSYM
  xc
    gga revPBE
  end
  Relativity
    Level None
  End
EndEngine
eor

$AMSBIN/nmr <<eor
adffile NICS.results/adf.rkf
tape10file NICS.results/TAPE10
NMR
  Out Iso Tens
  GHOSTS
     3.0 4.0 5.0
     1.0 2.0 3.0
  SUBEND
END
eor
```

Example: NMR with B3LYP: PF3

Download NMR_B3LYP.run

```
#!/bin/sh


# This example shows how to do hybrid calculation of NMR chemical shifts.

# One needs of course a hybrid functional in the XC block key in ADF. One should
# also use SAVE TAPE10, such that it is an input file in the nmr module.

# In the input for the nmr module one can add the key ZSOAO2007 to approximate
# the effect of spin on the nucleus in the spin-orbit coupled calculations.

# In the last example spin-orbit coupling is included. Symmetry should be NOSYM.


AMS_JOBNAME=NR $AMSBIN/ams <<eor
System
  atoms
     P       0.00000000       0.00000000       1.00000000
     F      -0.71283358       1.23466398       1.81325568
     F      -0.71283358      -1.23466398       1.81325568
     F       1.42566716       0.00000000       1.81325568
  end
end

Task SinglePoint

Engine ADF
  title PF3-NMR-B3LYP
  basis
    core None
    type DZP
  end
  noprint sfo
  numericalquality Good
  save TAPE10
  xc
    hybrid B3LYP
  end
  Relativity
    Level None
  End
EndEngine
eor

$AMSBIN/nmr <<eor
adffile NR.results/adf.rkf
tape10file NR.results/TAPE10
NMR
  U1K  BEST
  Out  TENS
  Nuc  1  2
  SCF 1.0e-4
End
eor
```

(continues on next page)

(continued from previous page)

```
rm logfile

AMS_JOBNAME=SR $AMSBIN/ams <<eor
System
  atoms
     P       0.00000000       0.00000000       1.00000000
     F      -0.71283358       1.23466398       1.81325568
     F      -0.71283358      -1.23466398       1.81325568
     F       1.42566716       0.00000000       1.81325568
  end
end

Task SinglePoint

Engine ADF
  title PF3-NMR-B3LYP ZORA SCALAR
  basis
    core None
    type DZP
  end
  noprint sfo
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  save TAPE10
  xc
    hybrid B3LYP
  end
EndEngine
eor

$AMSBIN/nmr <<eor
adffile SR.results/adf.rkf
tape10file SR.results/TAPE10
NMR
  U1K  BEST
  Out  TENS
  Nuc  1  2
  SCF 1.0e-4
End
eor

rm logfile

AMS_JOBNAME=SO $AMSBIN/ams <<eor
System
  atoms
     P         1.000000    0.000000    0.000000
     F         1.813256    1.425667    0.000000
     F         1.813256   -0.712834    1.234664
     F         1.813256   -0.712834   -1.234664
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title PF3-NMR-B3LYP ZORA SPINORBIT
  basis
    core None
    type DZP
  end
  noprint sfo
  numericalquality Good
  relativity
    level spin-orbit
    formalism ZORA
  end
  rihartreefock
    fitsetquality Normal
  end
  save TAPE10
  symmetry nosym
  xc
    hybrid B3LYP
  end
EndEngine
eor

$AMSBIN/nmr <<eor
adffile SO.results/adf.rkf
tape10file SO.results/TAPE10
NMR
  USE  SO1C
  U1K  BEST
  Out  TENS
  Nuc  1  2
  SCF 1.0e-4
End
eor
rm logfile TAPE15

$AMSBIN/nmr <<eor
adffile SO.results/adf.rkf
tape10file SO.results/TAPE10
NMR
  U1K  BEST
  Out  TENS
  Nuc  1  2
  SCF 1.0e-4
End
eor
rm logfile TAPE15
```

Example: open shell NMR (pNMR) shielding: O2

Download O2_pNMR.run

```
#!/bin/sh

# Example calculation open shell NMR (pNMR) chemical shielding.

# This can only be done at the scalar relativistic ZORA level,
# symmetry NOSYM, all electron basis sets,
# and one needs to include the key ALLINONE in the input for NMR

# Needed are the ESR g-tensor, the ESR hyperfine interaction on the nuclei,
# and the orbital dependent part of the NMR chemical shielding.
# The calculation of the orbital dependent part of the NMR chemical shielding
# should be done at the end.
# The ESR g-tensor and ESR A-tensor will be written to adf.rkf, and the nmr
# program will use those to calculate a temperature dependent contribution
# to the NMR chemical shielding.
# Effects due to zero-field splitting are neglected (possibly important for S>1/2).

# The temperature dependent part can be huge and is highly dependent on the
# calculated A-tensor (hyperfine interaction), which is highly dependent
# on the used basis set and used XC-functional, especially if the spin-polarization␣
↪effects
# on the nuclei are important, like this is the case in this example for O2.
# For an accurate evaluation of the spin-polarization density at the nucleus it is
# important to use an all-electron basis set for the nuclei that one is interested in,
# avoiding the frozen core approximation.
# The basis sets in the directories ZORA/TZ2P-J, ZORA/QZ4P-J are
# recommended for accurate hyperfine calculations.
# For heavy elements the incorporation of a Gaussian finite nucleus model can be␣
↪important.
# However, one should have really large basis sets with tight basis functions to␣
↪observe this
# effect in calculations.
# The basis sets in the directories ZORA/TZ2P-J, ZORA/QZ4P-J are suitable for finite␣
↪nucleus calculations.

# Remark: do not use restricted open shell Kohn-Sham (ROKS), since that would␣
↪(incorrectly) give
# (almost) zero hyperfine interaction.

AMS_JOBNAME=SR "$AMSBIN/ams" << eor
Task SinglePoint
System
    Atoms
        O  0.000 0.000 -0.604
        O  0.000 0.000  0.604
    End
End

Engine ADF
    Basis
        Type TZ2P-J
        Core None
    End
    Occupations IntegerAufbau
```

(continues on next page)

(continued from previous page)

```
    Unrestricted Yes
    SpinPolarization 2
    NumericalQuality Good
    Symmetry NOSYM
    XC
        GGA PBE
    End
    SAVE TAPE10
EndEngine
eor

# Calculation of the ESR A-tensors, which are written to adf.rkf

"$AMSBIN/cpl" << eor
adffile SR.results/adf.rkf
tape10file SR.results/TAPE10
hyperfine
  atoms 1 2
END
eor
rm logfile TAPE15

# Calculation of the ESR g-tensor, which is written to adf.rkf

"$AMSBIN/nmr" << eor
adffile SR.results/adf.rkf
tape10file SR.results/TAPE10
NMR
  gfactors
  out iso tens
  u1k best
  calc all
END
eor
rm logfile TAPE15

# Calculation of the orbital dependent part of the NMR chemical shielding
# The ESR g-tensor and ESR A-tensors will be read from adf.rkf, and the nmr
# program will use those to calculate a temperature dependent contribution
# to the pNMR chemical shielding.
# ALLINONE is required.

"$AMSBIN/nmr" << eor
adffile SR.results/adf.rkf
tape10file SR.results/TAPE10
ALLINONE
NMR
    out iso tens
    atoms 1 2
    u1k best
    calc all
End
eor
rm logfile TAPE15
```

Example: NMR Spin-spin coupling constants: C2H2

Download CPL_C2H2.run

```
#!/bin/sh


# Non-relativistic calculation
# ============================

# A calculation of NMR nuclear spin-spin coupling constants (NSCCs).

# As explained in the ADF manual, the quality of a calculation for spin-spin
# coupling constants, using the program 'CPL', depends largely on the preceding
# ADF calculation, which produces the Kohn-Sham orbitals and orbital energies,
# used as a starting point.

# One of the quality-determining factors is the chosen basis set. It should be
# sufficiently flexible near the nucleus. Although the all-electron basis TZ2P
# is chosen in this example, it is recommendable to add more functions to the
# basis sets near the nucleus in case of heavy elements. One could start from a
# ZORA/QZ4P basis for example.

# The NOSYM symmetry currently needs to be specified in ADF to enable the CPL
# program to work correctly.


AMS_JOBNAME=first $AMSBIN/ams <<eor
System
  atoms
     C 0.0  0.0  0.0
     C 0.0  0.0  1.20692
     H 0.0  0.0  2.26672
     H 0.0  0.0 -1.0598
  end
end

Task SinglePoint

Engine ADF
  title C2H2 nrel
  beckegrid
    quality good
  end
  basis
    type TZ2P
    core None
  end
  save TAPE10
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
# The CPL program can run in parallel.

# The specification of what needs to be calculated is given in the nmrcoupling
# block key.

# coupling without the SD term
$AMSBIN/cpl <<eor
  adffile first.results/adf.rkf
  tape10file first.results/TAPE10
  nmrcoupling
    dso
    pso
    scf converge=1e-7
    nuclei 1 2 3 4
    nuclei 3 4
  end
eor
rm logfile


# In this first example, the SD subkey is left out, as this would lead to a very
# strong increase in the required CPU time. The SD subkey is included in the
# second CPL run. That subkey controls the calculation of the so-called spin-
# dipole term.

# The subkeys dso and pso specify that, respectively, the diamagnetic and
# paramagnetic orbital terms will be calculated. The often dominant Fermi
# contact term (FC) is calculated by default and therefore does not have to be
# specified explicitly.

# The scf convergence subkey, in this context, refers to the convergence for the
# solution of the coupled-perturbed Kohn-sham equations which need to be solved
# to obtain to spin-spin couplings.

# The following lines
#   nuclei 1 2 3 4
#   nuclei 3 4
# specify that one coupled-perturbed Kohn-Sham calculation is performed where
# nucleus number 1 (according to the ordering in the ADF output) is the
# perturbing nucleus, and nuclei 2, 3, and 4 are the perturbed nuclei, and
# another coupled-perturbed Kohn-Sham calculation is performed where nucleus 3
# is the perturbing nucleus and nucleus 4 is the perturbed nucleus.

# The second CPL run also includes the spin-dipole (SD) term, through the SD
# subkey.

# The output of the CPL program first contains a lot of general information, a
# summary of the specified input, and then produces the desired numbers:

# It prints separately the different contributions (FC, DSO, PSO, SD) if
# specified in input and sums them up to a total number. Experimental NSCCs
# between two nuclei A and B are usually reported as J(A,B) in Hertz. From a
# computational point of view, the so-called reduced NSCCs K(A,B) are more
# convenient for comparisons. CPL outputs both. In this example, the Fermi-
# contact term is indeed dominant.
```

(continues on next page)

(continued from previous page)

```
# The first part of the output refers to the line
#   nuclei 1 2 3 4
# then the same thing is done for the second similar line where nucleus 3 is the
# perturbing nucleus.

# The output for the second CPL run looks very similar, but now the SD term is
# added to the Fermi contact term, resulting in much longer execution times.


# == Scalar relativistic and spin-orbit calculations ==

# The CPL program also enables calculations using scalar relativistic effects
# (ZORA) and/or spin-orbit effects.

# Schematically, this requires the following changes to the input file with
# respect to a regular spin-orbit calculation and a non-relativistic CPL
# calculation:

# steep (1s) functions may need to be added to the standard basis sets. the
# full-potential option for ZORA is needed in the create runs and all further
# runs: relativistic zora scalar full the molecular ADF calculation should
# contain the line relativistic zora full spinorbit the CPL input is unmodified
# with respect to the example given here. Please check the ADF manual for
# details on relativistic input options.


# coupling including the SD term

$AMSBIN/cpl <<eor
  adffile first.results/adf.rkf
  tape10file first.results/TAPE10
  nmrcoupling
    dso
    pso
    sd
    scf converge=1e-7
    nuclei 1 2 3 4
    nuclei 3 4
  end
eor
rm logfile

# redo the same set of calculations, but with a different orientation
# of the molecule this time. Only the CPL calculation without the SD
# term is carried out in order to keep the execution time for this example
# within reasonable limits

AMS_JOBNAME=second $AMSBIN/ams <<eor
System
  atoms
     C 0.0  0.0     0.0
     C 0.0  1.20692 0.0
     H 0.0  2.26672 0.0
     H 0.0 -1.0598  0.0
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title C2H2 nrel
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
  end
  save TAPE10
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
  Relativity
    Level None
  End
EndEngine
eor

# coupling without the SD term

$AMSBIN/cpl <<eor
  adffile second.results/adf.rkf
  tape10file second.results/TAPE10
  nmrcoupling
    dso
    pso
    scf converge=1e-7
    nuclei 1 2 3 4
    nuclei 3 4
  end
eor
rm logfile
```

Example: NMR Spin-spin coupling constants, hybrid PBE0: HF

Download CPL_HF_hybrid.run

```
#!/bin/sh


# A calculation of NMR nuclear spin-spin coupling constants (NSCCs) for the
# hybrid PBE0.

# The hybrid PBE0 is chosen as exchange-correlation potential in the ADF
# calculation. The key 'usespcode' is required for consistency reasons of the
# PBE0 implementation in ADF and the kernel that is used in the 'CPL' program,
# that calculates NMR spin-spin coupling constants. Symmetry should be NOSYM.
# The basis sets used are specially optimized all-electron basis sets for NMR
# spin-spin coupling calculations (in the directory
# $AMSHOME/atomicdata/ZORA/jcpl), which have extra tight functions, compared to
```

(continues on next page)

(continued from previous page)

```
# a default ADF basis set. The integration accuracy is extra high (Quality
# VeryGood).


$AMSBIN/ams <<eor
System
  atoms
     F 0.0000 0.0000 0.0000
     H 0.0000 0.0000 0.9170
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type ZORA/jcpl
  end
  beckegrid
    quality verygood
  end
  scf
    converge 1e-8 1e-7
  end
  symmetry nosym
  usespcode
  xc
    hybrid PBE0
  end
EndEngine
eor


# The first call to cpl is as follows.
# The key 'gga' is included to use the first-order GGA potential instead of the
# first-order VWN potential. The Hartree-Fock part of the kernel is included
# automatically if a hybrid potential is used in the ADF calculation.


$AMSBIN/cpl <<eor
   adffile ams.results/adf.rkf
   gga
   nmrcoupling
   dso
   pso
   scf converge=1e-6 iterations=20
      nuclei 1 2
   End
eor


# The second CPL run also includes the spin-dipole (SD) term, through the SD
# subkey, which is much more time-consuming.


$AMSBIN/cpl <<eor
```

(continues on next page)

(continued from previous page)

```
   adffile ams.results/adf.rkf
   gga
   nmrcoupling
   dso
   pso
   sd
   scf converge=1e-6 iterations=20
      nuclei 1 2
   End
eor
```

Example: NMR Spin-spin coupling constants, finite nucleus: PbH4

Download `PbH4_finitenuc.run`

```
#!/bin/sh


# Example for a finite nucleus calculation and the calculation of NMR spin-spin
# coupling constants.

# One of the quality-determining factors for the calculation of NMR coupling
# constants is the chosen basis set, especially one needs enough tight s
# functions. If one has a large enough basis set in the core region one can see
# an effect of using a finite size of the nucleus instead of a point nucleus,
# especially for heavy nuclei. Such large basis sets can be found for some
# elements in $AMSRESOURCES/ADF/ZORA/jcpl, which are basis sets especially designed
# for NMR spin-spin coupling calculations. In this example first a basis set for
# Pb is made which has many tight s functions. The file can be found in the
# example directory



# The large basis set for Pb is used in ADF calculations on PbH4 and the
# calculation of the NMR spin-spin coupling constants.


AMS_JOBNAME=PbH4 $AMSBIN/ams <<eor
System
  atoms
     Pb      0.000000      0.000000      0.000000
     H      -1.023703      1.023703      1.023703
     H       1.023703     -1.023703      1.023703
     H      -1.023703     -1.023703     -1.023703
     H       1.023703      1.023703     -1.023703
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality verygood
  end
  basis
```

(continues on next page)

(continued from previous page)

```
    Core None
    Type DZP
    PerAtomType Symbol=Pb File=$AMSHOME/examples/adf/PbH4_finitenuc/Pb
    CreateOutput Yes
  end
  nuclearmodel Gaussian
  print Nuclei
  relativity
    level scalar
    formalism ZORA
  end
  save TAPE10
  scf
    converge 1.0e-8 1.0e-6
    iterations 100
  end
  usespcode
  xc
    gga PBE
  end
EndEngine
eor

## end scalar run, now do the coupling constant

$AMSBIN/cpl <<eor
adffile PbH4.results/adf.rkf
tape10file PbH4.results/TAPE10
gga
Print Nuclei
nmrcoupling
 scf converge=1e-5 iterations=25
 nuclei 1 2
end
eor
```

14.4.8 ESR/EPR g-tensor, A-tensor, Q-tensor, ZFS

Example: ESR g-tensor, A-tensor, Q-tensor, D-tensor: HfV

`Download ESR_HfV.run`

```
#!/bin/sh


# For the ESR g-tensor and D-tensor (zero-field splitting, ZFS) the effect of
# spin-orbit coupling is important. For the ESR A-tensor and Q-tensor (EFG)
# spin-orbit coupling is less important.

# In this example first spin-orbit coupling is taken into account
# perturbatively. Next spin-orbit coupling is taken into account self-
# consistent, using the COLLINEAR keyword.

# Note that an all-electron calculation is carried out. This is relevant for the
```

(continues on next page)

(continued from previous page)

```
# computation of the A-tensor, the nuclear magnetic dipole hyperfine
# interaction, where an accurate value of the spin-polarization density at the
# nucleus is important. For the g-tensor this plays a minor role. However, for
# the g-tensor calculation that includes spin-orbit coupling perturbatively, all
# electron basis sets are necessary.

# In the first ADF calculation the A-tensor (block key ESR) is calculated
# without the effect of spin-orbit coupling included. The zero-field splitting
# (key ZFS) is calculated by including spin-orbit coupling perturbatively.


AMS_JOBNAME=HfV $AMSBIN/ams <<eor
System
  atoms
     Hf   0.0 0.0 0.0
     V    0.0 0.0 2.033
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  esr
  end
  basis
    core None
    type TZ2P
  end
  qtens
  relativity
    level scalar
    formalism ZORA
  end
  save TAPE21 TAPE10
  spinpolarization 3
  symmetry NoSym
  unrestricted
  zfs
EndEngine
eor


# In the next calculation the module nmr calculates the g-tensor (subkey
# GFACTORS) using spin-orbit coupling and the external magnetic field as
# perturbation.


$AMSBIN/nmr <<eor
  adffile HfV.results/adf.rkf
  tape10file HfV.results/TAPE10
  nmr
   gfactors
   u1k best
   out iso tens
```

(continues on next page)

(continued from previous page)

```
  end
eor

rm logfile TAPE15

# The module cpl can calculate the A-tensor (key HYPERFINE) using spin-orbit
# coupling and the nuclear magnetic field as perturbation. Note that one needs
# to set the SCF convergence criterium to a small value.

$AMSBIN/cpl <<eor
  adffile HfV.results/adf.rkf
  tape10file HfV.results/TAPE10
  hyperfine
   atoms 1 2 :: calculates A-tensor for atom 1 and 2, input order
   SCF Converge=1e-7
  end
eor
rm logfile


# ADF can calculate the g-tensor and A-tensor (block key ESR) using only the
# nuclear or external magnetic field as perturbation, since spin-orbit coupling
# can be taken into account self-consistently. However, in this case, degenerate
# perturbation theory is used. The collinear approximation is used (and symmetry
# NOSYM) to account for spin-polarization effects.


AMS_JOBNAME=HfV_SO $AMSBIN/ams <<eor
System
  atoms
     Hf   0.0 0.0 0.0
     V    0.0 0.0 2.033
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  esr
  end
  basis
    core None
    type TZ2P
  end
  qtens
  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization collinear
  end
  symmetry NoSym
  unrestricted
EndEngine
eor
```

Example: ESR g-tensor, A-tensor, self consistent spin-orbit coupling: VO

Download VO_collinear.run

```
#!/bin/sh


# The ESR parameters of VO are calculated with the collinear approximation for
# unrestricted Spin-Orbit coupled calculations. In this example the VO-molecule
# has three unpaired electrons.

# You calculate Electron Spin Resonance properties with the keywords ESR and
# QTENS. ESR is a block-type key and is used to compute the G-tensor or the
# Nuclear Magnetic Dipole Hyperfine interaction. QTENS is a simple key and
# invokes the computation of the Nuclear Electric Quadrupole Hyperfine
# interaction.

# Proper usage of the key ESR requires that you do one of the following:

# A Spin-Orbit calculation, spin-restricted, with exactly one unpaired electron,
# or (b) A Spin-Orbit calculation, spin-unrestricted in the collinear
# approximation, or (c) No Spin-Orbit terms and spin-unrestricted. In case (a)
# and (b) you obtain the G-tensor. In case (b) and (c) you get the Magnetic
# Dipole Hyperfine interaction.

# Note: in case (a) the program also prints a Magnetic Dipole Hyperfine
# interaction data, but these have then been computed without the terms from the
# spin-density at the nucleus. Note: in case (b) and (c) one can have more than
# one unpaired electron. Note: in case (b) one has to use symmetry NOSYM.

# Two calculations are performed:

# Scalar relativistic spin-unrestricted (case c)
# Spin-Orbit relativistic spin-unrestricted collinear (case b)

# We first calculate the Dipole Hyperfine interaction:
# a spin- unrestricted calculation without Spin-Orbit coupling.

# Note that one has to use ALLPOINTS in the calculation for a linear molecule to
# get results for the nuclear magnetic dipole hyperfine interaction. For an
# accurate calculation of the hyperfine interaction the numerical quality is set
# to VeryGood.

AMS_JOBNAME=scalar $AMSBIN/ams <<eor
System
  atoms
     V 0 0 0
     O 0 0 1.589
  End
End

Task SinglePoint

Engine ADF
  AllPoints

  ESR
  End
```

(continues on next page)

(continued from previous page)

```
  QTens

  ZFS

  basis
    core None
    type TZ2P
  End

  numericalquality verygood

  relativity
    level scalar
    formalism ZORA
  end

  Unrestricted
  SpinPolarization 3
  XC
    GGA Becke Perdew
  End
EndEngine

eor

# Then a spin-orbit coupled spin-unrestricted calculation is performed using the
# collinear approximation. Note that symmetry NOSYM is used.

AMS_JOBNAME=so $AMSBIN/ams <<eor
System
  atoms
     V 0 0 0
     O 0 0 1.589
  End
End

Task SinglePoint

Engine ADF
  ESR
  End

  QTens

  basis
    core None
    type TZ2P
  End

  numericalquality verygood

  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization collinear
  end
```

(continues on next page)

(continued from previous page)

```
  SCF
    AccelerationMethod LISTf
  End

  symmetry nosym

  Unrestricted

  XC
    GGA Becke Perdew
  End
EndEngine

eor
```

Example: ESR g-tensor, A-tensor, perturbative spin-orbit coupling: HgF

Download ESR_HgF_2der.run

```
#!/bin/sh


# This example calculates the ESR g-tensor and A-tensor for HgF. In this example
# first spin-orbit coupling is taken into account perturbatively.

# Note that an all-electron calculation is carried out. This is relevant for the
# computation of the A-tensor, the nuclear magnetic dipole hyperfine
# interaction, where an accurate value of the spin-polarization density at the
# nucleus is important. For the g-tensor this plays a minor role. However, for
# the g-tensor calculation that includes spin-orbit coupling perturbatively, all
# electron basis sets are necessary.

# In the first example the module nmr calculates the g-tensor (subkey GFACTORS)
# using spin-orbit coupling and the external magnetic field as perturbation.


AMS_JOBNAME=HgF $AMSBIN/ams <<eor
System
  atoms
     Hg        0.0 0.0 0.0
     F         0.0 0.0 2.804
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality verygood
  end
  basis
    core None
    type TZ2P
  end
```

(continues on next page)

(continued from previous page)

```
  relativity
    level scalar
    formalism ZORA
  end
  save TAPE21 TAPE10
  scf
    converge 1e-7 1e-7
    iterations 500
  end
  spinpolarization 1
  symmetry NoSym
  unrestricted
  usespcode
  xc
    gga PBE
  end
EndEngine
eor

$AMSBIN/nmr <<eor
  tape10file HgF.results/TAPE10
  adffile HgF.results/adf.rkf
  nmr
   gfactors
   u1k best
   out iso tens
  end
eor

# In the second example the module cpl calculates the A-tensor (key HYPERFINE)
# using spin-orbit coupling and the nuclear magnetic field as perturbation. Note
# that one needs to set the SCF convergence criterium to a small value. For an
# accurate calculation of the A-tensor one needs a very large basis set in the
# core region (especially tight s-functions), especially for heavy nuclei. If
# one has such a large basis set in the core region, one can also see an effect
# of using a finite size of the nucleus instead of a point nucleus. Such large
# basis sets can be found for some elements in $AMSRESOURCES/ADF/ZORA/jcpl, which
# are basis sets especially designed for ESR A-tensor and NMR spin-spin coupling
# calculations.


AMS_JOBNAME=HgF_jcpl $AMSBIN/ams <<eor
System
  atoms
      Hg        0.0 0.0 0.0
      F         0.0 0.0 2.804
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality verygood
  end
  basis
    PerAtomType Symbol=Hg File=ZORA/jcpl/Hg
```

(continues on next page)

(continued from previous page)

```
    PerAtomType Symbol=F  File=ZORA/jcpl/F
  end
  nuclearmodel Gaussian
  relativity
    level scalar
    formalism ZORA
  end
  scf
    converge 1e-7 1e-7
    iterations 500
  end
  spinpolarization 1
  symmetry NoSym
  unrestricted
  usespcode
  xc
    gga PBE
  end
EndEngine
eor

$AMSBIN/cpl <<eor
  adffile HgF_jcpl.results/adf.rkf
  gga
  hyperfine
   atoms 1 2 :: calculates A-tensor for atom 1 and 2, input order
   SCF Converge=1e-7
  end
eor
```

Example: ESR spin-restricted and spin-unrestricted: TiF3

Download ESR_TiF3.run

```
#!/bin/sh


# You calculate Electron Spin Resonance properties with the keywords ESR and
# QTENS. ESR is a block-type key and is used to compute the G-tensor or the
# Nuclear Magnetic Dipole Hyperfine interaction. QTENS is a simple key and
# invokes the computation of the Nuclear Electric Quadrupole Hyperfine
# interaction.

# Proper usage of the key ESR requires that you do one of the following:

# (a) A Spin-Orbit calculation, spin-restricted, with exactly one unpaired
#     electron, or
# (b) A Spin-Orbit calculation, spin-unrestricted in the collinear
#     approximation, or
# (c) No Spin-Orbit terms and spin-unrestricted.

# In case (a) and (b) you obtain the G-tensor. In case (b) and (c) you get the
# Magnetic Dipole Hyperfine interaction.

# Note: in case (a) the program also prints a Magnetic Dipole Hyperfine
```

(continues on next page)

(continued from previous page)

```
# interaction data, but these have then been computed without the terms from the
# spin-density at the nucleus. Note: in case (b) and (c) one can have more than
# one unpaired electron. Note: in case (b) one has to use symmetry NOSYM.

# Five calculations are performed:
# - Scalar relativistic spin-restricted
# - Scalar relativistic open shell spin-restricted
# - Scalar relativistic spin-unrestricted
# - Spin-Orbit relativistic spin-restricted
# - Spin-Orbit relativistic spin-unrestricted collinear

# First a scalar relativistic spin-restricted calculation is performed. The
# results of this calculation are used as a fragment in the next spin-
# unrestricted calculation, using only 0 SCF iterations, which is a way to get
# the scalar relativistic spin-restricted open shell result for the magnetic
# dipole hyperfine interaction.


AMS_JOBNAME=TiF3_SR $AMSBIN/ams <<eor
System
  atoms
     Ti  0      0                  0
     F   1.780  0                  0
     F  -0.89   1.5415252187363007 0
     F  -0.89  -1.5415252187363007 0
  end
end

Task SinglePoint

Engine ADF
  title TiF3  scalar relativistic restricted
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  noprint sfo frag functions
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor

AMS_JOBNAME=TiF3_SR_0 $AMSBIN/ams <<eor
System
  atoms
     Ti  0      0                  0 adf.f=TiF3
     F   1.780  0                  0 adf.f=TiF3
     F  -0.89   1.5415252187363007 0 adf.f=TiF3
     F  -0.89  -1.5415252187363007 0 adf.f=TiF3
  end
end
```

(continues on next page)

(continued from previous page)

```

Task SinglePoint

Engine ADF
  title TiF3  scalar relativistic open shell restricted
  esr
  end
  fragments
     TiF3 TiF3_SR.results/adf.rkf
  end
  noprint sfo frag functions
  qtens
  relativity
    level scalar
    formalism ZORA
  end
  scf
    iterations 0
  end
  spinpolarization 1
  unrestricted
  xc
    gga Becke Perdew
  end
EndEngine
eor

# Next a spin-unrestricted SCF calculation is performed to get the scalar
# relativistic spin-unrestricted result for the magnetic dipole hyperfine
# interaction.


AMS_JOBNAME=TiF3_SR_unrestricted $AMSBIN/ams <<eor
System
  atoms
     Ti  0      0                  0 adf.f=TiF3
     F   1.780  0                  0 adf.f=TiF3
     F  -0.89   1.5415252187363007 0 adf.f=TiF3
     F  -0.89  -1.5415252187363007 0 adf.f=TiF3
  end
end

Task SinglePoint

Engine ADF
  title TiF3  relativistic open shell unrestricted
  esr
  end
  fragments
     TiF3   TiF3_SR.results/adf.rkf
  end
  noprint sfo frag functions
  qtens
  relativity
    level scalar
    formalism ZORA
  end
```

(continues on next page)

(continued from previous page)

```
  spinpolarization 1
  unrestricted
  xc
    gga Becke Perdew
  end
EndEngine
eor

# Then, for the same molecule, we compute the G-tensor in a Spin-Orbit run
# (spin-restricted).

# The here-computed and printed Dipole Hyperfine interaction misses the terms
# from the spin-density at the nucleus: compare with the outcomes from the first
# calculation.

# In each of the calculations, the QTENS key invokes the computation of the
# Electric Quadrupole Hyperfine interaction.

# Note that an all-electron calculation is carried out. This is relevant for the
# computation of the A-tensor, the nuclear magnetic dipole hyperfine
# interaction, where an accurate value of the spin-polarization density at the
# nucleus is important. For the G-tensor (and also for the Q-tensor) this plays
# a minor role, but for reasons of consistency both calculations use the same
# basis set and (absence of) frozen core.


AMS_JOBNAME=TiF3_SO $AMSBIN/ams <<eor
System
  atoms
     Ti  0      0                  0 adf.f=TiF3
     F   1.780  0                  0 adf.f=TiF3
     F  -0.89   1.5415252187363007 0 adf.f=TiF3
     F  -0.89  -1.5415252187363007 0 adf.f=TiF3
  end
end

Task SinglePoint

Engine ADF
  title TiF3  relativistic spinorbit open shell restricted
  esr
  end
  fragments
     TiF3   TiF3_SR.results/adf.rkf
  end
  noprint sfo frag functions
  qtens
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
# Finally a spin-orbit coupled spin-unrestricted calculation is performed using
# the collinear approximation. Note that symmetry NOSYM is used.


AMS_JOBNAME=TiF3_SO_unrestricted $AMSBIN/ams <<eor
System
  atoms
     Ti   0       0                   0 adf.f=TiF3
     F    1.780   0                   0 adf.f=TiF3
     F   -0.89    1.5415252187363007  0 adf.f=TiF3
     F   -0.89   -1.5415252187363007  0 adf.f=TiF3
  end
end

Task SinglePoint

Engine ADF
  title TiF3  relativistic spinorbit open shell unrestricted collinear
  esr
  end
  fragments
     TiF3   TiF3_SR.results/adf.rkf
  end
  noprint sfo frag functions
  qtens
  relativity
    level spin-orbit
    formalism ZORA
    spinorbitmagnetization collinear
  end
  symmetry nosym
  unrestricted
  xc
    gga Becke Perdew
  end
EndEngine
eor
```

Example: ESR, X2C and RA-X2C: PdH

Download PdH_ESR_X2C.run

```
#! /bin/sh


# This example calculates the ESR A-tensor and g-tensor of PdH, using the X2C
# and RA-X2C relativistic methods. Four calculations are performed:
# - Scalar relativistic X2C open shell spin-restricted
# - Scalar relativistic RA-X2C open shell spin-restricted
# - Spin-Orbit relativistic X2C spin-unrestricted collinear
# - Spin-Orbit relativistic RA-X2C spin-unrestricted collinear

# The spin-orbit coupled spin-unrestricted calculation is performed using the
# collinear approximation, symmetry NOSYM is required in that case. A finite
# nucleus is used, which will be important for the calculated A-tensor,
```

(continues on next page)

(continued from previous page)

```
# especially for heavy nuclei.

# == Scalar relativistic X2C open shell spin-restricted ==

AMS_JOBNAME=X2C_SR $AMSBIN/ams <<eor
System
  atoms
     Pd 0 0 0
     H  0 0 1.529
  end
end

Task SinglePoint

Engine ADF
  esr
  end
  basis
    core None
    type QZ4P
  end
  nuclearmodel gaussian
  relativity
    level scalar
    formalism X2C
  end
  spinpolarization 1
  unrestricted
EndEngine
eor

# == Scalar relativistic RA-X2C open shell spin-restricted ==

AMS_JOBNAME=RAX2C_SR $AMSBIN/ams <<eor
System
  atoms
     Pd 0 0 0
     H  0 0 1.529
  end
end

Task SinglePoint

Engine ADF
  esr
  end
  basis
    core None
    type QZ4P
  end
  nuclearmodel gaussian
  relativity
    level scalar
    formalism RA-X2C
  end
  spinpolarization 1
  unrestricted
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor

# == Spin-Orbit relativistic X2C spin-unrestricted collinear ==

AMS_JOBNAME=X2C_SO $AMSBIN/ams <<eor
System
  atoms
     Pd 0 0 0
     H  0 0 1.529
  end
end

Task SinglePoint

Engine ADF
  esr
  end
  basis
    core None
    type QZ4P
  end
  nuclearmodel gaussian
  relativity
    level spin-orbit
    formalism X2C
    SpinOrbitMagnetization collinear
  end
  symmetry nosym
  unrestricted
EndEngine
eor

# == Spin-Orbit relativistic RA-X2C spin-unrestricted collinear ==

AMS_JOBNAME=RAX2C_SO $AMSBIN/ams <<eor
System
  atoms
     Pd 0 0 0
     H  0 0 1.529
  end
end

Task SinglePoint

Engine ADF
  esr
  end
  basis
    core None
    type QZ4P
  end
  nuclearmodel gaussian
  relativity
    level spin-orbit
    formalism RA-X2C
    SpinOrbitMagnetization collinear
```

(continues on next page)

(continued from previous page)

```
  end
  symmetry nosym
  unrestricted
EndEngine
eor
```

Example: Zero-field splitting (ZFS), ESR D-tensor: NH

Download NH_ZFS.run

```
#! /bin/sh


# The zero-field splitting (ZFS) can be calculated for open shell molecules with
# electron spin S<=1, using the key ZFS.

# Only the spin-orbit contribution to ZFS is evaluated. Can be used in
# combination with LDA and GGAs. RELATIVISTIC ZORA is also required.


$AMSBIN/ams <<eor
System
  atoms
     N          0.000000    0.000000    0.007767
     H          0.000000    0.000000   -1.043445
  end
end

Task SinglePoint

Engine ADF
  basis
    type DZP
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 2.0
  title NH Zero-field splitting
  unrestricted
  xc
    gga BECKE PERDEW
  end
  zfs
EndEngine
eor
```

Example: ZFS D tensor, including direct electron spin-spin part: Phenylnitrene

Download Phenylnitrene_ZFS.run

```
#! /bin/sh


# The zero-field splitting (ZFS) can be calculated for open shell molecules with
# electron spin S<=1, using the key ZFS.

# The direct electron spin-spin part and the spin-orbit contribution to ZFS is
# evaluated. Can be used in combination with LDA and GGAs. RELATIVISTIC ZORA is
# also required. The direct electron spin-spin part is calculated if HARTREEFOCK
# is included as separate keyword. Both Coulomb and (Hartree-Fock like) exchange
# contributions to the direct electron spin-spin term are calculated. In the
# spin-orbit coupling no Hartree-Fock like exchange contributions are included.

$AMSBIN/ams <<eor
System
  atoms
     C  0.000000  0.000000  1.072671
     C  0.000000  1.232517  0.336828
     C  0.000000  1.216609 -1.046705
     C  0.000000  0.000000 -1.747681
     C  0.000000 -1.216609 -1.046705
     C  0.000000 -1.232517  0.336828
     N  0.000000  0.000000  2.400797
     H  0.000000  2.164354  0.892961
     H  0.000000  2.155064 -1.594614
     H  0.000000  0.000000 -2.833691
     H  0.000000 -2.155064 -1.594614
     H  0.000000 -2.164354  0.892961
  end
end

Task SinglePoint

Engine ADF
  hartreefock
  basis
    core None
    type TZ2P
  end
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 2.0
  unrestricted
  xc
    gga blyp
  end
  zfs
EndEngine
eor
```

14.4.9 EFG, Mössbauer

Example: Mössbauer spectroscopy: Ferrocene

Download Mossbauer.run

```
#! /bin/sh


# By default in ADF the electron density at the nuclei is calculated, no input
# key is required. The electron density at the nuclei could be used for the
# interpretation of isomer shifts in Mossbauer spectroscopy. The absolute
# electron density at a nucleus heavily depends on the accuracy of the basis set
# in the core region of this nucleus, especially if relativistic effects are
# included. Important is to use the same basis set, same exchange correlation
# functional, same numerical accuracy, if electron densities at nuclei in
# different molecules are compared. For the calculation of Mossbauer quadrupole
# splittings the key QTENS is required For 57 Fe quadrupole splittings will be
# written in units of mm/s, used in Mossbauer spectroscopy Example shows a
# calculation on ferrocene with anon-relativistic, and two with a scalar
# relativistic ZORA Hamiltonian using a different all electron basis set.

# First a non-relativistic calculation.

AMS_JOBNAME=NR $AMSBIN/ams <<eor
System
  symmetrize Yes
  atoms
     Fe         0.000000    0.000000    0.000000
     C          1.215650    0.000000    1.600813
     C          0.375656   -1.156152    1.600813
     C         -0.983481   -0.714541    1.600813
     C         -0.983481    0.714541    1.600813
     C          0.375656    1.156152    1.600813
     C          1.215650    0.000000   -1.600813
     C          0.375656    1.156152   -1.600813
     C         -0.983481    0.714541   -1.600813
     C         -0.983481   -0.714541   -1.600813
     C          0.375656   -1.156152   -1.600813
     H          2.310827    0.000000    1.629796
     H          0.714085   -2.197727    1.629796
     H         -1.869498   -1.358270    1.629796
     H         -1.869498    1.358270    1.629796
     H          0.714085    2.197727    1.629796
     H          2.310827    0.000000   -1.629796
     H          0.714085    2.197727   -1.629796
     H         -1.869498    1.358270   -1.629796
     H         -1.869498   -1.358270   -1.629796
     H          0.714085   -2.197727   -1.629796
  end
end

Task SinglePoint

Engine ADF
  title ferrocene
  exactdensity
```

(continues on next page)

(continued from previous page)

```
  basis
    core none
    type TZP
  end
  numericalquality Good
  qtens
  xc
    gga blyp
  end
  Relativity
    Level None
  End
EndEngine
eor


# Next the scalar relativistic ZORA calculations. ADF will also calculate the
# quadrupole splittings including the small component density, also called SR
# ZORA-4.


AMS_JOBNAME=SR $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     FE        0.000000    0.000000    0.000000
     C         1.215650    0.000000    1.600813
     C         0.375656   -1.156152    1.600813
     C        -0.983481   -0.714541    1.600813
     C        -0.983481    0.714541    1.600813
     C         0.375656    1.156152    1.600813
     C         1.215650    0.000000   -1.600813
     C         0.375656    1.156152   -1.600813
     C        -0.983481    0.714541   -1.600813
     C        -0.983481   -0.714541   -1.600813
     C         0.375656   -1.156152   -1.600813
     H         2.310827    0.000000    1.629796
     H         0.714085   -2.197727    1.629796
     H        -1.869498   -1.358270    1.629796
     H        -1.869498    1.358270    1.629796
     H         0.714085    2.197727    1.629796
     H         2.310827    0.000000   -1.629796
     H         0.714085    2.197727   -1.629796
     H        -1.869498    1.358270   -1.629796
     H        -1.869498   -1.358270   -1.629796
     H         0.714085   -2.197727   -1.629796
  end
end

Task SinglePoint

Engine ADF
  title ferrocene
  exactdensity
  basis
    core none
    type TZP
```

(continues on next page)

(continued from previous page)

```
  end
  numericalquality Good
  qtens
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga blyp
  end
EndEngine
eor


# Next a scalar relativistic calculation is performed with a much larger basis
# set (QZ4P) in the core region. Changing the basis set will have a large effect
# on the electron density at the nucleus and a noticeable effect on the
# calculated quadrupole splittings.


AMS_JOBNAME=SR_QZ4P $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     FE        0.000000    0.000000    0.000000
     C         1.215650    0.000000    1.600813
     C         0.375656   -1.156152    1.600813
     C        -0.983481   -0.714541    1.600813
     C        -0.983481    0.714541    1.600813
     C         0.375656    1.156152    1.600813
     C         1.215650    0.000000   -1.600813
     C         0.375656    1.156152   -1.600813
     C        -0.983481    0.714541   -1.600813
     C        -0.983481   -0.714541   -1.600813
     C         0.375656   -1.156152   -1.600813
     H         2.310827    0.000000    1.629796
     H         0.714085   -2.197727    1.629796
     H        -1.869498   -1.358270    1.629796
     H        -1.869498    1.358270    1.629796
     H         0.714085    2.197727    1.629796
     H         2.310827    0.000000   -1.629796
     H         0.714085    2.197727   -1.629796
     H        -1.869498    1.358270   -1.629796
     H        -1.869498   -1.358270   -1.629796
     H         0.714085   -2.197727   -1.629796
  end
end

Task SinglePoint

Engine ADF
  title ferrocene
  exactdensity
  basis
    core none
    type QZ4P
  end
```

(continues on next page)

(continued from previous page)

```
  numericalquality Good
  qtens
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga blyp
  end
EndEngine
eor
```

Example: Mössbauer with X2C: Hg compounds

Download Hg_Mossbauer_X2C.run

```
#! /bin/sh


# This example uses the relativistic spin-orbit coupled X2C method. By default
# in ADF the electron density at the nuclei is calculated, no input key is
# required. The electron density at the nuclei could be used for the
# interpretation of isomer shifts in Mossbauer spectroscopy. Typically one needs
# to perform a fit of the experimentally measured isomer shifts versus the
# calculated electron densities. The absolute electron density at a nucleus
# heavily depends on the accuracy of the basis set in the core region of this
# nucleus, especially if relativistic effects are included. Important is to use
# the same basis set, same exchange correlation functional, same numerical
# accuracy, if electron densities at nuclei in different molecules are compared.
# For the calculation of Mossbauer quadrupole splittings the key QTENS is
# required This example calculates the compounds Hg, HgF, HgF2, HgF4, and HgCl2.


AMS_JOBNAME=Hg_X2C $AMSBIN/ams <<eor
System
  atoms
     Hg 0 0 0
  end
end

Task SinglePoint

Engine ADF
  exactdensity
  basis
    core None
    type ZORA/DZ
  end
  nuclearmodel gaussian
  numericalquality good
  relativity
    level spin-orbit
    formalism X2C
  end
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor


AMS_JOBNAME=HgF_X2C $AMSBIN/ams <<eor
System
  atoms
     Hg 0 0 0
     F  0 0 2.007
  end
end

Task SinglePoint

Engine ADF
  exactdensity
  basis
    core None
    type ZORA/DZ
  end
  nuclearmodel gaussian
  numericalquality good
  qtens
  relativity
    Level spin-orbit
    Formalism X2C
    SpinOrbitMagnetization NonCollinear
  end
  symmetry nosym
  unrestricted
EndEngine
eor


AMS_JOBNAME=HgF2_X2C $AMSBIN/ams <<eor
System
  atoms
     Hg 0 0  0
     F  0 0  1.914
     F  0 0 -1.914
  end
end

Task SinglePoint

Engine ADF
  exactdensity
  basis
    core None
    type ZORA/DZ
  end
  nuclearmodel gaussian
  numericalquality good
  qtens
  relativity
    level spin-orbit
    formalism X2C
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor


AMS_JOBNAME=HgF4_X2C $AMSBIN/ams <<eor
System
  atoms
     Hg  0      0      0
     F   0     -1.885  0
     F   0      1.885  0
     F  -1.885  0      0
     F   1.885  0      0
  end
end

Task SinglePoint

Engine ADF
  exactdensity
  basis
    core None
    type ZORA/DZ
  end
  nuclearmodel gaussian
  numericalquality good
  qtens
  relativity
    level spin-orbit
    formalism X2C
  end
EndEngine
eor


AMS_JOBNAME=HgCl2_X2C $AMSBIN/ams <<eor
System
  atoms
     Hg 0 0  0
     Cl 0 0  2.252
     Cl 0 0 -2.252
  end
end

Task SinglePoint

Engine ADF
  exactdensity
  basis
    core None
    type ZORA/DZ
  end
  nuclearmodel gaussian
  numericalquality good
  qtens
  relativity
    level spin-orbit
    formalism X2C
```

(continues on next page)

(continued from previous page)

```
  end
EndEngine
eor
```

14.4.10 GW

Example: G_0W_0@PBE calculation: H2O

Download GW_H2O.run

```
#! /bin/sh

# GW calculation for Water. By default. The highest 5 occupied and lowest 5
# unoccupied states are calculated.

# We use an all-electron basis set since core-correlation effects are important

# For H2O, PBE will be a suitable reference since it is only weakly correlated and␣
↪the
# KS HOMO-LUMO gap will be reasonably large.

# RECOMMENDED: Good numerical quality

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
End

System
  Atoms
    O    2.220871067   0.026716792   0.000620476
    H    2.597492682  -0.411663274   0.766744858
    H    2.593135384  -0.449496183  -0.744782026
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type TZ2P
  end

  NumericalQuality GOOD
  RIHartreeFock fitsetQuality=Normal dependencyThreshold=1e-3
  Dependency bas=1e-4

  XC
    GGA PBE
  end

  GW
```

(continues on next page)

(continued from previous page)

```
    nstates 3
  END
EndEngine
eor
```

Example: G_0W_0@PBE0 calculation: O3

Download GW_O3.run

```
#! /bin/sh

# GW calculation on Ozone. By default. The highest 5 occupied and lowest 5
# unoccupied states are calculated.

# We use an all-electron basis set since core-correlation effects are important
# to capture all screening effects.

# Since for a GGA reference poles of the self-energy will be close to the highest
# (particle )QP solution, we use a hybrid reference. This shifts away the poles
# from the QP solution. We also use large imaginary time and imaginary frequency
# grids to ensure convergence of the QP energies despite the small HOMO-LUMO gap.

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
End

System
  Atoms
      O      0.000000      0.000000      0.000000
      O      1.086900      0.000000      0.660000
      O     -1.086900      0.000000      0.660000
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type TZ2P
  end

  NumericalQuality GOOD

  RIHartreeFock
    fitsetQuality Normal
  end

  MBPT
    nFrequency 18
    nTime 18
  end

```

(continues on next page)

(continued from previous page)

```
  XC
    Hybrid PBE0
  end

  GW
     nstates 2
  END
EndEngine
eor
```

Example: G_0W_0@PBE0 calculation: Phenol

Download GW_Phenol.run

```
#! /bin/sh

# GW calculation of Toluene. By default, the highest occupied and lowest
# unoccupied state are calculated. We use an all-electron basis set
# since core-correlation effects are important to capture all screening effects.

# Since we are interested in an accurate QP HOMO-LUMO gap, we use the aug-TZ2P
# basis set. To ensure numerical stability, we switch on the dependency key.

# We also use imaginary time and imaginary frequency grids of 16 points each.
# Note, that PBE is not a recommended starting point (better hybrid PBE0).
# Also note, that Good fitsetQuality is recommended

$AMSBIN/ams << eor

System
  Atoms
        C 4.555420 5.661760 4.489060
        C 4.584960 2.843420 4.498910
        C 3.378760 3.548270 4.498890
        C 3.359200 4.943960 4.500070
        C 5.758650 4.948670 4.502800
        C 5.789840 3.550210 4.500330
        H 4.622340 1.760920 4.495600
        H 3.631910 7.321310 4.523190
        H 2.474610 2.963630 4.498170
        H 2.384290 5.419790 4.496670
        H 6.695040 5.492670 4.498160
        H 6.727200 3.021210 4.497160
        O 4.537700 7.024010 4.500450
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type AUG/ATZ2P
  end
```

(continues on next page)

(continued from previous page)

```
  NumericalQuality Good
  Dependency bas=1e-03

  MBPT
    FitsetQuality Normal
    nFrequency 16
    nTime 16
    Dependency
  end

  XC
    GGA PBE
  end

  GW
  END

  Relativity
    Level None
  End
EndEngine
eor
```

Example: evGW@M06-2X calculation: H2O

Download evGW_H2O.run

```
#! /bin/sh

# eigenvalue-only self-consistent GW calculation for Water.

# We change here several options in the GW block to illustrate how this works

# We chose M06-2X as starting point
# We chose good numerical quality. This implies that 20 imaginary time and frequency␣
↪points each are used

# nStates 3
# ==> We want to print out the 3 highest occupied, and the 3 lowest unoccupied quasi-
↪particle states

# nDIIS 5
# ==> We use DIIS to converge the quasi-particle energies (linear mixing is possible␣
↪as well, but it is not recommended)
# ==> Per default, the DIIS algorithm in evGW considers the last 10 iterations. We␣
↪only want 5 here. If for some reason
# ==> your evGW calculation should not converge, this is the first parameter to look␣
↪into

# Converge HOMO=5e-3
# ==> We consider the procedure to be converged when the HOMO quasi-particle energy␣
↪between 2 iterations does not change
# ==> by more than 5 meV
```

(continues on next page)

(continued from previous page)

```

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
End

System
  Atoms
    O  0.0000 0.0000 0.0000
    H  0.7571 0.0000 0.5861
    H -0.7571 0.0000 0.5861
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type TZ2P
  End
  XC
     libxc M06-2X
  end
  MBPT
     nTime 20
     nFrequency 20
  End
  numericalQuality Good

  GW
     DIIS 5
     nStates 3
     Converge HOMO=5e-3
     selfconsistency evGW
  END

EndEngine
eor
```

Example: qsGW calculation: N2

Download qsGW_N2.run

```
#! /bin/sh

# Like in evGW in qsGW the quasi-particle energies are updated
# until self-consistency is reached.
# In qsGW, the density is updated as well, however, the self-energy is
# mapped to a static effective potential and the Dyson equation is
# solved by diagonalization instead of inversion.
# The results of a qsGW are independent of the choice of the underlying
# exchange-correlation functional and are usually the most accurate ones.
# The same is done in qsGW0, but the screened interaction is not updated.
```

(continues on next page)

(continued from previous page)

```
for k in 'qsGW' 'qsGW0' ; do

rm -r ams.results

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
End

System
  Atoms
    N 0.0000 0.0000 0.0000
    N 0.0000 0.0000 1.0977
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type TZ2P
  End
  symmetry nosym
  XC
     hartreefock
  end
  MBPT
     nTime 12
     nFrequency 12
  End
  numericalQuality Good
  GW
     DIIS 3
     nStates 3
     selfconsistency $k
  END

EndEngine
eor

done
```

Example: qsGW with frozen core: Br2

Download qsGW_Br2_frozenCore.run

```
#!/bin/sh

# Standard qsGW calculation using the frozen core approximation in the
# correlation part. An all electron basis set is required.
# The execution time will roughly stay the same, but numerical convergence
# is better since problems with analytical continuation of core states
```

(continues on next page)

(continued from previous page)

```
# are completely avoided. The frozen core approximation only affects the
# correlation part of the self-energy.
# The number of core states are being chosen using a default scheme.
# For Br, the frozen core is Argon.
# Symmetry NOSYM is required in case of MBPT%frozencore

# Note that the used DZ basis set is too small to get accurate results.
# Recommended is to use at least TZ2P.

$AMSBIN/ams << eor
System
  Atoms
    Br 0.0000 0.0000 0.0000
    Br 0.0000 0.0000 2.2811
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type DZ
  End
  symmetry nosym
  XC
     hartreefock
  end
  MBPT
     frozenCore True
  End
  numericalQuality Good
  GW
     selfconsistency qsGW
  END
EndEngine
eor
```

Example: G3W2 calculation: H2O

Download G3W2_H2O.run

```
#! /bin/sh

# G3W2 calculation H2O.
# G3W2 is a GW calculation plus a perturbative second-order statically
# screened exchange correction (second order expansion in the self-energy).
# Note, that there the self-energy is always static.
# Only a second-order screened exchange correction
# is evaluated after the GW calculation is completed.

$AMSBIN/ams << eor
Symmetry
  SymmetrizeTolerance 0.001
```

(continues on next page)

(continued from previous page)

```
End

System
  Atoms
    O    2.220871067   0.026716792   0.000620476
    H    2.597492682  -0.411663274   0.766744858
    H    2.593135384  -0.449496183  -0.744782026
  End
  Symmetrize Yes
End

task SinglePoint

Engine adf
  Basis
    Core None
    Type TZ2P
  end

  NumericalQuality GOOD

  XC
    hartreefock
  end

  GW
     selfenergy G3W2
     nstates 1
  END
EndEngine
eor
```

Example: GW with DRF: CH2O in Water

Download GW_CH2O_Embed.run

```
#! /bin/sh

# GW calculation on formaldehyde in water.
# In this example a single formaldehyde molecule and its first solvation shell
# which includes water molecules within 4 Angstroms, is simulated

# We use the G0W0 approximation for the QM system, and the DRF model for water.

$AMSBIN/ams << eor
System
  Atoms
    C       2.39264728       2.11086646      -3.70673394
    O       1.47987901       1.52348369      -4.24811419
    H       2.99243915       1.67294268      -2.84396986
    H       2.59427915       3.15468783      -4.03620406
  End
  Symmetrize No
End
```

(continues on next page)

(continued from previous page)

```
UseSymmetry No

Task SinglePoint

Engine ADF
  Basis
    Core None
    Type TZP
  End
  XC
    GGA PBE
  End
  GW
    SelfConsistency G0W0
  End
  DimQM
    DRF2
  End
  DimPar
    O
      Rad 0.90
      Char 0.6
      Static
    SubEnd
    H
      Rad 0.45
      Char -0.3
      Static
    SubEnd
    XYZ
      51
      H       3.54666213       5.35725337      -7.02041452
      O       2.04645779       7.07368154      -3.27252710
      H       1.78764102       7.78765869      -3.90371649
      H       1.71058215       6.30711583      -3.76280361
      O       4.88286035       4.00451983      -3.28588077
      H       5.38597457       3.44394310      -3.92640544
      H       5.19077667       4.91657693      -3.44227986
      O       0.18656256       4.23618795      -7.03761368
      H       0.89075012       4.62333203      -7.62947059
      H       0.40727875       3.22970332      -7.13319119
      O      -1.14702326       4.47659905      -2.16966035
      H      -1.81501249       3.71212364      -2.27877436
      H      -0.43286915       4.20138660      -2.75613800
      O       7.07024867       0.36614784      -2.04440869
      H       6.46762805      -0.30965402      -1.61572595
      H       6.70719633       1.14126953      -1.55520987
      O       3.57962906       3.67314699      -6.98526274
      H       3.10205286       2.94694403      -7.48923062
      H       4.25784146       3.17526726      -6.41592139
      O       3.13082943      -1.64764804      -5.76593595
      H       2.41673557      -1.74073399      -5.04455583
      H       3.81476755      -2.36835657      -5.73215731
      O       0.74188261       1.82800852      -6.78436630
      H       0.96547395       1.58943581      -5.85019084
      H      -0.03919267       1.19371203      -6.97282629
      O       4.12013607       0.69721169      -6.79199940
```

(continues on next page)

(continued from previous page)

```
      H       3.82718856      -0.16106544      -6.36689761
      H       4.70919651       1.13524018      -6.22423015
      O       3.63633475       1.77306085       0.08634103
      H       3.18738933       0.91138923      -0.17852169
      H       4.08266799       1.46290305       0.89402583
      O       1.57983242       3.47984534      -0.72435642
      H       2.40873386       3.14297655      -0.32712375
      H       1.59962298       4.49653225      -0.66765510
      O       5.92317302       2.67946724      -0.90296232
      H       5.13809949       2.26177170      -0.43308957
      H       5.56497636       3.33063271      -1.61014140
      O       0.38976232       5.05565287      -4.27026257
      H       0.57556838       4.43609806      -5.05893278
      H      -0.20875330       5.77787276      -4.64583021
      O       2.59583614      -0.59253360      -0.56623965
      H       1.68195751      -0.43168111      -0.15029539
      H       2.50988463      -1.59868716      -0.85813667
      O       0.59002147      -1.21174296      -4.24462128
      H       0.26277112      -0.97994479      -3.36169700
      H       0.84085343      -0.34198086      -4.42412382
      O       6.02045233       3.02011606      -5.80000628
      H       6.39638420       3.30267104      -6.68275756
      H       6.73263271       2.47966314      -5.34640323
      O       3.75450136       6.39026611      -7.12258679
      H       2.78990713       6.75181501      -7.10199835
    SubEnd
  End
EndEngine
eor
```

14.5 Transport properties

14.5.1 Charge transfer integrals (transport properties)

Example: Charge transfer integrals: AT base pair

Download AT_transferintegrals.run

```
#! /bin/sh


# ADF can calculate charge transfer integrals, that are needed in
# approximate methods that model charge transport properties. The molecular
# system typically should be build from 2 fragments. In this example charge
# transfer integrals are calculated between Adenine and Thymine. First these two
# molecules are calculated. In the fragment calculation full symmetry can be
# used. For precision reasons the ZlmFit quality is set to good.

AMS_JOBNAME=Adenine $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N       0.000000000000       0.656191000000       4.473450000000
```

(continues on next page)

(continued from previous page)

```
     C       0.000000000000       1.850911000000       5.098850000000
     N       0.000000000000       2.094911000000       6.411070000000
     C       0.000000000000       0.951291000000       7.115010000000
     C       0.000000000000      -0.355699000000       6.611740000000
     C       0.000000000000      -0.487619000000       5.203330000000
     N       0.000000000000       0.791131000000       8.484350000000
     C       0.000000000000      -0.567649000000       8.729290000000
     N       0.000000000000      -1.292469000000       7.631450000000
     N       0.000000000000      -1.672349000000       4.572610000000
     H       0.000000000000       2.715551000000       4.433920000000
     H       0.000000000000       1.540301000000       9.166150000000
     H       0.000000000000      -0.961519000000       9.739820000000
     H       0.000000000000      -2.515699000000       5.129900000000
     H       0.000000000000      -1.718459000000       3.541030000000
  end
end

Task SinglePoint

Engine ADF
  title Adenine fragment
  basis
    core None
    type DZ
  end
  zlmfit
    quality good
  end
EndEngine
eor


AMS_JOBNAME=Thymine $AMSBIN/ams <<eor
System
  symmetrize
  atoms
     N       0.000000000000       0.617991000000       1.666040000000
     C       0.000000000000       1.851251000000       1.046260000000
     N       0.000000000000       1.768641000000      -0.347380000000
     C       0.000000000000       0.582611000000      -1.042160000000
     C       0.000000000000      -0.621999000000      -0.417040000000
     C       0.000000000000      -0.627269000000       1.045880000000
     O       0.000000000000      -1.670479000000       1.720780000000
     O       0.000000000000       2.924531000000       1.636600000000
     C       0.000000000000      -1.937039000000      -1.138130000000
     H       0.000000000000       0.635221000000       2.733380000000
     H       0.000000000000       2.660141000000      -0.830100000000
     H       0.000000000000       0.676731000000      -2.127100000000
     H       0.880180000000      -2.533409000000      -0.860650000000
     H       0.000000000000      -1.793509000000      -2.225780000000
     H      -0.880180000000      -2.533409000000      -0.860650000000
  end
end

Task SinglePoint

Engine ADF
```

(continues on next page)

(continued from previous page)

```
  title Thymine fragment
  basis
    core None
    type DZ
  end
  zlmfit
    quality good
  end
EndEngine
eor


# Next the the base pair is calculated that consists of Adenine and Thymine. To
# calculate the charge transfer integrals, spatial overlap integrals and site
# energies, include the key TRANSFERINTEGRALS in the input for ADF. Symmetry
# NOSYM should be used.


$AMSBIN/ams <<eor
System
  atoms
     N       0.000000000000       0.656191000000       4.473450000000  adf.f=Adenine
     C       0.000000000000       1.850911000000       5.098850000000  adf.f=Adenine
     N       0.000000000000       2.094911000000       6.411070000000  adf.f=Adenine
     C       0.000000000000       0.951291000000       7.115010000000  adf.f=Adenine
     C       0.000000000000      -0.355699000000       6.611740000000  adf.f=Adenine
     C       0.000000000000      -0.487619000000       5.203330000000  adf.f=Adenine
     N       0.000000000000       0.791131000000       8.484350000000  adf.f=Adenine
     C       0.000000000000      -0.567649000000       8.729290000000  adf.f=Adenine
     N       0.000000000000      -1.292469000000       7.631450000000  adf.f=Adenine
     N       0.000000000000      -1.672349000000       4.572610000000  adf.f=Adenine
     H       0.000000000000       2.715551000000       4.433920000000  adf.f=Adenine
     H       0.000000000000       1.540301000000       9.166150000000  adf.f=Adenine
     H       0.000000000000      -0.961519000000       9.739820000000  adf.f=Adenine
     H       0.000000000000      -2.515699000000       5.129900000000  adf.f=Adenine
     H       0.000000000000      -1.718459000000       3.541030000000  adf.f=Adenine
     N       0.000000000000       0.617991000000       1.666040000000  adf.f=Thymine
     C       0.000000000000       1.851251000000       1.046260000000  adf.f=Thymine
     N       0.000000000000       1.768641000000      -0.347380000000  adf.f=Thymine
     C       0.000000000000       0.582611000000      -1.042160000000  adf.f=Thymine
     C       0.000000000000      -0.621999000000      -0.417040000000  adf.f=Thymine
     C       0.000000000000      -0.627269000000       1.045880000000  adf.f=Thymine
     O       0.000000000000      -1.670479000000       1.720780000000  adf.f=Thymine
     O       0.000000000000       2.924531000000       1.636600000000  adf.f=Thymine
     C       0.000000000000      -1.937039000000      -1.138130000000  adf.f=Thymine
     H       0.000000000000       0.635221000000       2.733380000000  adf.f=Thymine
     H       0.000000000000       2.660141000000      -0.830100000000  adf.f=Thymine
     H       0.000000000000       0.676731000000      -2.127100000000  adf.f=Thymine
     H       0.880180000000      -2.533409000000      -0.860650000000  adf.f=Thymine
     H       0.000000000000      -1.793509000000      -2.225780000000  adf.f=Thymine
     H      -0.880180000000      -2.533409000000      -0.860650000000  adf.f=Thymine
  end
end

Task SinglePoint

Engine ADF
```

(continues on next page)

(continued from previous page)

```
  title AT
  fragments
     Adenine Adenine.results/adf.rkf
     Thymine Thymine.results/adf.rkf
  end
  symmetry NOSYM
  TransferIntegrals
  zlmfit
    quality good
  end
EndEngine
eor

# After the calculation has finished in the output one will find the charge
# transfer (overlap integrals and site energies) that are needed to calculate
# hole mobility or electron mobility calculations:

#␣
↪===============================================================================================
# Electronic coupling V (also known as effective (generalized) transfer integrals J_
↪eff)
# V = (J-S(e1+e2)/2)/(1-S^2)
#
# V for hole transfer:            0.000 eV
# V for electron transfer:       -0.036 eV
#
# The effective transfer integral, or electronic coupling, is calculated from these␣
↪components:
#
# e1(hole) Site energy HOMO fragment 1:      -6.88 eV
# e2(hole) Site energy HOMO fragment 2:      -6.46 eV
# J(hole) Charge transfer integral HOMO fragment 1 - HOMO fragment 2:       0.000 eV
# S(hole) Overlap integral HOMO fragment 1 - HOMO fragment 2:       0.000
#
# e1(electron) Site energy LUMO fragment 1:      -2.24 eV
# e2(electron) Site energy LUMO fragment 2:      -2.62 eV
# J(electron) Charge transfer integral LUMO fragment 1 - LUMO fragment 2:      -0.046␣
↪eV
# S(electron) Overlap integral LUMO fragment 1 - LUMO fragment 2:       0.004
#␣
↪===============================================================================================
```

Example: Charge transfer integrals with FDE: water dimer

Download ElectronTransfer_FDE_H2O.run

```
#!/bin/sh


# == Expert Option ==

# The electron transfer calculation of a water dimer radical cation in this
# example is aimed at:

# - calculate site energies and couplings of FDE-derived charge-localized states
```

(continues on next page)

(continued from previous page)

```
# - calculate the charge-transfer excitation energy from a two-state model that
#   includes the two charge-localized states involved in the calculation
# - the output also includes a rough evaluation of the error introduced by the
#   density fitting on the site energies and coupling


# First the isolated neutral fragments are obtained. Symmetry NOSYM is used.
# Next in the first FDE calculation the localized state D+A is calculated, which
# means that the first water molecule has charge +1, and the second water
# molecule is neutral. The resulting adf.rkf files must be renamed to fragA1.t21
# and fragA2.t21. In the second FDE calculation the localized state DA+ is
# calculated, now the second water molecule has charge +1, and the first water
# molecule is neutral. The resulting adf.rkf files must be renamed to fragB1.t21
# and fragB2.t21. The FDE freeze and thaw cycle is done manually, and a spin-
# unrestricted calculation is performed.

# The electron transfer calculation is next. The files fragA1.t21, fragA2.t21,
# fragB1.t21, and fragB2.t21 must exist and must have these names. The program
# must be execute in serial mode. Hybrids are not supported. NumericalQuality
# good is needed for accuracy.


# == The output of this example is discussed here ==


# ============  Electron Transfer RESULTS ===================

# Electronic Coupling =         0.000000 eV
# Electronic Coupling =        -0.000006 cm-1
# H11-H22             =        -1.396836 eV
# Excitation Energy   =         1.396836 eV
# Overlap             =         0.000000
# H11 H22 H12 =  -152.443044906236  -152.391712133030  -151.743951196449 Eh
# S11 S22 S12 =     0.981761438554     0.980941502465    -0.000000000038


# =========== END Electron Transfer RESULTS ================

# Due to symmetry, the overlap is almost diagonal (Overlap = 0.00), thus the
# transition density is evaluated with one less electron as explained in Ref.
# [JCP 138 (2013) 054101, https://doi.org/10.1063/1.4789418].

# The electronic coupling between the state with a positive charge localized on
# one water molecule and another with the charge localized on the other water
# molecule is given by 'Electronic Coupling' and is reported in eV and cm^-1.

# 'H11-H22' is the difference of the site energies in eV. Values of the site
# energies are given by the first two values of 'H11 H22 H12' in atomic units.

# 'Excitation Energy' reports the value of the transfer excitation energy as
# calculated by diagonalization of the 2X2 generalized eigenvalue problem in
# the basis of the charge-localized states, see Refs.
# [JCP 135 (2011) 234103, https://doi.org/10.1063/1.3666005] and
# [JCP 138 (2013) 054101, https://doi.org/10.1063/1.4789418].

# 'S11 S22 S12' are the values of the non-normalized overlaps.
```

(continues on next page)

(continued from previous page)

```
# ==================================================
#           OBTAIN ISOLATED FRAGMENTS
# ==================================================


AMS_JOBNAME=Iso1 $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000
     H        -0.9358409558         .2646136961      0.0000000000
     H        -0.0304663436       -0.9828924420      0.0000000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core None
    type TZP
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  symmetry NOSYM
  title Isolated water molecule 1
  xc
    gga PW91
  end
EndEngine
eor

mv Iso1.results/adf.rkf t21.iso.rho1

AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000       -2.9053396088        0.0000000000
     H        -0.4092227596       -3.3374838250       -0.7701260000
     H        -0.4092227596       -3.3374838250        0.7701260000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
```

(continues on next page)

(continued from previous page)

```
    core None
    type TZP
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  symmetry NOSYM
  title isolated water molecule 2
  xc
    gga PW91
  end
EndEngine
eor
mv ams.results prev.results

mv Iso2.results/adf.rkf t21.iso.rho2

# ==================================================
#   FIRST FDE CALCULATION: STATE D+A
#   note: the freeze and thaw is done manually
#         as we have open-shell fragments
# ==================================================


AMS_JOBNAME=Emb1 $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2 type=fde
  end
  noprint BAS FUNCTIONS
```

(continues on next page)

(continued from previous page)

```
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  spinpolarization 1
  symmetry NOSYM
  title MODCO: Fragment no.          1;  (polarized)
  unrestricted
  xc
    gga PW91
  end
EndEngine
eor

mv Emb1.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2 $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  symmetry NOSYM
  title MODCO: Fragment no.          2;  (polarized)
  xc
    gga PW91
  end
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

mv Emb2.results/adf.rkf t21.emb.rho2


AMS_JOBNAME=Emb1r $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

EngineRestart t21.emb.rho1

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  spinpolarization 1
  symmetry NOSYM
  title MODCO: Fragment no.          1; relaxed
  unrestricted
  xc
    gga PW91
  end
EndEngine
eor

mv Emb1r.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2r $AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     O        0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H       -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H       -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O        0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H       -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H       -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.emb.rho2 subfrag=active
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  symmetry NOSYM
  title MODCO: Fragment no.           2; relaxed
  xc
    gga PW91
  end
EndEngine
eor

mv Emb2.results/adf.rkf t21.emb.rho2

mv t21.emb.rho1 fragA1.t21
mv t21.emb.rho2 fragA2.t21


# ==================================================
#   SECOND FDE CALCULATION: STATE DA+
#   note: the freeze and thaw is done manually
#         as we have open-shell fragments
#   note: now rho1 is the second fragment
#         as rho2 is the cation
# ==================================================


AMS_JOBNAME=Emb2p $AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1 type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  spinpolarization 1
  symmetry NOSYM
  title MODCO: Fragment no.          2; (polarized)
  unrestricted
  xc
    gga PW91
  end
EndEngine
eor

mv Emb2p.results/adf.rkf t21.emb.rho2


AMS_JOBNAME=Emb1p $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  symmetry NOSYM
  title MODCO: Fragment no.          1;  (polarized)
  xc
    gga PW91
  end
EndEngine
eor

mv Emb1p.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2pr $AMSBIN/ams <<eor
System
  atoms
     O          0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H         -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H         -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O          0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H         -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H         -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

EngineRestart t21.emb.rho2

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
```

(continues on next page)

(continued from previous page)

```
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
  end
  spinpolarization 1
  symmetry NOSYM
  title MODCO: Fragment no.          2; relaxed
  unrestricted
  xc
    gga PW91
  end
EndEngine
eor

mv Emb2pr.results/adf.rkf t21.emb.rho2

AMS_JOBNAME=Embp1r $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 90
```

(continues on next page)

(continued from previous page)

```
  end
  symmetry NOSYM
  title MODCO: Fragment no.          1; relaxed
  xc
    gga PW91
  end
EndEngine
eor

mv Embp1r.results/adf.rkf t21.emb.rho1

# =======================================
# NOW THE adf.rkf FILES MUST BE RENAMED
#
#  TO: fragA1.t21, fragA2.t21
#       .... up to a max of fragA999.t21
# AND: fragB1.t21, fragB2.t21
#       .... up to a max of fragB999.t21
# =======================================

mv t21.emb.rho1 fragB1.t21
mv t21.emb.rho2 fragB2.t21

# =======================================
# Electron Transfer calculation is next
#
# 1) note the ELECTRONTRANSFER block keyword
# 2) the program must be execute in
#    serial mode (parallel mode not
#    supported)
# 3) hybrids not supported
# 4) NumericalQuality good is
#    needed for accuracy
# =======================================

AMS_JOBNAME=ET $AMSBIN/ams <<eor
System
  atoms
     O         0.0000000000        0.0000000000      0.0000000000   adf.f=rho1
     H        -0.9358409558         .2646136961      0.0000000000   adf.f=rho1
     H        -0.0304663436       -0.9828924420      0.0000000000   adf.f=rho1
     O         0.0000000000       -2.9053396088      0.0000000000   adf.f=rho2
     H        -0.4092227596       -3.3374838250     -0.7701260000   adf.f=rho2
     H        -0.4092227596       -3.3374838250      0.7701260000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

Engine ADF
  electrontransfer
    numfrag 2
  end
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
```

(continues on next page)

(continued from previous page)

```
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  Relativity Level=None
  numericalquality good
  scf
    iterations 0
  end
  spinpolarization 1
  symmetry nosym
  title ElectronTransfer calculation
  unrestricted
  xc
    gga PW91
  end
EndEngine
eor
```

Example: Charge Recombination Calculation of Toluene and TCNE

Download `ChargeSeparation_Toluene_TCNE.run`

```
#!/bin/sh


# Charge Recombination Calculation of Toluene and TCNE
# tol + TCNE ==> tol+ + TCNE-
# aimed at:

# 1) calculate site energies and couplings of fde-derived neutral states
#     (Diabat 1) and charge-localized states (Diabat 2)
# 2) calculate the charge-transfer excitation energy from a two-state model that
#    includes the two Diabatic states involved in the calculation

# First the isolated neutral fragments are obtained. Symmetry NOSYM is used.
# Resulting files: t21.iso.rho1 and t21.iso.rho2. Next in the first FDE
# calculations: state Toluene (neutral) and TCNE (neutral), resulting files:
# fragA1.t21 and fragA2.t21. The freeze and thaw cycles are done manually since
# open-shell fragments were used. Next in the second FDE calculations: state
# Toluene+ and TCNE-, resulting files fragB1.t21 and fragB2.t21. Again freeze
# and thaw cycles are done manually since open-shell fragments were used.

# In the final charge separation calculation the ELECTRONTRANSFER block keyword
# The program must be executed in serial mode.

#  NOTE: For production-quality results use a larger basis set


# ====================================================
#           OBTAIN ISOLATED FRAGMENTS
# ====================================================
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=Iso1 $AMSBIN/ams <<eor
System
  atoms
     C        -1.07097000000000     -2.32168000000000      0.00000000000000
     C        -1.39599000000000     -1.69511000000000      1.20863700000000
     C        -1.39599000000000     -1.69511000000000     -1.20864000000000
     H        -0.57394000000000     -3.29447000000000      0.00000000000000
     C        -2.04242000000000     -0.45596000000000      1.20579700000000
     H        -1.14539000000000     -2.17411000000000      2.15797500000000
     C        -2.04242000000000     -0.45596000000000     -1.20580000000000
     H        -1.14539000000000     -2.17411000000000     -2.15798000000000
     C        -2.38186000000000      0.18220800000000      0.00000000000000
     H        -2.29006000000000      0.02569300000000      2.15529900000000
     H        -2.29006000000000      0.02569300000000     -2.15530000000000
     C        -3.12057000000000      1.49908200000000      0.00000000000000
     H        -2.88337000000000      2.09962700000000      0.89190300000000
     H        -2.88337000000000      2.09962700000000     -0.89190000000000
     H        -4.21273000000000      1.33541500000000      0.00000000000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core none
    type DZ
  end
  noprint BAS FUNCTIONS
  symmetry NOSYM
  title Isolated Toluene
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Iso1.results/adf.rkf t21.iso.rho1

AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
System
  atoms
     C         1.83245600000000     -0.14587000000000      0.00000000000000
     C         1.20994900000000      1.07690800000000      0.00000000000000
     C         2.18809800000000     -0.80630000000000     -1.21971000000000
     C         2.18809800000000     -0.80630000000000      1.21971000000000
     C         0.88208300000000      1.75218800000000     -1.21935000000000
     C         0.88208300000000      1.75218800000000      1.21934500000000
     N         2.50290100000000     -1.35237000000000     -2.19775000000000
     N         2.50290100000000     -1.35237000000000      2.19774900000000
     N         0.61754300000000      2.32819700000000     -2.19523000000000
     N         0.61754300000000      2.32819700000000      2.19522600000000
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core none
    type DZ
  end
  noprint BAS FUNCTIONS
  symmetry NOSYM
  title Isolated TCNE
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Iso2.results/adf.rkf t21.iso.rho2

# ==============================================================
#   FIRST FDE CALCULATION: STATE tol(neutral) and TCNE (neutral)
#   note: the freeze and thaw is done manually
#         as we have open-shell fragments
# ==============================================================

AMS_JOBNAME=Emb1 $AMSBIN/ams <<eor
System
  atoms
     C       -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H       -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C       -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C       -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     H        -2.88337000000000      2.09962700000000       0.89190300000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000      -0.89190000000000    adf.
↪f=rho1
     H        -4.21273000000000      1.33541500000000       0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000       0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000       0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000       1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000       1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000       2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000       2.19522600000000    adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2 type=fde
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title tol(neutral), First freeze--and--thaw cycle
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb1.results/adf.rkf t21.emb.rho1
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=Emb2 $AMSBIN/ams <<eor
System
  atoms
     C        -1.07097000000000     -2.32168000000000       0.00000000000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000       1.20863700000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000      -1.20864000000000    adf.
↪f=rho1
     H        -0.57394000000000     -3.29447000000000       0.00000000000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000       1.20579700000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000       2.15797500000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000      -1.20580000000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000      -2.15798000000000    adf.
↪f=rho1
     C        -2.38186000000000      0.18220800000000       0.00000000000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000       2.15529900000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000      -2.15530000000000    adf.
↪f=rho1
     C        -3.12057000000000      1.49908200000000       0.00000000000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000       0.89190300000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000      -0.89190000000000    adf.
↪f=rho1
     H        -4.21273000000000      1.33541500000000       0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000       0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000       0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000       1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000       1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000       2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000       2.19522600000000    adf.
↪f=rho2
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title TCNE(neutral), First freeze--and--thaw cycle
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb2.results/adf.rkf t21.emb.rho2


it=1
while [ $it -le 2 ]; do

AMS_JOBNAME=Emb1.$it $AMSBIN/ams <<eor
System
  atoms
     C       -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H       -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C       -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     C        -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
     H        -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000    adf.
↪f=rho2
  end
end

Task SinglePoint

EngineRestart t21.emb.rho1

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title tol(neutral), Relaxation cycles
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor

mv Emb1.$it.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2.$it $AMSBIN/ams <<eor
System
  atoms
     C        -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H        -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C        -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C        -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
     H        -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000    adf.
↪f=rho2
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

EngineRestart t21.emb.rho2

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title TCNE(neutral), Relaxation cycles
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb2.$it.results/adf.rkf t21.emb.rho2


it=$(($it+1))
done

mv t21.emb.rho1 fragA1.t21
mv t21.emb.rho2 fragA2.t21


# ==================================================
#   SECOND FDE CALCULATION: STATE tol+ TCNE-
#   note: the freeze and thaw is done manually
#         as we have open-shell fragments
# ==================================================

AMS_JOBNAME=Emb1p $AMSBIN/ams <<eor
System
  atoms
     C        -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     H        -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C        -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C        -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
     H        -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000    adf.
↪f=rho2
  end
  charge 1
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
```

(continues on next page)

(continued from previous page)

```
    xcnadd PW91
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2 type=fde
  end
  noprint BAS FUNCTIONS
  spinpolarization 1
  symmetry NOSYM
  title tol+, First freeze--and--thaw cycle
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb1p.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2p $AMSBIN/ams <<eor
System
  atoms
     C       -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H       -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C       -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C       -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H       -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H       -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
     H       -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C        1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C        1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
```

(continues on next page)

(continued from previous page)

```
     C         2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000    adf.
↪f=rho2
  end
  charge -1
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  spinpolarization -1
  symmetry NOSYM
  title TCNE-, First freeze--and--thaw cycle
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb2p.results/adf.rkf t21.emb.rho2


it=1
while [  $it -le 2 ]; do

AMS_JOBNAME=Emb1p.$it $AMSBIN/ams <<eor
System
  atoms
     C        -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     C        -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C        -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H        -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C        -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H        -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C        -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H        -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C        -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H        -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
     H        -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000    adf.
↪f=rho2
  end
  charge 1
end

Task SinglePoint

EngineRestart t21.emb.rho1

Engine ADF
```

(continues on next page)

(continued from previous page)

```
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  spinpolarization 1
  symmetry NOSYM
  title tol+, Relaxation cycles
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb1p.$it.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2p.$it $AMSBIN/ams <<eor
System
  atoms
     C       -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H       -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C       -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C       -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H       -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H       -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
```

(continues on next page)

(continued from previous page)

```
     H        -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C         1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C         1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
     C         2.18809800000000     -0.80630000000000      1.21971000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000    adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000    adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000    adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000    adf.
↪f=rho2
  end
  charge -1
end

Task SinglePoint

EngineRestart t21.emb.rho2

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    pw91k
    xcnadd PW91
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  spinpolarization -1
  symmetry NOSYM
  title TCNE-, Relaxation cycles
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb2p.$it.results/adf.rkf t21.emb.rho2
```

(continues on next page)

(continued from previous page)

```
it=$(($it+1))
done


mv t21.emb.rho1 fragB1.t21
mv t21.emb.rho2 fragB2.t21


# ======================================
# Charge Separation calculation is next
#
# 1) note the ELECTRONTRANSFER block keyword
# 2) the program must be execute in
#    serial mode (parallel mode not
#    supported)
# 3) hybrids not supported
# ======================================

AMS_JOBNAME=ET $AMSBIN/ams <<eor
System
  atoms
     C       -1.07097000000000     -2.32168000000000      0.00000000000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000      1.20863700000000    adf.
↪f=rho1
     C       -1.39599000000000     -1.69511000000000     -1.20864000000000    adf.
↪f=rho1
     H       -0.57394000000000     -3.29447000000000      0.00000000000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000      1.20579700000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000      2.15797500000000    adf.
↪f=rho1
     C       -2.04242000000000     -0.45596000000000     -1.20580000000000    adf.
↪f=rho1
     H       -1.14539000000000     -2.17411000000000     -2.15798000000000    adf.
↪f=rho1
     C       -2.38186000000000      0.18220800000000      0.00000000000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000      2.15529900000000    adf.
↪f=rho1
     H       -2.29006000000000      0.02569300000000     -2.15530000000000    adf.
↪f=rho1
     C       -3.12057000000000      1.49908200000000      0.00000000000000    adf.
↪f=rho1
     H       -2.88337000000000      2.09962700000000      0.89190300000000    adf.
↪f=rho1
     H       -2.88337000000000      2.09962700000000     -0.89190000000000    adf.
↪f=rho1
     H       -4.21273000000000      1.33541500000000      0.00000000000000    adf.
↪f=rho1
     C        1.83245600000000     -0.14587000000000      0.00000000000000    adf.
↪f=rho2
     C        1.20994900000000      1.07690800000000      0.00000000000000    adf.
↪f=rho2
     C        2.18809800000000     -0.80630000000000     -1.21971000000000    adf.
↪f=rho2
```

(continues on next page)

(continued from previous page)

```
     C         2.18809800000000     -0.80630000000000      1.21971000000000     adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000     -1.21935000000000     adf.
↪f=rho2
     C         0.88208300000000      1.75218800000000      1.21934500000000     adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000     -2.19775000000000     adf.
↪f=rho2
     N         2.50290100000000     -1.35237000000000      2.19774900000000     adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000     -2.19523000000000     adf.
↪f=rho2
     N         0.61754300000000      2.32819700000000      2.19522600000000     adf.
↪f=rho2
  end
end

Task SinglePoint

Engine ADF
  electrontransfer
    debug
    numfrag 2
  end
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  scf
    iterations 0
  end
  spinpolarization 0
  symmetry NOSYM
  title ChargeSeparation Calculation
  unrestricted
  xc
    gga PW91
  end
  Relativity Level=None
EndEngine
eor
```

Example: XCDFT: Charge Separation of an ethylene dimer

Download `XCDFT_FDE_ET_ethylene.run`

```
#!/bin/sh

#␣
↪=============================================================================================
#
#   Charge Separation Calculation of a ethylene dimer:
#   C2H4* + C2H4 --> C2H4+ + C2H4-
#   aimed at:
#
#   1) calculate site energies and couplings of fde-derived of a localized exciton
#      (Diabat 1) and charge-separated state (Diabat 2). The exciton is obtained by␣
↪using
#      XCDFT method.
#
#   2) calculate the charge-transfer excitation energy from a two-state model that
#        includes the two charge-localized states involved in the calculation
#
#   First the isolated (restricted) neutral fragments are obtained. Symmetry NOSYM is␣
↪used.
#   In addition, an isolated unrestricted neutral calculation of the fragment to be␣
↪excited is performed.
#   Resulting files: t21.iso.rho1, t21.iso.rho2 and gs.t21. Next in the first FDE␣
↪calculation:
#   state C2H4* (neutral-XCDFT excitation) and C2H4(neutral), resulting files:
#   fragA1.t21 and fragA2.t21. The freeze and thaw cycles are done manually since
#   open-shell fragments were used. Next in the second FDE calculation: state C2H4+␣
↪and C2H4-,
#   resulting files fragB1.t21 and fragB2.t21. Again freeze and thaw cycles are done␣
↪manually since
#   open-shell fragments were used.
#
#   In the final charge separation calculation the ELECTRONTRANSFER block keyword is␣
↪used.
#   The files fragA1.t21, fragA2.t21, fragB1.t21, and fragB2.t21 must exist and must␣
↪have these names.
#␣
↪================================================================================================

# ==================================================
#          OBTAIN ISOLATED FRAGMENTS
# ==================================================


AMS_JOBNAME=Iso1 $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000
     H        -0.930241    1.239503    0.000000
     C         0.000000    0.667450    0.000000
     C         0.000000   -0.667450    0.000000
     H         0.930241   -1.239503    0.000000
     H        -0.930241   -1.239503    0.000000
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core None
    type TZP
  end
  noprint BAS FUNCTIONS
  symmetry NOSYM
  title MODCO: Fragment no.          1; (isolated)
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Iso1.results/adf.rkf t21.iso.rho1

AMS_JOBNAME=Iso2 $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    8.000000
     H        -0.930241    1.239503    8.000000
     C         0.000000    0.667450    8.000000
     C         0.000000   -0.667450    8.000000
     H         0.930241   -1.239503    8.000000
     H        -0.930241   -1.239503    8.000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core None
    type TZP
  end
  noprint BAS FUNCTIONS
  symmetry NOSYM
  title MODCO: Fragment no.          2; (isolated)
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Iso2.results/adf.rkf t21.iso.rho2
```

(continues on next page)

(continued from previous page)

```
# ===================================================
#       OBTAIN ISOLATED UNRESTRICTED FRAGMENT
# ===================================================
AMS_JOBNAME=Iso1u $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000
     H        -0.930241    1.239503    0.000000
     C         0.000000    0.667450    0.000000
     C         0.000000   -0.667450    0.000000
     H         0.930241   -1.239503    0.000000
     H        -0.930241   -1.239503    0.000000
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  basis
    core None
    type TZP
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title Fragment 1; (isolated_unrestricted)
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

# ===================================================
#    RENAME TAPE21 file as gs.t21, as required in
#    XCDFT methodology.
# ===================================================
mv Iso1u.results/adf.rkf gs.t21


# ===================================================
#   FIRST FDE CALCULATION: STATE D* (XCDFT) + A
#   note: the freeze and thaw is done manually
#         as we have open-shell fragments
# ===================================================

AMS_JOBNAME=Emb1  $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000   adf.f=rho1
     H        -0.930241    1.239503    0.000000   adf.f=rho1
     C         0.000000    0.667450    0.000000   adf.f=rho1
     C         0.000000   -0.667450    0.000000   adf.f=rho1
```

(continues on next page)

(continued from previous page)

```
     H         0.930241   -1.239503    0.000000   adf.f=rho1
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  cdft
    allatoms
    constraints 1.0
    excitedcdft
    initialmultipliers 0.0
    maxiter 250
    metric
    natomsperset 36
    nconstraints 1
    print LOW
    stepsize 0.4
    threshold 1.0e-9
  end
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2 type=fde
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title MODCO: Fragment no.          1; (polarized)
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb1.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2 $AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
      H         0.930241    1.239503    0.000000   adf.f=rho1
      H        -0.930241    1.239503    0.000000   adf.f=rho1
      C         0.000000    0.667450    0.000000   adf.f=rho1
      C         0.000000   -0.667450    0.000000   adf.f=rho1
      H         0.930241   -1.239503    0.000000   adf.f=rho1
      H        -0.930241   -1.239503    0.000000   adf.f=rho1
      H         0.930241    1.239503    8.000000   adf.f=rho2
      H        -0.930241    1.239503    8.000000   adf.f=rho2
      C         0.000000    0.667450    8.000000   adf.f=rho2
      C         0.000000   -0.667450    8.000000   adf.f=rho2
      H         0.930241   -1.239503    8.000000   adf.f=rho2
      H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  spinpolarization 0
  symmetry NOSYM
  title MODCO: Fragment no.          2;  (polarized)
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb2.results/adf.rkf t21.emb.rho2


it=1
while [ $it -le 2 ]; do

AMS_JOBNAME=Emb1.$it $AMSBIN/ams <<eor
System
  atoms
      H         0.930241    1.239503    0.000000   adf.f=rho1
      H        -0.930241    1.239503    0.000000   adf.f=rho1
      C         0.000000    0.667450    0.000000   adf.f=rho1
      C         0.000000   -0.667450    0.000000   adf.f=rho1
      H         0.930241   -1.239503    0.000000   adf.f=rho1
```

(continues on next page)

(continued from previous page)

```
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
end

Task SinglePoint

EngineRestart t21.emb.rho1

Engine ADF
  allow PARTIALSUPERFRAGS
  cdft
    allatoms
    constraints 1.0
    excitedcdft
    initialmultipliers 0.0
    maxiter 250
    metric
    natomsperset 36
    nconstraints 1
    print LOW
    stepsize 0.4
    threshold 1.0e-9
  end
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    energy
    fullgrid
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  spinpolarization 0
  symmetry NOSYM
  title MODCO: Fragment no.           1; relaxed
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb1.$it.results/adf.rkf t21.emb.rho1
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=Emb2.$it $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000   adf.f=rho1
     H        -0.930241    1.239503    0.000000   adf.f=rho1
     C         0.000000    0.667450    0.000000   adf.f=rho1
     C         0.000000   -0.667450    0.000000   adf.f=rho1
     H         0.930241   -1.239503    0.000000   adf.f=rho1
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
end

Task SinglePoint

EngineRestart t21.emb.rho2

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    energy
    fullgrid
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  spinpolarization 0
  symmetry NOSYM
  title MODCO: Fragment no.          2; relaxed
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb2.$it.results/adf.rkf t21.emb.rho2


it=$(($it+1))
done
```

(continues on next page)

(continued from previous page)

```
# ======================================
# NOW THE TAPE21 FILES MUST BE RENAMED
#
#  TO: fragA1.t21, fragA2.t21
#      .... up to a max of fragA999.t21
# AND: fragB1.t21, fragB2.t21
#      .... up to a max of fragB999.t21
# ======================================

mv t21.emb.rho1 fragA1.t21
mv t21.emb.rho2 fragA2.t21



# ==================================================
#   SECOND FDE CALCULATION: STATE D+ + A-
#   note: the freeze and thaw is done manually
#         as we have open-shell fragments
# ==================================================

AMS_JOBNAME=Emb1p $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000   adf.f=rho1
     H        -0.930241    1.239503    0.000000   adf.f=rho1
     C         0.000000    0.667450    0.000000   adf.f=rho1
     C         0.000000   -0.667450    0.000000   adf.f=rho1
     H         0.930241   -1.239503    0.000000   adf.f=rho1
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.iso.rho2 type=fde
  end
  noprint BAS FUNCTIONS
```

(continues on next page)

(continued from previous page)

```
  spinpolarization -1
  symmetry NOSYM
  title MODCO: Fragment no.          1; (polarized)
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor

mv Emb1p.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2p $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000   adf.f=rho1
     H        -0.930241    1.239503    0.000000   adf.f=rho1
     C         0.000000    0.667450    0.000000   adf.f=rho1
     C         0.000000   -0.667450    0.000000   adf.f=rho1
     H         0.930241   -1.239503    0.000000   adf.f=rho1
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
  charge -1
end

Task SinglePoint

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  spinpolarization 1
  symmetry NOSYM
  title MODCO: Fragment no.          2; (polarized)
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor

mv Emb2p.results/adf.rkf t21.emb.rho2

it=1
while [ $it -le 2 ]; do

AMS_JOBNAME=Emb1p.$it $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000   adf.f=rho1
     H        -0.930241    1.239503    0.000000   adf.f=rho1
     C         0.000000    0.667450    0.000000   adf.f=rho1
     C         0.000000   -0.667450    0.000000   adf.f=rho1
     H         0.930241   -1.239503    0.000000   adf.f=rho1
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
  charge 1
end

Task SinglePoint

EngineRestart t21.emb.rho1

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    energy
    fullgrid
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.iso.rho1
     rho2 t21.emb.rho2 subfrag=active type=fde
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  spinpolarization -1
  symmetry NOSYM
  title MODCO: Fragment no.          1; relaxed
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor

mv Emb1p.$it.results/adf.rkf t21.emb.rho1

AMS_JOBNAME=Emb2p.$it $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000   adf.f=rho1
     H        -0.930241    1.239503    0.000000   adf.f=rho1
     C         0.000000    0.667450    0.000000   adf.f=rho1
     C         0.000000   -0.667450    0.000000   adf.f=rho1
     H         0.930241   -1.239503    0.000000   adf.f=rho1
     H        -0.930241   -1.239503    0.000000   adf.f=rho1
     H         0.930241    1.239503    8.000000   adf.f=rho2
     H        -0.930241    1.239503    8.000000   adf.f=rho2
     C         0.000000    0.667450    8.000000   adf.f=rho2
     C         0.000000   -0.667450    8.000000   adf.f=rho2
     H         0.930241   -1.239503    8.000000   adf.f=rho2
     H        -0.930241   -1.239503    8.000000   adf.f=rho2
  end
  charge -1
end

Task SinglePoint

EngineRestart t21.emb.rho2

Engine ADF
  allow PARTIALSUPERFRAGS
  eprint
    sfo NOEIG NOOVL NOORBPOP
  end
  fde
    energy
    fullgrid
    ggapotcfd pw91c
    ggapotxfd pw91x
    pw91k
  end
  fragments
     rho1 t21.emb.rho1 subfrag=active type=fde
     rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  spinpolarization 1
  symmetry NOSYM
  title MODCO: Fragment no.          2; relaxed
  unrestricted
  xc
    model SAOP
  end
  Relativity Level=None
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
mv Emb2p.$it.results/adf.rkf t21.emb.rho2

it=$(($it+1))
done
# ======================================
# NOW THE TAPE21 FILES MUST BE RENAMED
#
#  TO: fragA1.t21, fragA2.t21
#      .... up to a max of fragA999.t21
# AND: fragB1.t21, fragB2.t21
#      .... up to a max of fragB999.t21
# ======================================

mv t21.emb.rho1 fragB1.t21
mv t21.emb.rho2 fragB2.t21

# ======================================
# Electron Transfer calculation is next
#
# 1) note the ELECTRONTRANSFER block keyword
# 2) hybrids not supported
# 3) NumericalQuality GOOD is needed for accuracy
# ======================================

AMS_JOBNAME=ET $AMSBIN/ams <<eor
System
  atoms
     H         0.930241    1.239503    0.000000 adf.f=rho1
     H        -0.930241    1.239503    0.000000 adf.f=rho1
     C         0.000000    0.667450    0.000000 adf.f=rho1
     C         0.000000   -0.667450    0.000000 adf.f=rho1
     H         0.930241   -1.239503    0.000000 adf.f=rho1
     H        -0.930241   -1.239503    0.000000 adf.f=rho1
     H         0.930241    1.239503    8.000000 adf.f=rho2
     H        -0.930241    1.239503    8.000000 adf.f=rho2
     C         0.000000    0.667450    8.000000 adf.f=rho2
     C         0.000000   -0.667450    8.000000 adf.f=rho2
     H         0.930241   -1.239503    8.000000 adf.f=rho2
     H        -0.930241   -1.239503    8.000000 adf.f=rho2
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality Good
  end
  electrontransfer
    numfrag 2
  end
  eprint
    scf NOPOP
    sfo NOEIG NOOVL NOORBPOP
  end
  fragments
     rho1 t21.iso.rho1
```

(continues on next page)

(continued from previous page)

```
    rho2 t21.iso.rho2
  end
  noprint BAS FUNCTIONS
  numericalquality Good
  scf
    iterations 0
  end
  spinpolarization 0
  symmetry nosym
  title ET calculation at R=0.0 Angstrom
  unrestricted
  xc
    gga PW91
  end
  Relativity Level=None
EndEngine

eor
```

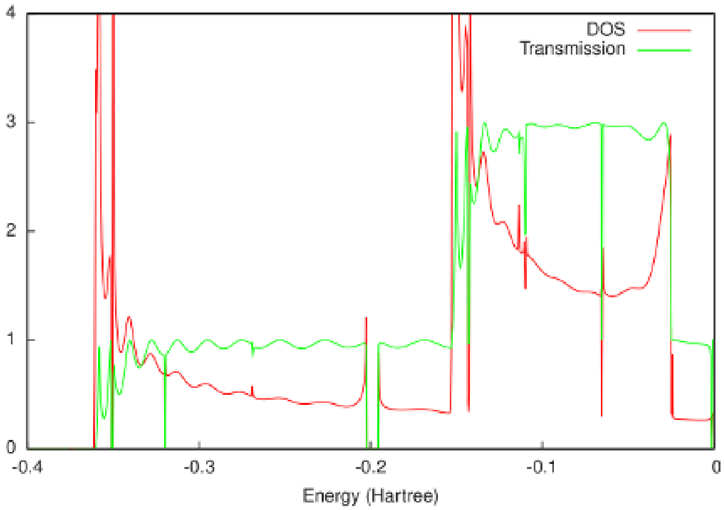
14.5.2 Non-self-consistent Green's function calculation

Example: DOS and transmission: Aluminium

`Download green_Al.run`

As an example of a non-self-consistent Green's function calculation, we will look at the density of states (DOS) and transmission of an infinite 1D chain of Aluminum atoms.

The final resulting DOS and transmission are shown in the following figure:



As would be expected for a 1D system, the DOS shows Van Hove singularities at the band edges. Apart from oscillations due to the finite size of the system in ADF, the transmission only reaches integer values. Between approximately -0.35 and -0.15 Hartree, only the sigma channel contributes to the transmission. Above -0.15 Hartree also the two pi channels start to contribute.

```
#!/bin/sh
```

(continues on next page)

(continued from previous page)

```
# As an example of a non-self-consistent Green's function calculation, we will
# look at the density of states (DOS) and transmission of an infinite 1D chain
# of Aluminum atoms.

# First we need to perform a single-point calculation with ADF on a principal
# layer, consisting, in this case, of four atoms. Since bulk Aluminum has an FCC
# structure with a lattice constant of 4.05 Angstrom, the nearest neighbor
# distance is approximately 2.83 Angstrom. green requires SYMMETRY NOSYM, so we
# have the following input file for the principal layer:

AMS_JOBNAME=layer $AMSBIN/ams <<eor
System
  atoms
     Al       -4.290000    0.000000    0.000000
     Al       -1.430000    0.000000    0.000000
     Al        1.430000    0.000000    0.000000
     Al        4.290000    0.000000    0.000000
  end
end

Task SinglePoint

Engine ADF
  basis
    core Large
    type DZP
  end
  scf
    converge 1.0e-8
  end
  symmetry NOSYM
  title Principal layer
  xc
    lda SCF VWN
  end
EndEngine
eor

# The bulk contact geometry consists of three principal layers:

AMS_JOBNAME=bulk $AMSBIN/ams <<eor
System
  atoms
     Al      -15.730000    0.000000    0.000000 adf.f=left
     Al      -12.870000    0.000000    0.000000 adf.f=left
     Al      -10.010000    0.000000    0.000000 adf.f=left
     Al       -7.150000    0.000000    0.000000 adf.f=left
     Al       -4.290000    0.000000    0.000000 adf.f=center
     Al       -1.430000    0.000000    0.000000 adf.f=center
     Al        1.430000    0.000000    0.000000 adf.f=center
     Al        4.290000    0.000000    0.000000 adf.f=center
     Al        7.150000    0.000000    0.000000 adf.f=right
     Al       10.010000    0.000000    0.000000 adf.f=right
     Al       12.870000    0.000000    0.000000 adf.f=right
     Al       15.730000    0.000000    0.000000 adf.f=right
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  fragments
     left    layer.results/adf.rkf
     center  layer.results/adf.rkf
     right   layer.results/adf.rkf
  end
  scf
    accelerationmethod LISTi
    converge 1.0e-10
  end
  symmetry NOSYM
  title Bulk
  xc
    lda SCF VWN
  end
EndEngine
eor


# Notice that we have increased the number of SCF iterations. The combination of
# SYMMETRY NOSYM with a 1D chain of metal atoms generally leads to convergence
# problems. This is the main reason why the principal layer consists of only
# four atoms. Fortunately, for larger 3D contacts, the convergence is generally
# better.

# From the bulk TAPE21 file green can calculate the self-energies of the left
# and right contacts. As discussed in the introduction, the self-energy of the
# left contact needs the center and right fragments of the bulk calculation, and
# the self-energy of the right contact needs the center and left fragments.
# Since we need a self-energy matrix for every energy for which we want to
# calculate the DOS and transmission, already here we have to specify the energy
# range. We take 1000 points between -0.4 and 0 Hartree.

$AMSBIN/green <<eor
SURFACE bulk.results/adf.rkf
    FRAGMENTS center right
END
EPS -0.4 0 1000
ETA 1e-6
eor

mv SURFACE left.kf

$AMSBIN/green <<eor
SURFACE bulk.results/adf.rkf
    FRAGMENTS center left
END
EPS -0.4 0 1000
ETA 1e-6
eor

mv SURFACE right.kf
```

(continues on next page)

(continued from previous page)

```
# Since we want to calculate the DOS and transmission of bare aluminum, we can
# reuse the bulk.t21 file for the extended molecule. We couple the left self-
# energy to the 'left' fragment and the right self-energy to the 'right'
# fragment in bulk.t21. Since we performed restricted ADF calculations, there is
# no difference between spin-A and spin-B and we can omit spin-B from the
# calculation.

$AMSBIN/green <<eor
DOS bulk.results/adf.rkf
TRANS bulk.results/adf.rkf
EPS -0.4 0 1000
ETA 1e-6
LEFT left.kf
    FRAGMENT left
END
RIGHT right.kf
    FRAGMENT right
END
NOSAVE DOS_B, TRANS_B
eor

# As would be expected for a 1D system, the DOS shows Van Hove singularities at
# the band edges. Apart from oscillations due to the finite size of the system
# in ADF, the transmission only reaches integer values. Between approximately
# -0.35 and -0.15 Hartree, only the sigma channel contributes to the
# transmission. Above -0.15 Hartree also the two pi channels start to
# contribute.

echo ""
echo "Contents of DOS_A:"
cat DOS_A
echo "END"
echo ""
echo "Contents of TRANS_A:"
cat TRANS_A
echo "END"
```

Example: Gold electrodes

`Download green_Au.run`

In this example of *green*, the self-energies are calculated of gold electrodes, the material most often used in molecular electronics. In *the example for the Benzenedithiol junction* (page 979) these self-energies will be used to calculate the DOS and transmission of a benzenedithiol junction. The geometry of the electrodes is shown in Fig. 1.

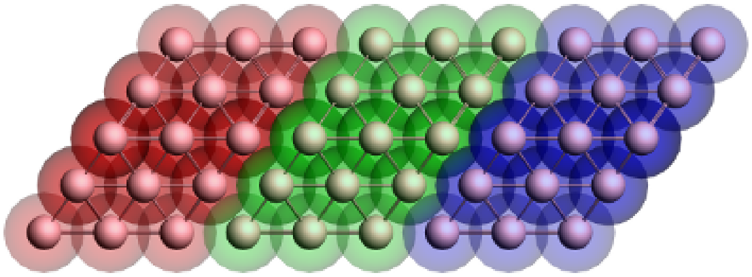
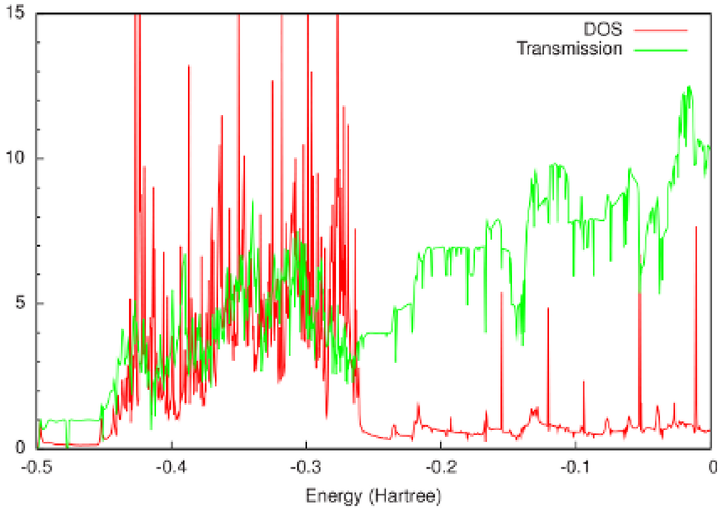


Figure 1: Geometry of the gold contact used in the calculation of the self-energy. The lead consists of two surface layers,

left (red) and right (blue), and a bulk layer (green). Each principal layer in turn consists of three atomic layers. This should be sufficient to ensure that the Hamiltonian of the central (green) layer is a bulk Hamiltonian.

The final results are shown in the following figure:



From this figure it can be seen that around the Fermi energy (-0.2 Hartree), both the DOS and the transmission of gold are relatively constant. This feature makes gold an attractive material for electrodes, since one can expect that the transmission of a molecular junction will be dominated by the molecular properties.

```
#!/bin/sh

# In this example of green, the self-energies are calculated of gold electrodes,
# the material most often used in molecular electronics.
# In the example for the Benzenedithiol junction, green_BDT.run, these self-energies
# will be used to calculate the DOS and transmission of a benzenedithiol junction.

# Each principal layer contains 3x3x3=27 gold atoms.
# For the calculation of the self-energies three principal layers are needed,
# and therefore 81 gold atoms in total.
# To keep the runtimes manageable it is therefore important to choose the basis set
# as small as possible.
# For transport calculations, a DZ basis with a large frozen core is generally␣
↪sufficient.
# Unfortunately, even with the largest frozen core (Au.4f),
# the basis set for Au still contains 19 electrons.
# A significant speedup can be obtained by limiting this to 11 electrons
# (only the outer *d* and *s* shells).
# Be advised that even with this reduction the total runtime of calculation can be␣
↪long.

# To facilitate the calculation of the electrodes,
# first a gold atom fragment will be calculated with the smallest possible basis.
# The sample directory contains the required Au.5p and Au.5p.dirac files.
# Note that for gold relativistic effects are important.
# Therefore Scalar ZORA will be used throughout this example.

cp $AMSHOME/examples/adf/green_Al/Au.5p .
cp $AMSHOME/examples/adf/green_Al/Au.5p.dirac .

$AMSBIN/dirac < Au.5p.dirac
```

(continues on next page)

(continued from previous page)

```
mv TAPE12 t12.rel

AMS_JOBNAME=Au $AMSBIN/ams <<eor
System
   Atoms
      Au 0.0 0.0 0.0
   End
end
Task SinglePoint
Engine ADF
  corepotentials t12.rel
  end
  create Au   file=Au.5p
  relativity
    level scalar
    formalism ZORA
  end
  xc
    lda SCF VWN
  end
EndEngine
eor

# A principal layer of gold consists of three atomic layers,
# which should be sufficient due to the small screening length.
# An atomic layer contains 3x3=9 atoms in a (111) FCC configuration.
# This allows one to use the top-, bride-, and hollow-site binding configurations for␣
↪a molecule.
# For the following calculations it is necessary to first construct a fragment
# of a principal layer.

AMS_JOBNAME=layer $AMSBIN/ams <<eor
System
  atoms
     Au       -2.355588   -6.662612    0.000000
     Au       -2.355589   -4.164133   -1.442498
     Au       -2.355589   -4.164133    1.442498
     Au       -2.355589   -1.665653   -2.884996
     Au       -2.355589   -1.665653    0.000000
     Au       -2.355589   -1.665653    2.884996
     Au       -2.355589    0.832826   -1.442498
     Au       -2.355589    0.832826    1.442498
     Au       -2.355589    3.331306    0.000000
     Au        0.000000   -4.996959    0.000000
     Au        0.000000   -2.498480   -1.442498
     Au        0.000000   -2.498480    1.442498
     Au        0.000000    0.000000   -2.884996
     Au        0.000000    0.000000    0.000000
     Au        0.000000    0.000000    2.884996
     Au        0.000000    2.498480   -1.442498
     Au        0.000000    2.498480    1.442498
     Au        0.000000    4.996959    0.000000
     Au        2.355589   -3.331306    0.000000
     Au        2.355589   -0.832826   -1.442498
     Au        2.355589   -0.832826    1.442498
     Au        2.355589    1.665653   -2.884996
     Au        2.355589    1.665653    0.000000
```

(continues on next page)

(continued from previous page)

```
     Au        2.355589    1.665653    2.884996
     Au        2.355589    4.164133   -1.442498
     Au        2.355589    4.164133    1.442498
     Au        2.355588    6.662612    0.000000
  end
end

Task SinglePoint

Engine ADF
  fragments
     Au  Au.results/adf.rkf
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Principal layer
  xc
    lda SCF VWN
  end
EndEngine
eor

# Three principal layers are stacked together to calculate the self-energies (see Fig.
↪ 1).
# The names of the fragments are significant, since one needs to refer to them by␣
↪name in
# the calculation of the self-energies.

AMS_JOBNAME=bulk $AMSBIN/ams <<eor
System
  atoms
     Au       -9.422355  -11.659571    0.000000 adf.f=left
     Au       -9.422356   -9.161092   -1.442498 adf.f=left
     Au       -9.422356   -9.161092    1.442498 adf.f=left
     Au       -9.422356   -6.662612   -2.884996 adf.f=left
     Au       -9.422356   -6.662612    0.000000 adf.f=left
     Au       -9.422356   -6.662612    2.884996 adf.f=left
     Au       -9.422356   -4.164133   -1.442498 adf.f=left
     Au       -9.422356   -4.164133    1.442498 adf.f=left
     Au       -9.422356   -1.665653    0.000000 adf.f=left
     Au       -7.066767   -9.993918    0.000000 adf.f=left
     Au       -7.066767   -7.495439   -1.442498 adf.f=left
     Au       -7.066767   -7.495439    1.442498 adf.f=left
     Au       -7.066767   -4.996959   -2.884996 adf.f=left
     Au       -7.066767   -4.996959    0.000000 adf.f=left
     Au       -7.066767   -4.996959    2.884996 adf.f=left
     Au       -7.066767   -2.498479   -1.442498 adf.f=left
     Au       -7.066767   -2.498479    1.442498 adf.f=left
     Au       -7.066767    0.000000    0.000000 adf.f=left
     Au       -4.711178   -8.328265    0.000000 adf.f=left
     Au       -4.711178   -5.829785   -1.442498 adf.f=left
     Au       -4.711178   -5.829785    1.442498 adf.f=left
     Au       -4.711178   -3.331306   -2.884996 adf.f=left
     Au       -4.711178   -3.331306    0.000000 adf.f=left
```

(continues on next page)

(continued from previous page)

```
    Au       -4.711178   -3.331306    2.884996 adf.f=left
    Au       -4.711178   -0.832826   -1.442498 adf.f=left
    Au       -4.711178   -0.832826    1.442498 adf.f=left
    Au       -4.711179    1.665653    0.000000 adf.f=left
    Au       -2.355588   -6.662612    0.000000 adf.f=center
    Au       -2.355589   -4.164133   -1.442498 adf.f=center
    Au       -2.355589   -4.164133    1.442498 adf.f=center
    Au       -2.355589   -1.665653   -2.884996 adf.f=center
    Au       -2.355589   -1.665653    0.000000 adf.f=center
    Au       -2.355589   -1.665653    2.884996 adf.f=center
    Au       -2.355589    0.832826   -1.442498 adf.f=center
    Au       -2.355589    0.832826    1.442498 adf.f=center
    Au       -2.355589    3.331306    0.000000 adf.f=center
    Au        0.000000   -4.996959    0.000000 adf.f=center
    Au        0.000000   -2.498480   -1.442498 adf.f=center
    Au        0.000000   -2.498480    1.442498 adf.f=center
    Au        0.000000    0.000000   -2.884996 adf.f=center
    Au        0.000000    0.000000    0.000000 adf.f=center
    Au        0.000000    0.000000    2.884996 adf.f=center
    Au        0.000000    2.498480   -1.442498 adf.f=center
    Au        0.000000    2.498480    1.442498 adf.f=center
    Au        0.000000    4.996959    0.000000 adf.f=center
    Au        2.355589   -3.331306    0.000000 adf.f=center
    Au        2.355589   -0.832826   -1.442498 adf.f=center
    Au        2.355589   -0.832826    1.442498 adf.f=center
    Au        2.355589    1.665653   -2.884996 adf.f=center
    Au        2.355589    1.665653    0.000000 adf.f=center
    Au        2.355589    1.665653    2.884996 adf.f=center
    Au        2.355589    4.164133   -1.442498 adf.f=center
    Au        2.355589    4.164133    1.442498 adf.f=center
    Au        2.355588    6.662612    0.000000 adf.f=center
    Au        4.711179   -1.665653    0.000000 adf.f=right
    Au        4.711178    0.832826   -1.442498 adf.f=right
    Au        4.711178    0.832826    1.442498 adf.f=right
    Au        4.711178    3.331306   -2.884996 adf.f=right
    Au        4.711178    3.331306    0.000000 adf.f=right
    Au        4.711178    3.331306    2.884996 adf.f=right
    Au        4.711178    5.829785   -1.442498 adf.f=right
    Au        4.711178    5.829785    1.442498 adf.f=right
    Au        4.711178    8.328265    0.000000 adf.f=right
    Au        7.066767    0.000000    0.000000 adf.f=right
    Au        7.066767    2.498479   -1.442498 adf.f=right
    Au        7.066767    2.498479    1.442498 adf.f=right
    Au        7.066767    4.996959   -2.884996 adf.f=right
    Au        7.066767    4.996959    0.000000 adf.f=right
    Au        7.066767    4.996959    2.884996 adf.f=right
    Au        7.066767    7.495439   -1.442498 adf.f=right
    Au        7.066767    7.495439    1.442498 adf.f=right
    Au        7.066767    9.993918    0.000000 adf.f=right
    Au        9.422356    1.665653    0.000000 adf.f=right
    Au        9.422356    4.164133   -1.442498 adf.f=right
    Au        9.422356    4.164133    1.442498 adf.f=right
    Au        9.422356    6.662612   -2.884996 adf.f=right
    Au        9.422356    6.662612    0.000000 adf.f=right
    Au        9.422356    6.662612    2.884996 adf.f=right
    Au        9.422356    9.161092   -1.442498 adf.f=right
    Au        9.422356    9.161092    1.442498 adf.f=right
```

(continues on next page)

(continued from previous page)

```
     Au        9.422355   11.659571    0.000000 adf.f=right
  end
end

Task SinglePoint

Engine ADF
  fragments
     left    layer.results/adf.rkf
     center  layer.results/adf.rkf
     right   layer.results/adf.rkf
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Bulk gold
  xc
    lda SCF VWN
  end
EndEngine
eor

# Similar to the other examples, the self-energies of the left and right contacts
# is calculated for 1000 energy points between -0.5 and 0 Hartree.
# This results in two keyfiles of approximately 2.5 GB each.
# Since the self-energies are independent of whatever is placed between the contacts,
# they can be reused many times.

$AMSBIN/green << eor
SURFACE bulk.results/adf.rkf
    FRAGMENTS center right
END
EPS -0.5 0 1000
ETA 1e-6
eor
mv SURFACE left.kf

$AMSBIN/green << eor
SURFACE bulk.results/adf.rkf
    FRAGMENTS center left
END
EPS -0.5 0 1000
ETA 1e-6
eor
mv SURFACE right.kf

# In order to interpret transmissions calculated with these self-energies,
# it is necessary to know the location of the Fermi energy.
# An estimate for the Fermi energy can be obtained from the bulk SCF calculation
# by taking the average of the HOMO and LUMO energies, which in this case equals -0.
↪195 Hartree.

# Usually the self-energies will be used to calculate the transmission of a molecular␣
↪junction.
# However, it is instructive to use a principal layer of gold as the "molecule" and␣
↪study
```

(continues on next page)

(continued from previous page)

```
# the DOS and transmission of bulk gold.

$AMSBIN/green << eor
DOS bulk.results/adf.rkf
TRANS bulk.results/adf.rkf
EPS -0.5 0 1000
ETA 1e-6
LEFT left.kf
    FRAGMENT left
END
RIGHT right.kf
    FRAGMENT right
END
NOSAVE DOS_B, TRANS_B
eor

echo ""
echo "Contents of DOS_A:"
cat DOS_A
echo "END"
echo ""
echo "Contents of TRANS_A:"
cat TRANS_A
echo "END"
```

Example: Benzenedithiol junction: Wide-Band-Limit

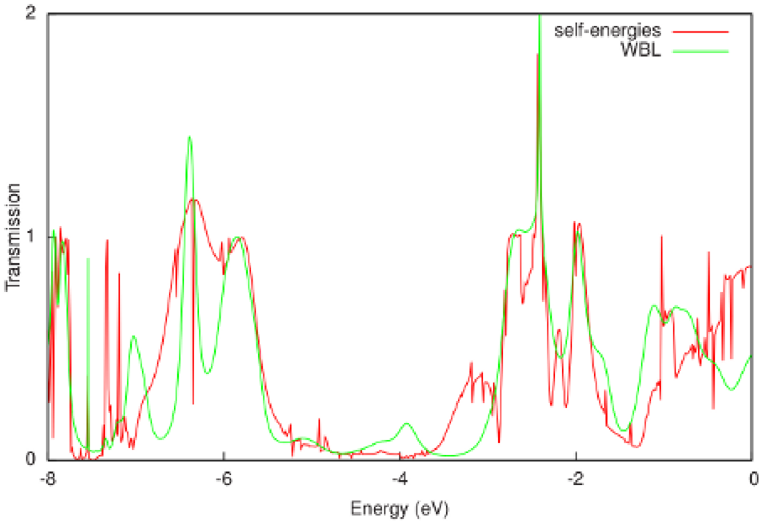
`Download green_WBL.run`

In the wide-band limit (WBL) the coupling to the leads is assumed to be independent of energy. Therefore one does not need to calculate any self-energies. This also means that the eigenspace of the Green's function is independent of energy. It can therefore be diagonalized in advance, greatly speeding up the calculation of the DOS and the transmission.

In the example $AMSHOME/examples/adf/green_Al/green_WBL/green_WBL.run of *green*, the transmission of *benzenedithiol junction* (page 979) in the wide-band limit (WBL) is calculated. In order to model the molecule-metal interface, we do need to include a few gold layers in the calculation. However, unlike *before* (page 967), only a single atomic layer as the principal layer is used.

Because a single atomic layer is an unnatural configuration for gold, a minor amount of smearing is necessary to make the calculation converge. The molecule is sandwiched in between the electrodes just like before (see Fig. 2 in *the example for benzenedithiol* (page 979)). However, this time each atomic layer of gold gets its own fragment. The reason for this configuration is that if the WBL is used on the entire gold contact the result is an an unphysical coupling to the leads; even the gold atoms contacting the molecule would have a direct coupling to the environment. A much better result can be obtained by only using the WBL on the back-most atomic layer and letting the electrons propagate naturally through the rest of the contact. Because the WBL is computationally so inexpensive, we can easily calculate the DOS and transmission for 10,000 points instead of 1000.

A comparison of the resulting transmission with the calculation with self-energies is shown in the following figure:



The WBL shows good agreement with the non-WBL transmission around the Fermi energy (-0.195 Hartree or -5.306 eV). Note that the quality of the WBL depends on the choice of the coupling (ETA). For this particular contact geometry we obtain good agreement for ETA = 0.02 Hartree, but a better value may be found for other electrodes. Finally, the WBL can be incrementally improved by adding more gold layers to the extended molecule. For many layers it converges to the calculation with full self-energies.

```
#!/bin/sh

# In the wide-band limit (WBL) the coupling to the leads is assumed to be independent␣
↪of energy.
# Therefore one does not need to calculate any self-energies.
# This also means that the eigenspace of the Green's function is independent of␣
↪energy.
# It can therefore be diagonalized in advance,
# greatly speeding up the calculation of the DOS and the transmission.

# In the example $AMSHOME/examples/adf/green_Al/green_WBL.run of green,
# the transmission of  benzenedithiol junction, example green_BDT.run,
# in the wide-band limit (WBL) is calculated.
# In order to model the molecule-metal interface, we do need to include a few gold␣
↪layers
# in the calculation.
# However, unlike in green_Au.run, only a single atomic layer as the principal layer␣
↪is used.

# In green_BDT.run 3 layers of gold atoms from 1 fragment.
# In this example green_WBL.run each layer of gold atoms has its own fragment.
# Because a single atomic layer is an unnatural configuration for gold,
# a minor amount of smearing is necessary to make the calculation converge.
# The molecule is sandwiched in between the electrodes just like before
# (see the example for benzenedithiol, example green_BDT.run).
# However, this time each atomic layer of gold gets its own fragment.
# The reason for this configuration is that if the WBL is used on the
# entire gold contact the result is an an unphysical coupling to the leads;
# even the gold atoms contacting the molecule would have a direct coupling to the␣
↪environment.
# A much better result can be obtained by only using the WBL on the back-most atomic␣
↪layer
# and letting the electrons propagate naturally through the rest of the contact.
```

(continues on next page)

(continued from previous page)

```
# Because the WBL is computationally so inexpensive, we can easily calculate
# the DOS and transmission for 10,000 points instead of 1000.

cp $AMSHOME/examples/adf/green_Al/Au.5p .
cp $AMSHOME/examples/adf/green_Al/Au.5p.dirac .

$AMSBIN/dirac < Au.5p.dirac
mv TAPE12 t12.rel

AMS_JOBNAME=Au $AMSBIN/ams <<eor
System
   Atoms
      Au 0.0 0.0 0.0
   End
end
Task SinglePoint
Engine ADF
  corepotentials t12.rel
  end
  create Au   file=Au.5p
  relativity
    level scalar
    formalism ZORA
  end
  xc
    lda SCF VWN
  end
EndEngine
eor

AMS_JOBNAME=layer $AMSBIN/ams <<eor
System
  atoms
     Au        0.000000   -4.996959    0.000000
     Au        0.000000   -2.498480   -1.442498
     Au        0.000000   -2.498480    1.442498
     Au        0.000000    0.000000   -2.884996
     Au        0.000000    0.000000    0.000000
     Au        0.000000    0.000000    2.884996
     Au        0.000000    2.498480   -1.442498
     Au        0.000000    2.498480    1.442498
     Au        0.000000    4.996959    0.000000
  end
end

Task SinglePoint

Engine ADF
  fragments
     Au  Au.t21
  end
  occupations Smearq=0.001
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
```

(continues on next page)

(continued from previous page)

```
  title Principal layer
  xc
    lda SCF VWN
  end
EndEngine
eor

AMS_JOBNAME=molecule $AMSBIN/ams <<eor
System
  atoms
     C        -1.400000    0.000000    0.000000
     C        -0.700000    0.000000   -1.200000
     C        -0.700000    0.000000    1.200000
     C         0.700000    0.000000   -1.200000
     C         0.700000    0.000000    1.200000
     C         1.400000    0.000000    0.000000
     H        -1.200000    0.000000   -2.200000
     H        -1.200000    0.000000    2.200000
     H         1.200000    0.000000   -2.200000
     H         1.200000    0.000000    2.200000
     S        -3.200000    0.000000    0.000000
     S         3.200000    0.000000    0.000000
  end
end

Task SinglePoint

Engine ADF
  basis
     type DZP
     core Large
     createOutput None
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Benzenedithiol
  xc
    lda SCF VWN
  end
EndEngine
eor

AMS_JOBNAME=fock $AMSBIN/ams <<eor
System
  atoms
     Au       -9.911177   -6.662612    0.000000 adf.f=left
     Au       -9.911178   -4.164133   -1.442498 adf.f=left
     Au       -9.911178   -4.164133    1.442498 adf.f=left
     Au       -9.911178   -1.665653   -2.884996 adf.f=left
     Au       -9.911178   -1.665653    0.000000 adf.f=left
     Au       -9.911178   -1.665653    2.884996 adf.f=left
     Au       -9.911178    0.832826   -1.442498 adf.f=left
     Au       -9.911178    0.832826    1.442498 adf.f=left
     Au       -9.911178    3.331306    0.000000 adf.f=left
```

(continues on next page)

(continued from previous page)

```
    Au       -7.555589   -4.996959    0.000000 adf.f=left.2
    Au       -7.555589   -2.498480   -1.442498 adf.f=left.2
    Au       -7.555589   -2.498480    1.442498 adf.f=left.2
    Au       -7.555589    0.000000   -2.884996 adf.f=left.2
    Au       -7.555589    0.000000    0.000000 adf.f=left.2
    Au       -7.555589    0.000000    2.884996 adf.f=left.2
    Au       -7.555589    2.498480   -1.442498 adf.f=left.2
    Au       -7.555589    2.498480    1.442498 adf.f=left.2
    Au       -7.555589    4.996959    0.000000 adf.f=left.2
    Au       -5.200000   -3.331306    0.000000 adf.f=left.3
    Au       -5.200000   -0.832826   -1.442498 adf.f=left.3
    Au       -5.200000   -0.832826    1.442498 adf.f=left.3
    Au       -5.200000    1.665653   -2.884996 adf.f=left.3
    Au       -5.200000    1.665653    0.000000 adf.f=left.3
    Au       -5.200000    1.665653    2.884996 adf.f=left.3
    Au       -5.200000    4.164133   -1.442498 adf.f=left.3
    Au       -5.200000    4.164133    1.442498 adf.f=left.3
    Au       -5.200001    6.662612    0.000000 adf.f=left.3
    C        -1.400000    0.000000    0.000000 adf.f=molecule
    C        -0.700000    0.000000   -1.200000 adf.f=molecule
    C        -0.700000    0.000000    1.200000 adf.f=molecule
    C         0.700000    0.000000   -1.200000 adf.f=molecule
    C         0.700000    0.000000    1.200000 adf.f=molecule
    C         1.400000    0.000000    0.000000 adf.f=molecule
    H        -1.200000    0.000000   -2.200000 adf.f=molecule
    H        -1.200000    0.000000    2.200000 adf.f=molecule
    H         1.200000    0.000000   -2.200000 adf.f=molecule
    H         1.200000    0.000000    2.200000 adf.f=molecule
    S        -3.200000    0.000000    0.000000 adf.f=molecule
    S         3.200000    0.000000    0.000000 adf.f=molecule
    Au        5.200001   -6.662612    0.000000 adf.f=right.3
    Au        5.200000   -4.164133   -1.442498 adf.f=right.3
    Au        5.200000   -4.164133    1.442498 adf.f=right.3
    Au        5.200000   -1.665653   -2.884996 adf.f=right.3
    Au        5.200000   -1.665653    0.000000 adf.f=right.3
    Au        5.200000   -1.665653    2.884996 adf.f=right.3
    Au        5.200000    0.832826   -1.442498 adf.f=right.3
    Au        5.200000    0.832826    1.442498 adf.f=right.3
    Au        5.200000    3.331306    0.000000 adf.f=right.3
    Au        7.555589   -4.996959    0.000000 adf.f=right.2
    Au        7.555589   -2.498480   -1.442498 adf.f=right.2
    Au        7.555589   -2.498480    1.442498 adf.f=right.2
    Au        7.555589    0.000000   -2.884996 adf.f=right.2
    Au        7.555589    0.000000    0.000000 adf.f=right.2
    Au        7.555589    0.000000    2.884996 adf.f=right.2
    Au        7.555589    2.498480   -1.442498 adf.f=right.2
    Au        7.555589    2.498480    1.442498 adf.f=right.2
    Au        7.555589    4.996959    0.000000 adf.f=right.2
    Au        9.911178   -3.331306    0.000000 adf.f=right
    Au        9.911178   -0.832826   -1.442498 adf.f=right
    Au        9.911178   -0.832826    1.442498 adf.f=right
    Au        9.911178    1.665653   -2.884996 adf.f=right
    Au        9.911178    1.665653    0.000000 adf.f=right
    Au        9.911178    1.665653    2.884996 adf.f=right
    Au        9.911178    4.164133   -1.442498 adf.f=right
    Au        9.911178    4.164133    1.442498 adf.f=right
    Au        9.911177    6.662612    0.000000 adf.f=right
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  fragments
     left        layer.t21
     left.2      layer.t21
     left.3      layer.t21
     molecule    molecule.t21
     right.3     layer.t21
     right.2     layer.t21
     right       layer.t21
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Benzenedithiol
  xc
    lda SCF VWN
  end
EndEngine
eor

$AMSBIN/green << eor
DOS fock.results/adf.rkf
TRANS fock.results/adf.rkf
EPS -0.5 0 10000
ETA 1e-6
LEFT
    FRAGMENT left
    ETA 2e-2
END
RIGHT
    FRAGMENT right
    ETA 2e-2
END
NOSAVE DOS_B, TRANS_B
eor

echo ""
echo "Contents of DOS_A:"
cat DOS_A
echo "END"
echo ""
echo "Contents of TRANS_A:"
cat TRANS_A
echo "END"
```

Example: Benzenedithiol junction

`Download green_BDT.run`

In this example of *green*, the DOS and transmission of a benzenedithiol molecule between gold electrodes is calculated. The calculation uses the self-energies obtained in *the example for the gold electrodes* (page 967). Note that this is a relatively expensive calculation.

First a fragment of the isolated molecule is constructed. Next the molecule is sandwiched between the electrodes in the configuration of Fig. 2. For this the fragment of the principal layer obtained in *the example for the gold electrodes* (page 967) is needed.

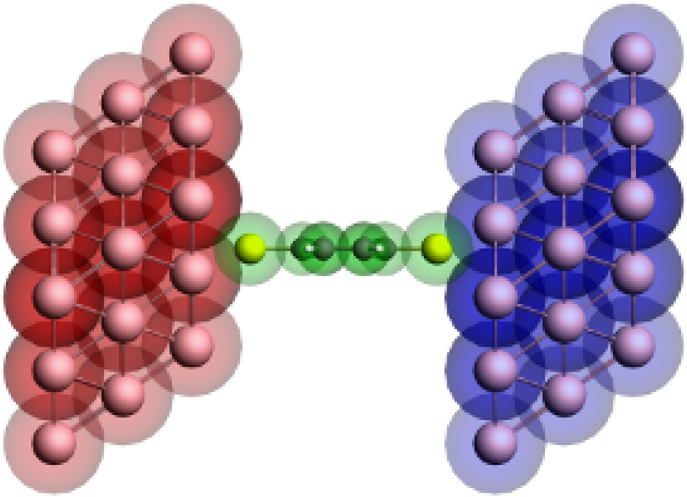
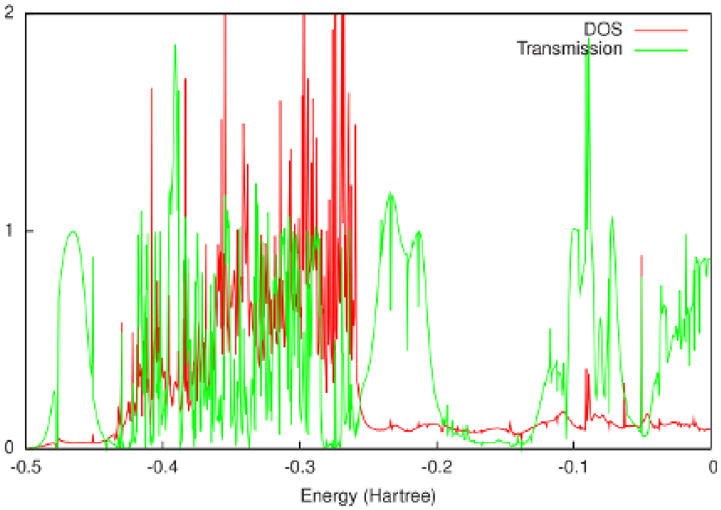


Figure 2: Geometry of the extended molecule used in the calculation of a benzenedithiol junction. The molecule is shown in green, while the left and right contact regions are shown in red and blue, respectively. Note that the red region corresponds to the blue surface layer in Figure 1 of *the example for the gold electrodes* (page 967) and vice versa.

The final results are shown in the following figure:



The Fermi energy of the electrodes is -0.2 Hartree (see *the example for the gold electrodes* (page 967)). This is just above the HOMO of the molecular junction. Consistent with literature, the HOMO and lower orbitals are combined into a broad peak just below the Fermi energy, while the LUMO is much sharper and situated about 2 eV above the Fermi energy.

The current can be calculated by integrating the transmission around the Fermi energy. At low temperatures this means that the differential conductance is equal to the transmission times the quantum of conductance.

```
#!/bin/sh

# In this example of green, the DOS and transmission of a benzenedithiol molecule
# between gold electrodes is calculated.
# The calculation uses the self-energies obtained in the example for the gold␣
↪electrodes,
# green_Au.run.
# Note that this is a relatively expensive calculation.

cp $AMSHOME/examples/adf/green_Al/Au.5p .
cp $AMSHOME/examples/adf/green_Al/Au.5p.dirac .

$AMSBIN/dirac < Au.5p.dirac
mv TAPE12 t12.rel

AMS_JOBNAME=Au $AMSBIN/ams <<eor
System
   Atoms
      Au 0.0 0.0 0.0
   End
end
Task SinglePoint
Engine ADF
  corepotentials t12.rel
  end
  create Au   file=Au.5p
  relativity
    level scalar
    formalism ZORA
  end
  xc
    lda SCF VWN
  end
EndEngine
eor

AMS_JOBNAME=layer $AMSBIN/ams <<eor
System
  atoms
     Au       -2.355588   -6.662612    0.000000
     Au       -2.355589   -4.164133   -1.442498
     Au       -2.355589   -4.164133    1.442498
     Au       -2.355589   -1.665653   -2.884996
     Au       -2.355589   -1.665653    0.000000
     Au       -2.355589   -1.665653    2.884996
     Au       -2.355589    0.832826   -1.442498
     Au       -2.355589    0.832826    1.442498
     Au       -2.355589    3.331306    0.000000
     Au        0.000000   -4.996959    0.000000
     Au        0.000000   -2.498480   -1.442498
     Au        0.000000   -2.498480    1.442498
     Au        0.000000    0.000000   -2.884996
     Au        0.000000    0.000000    0.000000
     Au        0.000000    0.000000    2.884996
     Au        0.000000    2.498480   -1.442498
     Au        0.000000    2.498480    1.442498
     Au        0.000000    4.996959    0.000000
```

(continues on next page)

(continued from previous page)

```
      Au        2.355589   -3.331306    0.000000
      Au        2.355589   -0.832826   -1.442498
      Au        2.355589   -0.832826    1.442498
      Au        2.355589    1.665653   -2.884996
      Au        2.355589    1.665653    0.000000
      Au        2.355589    1.665653    2.884996
      Au        2.355589    4.164133   -1.442498
      Au        2.355589    4.164133    1.442498
      Au        2.355588    6.662612    0.000000
  end
end

Task SinglePoint

Engine ADF
  fragments
      Au  Au.results/adf.rkf
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Principal layer
  xc
    lda SCF VWN
  end
EndEngine
eor

AMS_JOBNAME=bulk $AMSBIN/ams <<eor
System
  atoms
      Au       -9.422355  -11.659571    0.000000 adf.f=left
      Au       -9.422356   -9.161092   -1.442498 adf.f=left
      Au       -9.422356   -9.161092    1.442498 adf.f=left
      Au       -9.422356   -6.662612   -2.884996 adf.f=left
      Au       -9.422356   -6.662612    0.000000 adf.f=left
      Au       -9.422356   -6.662612    2.884996 adf.f=left
      Au       -9.422356   -4.164133   -1.442498 adf.f=left
      Au       -9.422356   -4.164133    1.442498 adf.f=left
      Au       -9.422356   -1.665653    0.000000 adf.f=left
      Au       -7.066767   -9.993918    0.000000 adf.f=left
      Au       -7.066767   -7.495439   -1.442498 adf.f=left
      Au       -7.066767   -7.495439    1.442498 adf.f=left
      Au       -7.066767   -4.996959   -2.884996 adf.f=left
      Au       -7.066767   -4.996959    0.000000 adf.f=left
      Au       -7.066767   -4.996959    2.884996 adf.f=left
      Au       -7.066767   -2.498479   -1.442498 adf.f=left
      Au       -7.066767   -2.498479    1.442498 adf.f=left
      Au       -7.066767    0.000000    0.000000 adf.f=left
      Au       -4.711178   -8.328265    0.000000 adf.f=left
      Au       -4.711178   -5.829785   -1.442498 adf.f=left
      Au       -4.711178   -5.829785    1.442498 adf.f=left
      Au       -4.711178   -3.331306   -2.884996 adf.f=left
      Au       -4.711178   -3.331306    0.000000 adf.f=left
      Au       -4.711178   -3.331306    2.884996 adf.f=left
```

(continues on next page)

(continued from previous page)

```
      Au       -4.711178   -0.832826   -1.442498 adf.f=left
      Au       -4.711178   -0.832826    1.442498 adf.f=left
      Au       -4.711179    1.665653    0.000000 adf.f=left
      Au       -2.355588   -6.662612    0.000000 adf.f=center
      Au       -2.355589   -4.164133   -1.442498 adf.f=center
      Au       -2.355589   -4.164133    1.442498 adf.f=center
      Au       -2.355589   -1.665653   -2.884996 adf.f=center
      Au       -2.355589   -1.665653    0.000000 adf.f=center
      Au       -2.355589   -1.665653    2.884996 adf.f=center
      Au       -2.355589    0.832826   -1.442498 adf.f=center
      Au       -2.355589    0.832826    1.442498 adf.f=center
      Au       -2.355589    3.331306    0.000000 adf.f=center
      Au        0.000000   -4.996959    0.000000 adf.f=center
      Au        0.000000   -2.498480   -1.442498 adf.f=center
      Au        0.000000   -2.498480    1.442498 adf.f=center
      Au        0.000000    0.000000   -2.884996 adf.f=center
      Au        0.000000    0.000000    0.000000 adf.f=center
      Au        0.000000    0.000000    2.884996 adf.f=center
      Au        0.000000    2.498480   -1.442498 adf.f=center
      Au        0.000000    2.498480    1.442498 adf.f=center
      Au        0.000000    4.996959    0.000000 adf.f=center
      Au        2.355589   -3.331306    0.000000 adf.f=center
      Au        2.355589   -0.832826   -1.442498 adf.f=center
      Au        2.355589   -0.832826    1.442498 adf.f=center
      Au        2.355589    1.665653   -2.884996 adf.f=center
      Au        2.355589    1.665653    0.000000 adf.f=center
      Au        2.355589    1.665653    2.884996 adf.f=center
      Au        2.355589    4.164133   -1.442498 adf.f=center
      Au        2.355589    4.164133    1.442498 adf.f=center
      Au        2.355588    6.662612    0.000000 adf.f=center
      Au        4.711179   -1.665653    0.000000 adf.f=right
      Au        4.711178    0.832826   -1.442498 adf.f=right
      Au        4.711178    0.832826    1.442498 adf.f=right
      Au        4.711178    3.331306   -2.884996 adf.f=right
      Au        4.711178    3.331306    0.000000 adf.f=right
      Au        4.711178    3.331306    2.884996 adf.f=right
      Au        4.711178    5.829785   -1.442498 adf.f=right
      Au        4.711178    5.829785    1.442498 adf.f=right
      Au        4.711178    8.328265    0.000000 adf.f=right
      Au        7.066767    0.000000    0.000000 adf.f=right
      Au        7.066767    2.498479   -1.442498 adf.f=right
      Au        7.066767    2.498479    1.442498 adf.f=right
      Au        7.066767    4.996959   -2.884996 adf.f=right
      Au        7.066767    4.996959    0.000000 adf.f=right
      Au        7.066767    4.996959    2.884996 adf.f=right
      Au        7.066767    7.495439   -1.442498 adf.f=right
      Au        7.066767    7.495439    1.442498 adf.f=right
      Au        7.066767    9.993918    0.000000 adf.f=right
      Au        9.422356    1.665653    0.000000 adf.f=right
      Au        9.422356    4.164133   -1.442498 adf.f=right
      Au        9.422356    4.164133    1.442498 adf.f=right
      Au        9.422356    6.662612   -2.884996 adf.f=right
      Au        9.422356    6.662612    0.000000 adf.f=right
      Au        9.422356    6.662612    2.884996 adf.f=right
      Au        9.422356    9.161092   -1.442498 adf.f=right
      Au        9.422356    9.161092    1.442498 adf.f=right
      Au        9.422355   11.659571    0.000000 adf.f=right
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  fragments
     left    layer.results/adf.rkf
     center  layer.results/adf.rkf
     right   layer.results/adf.rkf
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Bulk gold
  xc
    lda SCF VWN
  end
EndEngine
eor

$AMSBIN/green << eor
SURFACE bulk.results/adf.rkf
    FRAGMENTS center right
END
EPS -0.5 0 1000
ETA 1e-6
eor
mv SURFACE left.kf

$AMSBIN/green << eor
SURFACE bulk.results/adf.rkf
    FRAGMENTS center left
END
EPS -0.5 0 1000
ETA 1e-6
eor
mv SURFACE right.kf

# A fragment of the isolated molecule is constructed:

AMS_JOBNAME=molecule $AMSBIN/ams <<eor
System
  atoms
     C        -1.400000    0.000000    0.000000
     C        -0.700000    0.000000   -1.200000
     C        -0.700000    0.000000    1.200000
     C         0.700000    0.000000   -1.200000
     C         0.700000    0.000000    1.200000
     C         1.400000    0.000000    0.000000
     H        -1.200000    0.000000   -2.200000
     H        -1.200000    0.000000    2.200000
     H         1.200000    0.000000   -2.200000
     H         1.200000    0.000000    2.200000
     S        -3.200000    0.000000    0.000000
```

(continues on next page)

(continued from previous page)

```
    S         3.200000    0.000000    0.000000
  end
end

Task SinglePoint

Engine ADF
  basis
     type DZP
     core Large
     createOutput None
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Benzenedithiol
  xc
    lda SCF VWN
  end
EndEngine
eor

# Next the molecule is sandwiched between the electrodes:

AMS_JOBNAME=fock $AMSBIN/ams <<eor
System
  atoms
     Au       -9.911177   -6.662612    0.000000 adf.f=left
     Au       -9.911178   -4.164133   -1.442498 adf.f=left
     Au       -9.911178   -4.164133    1.442498 adf.f=left
     Au       -9.911178   -1.665653   -2.884996 adf.f=left
     Au       -9.911178   -1.665653    0.000000 adf.f=left
     Au       -9.911178   -1.665653    2.884996 adf.f=left
     Au       -9.911178    0.832826   -1.442498 adf.f=left
     Au       -9.911178    0.832826    1.442498 adf.f=left
     Au       -9.911178    3.331306    0.000000 adf.f=left
     Au       -7.555589   -4.996959    0.000000 adf.f=left
     Au       -7.555589   -2.498480   -1.442498 adf.f=left
     Au       -7.555589   -2.498480    1.442498 adf.f=left
     Au       -7.555589    0.000000   -2.884996 adf.f=left
     Au       -7.555589    0.000000    0.000000 adf.f=left
     Au       -7.555589    0.000000    2.884996 adf.f=left
     Au       -7.555589    2.498480   -1.442498 adf.f=left
     Au       -7.555589    2.498480    1.442498 adf.f=left
     Au       -7.555589    4.996959    0.000000 adf.f=left
     Au       -5.200000   -3.331306    0.000000 adf.f=left
     Au       -5.200000   -0.832826   -1.442498 adf.f=left
     Au       -5.200000   -0.832826    1.442498 adf.f=left
     Au       -5.200000    1.665653   -2.884996 adf.f=left
     Au       -5.200000    1.665653    0.000000 adf.f=left
     Au       -5.200000    1.665653    2.884996 adf.f=left
     Au       -5.200000    4.164133   -1.442498 adf.f=left
     Au       -5.200000    4.164133    1.442498 adf.f=left
     Au       -5.200001    6.662612    0.000000 adf.f=left
     C        -1.400000    0.000000    0.000000 adf.f=molecule
```

(continues on next page)

(continued from previous page)

```
     C        -0.700000    0.000000   -1.200000 adf.f=molecule
     C        -0.700000    0.000000    1.200000 adf.f=molecule
     C         0.700000    0.000000   -1.200000 adf.f=molecule
     C         0.700000    0.000000    1.200000 adf.f=molecule
     C         1.400000    0.000000    0.000000 adf.f=molecule
     H        -1.200000    0.000000   -2.200000 adf.f=molecule
     H        -1.200000    0.000000    2.200000 adf.f=molecule
     H         1.200000    0.000000   -2.200000 adf.f=molecule
     H         1.200000    0.000000    2.200000 adf.f=molecule
     S        -3.200000    0.000000    0.000000 adf.f=molecule
     S         3.200000    0.000000    0.000000 adf.f=molecule
     Au        5.200001   -6.662612    0.000000 adf.f=right
     Au        5.200000   -4.164133   -1.442498 adf.f=right
     Au        5.200000   -4.164133    1.442498 adf.f=right
     Au        5.200000   -1.665653   -2.884996 adf.f=right
     Au        5.200000   -1.665653    0.000000 adf.f=right
     Au        5.200000   -1.665653    2.884996 adf.f=right
     Au        5.200000    0.832826   -1.442498 adf.f=right
     Au        5.200000    0.832826    1.442498 adf.f=right
     Au        5.200000    3.331306    0.000000 adf.f=right
     Au        7.555589   -4.996959    0.000000 adf.f=right
     Au        7.555589   -2.498480   -1.442498 adf.f=right
     Au        7.555589   -2.498480    1.442498 adf.f=right
     Au        7.555589    0.000000   -2.884996 adf.f=right
     Au        7.555589    0.000000    0.000000 adf.f=right
     Au        7.555589    0.000000    2.884996 adf.f=right
     Au        7.555589    2.498480   -1.442498 adf.f=right
     Au        7.555589    2.498480    1.442498 adf.f=right
     Au        7.555589    4.996959    0.000000 adf.f=right
     Au        9.911178   -3.331306    0.000000 adf.f=right
     Au        9.911178   -0.832826   -1.442498 adf.f=right
     Au        9.911178   -0.832826    1.442498 adf.f=right
     Au        9.911178    1.665653   -2.884996 adf.f=right
     Au        9.911178    1.665653    0.000000 adf.f=right
     Au        9.911178    1.665653    2.884996 adf.f=right
     Au        9.911178    4.164133   -1.442498 adf.f=right
     Au        9.911178    4.164133    1.442498 adf.f=right
     Au        9.911177    6.662612    0.000000 adf.f=right
  end
end

Task SinglePoint

Engine ADF
  fragments
     left        layer.results/adf.rkf
     molecule    molecule.results/adf.rkf
     right       layer.results/adf.rkf
  end
  relativity
    level scalar
    formalism ZORA
  end
  symmetry NOSYM
  title Benzenedithiol
  xc
    lda SCF VWN
```

(continues on next page)

(continued from previous page)

```
  end
EndEngine
eor

# The DOS and transmission can now be calculated:

$AMSBIN/green << eor
DOS fock.results/adf.rkf
TRANS fock.results/adf.rkf
EPS -0.5 0 1000
ETA 1e-6
LEFT left.kf
    FRAGMENT left
END
RIGHT right.kf
    FRAGMENT right
END
NOSAVE DOS_B, TRANS_B
eor

echo ""
echo "Contents of DOS_A:"
cat DOS_A
echo "END"
echo ""
echo "Contents of TRANS_A:"
cat TRANS_A
echo "END"
```

14.6 Analysis

14.6.1 Fragment orbitals, bond energy decomposition

Example: Compound Fragments: Ni(CO)4

Download `Frags_NiCO4.run`

```
#! /bin/sh


# An illustration of the fragment feature of ADF.

# A transition metal complex is built from a Nickel atom and four CO fragments.
# The outcomes allows for an analysis (of molecular orbitals and the Bonding
# energy) in terms of the fragment properties. It is a Single Point calculation.
# Geometry optimization would not have been possible in this set-up because an
# optimization requires that only single-atom fragments are used.

# The three atoms are created first: C, O, and Ni. For Carbon and Oxygen a type-
# DZ basis set is used (double-zeta) using the Basis key, while Ni gets a type-
# TZP basis (triple-zeta plus polarization).
```

(continues on next page)

(continued from previous page)

```
# == CO ==


# The CO molecule, to serve as a fragment template in Ni(CO)4 , is computed from
# the atomic fragments C and O. The coordinate values (atoms) are in bohr,
# rather than in Angstrom because the unit-of-length is redefined by the key
# units with subkey length.

# The key scf is used to specify a somewhat tighter convergence criterion than
# the default, just to illustrate how to do this (normal settings are quite
# adequate).

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/Ni.2p
mv TAPE12 t12.rel

AMS_JOBNAME=Ni $AMSBIN/ams <<eor > tmp
System
  Atoms
    Ni 0.0 0.0 0.0
  End
End
Task SinglePoint
Engine ADF
  create Ni  $AMSRESOURCES/ADF/ZORA/TZP/Ni.2p
  corepotentials t12.rel
     Ni 1
  end
EndEngine
eor


AMS_JOBNAME=CO $AMSBIN/ams <<eor
System
  atoms [bohr]
     C   0  0  0
     O   0  0  2.15617844
  end
end

Task SinglePoint

Engine ADF
  title CO (as fragment for NiCO4)
  eprint
    sfo eig ovl
  end
  basis
    core Small
    type DZ
  end
  scf
    converge 1e-8
  end
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
# One needs to include the subkey SFO of the key EPRINT with arguments eig and
# ovl in order to get the SFO MO coefficients and SFO overlap matrix printed on
# standard output.


# == Main calculation ==


# Apart from the title, the input contains comment. This does not specify
# computational parameters but is only echoed in the output header, similar to
# the title. Contrary to the title, however, such comments are not preserved,
# apart from their echo in output and they are not written to adf.rkf or any
# other result file.

# The atomic coordinates (atoms) are given in bohr (Units).

# The Atoms records contain also a specification of the fragments to which the
# respective atoms belong: four different CO fragments. No fragment is specified
# for the Ni atom, which implies that it is a fragment on its own.

# The numbers at the very left of the records (1 through 9, with (optionally) a
# period after them), have no relevance. You can set them for ease of reference
# or counting.


AMS_JOBNAME=NiCO4 $AMSBIN/ams <<eor
System
  atoms [bohr]
     Ni   0.0           0.0            0.0
     C    2.0053211    2.0053211     2.0053211     adf.f=CO|1
     C   -2.0053211   -2.0053211     2.0053211     adf.f=CO|2
     C    2.0053211   -2.0053211    -2.0053211     adf.f=CO|3
     C   -2.0053211    2.0053211    -2.0053211     adf.f=CO|4
     O    3.2501913    3.2501913     3.2501913     adf.f=CO|1
     O   -3.2501913   -3.2501913     3.2501913     adf.f=CO|2
     O    3.2501913   -3.2501913    -3.2501913     adf.f=CO|3
     O   -3.2501913    3.2501913    -3.2501913     adf.f=CO|4
  end
end

Task SinglePoint

Engine ADF
  title Ni(CO)4,  from fragments Ni and CO
  eprint
    sfo eig ovl
  end
  fragments
     CO  CO.results/adf.rkf
     Ni  Ni.results/adf.rkf
  end
EndEngine
eor
```

Example: Fragments: PtCl4H2 2-

Download Frags_PtCl4H2.run

```
#! /bin/sh

# The (scalar) ZORA relativistic option formalism) is used because of the
# presence of the heavy Pt atom. The complex is built from fragments H2 and
# PtCl_4^2- .

# The calculations of the molecule and larger fragments are performed with
# GGA's.


# == fragments H2 and PtCl_4^2- ==

# The two fragments H2 and PtCl_4^2- are first calculated, from which we are
# going to build the final complex.


AMS_JOBNAME=H2 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     H       0.0              0.0              0.84
     H       0.0              0.0             -0.84
  end
end

Task SinglePoint

Engine ADF
  title H2
  basis
    type DZP
    CreateOutput Yes
  end
  noprint sfo,frag,functions
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine
eor

AMS_JOBNAME=PtCl4 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     Pt    0            0           0
     Cl    4.361580     0.000000    0
     Cl    0.000000     4.361580    0
     Cl   -4.361580     0.000000    0
     Cl    0.000000    -4.361580    0
  end
  charge -2
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title PtCl4 (2-)
  basis
    PerAtomType Symbol=Pt File=ZORA/DZ/Pt.4d
    type DZP
    CreateOutput Yes
  end
  noprint sfo,frag,functions
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine
eor


# The key charge is used to specify the net total charge. The default for the
# net total charge is the sum-of-fragment-charges. The fragments (Pt and Cl
# atoms) have been computed neutrally, but we want to calculate the PtCl4
# complex as a 2- ion.


# == Main calculation ==

# Finally we compute PtCl4 H2 2- from the fragments PtCl_4^2- and H2/


AMS_JOBNAME=PtCl4H2 $AMSBIN/ams <<eor
System
  atoms [Bohr]
     Pt    0              0             0             adf.f=PtCl4
     Cl    0.000000     -4.361580      0.00000000    adf.f=PtCl4
     Cl    0.000000      4.361580      0.00000000    adf.f=PtCl4
     Cl   -4.361580      0.000000      0.00000000    adf.f=PtCl4
     Cl    4.361580      0.000000      0.00000000    adf.f=PtCl4
     H     0.0           0.0           5.58          adf.f=H2
     H     0.0           0.0           7.26          adf.f=H2
  end
  charge -2
end

Task SinglePoint

Engine ADF
  title PtCl4 H2
  eprint
    sfo eig ovl
  end
  fragments
     PtCl4     PtCl4.results/adf.rkf
     H2        H2.results/adf.rkf
```

(continues on next page)

(continued from previous page)

```
  end
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga becke perdew
  end
EndEngine
eor

# Note the adf.f=fragment specification in the Atoms block. No fragment-numbering
# suffix (|n) is required because there is only one fragment of each fragment
# type.
```

Example: Spin-unrestricted Fragments: H2

Download UnrFrag_H2.run

```
#! /bin/sh


# This is a small but important example to illustrate what goes into an accurate
# calculation of the 'true' bond energy of a molecule. The (ADF-specific)
# problem is that in a straightforward molecular calculation, the bond energy is
# computed as the energy difference between at the one hand the molecule, and at
# the other hand the isolated spherically symmetric spin-restricted atoms.
# This spherically symmetric spin-restricted reference (comparison) state is usually
# not the physical ground state of the reference system (isolated atoms) and
# hence the computed energy difference has no direct relation to experimental
# data. To account for the true atomic ground states, one has to add correction
# terms. Study this sample carefully to make sure that you fully understand the
# steps to take and consult the User's Guide for details. See also the this
# document for a discussion of multiplet states.

# See also the example, SD_Cr(NH3)6.

# The H2 case consists of a sequence of simple calculations to demonstrate the
# Unrestricted Fragments options.
# In the first part real spin-unrestricted fragments are used.
# In the second part simulate spin-unrestricted fragments are used.

# Unrestricted fragments
# ======================

# If one uses unrestricted fragments one needs to calculate the unrestricted
# H atom twice, one in which the 1s alpha orbital is occupied, and one in which
# the 1s beta orbital is occupied.

AMS_JOBNAME=H_A $AMSBIN/ams <<eor
System
  atoms
     H  0.0 0.0 0.0
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  eprint
    sfo eig ovl
  end
  basis
    type DZP
  end
  spinpolarization 1
  unrestricted
EndEngine
eor

AMS_JOBNAME=H_B $AMSBIN/ams <<eor
System
  atoms
     H  0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  eprint
    sfo eig ovl
  end
  basis
    type DZP
  end
  spinpolarization -1
  unrestricted
EndEngine
eor

AMS_JOBNAME=H2_AB $AMSBIN/ams <<eor
System
  atoms
     H   0   0   0.375  adf.f=H_A
     H   0   0  -0.375  adf.f=H_B
  end
end

Task SinglePoint

Engine ADF
  title H2 from unrestricted fragments
  eprint
    sfo eig ovl
  end
  fragments
     H_A  H_A.results/adf.rkf
     H_B  H_B.results/adf.rkf
  end
  spinpolarization 0
  unrestricted
```

(continues on next page)

(continued from previous page)

```
  unrestrictedfragments
EndEngine
eor

# In the unrestricted fragment different numbers of spin-alpha and spin-beta␣
↪electrons usually
# result in different spatial orbitals and different energy eigenvalues for
# spin-alpha and spin-beta when you go to self-consistency.
# For H2 calculated from real unrestricted fragments this will mean that the SFO␣
↪analysis of
# the H2 molecular orbitals looks a bit unsymmetric.
# However, if one simulates the unrestricted fragments with the key FRAGOCCUPATIONS
# the SFO analysis for H2 will be symmetric.


# Simulated unrestricted fragments with the key FRAGOCCUPATIONS
# ==============================================================

# In this approach spin-restricted fragments are used, in which
# one has the same spatial orbitals and energy eigenvalues for
# spin-alpha and spin-beta.
# The energy difference between an unrestricted
# fragment as it is used with the key FRAGOCCUPATIONS and a self-consistent
# unrestricted fragment is computed. This turns out to be quite small,
# confirming that this approach with simulated unrestricted fragments,
# although not formally exact, is also adequate for practical purposes.

$AMSBIN/dirac -n1 < $AMSRESOURCES/Dirac/H
mv TAPE12 t12.rel

AMS_JOBNAME=H_R $AMSBIN/ams <<eor
System
  Atoms
    H 0.0 0.0 0.0
  End
end
Task SinglePoint
Engine ADF
  create H    file=$AMSRESOURCES/ADF/ZORA/DZP/H
  corepotentials t12.rel
     H 1
  end
EndEngine
eor

AMS_JOBNAME=H_U_noscf $AMSBIN/ams <<eor
System
  atoms
     H 0 0 0
  end
end

Task SinglePoint

Engine ADF
  title H unrestr., not self-consistent (as used in unr.frag. calcs)
  eprint
```

(continues on next page)

(continued from previous page)

```
    sfo eig ovl
  end
  fragments
     H  H_R.results/adf.rkf
  end
  scf
    iterations 0
  end
  spinpolarization 1
  unrestricted
EndEngine

eor

# By setting the scf iterations to zero (a value of one (1) would give the same
# result) we prevent cycling to self-consistency. The energy of the 'final' one-
# electron orbitals is consequently computed in the start-up potential, i.e. the
# field of the restricted (basic) atom, where spin-alpha and spin-beta are equally
# occupied, in this case by 0.5 electron each. The not-self-consistent,
# unrestricted H atom is precisely the 'unrestricted' fragment as it can be used
# in an adf calculation with the FRAGOCCUPATIONS key. The fragment file must be
# the TAPE21 result file from a restricted run, but at start-up you can specify
# that the Fragment Orbitals are, for purposes of reference and comparison,
# occupied in an unrestricted way in the final molecule.

# A calculation that uses restricted fragments right away computes the bonding
# energy relative to the restricted fragments. The difference between using
# restricted and unrestricted fragments is the 'bonding' energy computed in the
# run above.

AMS_JOBNAME=H_U_scf $AMSBIN/ams <<eor
System
  atoms
     H  0 0 0
  end
end

Task SinglePoint

Engine ADF
  title H  unr. self-consistent from unr.0
  eprint
    sfo eig ovl
  end
  fragments
     H  H_R.results/adf.rkf
  end
  fragoccupations
     H
     s  1 // 0
     subend
  end
  spinpolarization 1
  unrestricted
EndEngine

eor
```

(continues on next page)

(continued from previous page)

```
# Here we start with the unrestricted fragment and relax to self-consistency.
# The 'bonding energy', i.e. the relaxation energy, is very small, demonstrating
# that using non-self-consistent unrestricted fragments involves only a small
# error (which, moreover, can be computed as shown here).

# The key UnRestricted sets the spin-unrestricted mode. The key Charge is used
# to specify a net total charge of zero and a net total spin polarization by an
# excess of 1.0 spin-alpha electrons over spin-beta.

AMS_JOBNAME=H2_R $AMSBIN/ams <<eor
System
  atoms
     H   0  0   0.375
     H   0  0  -0.375
  end
end

Task SinglePoint

Engine ADF
  title H2 restricted, from restricted fragments
  eprint
    sfo eig ovl
  end
  fragments
     H  H_R.results/adf.rkf
  end
EndEngine

eor

# This is the simplest approach, using restricted fragments. The bonding energy
# must be corrected because the reference (restricted H atoms, with 0.5
# electrons in spin-alpha and 0.5 in spin-beta) is far from the true H-atom
# ground state: see the previous runs on the single H atom.

AMS_JOBNAME=H2_U $AMSBIN/ams <<eor
System
  atoms
     H.1   0   0   0.375
     H.2   0   0  -0.375
  end
end

Task SinglePoint

Engine ADF
  title H2 from simulated unrestricted fragments using the FRAGOCCUPATIONS key
  eprint
    sfo eig ovl
  end
  fragments
     H.1  H_R.results/adf.rkf
     H.2  H_R.results/adf.rkf
  end
  fragoccupations
```

(continues on next page)

(continued from previous page)

```
     H.1
     s  1 // 0
     subend
     H.2
     s  0 // 1
     subend
  end
  irrepoccupations
     sigma   2
  end
  modifystartpotential
     H.1  1 // 0
     H.2  0 // 1
  end
EndEngine

eor

# This should be a fair approximation (in the LDA model) to the bonding energy
# of H2 with respect to the unrestricted H atoms. The difference between the
# bonding energies of this and the previous run should be very close to the
# energy of the not-self-consistent unrestricted H-atom with respect to the
# restricted basic atom.

# == Excited state ==

AMS_JOBNAME=H2_excited $AMSBIN/ams <<eor
System
  atoms
     H    0 .0     0.375
     H    0 .0    -0.375
  end
end

Task SinglePoint

Engine ADF
  title H2  excited
  eprint
    sfo eig ovl
  end
  fragments
     H  H_R.results/adf.rkf
  end
  fragoccupations
     H
     s  1 // 0
     subend
  end
  irrepoccupations
     sigma.g   1 // 0
     sigma.u   1 // 0
  end
  spinpolarization 2
  unrestricted
EndEngine
```

(continues on next page)

(continued from previous page)

```
eor

# Finally the calculation of an excited state, with respect to unrestricted
# fragments. The excitation energy is obtained by comparing the energy with the
# energy of the ground state calculation. This difference compares reasonably,
# but not accurately, to the difference in one-electron ground state energies of
# the involved orbitals (Koopman's theorem).

# Note that excitation energies can also be calculated with Time-Dependent DFT,
# using the EXCITATIONS key of ADF. See related sample runs.
```

Example: Bond Energy analysis open-shell fragments: PCCP

Download PCCP_Unr_BondEnergy.run

```
#! /bin/sh


# This example illustrates advanced usage of the bond energy decomposition
# scheme used in ADF.

# Remark: this calculation simulates unrestricted fragments using the FRAGOCCUPATIONS
# key. Real unrestricted fragments can also be used in ADF.

# A proper decomposition of an electron-pair bond energy requires specifying
# opposite spins for the unpaired electrons of the respective radical fragments,
# which can be done with the input key FragOccupations. The specified alpha- and
# beta-spin configurations of the radical fragments are shown in the output
# section B U I L D.

# Please note that if one neglects explicitly specifying opposite spins for the
# unpaired electrons of the fragments, each of them is treated as being half an
# alpha and half a beta electron and consequently, they enter into a spurious
# Pauli repulsive interaction. This results, among others, into the Pauli
# repulsion term being too repulsive and the orbital interaction term being too
# much stabilizing.

# The example consists of an analysis of the C-C single bond between two CP
# radicals in the four-atomic molecule PCCP. The CP fragment calculations used
# to provide the adf.rkf for the overall PCCP calculation are done here in
# the restricted mode ('cp_fpccp_asr'). The proper spins
# are then specified in the calculation of the overall molecule using the
# FragOccupations key ('pccp_fa1_as'). Note that this implies a slight
# approximation because the bond energy computed in this way refers to the
# energy difference between closed-shell PCCP and two CP radicals that are
# described by orbitals from a spin-restricted SCF calculation, which have been
# given an unrestricted occupation. In other words, the set of alpha- and beta-
# spin orbitals are identical and the effect of spin polarization is missing. In
# practice, this leads to minor energy differences with respect to the correct
# bond energy, that is, the energy difference between closed-shell PCCP and two
# CP radicals treated in the unrestricted mode, i.e., for which the set of
# alpha- and beta-spin orbitals are allowed to relax toward different solutions
# in the SCF procedure. This correction term can be computed directly by
# carrying out an unrestricted computation of the CP radical ('cp_fpccp_asu') using␣
↪the
```

(continues on next page)

(continued from previous page)

```
# restricted CP radical ('cp_fpccp_asr') as a fragment.

# The pure orbital interaction effect of forming the electron bonding combination
# of the two radicals can be isolated from the full orbital interaction by carrying␣
↪out a
# separate calculation. In this calculation (pccp_fa1_pb) the bond energy analysis is␣
↪performed in
# the absence of any virtual CP fragment orbitals, using the key REMOVEFRAGORBITALS.


AMS_JOBNAME=CP $AMSBIN/ams <<eor
System
  atoms
     C          .0000    .0000    .6681
     P          .0000    .0000   2.2555
  end
end

Task SinglePoint

Engine ADF
  title cp_fpccp_asr
  eprint
    sfo eig ovl
  end
  basis
    core Large
    type TZ2P
  end
  numericalquality Good
  xc
    gga BECKE PERDEW
  end
EndEngine
eor

AMS_JOBNAME=CP_Unrestricted $AMSBIN/ams <<eor
System
  atoms
     C          .0000    .0000    .6681   adf.f=CP
     P          .0000    .0000   2.2555   adf.f=CP
  end
end

Task SinglePoint

Engine ADF
  title cp_fpccp_asu
  eprint
    sfo eig ovl
  end
  fragments
     CP   CP.results/adf.rkf
  end
  fragoccupations
     CP
     SIGMA 3//2
```

(continues on next page)

(continued from previous page)

```
     PI    2//2
     SUBEND
  end
  numericalquality Good
  spinpolarization 1
  unrestricted
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=PCCP_pb $AMSBIN/ams <<eor
System
  atoms
     P          .0000    .0000   2.2555  adf.f=CP_A
     C          .0000    .0000    .6681  adf.f=CP_A
     C          .0000    .0000   -.6681  adf.f=CP_B
     P          .0000    .0000  -2.2555  adf.f=CP_B
  end
end

Task SinglePoint

Engine ADF
  title pccp_fa1_pb
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     CP_A   CP.results/adf.rkf
     CP_B   CP.results/adf.rkf
  end
  fragoccupations
     CP_A
     SIGMA 3//2
     PI    2//2
     SUBEND
     CP_B
     SIGMA 2//3
     PI    2//2
     SUBEND
  end
  numericalquality Good
  removeallfragvirtuals
  symmetry C(LIN)
  xc
    gga BECKE PERDEW
  end
  print VDDanalysis
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=PCCP $AMSBIN/ams <<eor
System
  atoms
     P          .0000    .0000   2.2555  adf.f=CP_A
     C          .0000    .0000    .6681  adf.f=CP_A
     C          .0000    .0000   -.6681  adf.f=CP_B
     P          .0000    .0000  -2.2555  adf.f=CP_B
  end
end

Task SinglePoint

Engine ADF
  title pccp_fa1_as
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     CP_A   CP.results/adf.rkf
     CP_B   CP.results/adf.rkf
  end
  fragoccupations
     CP_A
     SIGMA 3//2
     PI    2//2
     SUBEND
     CP_B
     SIGMA 2//3
     PI    2//2
     SUBEND
  end
  numericalquality Good
  symmetry C(LIN)
  xc
    gga BECKE PERDEW
  end
  print VDDanalysis
EndEngine
eor

AMS_JOBNAME=CP_B3LYP $AMSBIN/ams <<eor
System
  atoms
     C          .0000    .0000    .6681
     P          .0000    .0000   2.2555
  end
end

Task SinglePoint

Engine ADF
  title B3LYP restricted
  basis
    core None
    type TZ2P
```

(continues on next page)

(continued from previous page)

```
  end
  numericalquality Good
  xc
    hybrid B3LYP
  end
EndEngine
eor

AMS_JOBNAME=CP_B3LYP_Unrestricted $AMSBIN/ams <<eor
System
  atoms
     C          .0000    .0000    .6681  adf.f=CP
     P          .0000    .0000   2.2555  adf.f=CP
  end
end

Task SinglePoint

Engine ADF
  title B3LYP unrestricted
  fragments
     CP   CP_B3LYP.results/adf.rkf
  end
  fragoccupations
     CP
     SIGMA 7//6
     PI    4//4
     Subend
  end
  numericalquality Good
  spinpolarization 1
  unrestricted
  xc
    hybrid B3LYP
  end
EndEngine
eor

AMS_JOBNAME=PCCP_B3LYP_pb  $AMSBIN/ams <<eor
System
  atoms
     P          .0000    .0000   2.2555  adf.f=CP_A
     C          .0000    .0000    .6681  adf.f=CP_A
     C          .0000    .0000   -.6681  adf.f=CP_B
     P          .0000    .0000  -2.2555  adf.f=CP_B
  end
end

Task SinglePoint

Engine ADF
  title PCCP B3LYP PAIRBONDING
  fragments
     CP_A   CP_B3LYP.results/adf.rkf
     CP_B   CP_B3LYP.results/adf.rkf
  end
  fragoccupations
```

(continues on next page)

(continued from previous page)

```
      CP_A
      SIGMA 7//6
      PI    4//4
      Subend
      CP_B
      SIGMA 6//7
      PI    4//4
      Subend
  end
  numericalquality Good
  removeallfragvirtuals
  xc
    hybrid B3LYP
  end
  print VDDanalysis
EndEngine
eor


AMS_JOBNAME=PCCP_B3LYP $AMSBIN/ams <<eor
System
  atoms
      P          .0000    .0000   2.2555  adf.f=CP_A
      C          .0000    .0000    .6681  adf.f=CP_A
      C          .0000    .0000   -.6681  adf.f=CP_B
      P          .0000    .0000  -2.2555  adf.f=CP_B
  end
end

Task SinglePoint

Engine ADF
  title PCCP B3LYP
  fragments
      CP_A   CP_B3LYP.results/adf.rkf
      CP_B   CP_B3LYP.results/adf.rkf
  end
  fragoccupations
      CP_A
      SIGMA 7//6
      PI    4//4
      Subend
      CP_B
      SIGMA 6//7
      PI    4//4
      Subend
  end
  numericalquality Good
  xc
    hybrid B3LYP
  end
  print VDDanalysis
EndEngine
eor
```

Example: Analysis of NaCl using ionic fragments: Na+ and Cl-

Download `NaCl_ionicbonding.run`

```
#! /bin/sh


# This example illustrates advanced usage of the bond energy decomposition
# scheme used in ADF.

# This example shows the energy decomposition analysis of NaCl with
# ionic fragments: Na+ and Cl-
# See also the example with neutral fragments: Na and Cl, both open shell atoms
#
# If the key REMOVEALLFRAGORBITALS is used one can approximately
# see what the electrostatic effect is on the fragment orbitals if one places
# the two ions together.
#
# Another way to see what the electrostatic effect is on the fragment orbitals is
# to use the key 'PRINT FMATSFO' in the full molecule calculation, where one can
# look at the expectation value of the Fock matrix of a fragment orbital in the
# presence of the other fragment.


AMS_JOBNAME=Cl_min $AMSBIN/ams <<eor
System
  atoms
     Cl  0.0 0.0 0.0
  end
  charge -1
end

Task SinglePoint

Engine ADF
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga BECKE PERDEW
  end
EndEngine
eor

AMS_JOBNAME=Na_plus $AMSBIN/ams <<eor
System
  atoms
     Na 0.0 0.0 0.0
  end
  charge 1
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga BECKE PERDEW
  end
EndEngine
eor

AMS_JOBNAME=NaCl_es $AMSBIN/ams <<eor
System
  atoms
     Na   0.0 0.0 0.0   adf.f=Na
     Cl   0.0 0.0 2.361 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     Na  Na_plus.results/adf.rkf
     Cl  Cl_min.results/adf.rkf
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  removeallfragvirtuals
  title ionic bonding
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=NaCl $AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     Na 0.0 0.0 0.0 adf.f=Na
     Cl  0.0 0.0 2.361 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     Na  Na_plus.results/adf.rkf
     Cl  Cl_min.results/adf.rkf
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  title ionic bonding
  xc
    gga BECKE PERDEW
  end
EndEngine
eor
```

Example: Electron Pair bonding in NaCl: open shell fragments

Download NaCl_pairbonding.run

```
#! /bin/sh


# This example illustrates advanced usage of the bond energy decomposition
# scheme used in ADF.

# This example shows the energy decomposition analysis of NaCl with
# neutral fragments: Na and Cl, both open shell atoms
# See also the example with ionic fragments: Na+ and Cl-.

# A proper decomposition of an electron-pair bond energy requires specifying
# opposite spins for the unpaired electrons of the respective radical fragments,
# which can be done with the input key FragOccupations. The specified alpha- and
# beta-spin configurations of the radical fragments are shown in the output
# section B U I L D.

# Please note that if one neglects explicitly specifying opposite spins for the
# unpaired electrons of the fragments, each of them is treated as being half an
# alpha and half a beta electron and consequently, they enter into a spurious
# Pauli repulsive interaction. This results, among others, into the Pauli
# repulsion term being too repulsive and the orbital interaction term being too
# much stabilizing.
```

(continues on next page)

(continued from previous page)

```
# The example consists of an analysis of the Na-Cl single bond between two
# open shell atoms. Two methods are described here.
# The first method uses spin-restricted fragments but simulates unrestricted fragments
# with the use of the FRAGOCCUPATIONS key.
# The second method uses real unrestricted atoms.

# Unrestricted fragments with fragoccuaptions
# -------------------------------------------

# The proper spins in this method are specified in the calculation of the overall
# molecule using the FragOccupations key. Note that this implies a slight
# approximation because the bond energy computed in this way refers to the
# energy difference between closed-shell NaCl and two open shell atoms that are
# described by orbitals from a spin-restricted SCF calculation, which have been
# given an unrestricted occupation. In other words, the set of alpha- and beta-
# spin orbitals are identical and the effect of spin polarization is missing. In
# practice, this leads to minor energy differences with respect to the correct
# bond energy, that is, the energy difference between closed-shell NaCl and the two
# open shell atoms treated in the unrestricted mode, i.e., for which the set of
# alpha- and beta-spin orbitals are allowed to relax toward different solutions
# in the SCF procedure. This correction term can be computed directly by
# carrying out an unrestricted computation of the Na and Cl atom using the spin
# restricted atoms as a fragment.

# Real unrestricted fragments
# ---------------------------

# The unrestricted calculated Na and Cl atom are used as fragments.

# Prepared for bonding
# --------------------

# Note that the Cl atom that is used as fragment is given an electron
# configuration that is so called 'prepared for bonding' in order to minimize
# the Pauli repulsion in the electron pair bond.
# In this case it means that there is a hole in the valence Cl p_z orbital,
# that will later form an electron pair bond with the valence s orbital of Na.

# Pair bonding
# ------------

# The pure orbital interaction effect of forming the electron bonding combination
# of the two radicals can be isolated from the full orbital interaction by carrying␣
↪out a
# separate calculation. In this calculation the bond energy analysis is performed in
# the absence of (almost) all virtual fragment orbitals, using the key␣
↪REMOVEALLFRAGVIRTUALS.
# Note that in the calculation there is a slight orbital relaxation term in addition␣
↪to the
# pure orbital interaction, because not only the bonding combination of the singly␣
↪occupied
# orbitals of the fragments are included, but also the anti-bonding combination.

AMS_JOBNAME=Na_r $AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     Na 0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=Cl_r $AMSBIN/ams <<eor
System
  atoms
     Cl  0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  irrepoccupations
     SIGMA.g 6
     SIGMA.u 2 1
     PI.u    8
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  symmetry D(LIN)
  xc
    gga BECKE PERDEW
  end
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=Na_u $AMSBIN/ams <<eor
System
  atoms
     Na 0.0 0.0 0.0 adf.f=Na
  end
end

Task SinglePoint

Engine ADF
  fragments
     Na  Na_r.results/adf.rkf
  end
  fragoccupations
     Na
     S 3 // 2
     P 3 // 3
     SubEnd
  end
  irrepoccupations
     S 3 // 2
     P 3 // 3
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 1
  title unrestricted Na
  unrestricted
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=Cl_u $AMSBIN/ams <<eor
System
  atoms
     Cl  0.0 0.0 0.0 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  fragments
     Cl   Cl_r.results/adf.rkf
  end
  fragoccupations
     Cl
     SIGMA.g 3 // 3
     SIGMA.u 1 // 2
     PI.u    4 // 4
     SubEnd
```

(continues on next page)

(continued from previous page)

```
  end
  irrepoccupations
     SIGMA.g 3 // 3
     SIGMA.u 1 // 2
     PI.u    4 // 4
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization -1
  symmetry D(LIN)
  title unrestricted Cl
  unrestricted
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=NaCl_pb $AMSBIN/ams <<eor
System
  atoms
     Na 0.0 0.0 0.0   adf.f=Na
     Cl 0.0 0.0 2.361 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     Na  Na_r.results/adf.rkf
     Cl  Cl_r.results/adf.rkf
  end
  fragoccupations
     Na
     S 3 // 2
     P 3 // 3
     SubEnd
     Cl
     SIGMA.g 3 // 3
     SIGMA.u 1 // 2
     PI.u    4 // 4
     SubEnd
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
```

(continues on next page)

(continued from previous page)

```
  end
  removeallfragvirtuals
  title fragoccupations pair bonding
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=NaCl $AMSBIN/ams <<eor
System
  atoms
     Na 0.0 0.0 0.0   adf.f=Na
     Cl 0.0 0.0 2.361 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     Na  Na_r.results/adf.rkf
     Cl  Cl_r.results/adf.rkf
  end
  fragoccupations
     Na
     S 3 // 2
     P 3 // 3
     SubEnd
     Cl
     SIGMA.g 3 // 3
     SIGMA.u 1 // 2
     PI.u    4 // 4
     SubEnd
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  title fragoccupations
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=NaCl_unrfrag_pb $AMSBIN/ams <<eor
System
  atoms
```

(continues on next page)

(continued from previous page)

```
     Na 0.0 0.0 0.0   adf.f=Na
     Cl 0.0 0.0 2.361 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     Na  Na_u.results/adf.rkf
     Cl  Cl_u.results/adf.rkf
  end
  numericalquality Good
  relativity
    level scalar
    formalism ZORA
  end
  removeallfragvirtuals
  spinpolarization 0
  title unrestricted fragments pair bonding
  unrestricted
  unrestrictedfragments
  xc
    gga BECKE PERDEW
  end
EndEngine
eor


AMS_JOBNAME=NaCl_unrfrag $AMSBIN/ams <<eor
System
  atoms
     Na 0.0 0.0 0.0   adf.f=Na
     Cl 0.0 0.0 2.361 adf.f=Cl
  end
end

Task SinglePoint

Engine ADF
  eprint
    orbpop 20  20
    end
    sfo eig ovl
  end
  fragments
     Na  Na_u.results/adf.rkf
     Cl  Cl_u.results/adf.rkf
  end
  numericalquality Good
  relativity
    level scalar
```

(continues on next page)

(continued from previous page)

```
    formalism ZORA
  end
  spinpolarization 0
  title unrestricted fragments
  unrestricted
  unrestrictedfragments
  xc
    gga BECKE PERDEW
  end
EndEngine
eor
```

Example: Bond Energy analysis meta-GGA, (meta-)hybrids: Zn2, Cr2, CrH

Download EDA_meta_gga_hybrid.run

```
#! /bin/sh


# This example illustrates the bond energy decomposition scheme using metaGGA or
# metahybrid or hybrid functionals in ADF.

# The first example is straightforward with closed shell atomic fragments: Zn_2.
# The second example has open shell atomic fragments: Cr_2, and the extra
# complication that spin symmetry breaking lowers the energy of the molecule,
# although the total S_z-value is zero. The third example has open shell atomic
# fragments, and the molecule is open shell: CrH.

# == Zn2 ==

# In the first example for Zn_2 the metahybrid TPSSh is used. In the bond energy
# analysis, the bond energy is split in a Pauli repulsion term, a steric
# interaction, and an orbital interaction.

AMS_JOBNAME=Zn2 $AMSBIN/ams <<eor
System
  atoms
     Zn 0.0 0.0 -1.6
     Zn 0.0 0.0  1.6
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type TZ2P
  end
  xc
    metahybrid TPSSh
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine

eor

# == Cr2 ==

# In the second example for Cr2 the metaGGA TPSS is used. You can use
# spin-unrestricted fragments in ADF. However, here simulated unrestricted
# fragments are used. A fair approximation to a computation with unrestricted
# fragments can be achieved with the key FRAGOCCUPATIONS. You tell ADF that you
# want to treat the fragments as if they were unrestricted; this causes the
# program to duplicate the one-electron orbitals of the fragment: one set for
# spin-alpha and one set for spin-beta. You can then specify occupation numbers
# for these spin-unrestricted fragments, and occupy spin-alpha orbitals
# differently from spin-beta orbitals. Especially for the Pauli-repulsion it is
# important that one chooses the spin-occupations on the different fragments
# such that they are 'prepared for bonding'.

# Of course, the unrestricted fragments that you use in this way, are not self-
# consistent: different numbers of spin-alpha and spin-beta electrons usually
# result in different spatial orbitals and different energy eigenvalues for
# spin-alpha and spin-beta when you go to self-consistency, while here you have
# spatially identical fragment orbitals. Nevertheless it is often a fair
# approximation which gives you a considerable extension of analysis
# possibilities.

# Spin-symmetry breaking is enforced by the use of the key ModifyStartPotential
# in combination with the key key UNRESTRICTED. In the ADF output one can find
# that there is spin-density on both of the atoms.


AMS_JOBNAME=Cr2 $AMSBIN/ams <<eor
System
  atoms
     Cr.1 0.0 0.0 -0.9
     Cr.2 0.0 0.0  0.9
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  fragoccupations
     Cr.1
     S 4 // 3
     P 6 // 6
     D 5 // 0
     SubEnd
     Cr.2
     S 3 // 4
     P 6 // 6
     D 0 // 5
     SubEnd
  end
```

(continues on next page)

(continued from previous page)

```
  modifystartpotential
     Cr.1 1 // 0
     Cr.2 0 // 1
  end
  basis
    core None
    type TZ2P
  end
  spinpolarization 0
  unrestricted
  xc
    metagga TPSS
  end
EndEngine

eor


# In order to calculate the effect of self-consistency one should calculate the
# Cr atom spin-unrestrictedly.


AMS_JOBNAME=Cr $AMSBIN/ams <<eor
System
  atoms
     Cr 0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  fragoccupations
     Cr
     S 4 // 3
     P 6 // 6
     D 5 // 0
     SubEnd
  end
  basis
    core None
    type TZ2P
  end
  spinpolarization 6
  unrestricted
  xc
    metagga TPSS
  end
EndEngine

eor


# == CrH ==
```

(continues on next page)

(continued from previous page)

```
# In this example for CrH the hybrid B3LYP is used.

AMS_JOBNAME=CrH $AMSBIN/ams <<eor
System
  atoms
     Cr 0.0 0.0 0.0
     H  0.0 0.0 1.65
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  fragoccupations
     Cr
     S 4 // 3
     P 6 // 6
     D 5 // 0
     SubEnd
     H
     S 0 // 1
     SubEnd
  end
  basis
    core None
    type TZ2P
  end
  spinpolarization 5
  unrestricted
  xc
    hybrid B3LYP
  end
EndEngine

eor


# In order to calculate the effect of self-consistency of spin-polarization on
# the atoms one should calculate the Cr and H atom spin-unrestrictedly.


AMS_JOBNAME=Cr_B3LYP $AMSBIN/ams <<eor
System
  atoms
     Cr 0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
```

(continues on next page)

(continued from previous page)

```
  end
  fragoccupations
     Cr
     S 4 // 3
     P 6 // 6
     D 5 // 0
     SubEnd
  end
  basis
    core None
    type TZ2P
  end
  spinpolarization 6
  unrestricted
  xc
    hybrid B3LYP
  end
EndEngine

eor


AMS_JOBNAME=H $AMSBIN/ams <<eor
System
  atoms
     H 0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  fragoccupations
     H
     S 1 // 0
     SubEnd
  end
  basis
    core None
    type TZ2P
  end
  spinpolarization 1
  unrestricted
  xc
    hybrid B3LYP
  end
EndEngine

eor
```

Example: unrestricted EDA: Cu(C2H4)2

Download EDA_Unr_C2H4_Cu_C2H4.run

```
#!/bin/sh


# An illustration of an unrestricted energy decomposition analysis (unrestricted EDA).

# The open shell Cu atom forms a complex with 2 ethylene molecules.

AMS_JOBNAME=Cu_A $AMSBIN/ams <<eor
System
  atoms
     Cu  0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  basis
    type DZ
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 1
  unrestricted
  xc
    dispersion GRIMME4
    gga BP86
  end
EndEngine

eor


AMS_JOBNAME=Ethylene $AMSBIN/ams <<eor
System
  atoms
     C       0.000000000000       0.708069500374       1.888278754032
     C       0.000000000000      -0.708069500374       1.888278754032
     H       0.924242618139       1.263978729882       2.061097077561
     H       0.924242618139      -1.263978729882       2.061097077561
     H      -0.924242618139       1.263978729882       2.061097077561
     H      -0.924242618139      -1.263978729882       2.061097077561
  end
end

Task SinglePoint

Engine ADF
  basis
    type DZ
  end
```

(continues on next page)

(continued from previous page)

```
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  xc
    dispersion GRIMME4
    gga BP86
  end
EndEngine

eor


$AMSBIN/ams <<eor
System
  atoms
      C       0.000000000000       0.708069500374       1.888278754032    adf.
↪f=Ethylene|1
      C       0.000000000000      -0.708069500374       1.888278754032    adf.
↪f=Ethylene|1
      H       0.924242618139       1.263978729882       2.061097077561    adf.
↪f=Ethylene|1
      H       0.924242618139      -1.263978729882       2.061097077561    adf.
↪f=Ethylene|1
      H      -0.924242618139       1.263978729882       2.061097077561    adf.
↪f=Ethylene|1
      H      -0.924242618139      -1.263978729882       2.061097077561    adf.
↪f=Ethylene|1
      Cu      0.000000000000       0.000000000000       0.000000000000    adf.f=Cu_A
      C       0.000000000000       0.708069500374      -1.888278754032    adf.
↪f=Ethylene|2
      C       0.000000000000      -0.708069500374      -1.888278754032    adf.
↪f=Ethylene|2
      H       0.924242618139       1.263978729882      -2.061097077561    adf.
↪f=Ethylene|2
      H       0.924242618139      -1.263978729882      -2.061097077561    adf.
↪f=Ethylene|2
      H      -0.924242618139       1.263978729882      -2.061097077561    adf.
↪f=Ethylene|2
      H      -0.924242618139      -1.263978729882      -2.061097077561    adf.
↪f=Ethylene|2
  end
end

Task SinglePoint

Engine ADF
  fragments
     Ethylene Ethylene.results/adf.rkf
     Cu_A Cu_A.results/adf.rkf
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
```

(continues on next page)

(continued from previous page)

```
  spinpolarization 1
  unrestricted
  unrestrictedfragments
  xc
    dispersion GRIMME4
    gga BP86
  end
EndEngine

eor
```

Example: unrestricted fragments: CH3I

Download EDA_Unr_CH3I.run

```
#!/bin/sh


# An illustration of an unrestricted energy decomposition analysis (unrestricted EDA).

# Prepared for bonding
# --------------------

# The electron configuration of the fragments is chosen such that the valence I p_z␣
↪orbital
# has 1 alpha electron, and the highest occupied Methyl orbital has 1 beta electron.
# Note that this electron configuration of the fragments means that they
# are so called 'prepared for bonding' in order to minimize the Pauli repulsion in
# the electron pair bond.

AMS_JOBNAME=I_A $AMSBIN/ams <<eor
System
  atoms
     I  0.0 0.0 0.0
  end
end

Task SinglePoint

Engine ADF
  irrepoccupations
     Sigma.g 2 // 2
     Sigma.u 1 // 0
     Pi.g    2 // 2
     Pi.u    2 // 2
     Delta.g 2 // 2
  end
  basis
    core Small
    type TZ2P
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
```

(continues on next page)

(continued from previous page)

```
  end
  spinpolarization 1
  symmetry D(lin)
  unrestricted
  xc
    gga BP86
  end
EndEngine

eor


AMS_JOBNAME=Methyl_B $AMSBIN/ams <<eor
System
  atoms
      C       0.000000000000       0.000000000000      -0.239316000000
      H      -0.521322100000      -0.902956360000      -0.562716000000
      H      -0.521322100000       0.902956360000      -0.562716000000
      H       1.042644200000      -0.000000000000      -0.562716000000
  end
end

Task SinglePoint

Engine ADF
  basis
    core Small
    type TZ2P
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization -1
  unrestricted
  xc
    gga BP86
  end
EndEngine

eor

# The fragments are prepared. Next the unrestricted EDA is performed,
# in which symmetry can be used.

AMS_JOBNAME=ch3i $AMSBIN/ams <<eor
System
  atoms
      C       0.000000000000       0.000000000000      -0.239316000000  adf.f=Methyl_B
      H      -0.521322100000      -0.902956360000      -0.562716000000  adf.f=Methyl_B
      H      -0.521322100000       0.902956360000      -0.562716000000  adf.f=Methyl_B
      H       1.042644200000      -0.000000000000      -0.562716000000  adf.f=Methyl_B
      I       0.000000000000       0.000000000000       1.927464000000  adf.f=I_A
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  fragments
     I_A I_A.results/adf.rkf
     Methyl_B Methyl_B.results/adf.rkf
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 0
  unrestricted
  unrestrictedfragments
  xc
    gga BP86
  end
EndEngine

eor

# Same calculation but now using ETS-NOCV.
# ETS-NOCV: energy analysis using the Natural Orbitals for Chemical Valence.
# In ETS-NOCV symmetry NOSYM is required.

AMS_JOBNAME=etsnocv $AMSBIN/ams <<eor
System
  atoms
     C       0.000000000000       0.000000000000      -0.239316000000  adf.f=Methyl_B
     H      -0.521322100000      -0.902956360000      -0.562716000000  adf.f=Methyl_B
     H      -0.521322100000       0.902956360000      -0.562716000000  adf.f=Methyl_B
     H       1.042644200000      -0.000000000000      -0.562716000000  adf.f=Methyl_B
     I       0.000000000000       0.000000000000       1.927464000000  adf.f=I_A
  end
end

Task SinglePoint

Engine ADF
  etsnocv
  end
  fragments
     I_A I_A.results/adf.rkf
     Methyl_B Methyl_B.results/adf.rkf
  end
  numericalquality good
  relativity
    level scalar
    formalism ZORA
  end
  spinpolarization 0
  symmetry nosym
  unrestricted
  unrestrictedfragments
  xc
    gga BP86
```

(continues on next page)

(continued from previous page)

```
  end
EndEngine

eor
```

Example: Spin-Orbit SFO analysis: TlH

Download TlH_SO_analysis.run

```
#! /bin/sh


# Application of the Spin-Orbit relativistic option (using double-group
# symmetry) to TlH with a detailed analysis of the spinors in terms of SFOs
# (Symmetrized Fragment Orbitals).

# In order to get the population analysis, one should have one scalar
# relativistic fragment, which is the whole molecule. The SFOs in this case are
# the scalar relativistic orbitals, which are already orthonormal, because one
# has only one fragment which is the whole molecule.

AMS_JOBNAME=Scalar $AMSBIN/ams <<eor
System
  atoms
     Tl       0.0             0.0             0.0
     H        0.0             0.0             1.870
  end
end

Task SinglePoint

Engine ADF
  title TlH, scalar relativistic zora
  beckegrid
    quality good
  end
  eprint
    sfo eig ovl
  end
  basis
    core None
    type TZ2P
    CreateOutput Yes
  end
  print SFO
  relativity
    level scalar
    formalism ZORA
  end
  xc
    gga BP86
  end
EndEngine

eor
```

(continues on next page)

(continued from previous page)

```
# In order to get the population analysis, one should have one scalar
# relativistic fragment, which is the whole molecule, which is TlH in this case.

AMS_JOBNAME=SpinOrbit $AMSBIN/ams <<eor
System
  atoms
     Tl       0.0             0.0             0.0    adf.f=TlH
     H        0.0             0.0             1.870  adf.f=TlH
  end
end

Task SinglePoint

Engine ADF
  title TlH from fragment TlH,  with SpinOrbit coupling
  beckegrid
    quality good
  end
  eprint
    sfo eig ovl
  end
  fragments
     TlH Scalar.results/adf.rkf
  end
  print SFO
  relativity
    level spin-orbit
    formalism ZORA
  end
  xc
    gga BP86
  end
EndEngine

eor

# The output gives something like:
# ================================================================================
#
#  =======================
#  Double group symmetry :  *   J1/2  *
#  =======================
#                                         ===  J1/2:1 ===
#  Spinors expanded in SFOs
# Spinor:             21                22                23                24
#  occup:            1.00              1.00              1.00              0.00
#  ------            ----              ----              ----              ----
#  SFO SIGMA
#    13.alpha:   0.7614+0.0000i    0.0096+0.0000i    0.0052+0.0000i   -0.0006+0.0000i
#    14.alpha:   0.0154+0.0000i   -0.9996+0.0000i    0.0208+0.0000i   -0.0077+0.0000i
#    15.alpha:  -0.0146+0.0000i    0.0185+0.0000i    0.9849+0.0000i    0.1625+0.0000i
#  SFO PI:x
#     8.beta :   0.4578+0.0000i    0.0091+0.0000i    0.0112+0.0000i    0.0030+0.0000i
#     9.beta :   0.0005+0.0000i   -0.0074+0.0000i   -0.1119+0.0000i    0.6910+0.0000i
#  SFO PI:y
#     8.beta :   0.0000+0.4578i    0.0000+0.0091i    0.0000+0.0112i    0.0000+0.0030i
```

(continues on next page)

(continued from previous page)

```
#     9.beta :   0.0000+0.0005i   0.0000-0.0074i   0.0000-0.1119i   0.0000+0.6910i
#
# ==================================================================================

# Left out are a lot of small numbers. The meaning is that a spinor of J_z=1/2
# symmetry can have SIGMA and PI character, for example, the 21st spinor with
# occupation number 1.0, is approximately (21 J_z=1/2) = 0.76 (13 SIGMA alpha) +
# 0.46 (8 PI:x beta) + i 0.46 (8 PI:y beta)

# Next in the SFO contributions per spinor the real and imaginary spin alpha
# part and real and imaginary spin beta part are all summed together to give a
# percentage of a certain SFO. are summed. For example the 21st spinor has
# almost 60% (13 SIGMA) character.

# ======================================
#
# SFO contributions (%) per spinor
# Spinor:      21      22      23      24

# occup:      1.00    1.00    1.00    0.00
# ------      ----    ----    ----    ----

# SFO SIGMA
#     13:    57.97    0.01    0.00    0.00
#     14:     0.02   99.92    0.04    0.01
#     15:     0.02    0.03   97.01    2.64
# SFO PI:x
#      8:    20.96    0.01    0.01    0.00
#      9:     0.00    0.01    1.25   47.75
# SFO PI:y
#      8:    20.96    0.01    0.01    0.00
#      9:     0.00    0.01    1.25   47.75
#
# ======================================
```

Example: Activation Strain Model Analysis using PyFrag

Download `pyfrag.run`

```
#!/bin/bash


# The trajectory is a concatenated xyz-file:

cat << eor > coords.xyz
C         -1.741968   -2.220880    0.000000
Pd        -2.137509   -0.237843    0.000000
H         -2.809570   -2.499547    0.000000
H         -1.265288   -2.629932    0.895677
H         -1.265288   -2.629932   -0.895677
H         -0.755099   -0.885698    0.00000

C         -1.741968   -2.220880    0.000000
Pd        -2.137509   -0.250000    0.000000
H         -2.809570   -2.499547    0.000000
```

(continues on next page)

(continued from previous page)

```
H        -1.265288   -2.629932    0.895677
H        -1.265288   -2.629932   -0.895677
H        -0.755099   -0.885698    0.00000
eor

# Input options for ADF:

# One should generally symmetrize the coordinates by setting AMS.System.
↪Symmetrize=Yes

cat << eor > adfinputfile
AMS.Task=SinglePoint
AMS.System.Charge=0
AMS.System.Symmetrize=Yes
ADF.NumericalQuality=VeryGood
ADF.XC.GGA=OPBE
ADF.basis.core=None
ADF.Symmetry=AUTO
ADF.Relativity.Level=Scalar
ADF.Relativity.Formalism=ZORA
ADF.eprint.sfo=eig ovl
eor

# Here we call the PyFrag script.

# PyFrag conducts a total of six single point calculations, one for each fragment and␣
↪the
# whole system for each of the two geometries in the trajectory, respectively.
# The intermediate results for each geometry are printed during the calculation.
# At the end of its run PyFrag summarizes all results in a table which is also␣
↪written
# to an output file PyFrag<SystemName>.txt

$AMSBIN/amspython $AMSHOME/scripting/standalone/pyfrag/PyFrag.py  \
--xyzpath coords.xyz \
--fragment 2 --fragment 1 3 4 5 6 --strain 0 --strain -554.1 \
--bondlength 1 6 1.093 \
--angle 2 1 6 180 \
--VDD 2 1 6 \
--hirshfeld frag1 \
--irrepOI AA \
--irrepOI AAA \
--orbitalenergy frag1 HOMO \
--orbitalenergy frag2 LUMO \
--orbitalenergy AA frag2 5 \
--population frag1 HOMO \
--population AA frag2 5 \
--overlap frag1 HOMO frag2 LUMO \
--overlap S frag1 5 AA frag2 4 \
--adfinput ADF.basis.type=DZP \
--adfinputfile adfinputfile
```

14.6.2 Localized orbitals, bond orders, charge analysis

Example: (Perturbed) localized molecular orbitals in twisted Ethene

Download LocMO_Resp.run

```
#!/bin/sh


# This is example for the calculation of localized molecular orbitals, and
# perturbed localized molecular orbitals. This example uses the Pipek-Mezey
# localization method (criterion PM in the LOCORB block key). With the
# AORESPONSE block key and the subkey opticalrotation the optical rotation
# dispersion is calculated. The PERTLOC key is specified in order to calculate
# perturbed localized molecular orbitals. The subkey gprime and beta of the key
# PERTLOC are needed to analyze the optical rotation. Note that PERTLOC only
# makes sense in combination with AORESPONSE and LOCORB.

$AMSBIN/ams <<eor
System
  atoms
     C 0.000000 0.000000 0.020000
     C 1.350716 0.000000 -0.000000
     H -0.540179 -0.745428 -0.565703
     H -0.540179 0.745428 0.565703
     H 1.890895 -0.935093 -0.035837
     H 1.890895 0.935093 0.035837
  end
end

Task SinglePoint

Engine ADF
  title twisted ethene PM localization and response prop analysis
  aoresponse
    frequencies .05 [Hartree]
    giao
    opticalrotation
  end
  locorb
     criterion PM
  end
  basis
    core None
    type DZP
  end
  pertloc
    alfa
    beta
    dynamic
    efield
    fulltens
    gprime
  end
  symmetry nosym
  Relativity
    Level None
```

(continues on next page)

(continued from previous page)

```
  End
EndEngine
eor
```

Example: Charge model 5 (CM5)

Download CM5_chargemodel.run

```
#!/bin/sh


# With the key CM5 charge model 5 (CM5) atomic charges are calculated. The CM5
# model is especially designed such that one can reasonably accurately predict
# dipole moments with only atomic CM5 charges.


AMS_JOBNAME=Acetone $AMSBIN/ams <<eor
System
  atoms
     O       0.000000    0.000000    1.396962
     C       0.000000    0.000000    0.188800
     H       0.871902    1.314824   -1.265092
     C       0.000000   -1.275657   -0.611044
     H       0.000000    2.142271    0.042196
     H       0.000000   -2.142271    0.042196
     H       0.871902   -1.314824   -1.265092
     C       0.000000    1.275657   -0.611044
     H      -0.871902    1.314824   -1.265092
     H      -0.871902   -1.314824   -1.265092
  end
end

Task SinglePoint

Engine ADF
  title Acetone
  AtomicChargesTypeForAMS CM5
  basis
    type TZP
    core none
  end
  numericalquality good
  xc
    metagga M06-L
  end
EndEngine

eor


AMS_JOBNAME=Nitrobenzene $AMSBIN/ams <<eor
System
  atoms
     C       0.000000    0.000000   -2.503225
     C       0.000000    1.205864   -1.812565
```

(continues on next page)

(continued from previous page)

```
     C       0.000000    1.214022   -0.427327
     C       0.000000    0.000000    0.239613
     C       0.000000   -1.214022   -0.427327
     C       0.000000   -1.205864   -1.812565
     H       0.000000   -2.131001    0.143627
     H       0.000000   -2.142362   -2.354096
     H       0.000000    2.142362   -2.354096
     H       0.000000    2.131001    0.143627
     H       0.000000    0.000000   -3.585546
     N       0.000000    0.000000    1.715151
     O       0.000000   -1.084613    2.278800
     O       0.000000    1.084613    2.278800
  end
end

Task SinglePoint

Engine ADF
  title Nitrobenzene
  AtomicChargesTypeForAMS cm5
  basis
    type TZP
    core none
  end
  numericalquality good
  xc
    metagga M06-L
  end
EndEngine

eor
```

Example: Bond Orders

Download BondOrder.run

```
#!/bin/sh


# With the key BONDORDER a bond order analysis is performed based on SFOs. Note
# that symmetry used in the calculation should be NOSYM.

AMS_JOBNAME=benzene $AMSBIN/ams <<eor
  Task SinglePoint

  System
    atoms
      C  0.0                  0.0                 0.0
      C  1.384765863418579    0.0                 0.0
      C  2.077148675918579    1.199242353439331   0.0
      C  1.384765863418579    2.398484706878662   0.0
      C  0.0                  2.398484706878662   0.0
      C -0.6923829317092896   1.199242353439331   0.0
      H  1.920830130577087   -0.9284905791282654  0.0
      H  3.149277210235596    1.199242353439331   0.0
```

(continues on next page)

(continued from previous page)

```
      H  1.920830130577087   3.326975345611572  0.0
      H -0.5360642671585083  3.326975345611572  0.0
      H -1.764511466026306   1.199242353439331  0.0
      H -0.5360642671585083 -0.9284905791282654 0.0
    end
  end

  Properties
    BondOrders Yes
  End

  Engine ADF
    title benzene BP/SZ bondorders PrintTolerance=0.05
    basis
     Type SZ
     Core None
    end
    symmetry NOSYM
    xc
      gga becke perdew
    end
    BondOrders
      PrintAll Yes
      PrintTolerance 0.05
    End
    noprint sfo
  EndEngine
eor


AMS_JOBNAME=ethelene $AMSBIN/ams <<eor
  Task SinglePoint

  System
    atoms
     C               0.0000    0.0000    0.0000
     C               1.3319    0.0000    0.0000
     H              -0.5723    0.9299    0.0000
     H              -0.5725   -0.9297    0.0000
     H               1.9043    0.9298    0.0000
     H               1.9042   -0.9298    0.0000
    end
  end

  Properties
    BondOrders Yes
  End

  Engine ADF
    title ethylene BP/SZ bondorders-standard output
    basis
     Type SZ
     Core None
    end
    symmetry NOSYM
    xc
     gga becke perdew
```

(continues on next page)

(continued from previous page)

```
    end
    noprint sfo
  EndEngine
eor

AMS_JOBNAME=Cr2 $AMSBIN/ams <<eor
  Task GeometryOptimization

  System
    atoms
     Cr 0.0 0.0 -0.8
     Cr 0.0 0.0  0.8
    end
  end

  GeometryOptimization
    Convergence
      Gradients 0.0001
    End
  End

  Properties
    BondOrders Yes
  End

  Engine ADF
    title Cr2 BP/TZP - bondorder - standard printout
    basis
     Type TZP
     Core None
    end
    symmetry NOSYM
    xc
     gga becke perdew
    end
    noprint sfo
  EndEngine
eor


AMS_JOBNAME=Mo2 $AMSBIN/ams <<eor
  Task GeometryOptimization

  System
    atoms
     Mo 0.0 0.0 -1.0
     Mo 0.0 0.0  1.0
    end
  end

  Properties
    BondOrders Yes
  End

  Engine ADF
    title Mo2 BP/TZP - bondorder - standard printout
    basis
```

(continues on next page)

(continued from previous page)

```
     Type TZP
     Core Small
    end
    symmetry NOSYM
    relativity
      level scalar
      formalism ZORA
    end
    xc
     gga becke perdew
    end
    noprint sfo
  EndEngine
eor


AMS_JOBNAME=NaCl $AMSBIN/ams <<eor
  Task SinglePoint

  System
    atoms
     Na 0.0 0.0 0.0
     Cl 0.0 0.0 2.377
    end
  end

  Properties
    BondOrders Yes
  End

  Engine ADF
    title NaCl BP/TZP - bondorder - standard printout
    basis
     Type DZP
     Core None
    end
    symmetry NOSYM
    xc
     gga becke perdew
    end
    BondOrders
      PrintAll Yes
    End
    noprint sfo
  EndEngine
eor
```

14.6.3 ETS-NOCV

Example: NOCV: ethylene – Ni-diimina

Download Diimina_NOCV.run

```
#!/bin/sh


# Example for calculation of ETS-NOCV for spin-restricted fragments. ETS-NOCV:
# energy analysis using the Natural Orbitals for Chemical Valence. The ethylene
# molecule and a Ni-diimina form a complex together. This example will be
# discussed first. The other example is H+ and CO form together HCO+ , this
# example is similar to the discussed example. All electron basis sets are
# required.

# First the two fragments are calculated.


AMS_JOBNAME=etfrag $AMSBIN/ams <<eor
System
  atoms
     C   -0.430177075 -1.815433265  0.860288229
     C   -0.363705637 -1.910722338 -0.515633302
     H    0.533109934 -2.284970854 -1.016904201
     H   -1.279922499 -1.884673940 -1.115144723
     H   -1.389295819 -1.753589602  1.377541080
     H    0.440296224 -2.041861443  1.484489314
  end
end

Task SinglePoint

Engine ADF
  title et-----Ni-diimina: ethylene run
  basis
    core Small
    type DZP
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor


AMS_JOBNAME=Nifrag $AMSBIN/ams <<eor
System
  atoms
     Ni   0.022615419  0.037783871  0.025751533
     N    0.386170317  1.871072585  0.306265538
     C    1.612863056  2.248007643  0.148716016
```

(continues on next page)

(continued from previous page)

```
      C    2.540686607  1.163409862 -0.183603690
      N    1.976290003  0.008161589 -0.301176178
      H   -0.288333328  2.609667211  0.546869047
      H    1.942601454  3.283060847  0.269249237
      H    3.613259273  1.338293482 -0.302134814
      H    2.621707427 -0.766258151 -0.517479818
      H   -1.351756655  0.253389698  0.386197419
  end
  charge 1
end

Task SinglePoint

Engine ADF
  title et-----Ni-diimina: Ni-diimina run
  basis
    core Small
    type DZP
  end
  symmetry NOSYM
  xc
    gga scf becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

# Next these fragments are used in the calculation of the full complex. The block
# ETSNOCV is used to analyze the bonding in the molecule with respect to the␣
↪fragments.
# The symmetry must be NOSYM.

$AMSBIN/ams <<eor
System
  atoms
      Ni   0.022615419  0.037783871  0.025751533 adf.f=k
      N    0.386170317  1.871072585  0.306265538 adf.f=k
      C    1.612863056  2.248007643  0.148716016 adf.f=k
      C    2.540686607  1.163409862 -0.183603690 adf.f=k
      N    1.976290003  0.008161589 -0.301176178 adf.f=k
      H   -0.288333328  2.609667211  0.546869047 adf.f=k
      H    1.942601454  3.283060847  0.269249237 adf.f=k
      H    3.613259273  1.338293482 -0.302134814 adf.f=k
      H    2.621707427 -0.766258151 -0.517479818 adf.f=k
      H   -1.351756655  0.253389698  0.386197419 adf.f=k
      C   -0.430177075 -1.815433265  0.860288229 adf.f=m
      C   -0.363705637 -1.910722338 -0.515633302 adf.f=m
      H    0.533109934 -2.284970854 -1.016904201 adf.f=m
      H   -1.279922499 -1.884673940 -1.115144723 adf.f=m
      H   -1.389295819 -1.753589602  1.377541080 adf.f=m
      H    0.440296224 -2.041861443  1.484489314 adf.f=m
  end
  charge 1
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title et-----Ni-diimina run

  ETSNOCV
  End

  fragments
     m etfrag.results/adf.rkf
     k Nifrag.results/adf.rkf
  end
  symmetry NOSYM
  xc
    gga scf becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor
```

Example: NOCV: H+ – CO

Download Hplus_CO_etsnocv.run

```
#! /bin/sh


AMS_JOBNAME=Hplus $AMSBIN/ams <<eor
System
  atoms
     H         -0.010992     0.000002     0.148581
  end
  charge 1
end

Task SinglePoint

Engine ADF
  title [H]+-----[CO] run from H+ and CO fragments, FULL electron calc.!
  Basis Type=SZ Core=None
  numericalquality Good
  scf
    converge 1E-6
    iterations 2500
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=CO $AMSBIN/ams <<eor
System
  atoms
     C         0.005191    0.000001    1.248610
     O         0.021536    0.000000    2.359675
  end
end

Task SinglePoint

Engine ADF
  title [CO] run
  Basis Type=SZ Core=None
  numericalquality Good
  scf
    converge 1E-6
    iterations 2500
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
EndEngine
eor


AMS_JOBNAME=HCO $AMSBIN/ams <<eor
System
  atoms
     H        -0.010992    0.000002    0.148581 adf.f=f1
     C         0.005191    0.000001    1.248610 adf.f=f2
     O         0.021536    0.000000    2.359675 adf.f=f2
  end
  charge 1
end

Task SinglePoint

Engine ADF
  title [H]+--[CO], etsnocv activated by etsnocv and print etslowdin
  etsnocv
    ekmin 1.5
    enocv 0.05
    rhokmin 1.e-3
  end
  fragments
     f1 Hplus.results/adf.rkf
     f2 CO.results/adf.rkf
  end
  numericalquality Good
  print etslowdin
  scf
    converge 1E-6
    iterations 800
  end
  symmetry NOSYM
```

(continues on next page)

(continued from previous page)

```
  xc
    gga becke perdew
  end
EndEngine
eor

$AMSBIN/densf <<eor
adffile HCO.results/adf.rkf
Grid Medium
End
NOCV
  1
  11
  2
  3
  9
  10
END
eor
mv TAPE41 nocv1.t41

$AMSBIN/densf <<eor
adffile HCO.results/adf.rkf
Grid Medium
End
NOCV
  THRESH 0.01
END
eor
mv TAPE41 nocv2.t41

$AMSBIN/densf <<eor
adffile HCO.results/adf.rkf
Grid Medium
End
NOCV
  ALL
END
eor
mv TAPE41 nocv3.t41
```

Example: NOCV: CH2 – Cr(CO)5

Download NOCV_CrCO5-CH2.run

```
#! /bin/sh


# Example for calculation of ETS-NOCV for spin-restricted fragments. ETS-NOCV:
# energy analysis using the Natural Orbitals for Chemical Valence. The CH2
# molecule and Cr(CO)5 are the fragments, which form Cr(CO)5 CH2 molecule.

# First the two fragments are calculated.
```

(continues on next page)

(continued from previous page)

```
AMS_JOBNAME=CH2 $AMSBIN/ams <<eor
System
  atoms
     C        -0.429104    1.732058   -0.225052
     H         0.407023    2.440417   -0.352323
     H        -1.385325    2.281354   -0.254124
  end
end

Task SinglePoint

Engine ADF
  title CrCO5--[CH2] run from CrCO5 and CH2 closed shell fragments,FULL electron calc.
↪!
  basis
    core None
    type DZP
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
EndEngine
eor



AMS_JOBNAME=CrCO5 $AMSBIN/ams <<eor
System
  atoms
     Cr       -0.248053   -0.169062    0.005810
     C        -0.072963   -2.080685    0.229583
     O         0.030811   -3.223220    0.361925
     C        -0.182894    0.049840    1.909128
     O        -0.142780    0.212309    3.050403
     C        -0.299940   -0.409118   -1.894730
     O        -0.331795   -0.521589   -3.042336
     C        -2.138631   -0.242152    0.075713
     O        -3.295036   -0.249916    0.115045
     C         1.624487    0.092244   -0.083118
     O         2.763411    0.288575   -0.140976
  end
end

Task SinglePoint

Engine ADF
  title [CrCO5] run
  basis
    Type DZP
    Core None
    PerAtomType Symbol=Cr Type=TZP
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
```

(continues on next page)

(continued from previous page)

```
EndEngine
eor


# Next these fragments are used in the calculation of the full complex. The keys
# ETSNOCV and 'PRINT etslowdin' are needed in this case to to analyze the
# bonding in the molecule with respect to the fragments. The symmetry must be
# NOSYM.


AMS_JOBNAME=CrCO5_CH2 $AMSBIN/ams <<eor
System
  atoms
     C        -0.429104    1.732058   -0.225052 adf.f=f1
     Cr       -0.248053   -0.169062    0.005810 adf.f=f2
     C        -0.072963   -2.080685    0.229583 adf.f=f2
     O         0.030811   -3.223220    0.361925 adf.f=f2
     C        -0.182894    0.049840    1.909128 adf.f=f2
     O        -0.142780    0.212309    3.050403 adf.f=f2
     C        -0.299940   -0.409118   -1.894730 adf.f=f2
     O        -0.331795   -0.521589   -3.042336 adf.f=f2
     C        -2.138631   -0.242152    0.075713 adf.f=f2
     O        -3.295036   -0.249916    0.115045 adf.f=f2
     C         1.624487    0.092244   -0.083118 adf.f=f2
     O         2.763411    0.288575   -0.140976 adf.f=f2
     H         0.407023    2.440417   -0.352323 adf.f=f1
     H        -1.385325    2.281354   -0.254124 adf.f=f1
  end
end

Task SinglePoint

Engine ADF
  title CrCO5--[CH2], etsnocv activated by etsnocv and print etslowdin
  etsnocv
    ekmin 1.5
    enocv 0.05
    rhokmin 1.e-3
  end
  fragments
     f1 CH2.results/adf.rkf
     f2 CrCO5.results/adf.rkf
  end
  print etslowdin
  symmetry NOSYM
  xc
    gga becke perdew
  end
EndEngine
eor
```

Example: NOCV: CH3 – CH3

Download CH3_CH3_etsnocv.run

```
#! /bin/sh


# Example for calculation of ETS-NOCV for simulated unrestricted fragments. ETS-NOCV:
# energy analysis using the Natural Orbitals for Chemical Valence. The ethane
# molecule is built from two methyl radicals

# First the two methyl fragments are calculated.
# If the FRAGOCCUPATIONS keyword is used in ethane the fragments should be calculated
# spin-restricted.


AMS_JOBNAME=frag1 $AMSBIN/ams <<eor
System
  atoms
     C         0.019664   -0.034069    0.009101
     H         0.039672   -0.069395    1.109620
     H         1.063205   -0.065727   -0.341092
     H        -0.474230   -0.953693   -0.341621
  end
end

Task SinglePoint

Engine ADF
  title CH3-CH3 built from CH3 radicals,  FULL electron calc.!
  basis
    Type DZP
    Core None
  end
  scf
    converge 1E-6
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor


AMS_JOBNAME=frag2 $AMSBIN/ams <<eor
System
  atoms
     C        -0.703210    1.217999   -0.497874
     H        -0.723753    1.252869   -1.598316
     H        -1.746567    1.250049   -0.147169
     H        -0.208833    2.137544   -0.147653
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title CH3 radical
  basis
    Type DZP
    Core None
  end
  scf
    converge 1E-6
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
  Relativity
    Level None
  End
EndEngine

eor

# Next these fragments are used in the calculation of the molecule ethane, using
# the key FRAGOCCUPATIONS to use an unrestricted fragment occupation for the
# methyl radicals, such that they are prepared for bonding. In the one fragment
# the singly occupied orbital will be an alpha-orbital, and in the other fragment
# the singly occupied orbital will be a beta-orbital, such that the calculated
# Pauli repulsion between the fragments will be small.

# The block key ETSNOCV used here is in its single line form.
# The symmetry must be NOSYM.

$AMSBIN/ams <<eor
System
  atoms
     C          0.019664   -0.034069    0.009101 adf.f=f1
     H          0.039672   -0.069395    1.109620 adf.f=f1
     H          1.063205   -0.065727   -0.341092 adf.f=f1
     H         -0.474230   -0.953693   -0.341621 adf.f=f1
     C         -0.703210    1.217999   -0.497874 adf.f=f2
     H         -0.723753    1.252869   -1.598316 adf.f=f2
     H         -1.746567    1.250049   -0.147169 adf.f=f2
     H         -0.208833    2.137544   -0.147653 adf.f=f2
  end
end

Task SinglePoint

Engine ADF
  title final [CH3]-[CH3], etsnocv activated by etsnocv
  etsnocv
    ekmin 1.5
    enocv 0.05
    rhokmin 1.e-3
    tvanalysis
  end
```

(continues on next page)

(continued from previous page)

```
  fragments
     f1 frag1.results/adf.rkf
     f2 frag2.results/adf.rkf
  end
  fragoccupations
     f1
     A 5 // 4
     subend
     f2
     A 4 // 5
     subend
  end
  scf
    converge 1E-6
  end
  symmetry NOSYM
  xc
    gga becke perdew
  end
  Relativity
    Level None
  End
EndEngine
eor

# Next densf calculations, to view the natural orbitals in this method, see
# also the the documentation for the densf analysis program and the ADF-GUI.
# Input is the adf.rkf of the molecular calculation.

$AMSBIN/densf <<eor
  adffile ams.results/adf.rkf
  Grid Medium
  End
  NOCV
   Alpha
     1
     2
     59
     60
   Beta
     1
     2
     59
     60
  END
eor
mv TAPE41 nocv1.t41

$AMSBIN/densf <<eor
  adffile ams.results/adf.rkf
  Grid Medium
  End
  NOCV
    THRESH 0.01
    RESTSUM
  END
eor
```

(continues on next page)

(continued from previous page)

```
mv TAPE41 nocv2.t41

$AMSBIN/densf <<eor
  adffile ams.results/adf.rkf
  Grid Medium
  End
  NOCV
    ALPHA
      SUM 1
    BETA
      SUM 1
    RESTSUM
  END
eor
mv TAPE41 nocv3.t41

$AMSBIN/densf <<eor
  adffile ams.results/adf.rkf
  Grid Medium
  End
  NOCV
    SUMBELOW 0.5
  END
eor
mv TAPE41 nocv4.t41
```

14.6.4 QTAIM

Example: QTAIM Analysis

Download Bader.run

```
#! /bin/sh


# The Bader atomic charges are calculated using a grid based method. Another
# possibility for Bader's analysis is to use the adf2aim utility such that a
# third party program Xaim can be used.

# With the QTAIM input block the ADF program will calculate Bader charges (AIM
# charges) using a grid based method.


AMS_JOBNAME=water $AMSBIN/ams <<eor
  Task SinglePoint

  System
    Atoms
      O          0.000000    0.000000   -0.001658
      H         -0.769048    0.000000    0.595209
      H          0.769048    0.000000    0.595209
    End
  End
```

(continues on next page)

(continued from previous page)

```
  Engine ADF
    Title Calculate Bader analysis for water
    basis
      Type TZP
      Core none
    End
    QTAIM
      AnalysisLevel Extended
      Spacing 0.1
    End
  EndEngine
eor


# Next a similar calculation for ferrocene is given


AMS_JOBNAME=ferrocene $AMSBIN/ams <<eor
  Task SinglePoint

  System
    Symmetrize Yes
    Atoms
      Fe        0.000000    0.000000    0.000000
      C         1.215650    0.000000    1.600813
      C         0.375656   -1.156152    1.600813
      C        -0.983481   -0.714541    1.600813
      C        -0.983481    0.714541    1.600813
      C         0.375656    1.156152    1.600813
      C         1.215650    0.000000   -1.600813
      C         0.375656    1.156152   -1.600813
      C        -0.983481    0.714541   -1.600813
      C        -0.983481   -0.714541   -1.600813
      C         0.375656   -1.156152   -1.600813
      H         2.310827    0.000000    1.629796
      H         0.714085   -2.197727    1.629796
      H        -1.869498   -1.358270    1.629796
      H        -1.869498    1.358270    1.629796
      H         0.714085    2.197727    1.629796
      H         2.310827    0.000000   -1.629796
      H         0.714085    2.197727   -1.629796
      H        -1.869498    1.358270   -1.629796
      H        -1.869498   -1.358270   -1.629796
      H         0.714085   -2.197727   -1.629796
    End
  End

  Engine ADF
    Title Calculate Bader analysis for ferrocene
    basis
      Type TZP
      Core none
    End
    QTAIM
      AnalysisLevel Extended
    End
    NumericalQuality Good
```

(continues on next page)

(continued from previous page)

```
  EndEngine
eor
```

Example: QTAIM Reactivity

Download Bader_Reactivity.run

```
#!/bin/sh


# With the CONCEPTUALDFT input block the ADF program will calculate
# certain conceptual DFT descriptors,
# such as atomic Fukui functions f+ and f-, the Koopmans dual descriptor, the
# Jenkin's metallicity and the atomic electronegativity by integrating
# corresponding properties over the atomic basin. Besides, MO populations per
# basin, the localization and delocalization indices (LI/DI), and the matrix of
# the condensed linear response function are computed.
# ConceptualDFT requires the symmetry to be NOSYM.


$AMSBIN/ams <<eor
  Task SinglePoint

  System
    Atoms
      C       1.06434610      -0.70982731      -0.00004522
      C      -0.17846421      -1.32042953      -0.00163715
      C      -0.19884953      -2.73466243      -0.00296149
      C       1.00825259      -3.45497625      -0.00262955
      C       2.24377213      -2.78457605      -0.00098836
      C       2.28117568      -1.37069091       0.00034973
      H      -1.10692850      -0.74587638      -0.00187350
      H      -1.15592647      -3.26173790      -0.00423999
      H       0.98594687      -4.54659148      -0.00365721
      H       3.17851894      -3.35030989      -0.00073866
      H       3.23233560      -0.83454684       0.00163171
    End
  End

  Engine ADF
    Title Calculate Bader analysis for benzene-radical
    basis
      Type TZ2P
      Core none
    End
    Unrestricted
    SpinPolarization  1
    BeckeGrid
       Quality good
    End
    ConceptualDFT
      AnalysisLevel Full
      Electronegativity true
    End
    Relativity
```

(continues on next page)

(continued from previous page)

```
      Level None # QTAIM energies are incompatible with the Scalar-ZORA relativistic␣
↪model
    End
    Symmetry NOSYM
    XC
       GGA PBE
    END
  EndEngine
eor
```

Example: IQA/QTAIM analysis

Download IQA.run

```
#! /bin/sh


# An IQA/QTAIM analysis is performed to analyze selected interactions
# in formic acid dimer

AMS_JOBNAME=IQA $AMSBIN/ams <<eor
System
  atoms
     C      -4.777280616716      -2.237679070420       0.027134922119
     O      -4.316098016052      -1.091055032354      -0.045982864295
     O      -4.095233471752      -3.376211708158      -0.036303239453
     H      -5.858352595246      -2.416180987684       0.158163319700
     H      -3.096816965327      -3.182195213950      -0.161436385033
     C      -1.043719708731      -1.744256936204      -0.434121365892
     O      -1.505389993838      -2.890945383925      -0.365122472158
     O      -1.725417179069      -0.605694491898      -0.367592891810
     H      -2.724119170709      -0.799847829461      -0.245243925045
     H       0.037480148811      -1.565749121998      -0.563863149179
  end
end

Task SinglePoint

Engine ADF
  title IQA Analysis for (HCOOH)2
  IQA
    Enabled True
    atomstodo 2 5 7 9
    print Normal
  end
  basis
    type TZP
    core None
  end
  symmetry NOSYM
  xc
    dispersion Grimme3
    gga BLYP
  end
```

(continues on next page)

(continued from previous page)

```
  Relativity
    Level None
  End
EndEngine

eor

# Another IQA analysis for a halogen bonded complex: F-Cl...NH3
# All interactions are calculated - Details printed (verbose mode)

AMS_JOBNAME=IQA2 $AMSBIN/ams <<eor
System
  atoms
     F       0.000001582319       0.000000000000      -3.410515944395
     Cl      0.000001582319       0.000000000000      -1.623156530960
     N       0.000001582319       0.000000000000       0.662184762067
     H      -0.482577891417       0.835852167201       0.991662308171
     H      -0.482577891417      -0.835852167201       0.991662308171
     H       0.965160529792       0.000000000000       0.991662308171
  end
end

Task SinglePoint

Engine ADF
  title IQA Analysis for F-Cl...NH3
  IQA
    Enabled True
    print Verbose
  end
  basis
    type TZ2P
    core None
  end
  symmetry NOSYM
  xc
    dispersion Grimme3
    gga BLYP
  end
  Relativity
    Level None
  End
EndEngine

eor


# IQA analysis for PF5

AMS_JOBNAME=IQA3 "$AMSBIN/ams" << eor

Task SinglePoint
System
    Atoms
        P   0.0            0.0            0.0
        F   0.0            0.0           -1.59029681775
        F   1.56093864075  0.0            0.0
```

(continues on next page)

(continued from previous page)

```
        F -0.78046932037  1.35181251664  0.0
        F  0.0            0.0            1.59029681775
        F -0.78046932037 -1.35181251664  0.0
    End
End

Engine ADF
  Title IQA Analysis PF5
  Basis
      Type DZP
      Core None
  End
  Symmetry NOSYM
  IQA
      Enabled Yes
      Print Verbose
      AtomsToDo 1 3
  End
  Relativity
    Level None
  End
EndEngine
eor
```

Example: QTAIM Aromaticity indices

Download Aromaticity.run

```
#! /bin/sh


# The aromaticity indices Iring and MCI can be calculated with ADF. Calculation
# of these aromaticity indices invokes the QTAIM analysis automatically and uses
# its results so all accuracy considerations for the QTAIM key are also valid
# here. The rings should be specified using the AROMATICITY input block, one
# line per ring. The atoms must be listed in the order they are connected in the
# ring. Note that although a partial QTAIM analysis is performed, its results
# are not printed to the output file.
# Relativistic hamiltonian cannot be used when computing AROMATICITY.


$AMSBIN/ams <<eor
System
  atoms
     C      -0.000000000000      -1.402840000000      -1.219800000000
     C       0.000000000000      -0.698820000000      -2.427560000000
     C       0.000000000000       0.698820000000      -2.427560000000
     C       0.000000000000       1.402840000000      -1.219800000000
     C       0.000000000000       0.708650000000      -0.000000000000
     C      -0.000000000000      -0.708650000000       0.000000000000
     C      -0.000000000000      -1.402840000000       1.219800000000
     C      -0.000000000000      -0.698820000000       2.427560000000
     C      -0.000000000000       0.698820000000       2.427560000000
     C       0.000000000000       1.402840000000       1.219800000000
     H      -0.000000000000      -2.486200000000      -1.238920000000
```

(continues on next page)

(continued from previous page)

```
     H      -0.000000000000      -1.237890000000      -3.366130000000
     H       0.000000000000       1.237890000000      -3.366130000000
     H       0.000000000000       2.486200000000      -1.238920000000
     H      -0.000000000000      -2.486200000000       1.238920000000
     H      -0.000000000000      -1.237890000000       3.366130000000
     H       0.000000000000       1.237890000000       3.366130000000
     H       0.000000000000       2.486200000000       1.238920000000
  end
end

Task SinglePoint

Engine ADF
  aromaticity
     1 2 3 4 5 6
     5 6 7 8 9 10
  end
  basis
    core None
    type TZP
  end
  xc
    gga PBE
  end
  Relativity Level=None
EndEngine

eor
```

14.6.5 DOS: Density of states

Example: Density of States: Cu4CO

Download DOS_Cu4CO.run

```
#! /bin/sh


# This sample illustrates the DOS property program to compute density-of-states
# data, for energy-dependent analysis.

# First, the Cu_4 CO molecule is calculated (ADF), using single-atom fragments.


$AMSBIN/ams <<eor
System
  atoms [Bohr]
     Cu     2.784      0.0                0.0
     Cu    -1.392    2.411014724135877  0.0
     Cu    -1.392   -2.411014724135877  0.0
     Cu     0.0      0.0               -3.9371705576466964
     C      0.0      0.0                2.65
     O      0.0      0.0                4.91
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  title Cu4CO (3,1) from atoms
  basis
    core small
    type TZP
  end
  print SFO
  xc
    gga PostSCF    Becke Perdew
  end
EndEngine
eor


# The PostSCF feature in the specification of the XC functional is used: the
# 'Becke-Perdew' GGA corrections are not included self-consistently but applied
# to the energy evaluation after the self-consistent LDA solution has been
# obtained.

# The utility program dos requires as input file a file named TAPE21 in the current␣
↪directory,
# or if one uses the key ADFFILE a (relative) path to an adf.rkf file.


$AMSBIN/dos <<eor
adffile ams.results/adf.rkf
file dostxt

energyrange npoint=36 e-start=-25 e-end=10

tdos

! Cu 3d partial DOS
gpdos
  a1   14:22 32:34
  a2    5:10
  e1:1 18:32 37:42
  e1:2 18:32 37:42
end

! The same but using BAS
gpdos
  BAS 17:34 57:74 97:114 137:154
end

! The same as above, but using much less complicated input
gpdos
  ATYPE Cu d
end

! Overlap PDOS Cu 3d and CO 2p
opdos
  ATYPE Cu 3d
```

(continues on next page)

(continued from previous page)

```
SUBEND
  ATOM 5 2p
  ATOM 6 2p
end

eor


echo "DOS result file"
echo "==============="
cat dostxt
echo "END of DOS result file"

# Here, the total density of states, as well as various partial densities of
# states, are computed. You may feed the results found in the dostxt file into a
# plotting program such as gnuplot. The result is not displayed here. See the
# ADF manual for more detailed info about the dos program.
```

14.6.6 Third party analysis software

Example: adf2aim: convert an ADF adf.rkf to WFN format (for Bader analysis)

Download AIM_HF.run

```
#! /bin/sh


# One can calculate Bade atomic charges and other Atoms in Molecule properties
# directly in ADF using a grid based method, see also an example that uses the
# grid based method. Another possibility for Bader's analysis, an example is
# described here, is to use the adf2aim utility such that a third party program
# Xaim can be used.

# The ADF utility adf2aim (original name rdt21, now part of the ADF package)
# developed by Xavi Lopez, Engelber Sans and Carles Bo converts an ADF adf.rkf to
# WFN format (for Bader analysis).

# The WFN file is an input file for the third party program Xaim (see
# http://www.quimica.urv.es/XAIM for details), which is a graphical user
# interface to programs that can perform the Bader analysis.

# Usage of adf2aim:


AMS_JOBNAME=HF $AMSBIN/ams <<eor
System
  atoms
     H  0.000  0.000  0.000
     F  0.000  0.000  0.917
  end
end

Task SinglePoint

```

(continues on next page)

(continued from previous page)

```
Engine ADF
EndEngine
eor

$AMSBIN/adf2aim HF.results/adf.rkf
echo ""
echo 'Contents of rdt21.res:'
cat rdt21.res | grep -v RunTime
echo 'Contents of WFN:'
cat WFN | grep -v RunTime

rm rdt21.res WFN


# In an unrestricted calculation adf2aim will ask if the MOs should be split in
# two files. In this example the answer is yes (y).


AMS_JOBNAME=HF_unrestricted $AMSBIN/ams <<eor
System
  atoms
     H  0.000  0.000  0.000
     F  0.000  0.000  0.917
  end
end

Task SinglePoint

Engine ADF
  spinpolarization 0
  unrestricted Yes
EndEngine
eor

$AMSBIN/adf2aim HF_unrestricted.results/adf.rkf <<eor
y
eor

echo ""
echo 'Contents of rdt21.res:'
cat rdt21.res| grep -v RunTime
echo 'Contents of WFN-alpha:'
cat WFN-alpha| grep -v RunTime
echo 'Contents of WFN-beta:'
cat WFN-beta| grep -v RunTime
```

Example: NBO analysis: adfnbo, gennbo

Download H2O_ADFNBO.run

```
#! /bin/sh


# Dr. Autschbach, SCM, and Prof. Weinhold have collaborated to prepare a simple
# in put file generator, called adfnbo, for the GENNBO program of Prof.
# Weinholds Natural Bond Orbital (NBO) 6.0 package: http://nbo6.chem.wisc.edu
# The GENNBO executable is included in the ADF distribution and can be enabled
# via the license file ( info@scm.com).


$AMSBIN/ams <<eor
System
  atoms
     O          0.000000    0.000000    0.000000
     H          0.900000    0.000000    0.000000
     H         -0.156283    0.886327    0.000000
  end
end

Task SinglePoint

Engine ADF
  aomat2file
  fullfock
  basis
    core NONE
    type DZ
  end
  save TAPE15
  symmetry NOSYM
  title simple NBO example for water
EndEngine
eor

# Note added: recommended is to use the key 'spherical' in the adfnbo input.
# Note added: recommended is not to include the key 'TESTJOB' in the adfnbo input
# for larger calculations. The key TESTJOB will include the NRT key in FILE47.
# The calculation of NRT (natural resonance theory) can be very expensive
# for larger systems (or may fail to converge)

# A File named FILE47 is generated by adfnbo which is an input file for the
# general NBO program gennbo. ADF needs to write some data to file, which is
# done by including these keywords in the adf input file:


# == GENNBO ==

# Next a brief summary of the capabilities of GENNBO is given (by Prof.
# Weinhold). GENNBO implements most capabilities of the full NBO 6.0 program
# suite as described on the NBO website: http://nbo6.chem.wisc.edu These include
# determination of natural atomic orbitals (NAOs), bond orbitals (NBOs), and
# localized MOs (NLMOs), as well as the associated NPA (atomic charges and
# orbital populations) and NRT (resonance structures, weightings, bond orders)
# valence descriptors, for a wide variety of uncorrelated and correlated
```

(continues on next page)

(continued from previous page)

```
# (variational, perturbative, or density functional) theoretical levels. GENNBO-
# supported options include all keywords except those explicitly requiring
# interactive communication with the host electronic structure system (viz.,
# $DEL deletions, NEDA, NCS, NJC). The GENNBO program typically sits
# conveniently on the PC desktop, ready to analyze (or re-analyze at will, with
# altered options) the final results of a complex ADF calculation performed on a
# remote cluster.

# GENNBO 'communicates' with the original ADF calculation through an archive
# file (JOB.47 file, preserving all necessary details of the final density) that
# is initially generated by ADF and subsequently becomes the input file for
# GENNBO. The .47 file contains a standard $NBO ... $END keylist that can be
# edited with a standard word processor or text editor to include chosen NBO
# keyword options, just as though they might have appeared in the original input
# stream of an interactive ADFNBO run. The stand-alone GENNBO program therefore
# allows many alternative NBO analysis options to be explored at leisure,
# without costly re-calculation of the wave function.

# Using the GENNBO executable is possible only if NBO6 is enabled in your
# license file by SCM (info@scm.com). In that case you will get access to an
# NBO 6.0 manual in electronic form that explains in detail how GENNBO can be
# used and how the output should be interpreted.


$AMSBIN/adfnbo <<eor
  ADFfile ams.results/adf.rkf
  TAPE15file ams.results/TAPE15
  write
  fock
  NBOkeylist NBONLMO=W AONBO=W AONLMO=W NLMOMO=W NAOMO=PVAL NRT STERIC CMO 3CBOND DIST
eor

echo " "
echo " "
echo "Contents of FILE47 ="
echo " "
echo " "
cat FILE47

$AMSBIN/gennbo6 FILE47

echo " "
echo " "
echo "Contents of adfnbo.37 = "
echo " "
echo " "
cat adfnbo.37

echo " "
echo " "
echo "Contents of adfnbo.39 ="
echo " "
echo " "

cat adfnbo.39

echo " "
```

(continues on next page)

(continued from previous page)

```
echo " "
echo "Contents of adfnbo.49 ="
echo " "
echo " "
cat adfnbo.49
```

Example: NBO analysis: EFG

Download AlCl3_efgnbo.run

```
#! /bin/sh


# Example shows an NBO analysis of an EFG calculation for AlCl3 .

# In the ADF input one then needs to include the QTENS (EFG calculation) and
# include the subkey EFG of the key AORESPONSE. The BeckeGrid quality is good,
# which may be needed to accurately calculate core properties. Other keywords
# are necessary because of the NBO analysis afterwards. Note that ADF, ADFNBO,
# and GENNBO have to run several times.

# Note added: recommended is not to include the key 'TESTJOB' in the adfnbo input
# for larger calculations. The key TESTJOB will include the NRT key in FILE47.
# The calculation of NRT (natural resonance theory) can be very expensive
# for larger systems (or may fail to converge)

AMS_JOBNAME=first $AMSBIN/ams <<eor
System
  atoms
     Al      0.000000      0.000000     -0.237368
     Cl      1.808813      0.000000      0.807083
     Cl      0.000000      0.000000     -2.326083
     Cl     -1.808813      0.000000      0.807083
  end
end

Task SinglePoint

Engine ADF
  aomat2file
  aoresponse
    efg atom=1 nbo=true
  end
  beckegrid
    quality good
  end
  fullfock
  basis
    core none
    type TZP
  end
  qtens
  save TAPE15
  symmetry NOSYM
  xc
```

(continues on next page)

(continued from previous page)

```
    gga revPBE
    lda vwn
  end
  Relativity
    Level None
  End
EndEngine
eor

# end ADF, run gennbo
# ===================


$AMSBIN/adfnbo <<eor
 adffile first.results/adf.rkf
 TAPE15file first.results/TAPE15
 write
 spherical
 fock
 TESTJOB
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48

$AMSBIN/gennbo6 FILE47

$AMSBIN/adfnbo <<eor
 adffile first.results/adf.rkf
 TAPE15file first.results/TAPE15
 copy
 spherical
 fock
 TESTJOB
eor

$AMSBIN/adfnbo <<eor
 adffile first.results/adf.rkf
 TAPE15file first.results/TAPE15
 read
 spherical
 fock
 TESTJOB
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48

# end gennbo, rerun ADF with NBO property analysis
# ================================================

AMS_JOBNAME=second  $AMSBIN/ams <<eor
System
  atoms
     Al      0.000000      0.000000     -0.237368
     Cl      1.808813      0.000000      0.807083
     Cl      0.000000      0.000000     -2.326083
     Cl     -1.808813      0.000000      0.807083
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  aoresponse
    efg
      atom 1
      nbo true
    end
  end
  beckegrid
    quality good
  end
  basis
    core none
    type TZP
  end
  qtens
  symmetry NOSYM
  xc
    gga revPBE
    lda vwn
  end
  Relativity
    Level None
  End
EndEngine
eor
```

Example: NBO analysis: NMR chemical shift

Download CH4_nmrnbo.run

```
#! /bin/sh


# Example shows an NBO analysis of an NMR shielding calculation for CH4.

# Some keywords are necessary because of the NBO analysis afterwards. A 'Good'
# integration grid is used. First the scalar relativistic calculation is
# performed and the scalar relativistic localized orbitals are made. Next the
# spin-orbit coupled ADF calculation is done, and a calculation of NMR shielding
# constants is performed with an analysis in scalar relativistic localized
# orbitals.


AMS_JOBNAME=Scalar $AMSBIN/ams <<eor
System
  atoms
     C  0.0000  0.0000  0.0000
     H  0.6316  0.6316  0.6316
     H  0.6316 -0.6316 -0.6316
     H -0.6316  0.6316 -0.6316
     H -0.6316 -0.6316  0.6316
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  aomat2file
  beckegrid
    quality good
  end
  fullfock
  basis
    core None
    type DZP
  end
  save TAPE15
EndEngine
eor

###### end scalar, run gennbo ##########

$AMSBIN/adfnbo <<eor
  ADFfile Scalar.results/adf.rkf
  TAPE15file Scalar.results/TAPE15
  write
  spherical
  :: read
  fock
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48
$AMSBIN/gennbo6 FILE47

# run adfnbo in COPY mode

$AMSBIN/adfnbo <<eor
  ADFfile Scalar.results/adf.rkf
  TAPE15file Scalar.results/TAPE15
  spherical
  fock
  copy
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48

##### end gennbo

AMS_JOBNAME=NOSYM $AMSBIN/ams <<eor
System
  atoms
     C  0.0000  0.0000  0.0000
     H  0.6316  0.6316  0.6316
     H  0.6316 -0.6316 -0.6316
     H -0.6316  0.6316 -0.6316
     H -0.6316 -0.6316  0.6316
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type DZP
  end
  noprint sfo
  save TAPE10
  symmetry nosym
EndEngine
eor

$AMSBIN/nmr <<eor
  adffile NOSYM.results/adf.rkf
  tape10file NOSYM.results/TAPE10
  fakeso
  nmr
   atoms 2 1
   u1k best
   out iso tens
  end
  analysis
   print 0.01
   canonical
   nbo
   components
  end
eor

### scalar done, redo computation with spin-orbit coupling

AMS_JOBNAME=Spinorbit $AMSBIN/ams <<eor
System
  atoms
     C  0.0000  0.0000  0.0000
     H  0.6316  0.6316  0.6316
     H  0.6316 -0.6316 -0.6316
     H -0.6316  0.6316 -0.6316
     H -0.6316 -0.6316  0.6316
  end
end

Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type DZP
  end
```

(continues on next page)

(continued from previous page)

```
  noprint sfo
  relativity
    level spin-orbit
  end
  save TAPE10
  symmetry nosym
EndEngine
eor


$AMSBIN/nmr <<eor
  adffile Spinorbit.results/adf.rkf
  tape10file Spinorbit.results/TAPE10
  nmr
   atoms 2 1
   u1k best
   out iso tens
  end
  analysis
   print 0.01
   canonical
   nbo
   components
  end
eor
```

Example: NBO analysis: NMR spin-spin coupling

Download `CPL_CH3OH_NBO.run`

```
#!/bin/sh


# Example shows an NBO analysis of an NMR spin-spin coupling constants
# calculation for CH3 OH.

# Some keywords are necessary because of the NBO analysis afterwards. A good
# quality integration is used. First the scalar relativistic calculation is
# performed and the scalar relativistic localized orbitals are made, and a
# calculation of NMR spin-spin coupling constants is performed with an analysis
# in scalar relativistic localized orbitals. Next the spin-orbit coupled ADF
# calculation is done, and a calculation of NMR spin-spin coupling constants is
# performed with an analysis in scalar relativistic localized orbitals.



AMS_JOBNAME=scalar $AMSBIN/ams <<eor
System
  atoms
     O       0.151078120000      -0.158942890000      -0.184382010000
     H       0.762854510000       0.480823600000       0.187867830000
     C       0.654254930000      -1.481762230000       0.026343630000
     H       1.616760580000      -1.581906770000      -0.455670800000
     H      -0.035909520000      -2.200223490000      -0.393433960000
     H       0.761359880000      -1.661537720000       1.087000640000
```

(continues on next page)

(continued from previous page)

```
  end
end

Task SinglePoint

Engine ADF
  aomat2file
  beckegrid
    quality good
  end
  fullfock
  basis
    core None
    type DZP
  end
  relativity
    level scalar
    formalism ZORA
  end
  save TAPE15
  symmetry nosym
  xc
    lda SCF VWN
  end
EndEngine
eor

###### end scalar, run gennbo ##########


rm -f adfnbo.kf

$AMSBIN/adfnbo <<eor
  ADFfile scalar.results/adf.rkf
  TAPE15file scalar.results/TAPE15
  write
  spherical
  :: read
  fock
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48
$AMSBIN/gennbo6 FILE47

# run adfnbo in COPY mode

$AMSBIN/adfnbo <<eor
  ADFfile scalar.results/adf.rkf
  TAPE15file scalar.results/TAPE15
  spherical
  fock
  copy
eor

# run adfnbo in READ mode: prepare locorb on TAPE21

$AMSBIN/adfnbo <<eor
```

(continues on next page)

(continued from previous page)

```
  ADFfile scalar.results/adf.rkf
  TAPE15file scalar.results/TAPE15
  spherical
  fock
  read
eor

rm -f adfnbo.37 adfnbo.39 adfnbo.49 adfnbo.48

##### end gennbo

## end scalar and gennbo run, now do the coupling constant

$AMSBIN/cpl <<eor
  adffile scalar.results/adf.rkf
  nmrcoupling
    xalpha
    dso
    pso
    sd
    scf converge=1e-5 iterations=10
    contributions 1e19 nbo
    nuclei 3  5 6
  end
eor

# redo the coupling constant without the SD term

rm -f TAPE10 TAPE15 TAPE13

$AMSBIN/cpl <<eor
  adffile scalar.results/adf.rkf
  nmrcoupling
    xalpha
    dso
    pso
    scf converge=1e-5 iterations=10
    contributions 1e19 nbo
    nuclei 3  5 6
  end
eor


### scalar CPL done, redo computation with spin-orbit coupling
for f in TAPE*; do rm -f $f ; done

AMS_JOBNAME=spinorbit $AMSBIN/ams <<eor
System
  atoms
     O       0.151078120000      -0.158942890000      -0.184382010000
     H       0.762854510000       0.480823600000       0.187867830000
     C       0.654254930000      -1.481762230000       0.026343630000
     H       1.616760580000      -1.581906770000      -0.455670800000
     H      -0.035909520000      -2.200223490000      -0.393433960000
     H       0.761359880000      -1.661537720000       1.087000640000
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  beckegrid
    quality good
  end
  basis
    core None
    type DZP
  end
  noprint sfo
  relativity
    level spin-orbit
    formalism ZORA
  end
  symmetry nosym
  xc
    lda SCF VWN
  end
EndEngine
eor


## end spinorbit run, now do the coupling constant

$AMSBIN/cpl <<eor
  adffile spinorbit.results/adf.rkf
  nmrcoupling
    xalpha
    dso
    pso
    sd
    scf converge=1e-5 iterations=10
    contributions 1e19 nbo
    nuclei 3  5 6
  end
eor
```

Example: Multiple excited state gradients: H2O

Download EGO_H2O_multi.run

```
#! /bin/sh


# Basic example of TDDFT calculation with different number of singlets and triplets␣
↪(LOWEST 10 8)
# and multiple gradient calculation (SING_GRADS, TRIP_GRADS)
# Could be useful for external programs like SHARC.

$AMSBIN/ams <<eor
System
  atoms
     O     0.0000000    0.0000000   -0.0668805
```

(continues on next page)

(continued from previous page)

```
     H    -0.7658756    0.0000000    0.5307937
     H     0.7658756    0.0000000    0.5307937
  end
end

Task SinglePoint
Properties
  Gradients True
End

Engine ADF
  title H2O TDDFT Multiple Gradients
  excitations
    lowest 10 8
  end
  excitedgo
    sing_grads
       A1 1 2
       A2 1
       B1 1
    end
    trip_grads
       A1 1 2 3
       A2 1
       B1 3
       B2 1
    end
  end
  basis
    type SZ
    core None
    CreateOutput Yes
  end
  numericalquality Good
  print SOMATRIX
  relativity
    level scalar
    formalism ZORA
  end
  sopert
     gscorr True
  end
  symmetry C(2v)
  tda
  xc
    gga BP86
  end
EndEngine
eor
```

Example: Calculation of overlap of primitive basis functions

Download CalcOverlapOnly.run

```
#! /bin/sh


# Example for calculating overlaps of primitive basis functions for almost
# coinciding nuclear coordinates, using the key CALCOVERLAPONLY. ADF will stop
# after the calculation of these overlaps. The overlap is printed in the output,
# but is also present on TAPE15 in binary format in Matrices%Smat.
# Could be useful for external programs like SHARC.

$AMSBIN/ams <<eor
System
  atoms
     C     0.1662008   -0.0114225   0.0220295  region=one
     S     1.7387976    0.003246   -0.0021948  region=one
     H    -0.3207087    0.9655128  -0.0963368  region=one
     H    -0.479404    -0.9324828  -0.0963367  region=one
     C     0.167746    -0.0109425   0.0203042  region=two
     S     1.7382268    0.002927   -0.0020154  region=two
     H    -0.329981     0.9728265  -0.0893183  region=two
     H    -0.4704213   -0.9353924  -0.088502   region=two
  end
  # Prevents AMS from stopping in case of almost-coinciding nuclei
  AllowCloseAtoms True
end

Task SinglePoint

Engine ADF
  Title CH2S overlap of near identical geometries
  CalcOverlapOnly
  Basis
    Type DZP
    Core None
  End
  Print SMAT
  Save TAPE15
EndEngine

eor
```

14.7 Accuracy and Efficiency

14.7.1 BSSE, SCF convergence, Frequencies

Example: Basis Set Superposition Error (BSSE): Cr(CO)5 +CO

Download BSSE_CrCO6.run

```
#! /bin/sh
```

(continues on next page)

(continued from previous page)

```
# A study of the Basis Set Superposition Error (BSSE) in the formation of
# Cr(CO)_6. from CO and Cr(CO)_5.
# This example uses scalar relativistic ZORA calculations.

# For the BSSE calculation special chemical elements must be created to describe
# the 'ghost' atoms, which have zero nuclear charge and mass. They do have basis
# functions (and fit functions), however, and they are used to calculate the
# lowering of the energy of the system to which the ghost atoms are added, due
# to the enlargement of the basis by the ghost basis. The ghost atom has the
# same basis and fit set as the normal element but no nuclear charge and no
# frozen core. The BASIS key recognizes elements denoted with Gh.atom in the
# ATOMS key as being ghost atoms. If the basis file specifies a frozen core ADF
# will treat it as if no frozen core is present.

# The following calculations are carried out:

# 1. CO from C and O. This yields the bond energy of CO with respect to
#    the (restricted) basic atoms.
# 2. CO from the fragments CO (as calculated in 1) and the ghost atom Cr
#    and 5 Carbon and 5 Oxygen ghost atoms. The ghost atomic fragments
#    provide basis and fit functions but do not contribute charge or
#    potential to the molecule. The bond energy of this calculation is
#    the energy lowering of CO due to the additional basis functions.
#    This is the BSSE for CO.
# 3. Cr(CO)5 from Cr and 5 CO's. This yields the ('normal') bond energy
#    with respect to the given fragments.
# 4. Cr(CO)5 from Cr(CO)5 as fragment (as calculated in 3) but with the
#    CO basis functions added on the position of the 6th CO ('ghost' CO).
#    The bond energy is the BSSE for Cr(CO)5 .
# 5. Cr(CO)6 with Cr(CO)5 and CO as fragments. The bond energy is the one
#    without BSSE. This bond energy can now be corrected by the sum of
#    superposition contributions of calculations 2 and 4.



# This series of calculations is carried out with basis set DZ.

# Finally, the whole thing might be redone with basis set TZP, to show that the
# BSSE decreases with larger basis.

# The calculations for the type DZ basis are contained in the sample script
# (with input- and output files). Those for type TZP bases can be set up easily
# and may be done as an exercise.

# For the first series of calculations, with basis type DZ, the input files are
# discussed below and the relevant parts are echoed from the output files where
# the energy decomposition and the total bond energy are printed.

# For the other series, using type TZP basis sets, only a summary of the results
# is given.

# =====================
# Computational details
# =====================

# The calculations in this example all use:
```

(continues on next page)

(continued from previous page)

```
# Small core DZ basis set Frozen core level for the Chromium atom: 2p, for
# Carbon and Oxygen: 1s Numerical integration precision 4.0 (in Create runs
# 10.0, the default) Default settings for model parameters such as density
# functional (key XC) and for the remaining computational settings For the BSSE
# calculations we first do the 'normal' calculations of CO and Cr(CO)5 ,
# yielding the fragment files t21.CO and t21.CrCO5. The input files for these
# calculations are not shown here.


# ===========
# BSSE for CO
# ===========

# For the CO BSSE calculation the standard CO must have been computed first. In
# the BSSE run a Cr(CO)5 ghost fragment basis set is then added to the 'normal'
# CO input. Important is the use of the BASIS key. In this case the BASIS key is
# used for the generation of the ghost atoms, it should have the same definition
# for the atoms as will be used later for the Cr(CO)5 fragment. The FRAGMENTS
# key is used for the fragment CO. The energy change (the printed 'bond energy'
# in the output) is the BSSE.


AMS_JOBNAME=CO $AMSBIN/ams <<eor
  Task SinglePoint
  System
    atoms
      C    0   0   1.86
      O    0   0   3.03
    end
  end
  Engine ADF
    title CO (normal run)
    basis
      Type DZ
      Core Small
    End
    symmetry C(lin)
  EndEngine
eor


# The input file for the CO-BSSE run is:

AMS_JOBNAME=CO_with_fake_CrC5O5 $AMSBIN/ams <<eor
  Task SinglePoint
  System
    atoms
     Gh.Cr    0       0       0
     Gh.C    -1.86    0       0
     Gh.C     1.86    0       0
     Gh.C     0       1.86    0
     Gh.C     0      -1.86    0
     Gh.C     0       0      -1.86
     Gh.O     3.03    0       0
     Gh.O    -3.03    0       0
     Gh.O     0       3.03    0
```

(continues on next page)

(continued from previous page)

```
     Gh.O      0       -3.03     0
     Gh.O      0        0      -3.03
        C      0        0       1.86       adf.f=CO
        O      0        0       3.03       adf.f=CO
    end
  end
  Engine ADF
    title  BSSE for CO due to Cr(CO)5 ghost
    noprint sfo,frag,functions
    basis
      Type DZ
      Core Small
    end
    fragments
      CO CO.results/adf.rkf
    end
    symmetry  C(4V)
  EndEngine
eor

# In the output we find in the Bond Energy section: The BSSE for CO is computed
# as 2.40 kcal/mol


# ================
# BSSE for Cr(CO)5
# ================

# In similar fashion the BSSE is computed for Cr(CO)_5 . In the BSSE run a ghost
# atoms C and O at the positions they will have in the Cr(CO)_6 molecule are
# added to the normal Cr(CO)_5 input:

AMS_JOBNAME=CrCO5 $AMSBIN/ams <<eor
  Task SinglePoint
  System
    atoms
      Cr     0        0        0
      C      1.86     0        0         adf.f=CO|1
      C     -1.86     0        0         adf.f=CO|2
      C      0        1.86     0         adf.f=CO|3
      C      0       -1.86     0         adf.f=CO|4
      C      0        0       -1.86      adf.f=CO|5
      O      3.03     0        0         adf.f=CO|1
      O     -3.03     0        0         adf.f=CO|2
      O      0        3.03     0         adf.f=CO|3
      O      0       -3.03     0         adf.f=CO|4
      O      0        0       -3.03      adf.f=CO|5
    end
  end
  Engine ADF
    title Cr(CO)5  (normal run)
    noprint sfo,frag,functions
    SCF
      mixing 0.1
    END
    basis
      Type DZ
```

(continues on next page)

(continued from previous page)

```
      Core Small
    end
    fragments
      CO CO.results/adf.rkf
    end
    symmetry C(4v)
  EndEngine
eor


AMS_JOBNAME=final $AMSBIN/ams <<eor
  Task SinglePoint
  System
    atoms
      Cr     0        0        0         adf.f=CrCO5
      C      1.86     0        0         adf.f=CrCO5
      C     -1.86     0        0         adf.f=CrCO5
      C      0        1.86     0         adf.f=CrCO5
      C      0       -1.86     0         adf.f=CrCO5
      C      0        0       -1.86      adf.f=CrCO5
      O      3.03     0        0         adf.f=CrCO5
      O     -3.03     0        0         adf.f=CrCO5
      O      0        3.03     0         adf.f=CrCO5
      O      0       -3.03     0         adf.f=CrCO5
      O      0        0       -3.03      adf.f=CrCO5
      Gh.C    0        0        1.86
      Gh.O    0        0        3.03
    end
  end
  Engine ADF
    title BSSE for Cr(CO)5 due to CO ghost
    noprint sfo,frag,functions
    basis
      Type DZ
      Core Small
    end
    fragments
      CrCO5   CrCO5.results/adf.rkf
    end
    symmetry C(4v)
  EndEngine
eor

# The Bond Energy result yields 1.97 kcal/mol for the BSSE.


# ============================================
# Bond Energy calculation with BSSE correction
# ============================================

# The bonding of CO to Cr(CO)5 is computed in the normal way: from fragments CO
# and Cr(CO)5 . The obtained value for the bond energy can then simply corrected
# for the two BSSE terms, (2.40+1.97=) 4.37 kcal/mol together.


# ===========================
# Relevance of Core Functions
```

(continues on next page)

(continued from previous page)

```
# ===========================

# The two BSSE runs can also be repeated, but now with the core
# orthogonalization functions omitted from the ghost bases. To to this one can
# not use the BASIS key, but one needs to explicitly 'create' the ghost atoms.
# This will not be done here, but only the results will be discussed. One may
# argue about whether these functions should be included in the ghost basis
# sets, but since they are very contracted around the ghost nuclei they are not
# expected to contribute significantly anyway and may then just as well be
# omitted. This has been explicitly verified by test examples. The Core
# Functions (the functions in the valence basis set that serve only for core-
# orthogonalization, for instance the 1S 5.40 in the Carbon basis set (see the
# $AMSHOME/atomicdata/ADF/ZORA/DZ/C.1s basis set file) are removed from the Create␣
↪data
# files used for the creation of the ghost atoms. This yields as BSSE values for
# CO and Cr(CO)5 respectively 2.32 and 1.97 kcal/mol (compare 2.40 and 1.97
# kcal/mol for the case with Core Functions included). The net total effect of
# including/removing the Core Functions is therefore
# (2.40-2.32)+(1.97-1.97)=0.08 kcal/mol. This is an order of magnitude smaller
# than the BSSE effect itself.


# ===================================
# BSSE and the size of the basis set
# ===================================

# BSSE effects should diminish with larger bases and disappear in the limit of a
# perfect basis. This can be studied by comparing the BSSE for basis DZ, see
# above, with the BSSE for basis TZP. The procedure is completely similar to the
# one above and yields: For the BSSE terms: 0.7 kcal/mol for CO (compare: 2.4
# kcal/mol for basis DZ), and 0.5 kcal/mol for Cr(CO)5 (1.9 for basis DZ) The
# total BSSE drops from 4.4 kcal/mol in basis DZ to 1.2 in basis TZP.


# =========
# Reference
# =========

# A systematic study with adf of the BSSE in metal-carbonyl complexes can be
# found in Rosa, A., et al., Basis Set Effects in Density Functional
# Calculations on the Metal-Ligand and Metal-Metal Bonds of Cr(CO)5-CO and
# (CO)5. Journal of Physical Chemistry, 1996, 100: p. 5690-5696.


$AMSBIN/ams <<eor
  Task SinglePoint
  System
    atoms
     Cr     0        0        0        adf.f=CrCO5
     C      1.86     0        0        adf.f=CrCO5
     C     -1.86     0        0        adf.f=CrCO5
     C      0        1.86     0        adf.f=CrCO5
     C      0       -1.86     0        adf.f=CrCO5
     C      0        0       -1.86     adf.f=CrCO5
     O      3.03     0        0        adf.f=CrCO5
     O     -3.03     0        0        adf.f=CrCO5
     O      0        3.03     0        adf.f=CrCO5
```

(continues on next page)

(continued from previous page)

```
     O     0       -3.03    0        adf.f=CrCO5
     O     0        0      -3.03     adf.f=CrCO5
     C     0        0       1.86     adf.f=CO
     O     0        0       3.03     adf.f=CO
    end
  end
  Engine ADF
    symmetry C(4V)
    title Bond energy without BSSE for Cr(CO)6 made of Cr(CO)5 and CO
    noprint sfo,frag,functions
    basis
      Type DZ
      Core Small
    end
    fragments
     CrCO5   CrCO5.results/adf.rkf
     CO      CO.results/adf.rkf
    end
  EndEngine
eor
```

Example: Troubleshooting SCF convergence: Ti2O4

Download SCF_Ti2O4.run

```
#! /bin/sh


# One can run into SCF convergence problems when calculating certain types of
# systems. Some of the notorious examples are transition metal oxides and
# lanthanide compounds. Below, several approaches to solving the SCF convergence
# problem are demonstrated.


AMS_JOBNAME=Default $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (default case)
  basis
    core Small
    type DZ
  end
  scf
```

(continues on next page)

(continued from previous page)

```
    converge 1.0e-6 1.0e-6
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor


AMS_JOBNAME=MESA $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test MESA
  basis
    core Small
    type DZ
  end
  scf
    accelerationmethod MESA
    converge 1.0e-6 1.0e-6
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor


AMS_JOBNAME=LISTb $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (LISTb by Alex Wang)
  basis
```

(continues on next page)

(continued from previous page)

```
    core Small
    type DZ
  end
  scf
    accelerationmethod LISTb
    converge 1.0e-6 1.0e-6
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor


AMS_JOBNAME=LISTf $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (LISTf by Alex Wang)
  basis
    core Small
    type DZ
  end
  scf
    accelerationmethod LISTf
    converge 1.0e-6 1.0e-6
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor


AMS_JOBNAME=fDIIS $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end
```

(continues on next page)

(continued from previous page)

```
Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (fDIIS by Alex Wang)
  basis
    core Small
    type DZ
  end
  scf
    accelerationmethod fDIIS
    converge 1.0e-6 1.0e-6
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor



# A-DIIS method. The A-DIIS method combines the strength of the ARH and DIIS
# methods. It does not require energy evaluation so it is much cheaper than the
# ARH and Energy-DIIS methods.


AMS_JOBNAME=ADIIS $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (A-DIIS by Hu and Yang)
  basis
    core Small
    type DZ
  end
  occupations KeepOrbitals=100
  scf
    accelerationmethod ADIIS
    converge 1.0e-6 1.0e-6
    iterations 300
    mixing 0.05
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor
```

(continues on next page)

(continued from previous page)

```
# This example uses the LISTi method. LISTi is very
# similar to the usual DIIS but typically it performs much better. It is also
# computationally less expensive and scales better in parallel even though DIIS
# is rarely a scaling bottleneck.


AMS_JOBNAME=LISTi $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (LISTi)
  basis
    core Small
    type DZ
  end
  scf
    accelerationmethod LISTi
    iterations 300
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor



# This example uses the Augmented Roothaan-Hall (ARH) method.
# The basic idea of this method is that the density matrix is
# optimized directly to minimize the total energy. Important: the ARH method can
# be used with SYMMETRY NOSYM only.


AMS_JOBNAME=ARH $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
```

(continues on next page)

(continued from previous page)

```
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (Augmented Roothaan-Hall)
  basis
    core Small
    type DZ
  end
  scf
    arh
    end
    iterations 300
    mixing 0.05
  end
  symmetry NOSYM
  xc
    gga Becke Perdew
  end
EndEngine
eor



# This is an extension to the so-called 'electron smearing' method.
# In this method, the electrons are distributed among orbitals around Fermi-
# level using a pseudo-thermal distribution function. Although the result with
# fractional occupation number has no physical sense, the method can be used to
# achieve integer occupation numbers by reducing the smearing parameter step-
# wise.
# A few notes:

# You can specify up to ten comma-delimited values after Smear= (no spaces are
# allowed). ADF will start from the first value and try to converge SCF using
# it. If it succeeds, the next value will be picked and so on.

# Because the whole process may require many cycles to converge it is important
# to set the number of SCF cycles to a large value to avoid a premature
# termination.


AMS_JOBNAME=Smear $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
```

(continues on next page)

(continued from previous page)

```
  title Ti2O4 SCF aid test (Smear)
  debug DIIS SDIIS
  basis
    core Small
    type DZ
  end
  occupations Smear=0.2,0.1,0.07,0.05,0.03,0.02,0.01,0.007,0.005,0.001
  scf
    iterations 300
    oldscf
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor


# This example demonstrates the use of the Occupations Steep= option (see
# the User's Guide for details).

# One difference is, obviously, in the Occupations keyword. The other difference
# is more subtle. For stable convergence, it is often essential to switch off
# DIIS and set the mixing parameter to a low value. Of course, it will make
# convergence quite (sometimes very) slow. Ultimately you should get either an
# aufbau configuration or a configuration with exactly degenerate HOMO. In this
# example, the result is an aufbau solution.

# Both methods should, in principle, give the same result, which is the case in
# this example.


AMS_JOBNAME=Steep $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (Steep)
  basis
    core Small
    type DZ
  end
  occupations Steep=0.5,0.3
  scf
    diis
      n 0
    end
```

(continues on next page)

(continued from previous page)

```
    iterations 300
    mixing 0.05
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor


# This example uses the so called Energy-DIIS method. Please note that similar
# to ARH and unlike the standard SCF procedure in ADF this method requires
# energy evaluation at each SCF cycle, which makes it significantly slower
# compared to energy-free SCF.


AMS_JOBNAME=EDIIS $AMSBIN/ams <<eor
System
  atoms
     Ti  1.730   0.000   0.000
     Ti -1.730   0.000   0.000
     O   0.000   1.224   0.000
     O   0.000  -1.224   0.000
     O   3.850   0.000   0.000
     O  -3.850   0.000   0.000
  end
end

Task SinglePoint

Engine ADF
  title Ti2O4 SCF aid test (Energy-DIIS)
  basis
    core Small
    type DZ
  end
  scf
    converge 1.0e-6 1.0e-6
    ediis
    iterations 300
    mixing 0.05
  end
  xc
    gga Becke Perdew
  end
EndEngine
eor
```

Example: Rescan frequencies: NH3

Download Freq_NH3_Scan.run

```
#! /bin/sh

# Sometimes spurious imaginary frequencies are calculated where one would expect
# a very low (nearly zero) frequency. Most frequently this happens when there is
# a barrier-free rotation of, for example, methyl groups. The VibrationalAnalysis
# task (using ModeRefinement) allows one to rescan calculated frequencies in order
# to find out if they were calculated accurately.

# In this example analytical frequencies are calculated. Next recalculation of
# certain NH3 frequencies are performed by scanning along normal modes from a
# restart file. In this calculation the frequencies are calculated numerically
# with finite displacements using symmetry.

AMS_JOBNAME=NH3 $AMSBIN/ams <<eor
System
  Symmetrize Yes
  atoms
     N                0.0000    0.0000    0.0000
     H                0.4729    0.8190    0.3821
     H               -0.9457    0.0000    0.3821
     H                0.4729   -0.8190    0.3821
  end
end

Task SinglePoint

Properties
  NormalModes Yes
End

Engine ADF
  title NH3 analytic frequencies
  beckegrid quality=Good
  basis
    type TZP
    core Small
  end
EndEngine
eor

AMS_JOBNAME=ScanFreq $AMSBIN/ams <<eor
LoadSystem
  File NH3.results/adf.rkf
End
LoadEngine NH3.results/adf.rkf

Task VibrationalAnalysis

VibrationalAnalysis
    Type ModeRefinement
    NormalModes
       ModeFile NH3.results/adf.rkf
       ModeSelect
          FreqRange 0 4000
```

(continues on next page)

(continued from previous page)

```
        End
    End
End
eor
```

14.7.2 Speed

Example: Multiresolution

Download Multiresolution_H2O.run

```
#! /bin/sh

# This example demonstrates how to use different levels of numerical precision
# for different regions, with the aim of increasing computational efficiency.

# Let us assume that we are interested in having an accurate description only
# for a subregion of a large chemical system (in this simple example, the
# central water molecule). The system can be divided into sub-regions and
# different levels of numerical accuracy can be used for each of these sub-
# regions.

# In this example we will tweak:
# - the basis set (Basis)
# - the numerical integration (BeckeGrid)
# - the density fitting for Coulomb potential (ZlmFit)
# - the fit-set used in the Hartree-Fock Resolution of identity (RIHartreeFock)

# Note: For the regions for which no quality has been explicitly defined
# through a QualityPerRegion keyword, the quality defined in NumericalQuality
# will be used (Normal, in this example).


$AMSBIN/ams <<eor
Task SinglePoint

System
  Atoms
     O    0.00000000    0.00000000    0.00000000   region=Accurate
     H    0.40399229   -0.65578342    0.63241539   region=Accurate
     H    0.81410032    0.55624569   -0.23129097   region=Accurate
     O   -3.02535626   -0.08473104   -0.47678489
     H   -2.56531481    0.62644005    0.07759386
     H   -2.25289644   -0.61700366   -0.80790649
     O    2.95394790   -0.54939973   -0.38206034
     H    3.91427727   -0.21304908   -0.44738291
     H    2.87780992   -1.13241278   -1.20853726
     O   -5.95425742   -0.56764616   -0.02016682   region=Far
     H   -5.26308282   -0.46969096    0.69255963   region=Far
     H   -5.42117992   -0.54361203   -0.86443121   region=Far
     O    6.25171470   -0.62004899   -0.03702467   region=Far
     H    6.16508647   -1.38696453    0.58541903   region=Far
     H    7.09161199   -0.16700550    0.23679419   region=Far
  End
```

(continues on next page)

(continued from previous page)

```
End

Engine ADF

   XC
      LibXC B3LYP
   End

   ! ========================================================
   ! Set different basis sets for atoms in different regions:
   ! ========================================================

   Basis
      Type DZ
      Core None
      PerRegion Region=Accurate Type=TZP
      PerRegion Region=Far      Type=SZ
   End

   ! =============================================================
   ! Set the NumerialQuality to be used for the atoms that are not
   ! explicitly defined through a QualityPerRegion keyword
   ! =============================================================

   NumericalQuality Normal

   # ======================
   # Numerical integration:
   # ======================

   BeckeGrid
      QualityPerRegion Region=Accurate Quality=Good
      QualityPerRegion Region=Far      Quality=Basic
   End

  # ======================================
  # Density fitting for Coulomb potential:
  # ======================================

   ZlmFit
      QualityPerRegion Region=Accurate Quality=Good
      QualityPerRegion Region=Far      Quality=Basic
   End

  # ============================================================
  # Hartree-Fock Resolution of identity (for hybrid functionals)
  # ============================================================

   RIHartreeFock
      QualityPerRegion Region=Accurate Quality=Good
      QualityPerRegion Region=Far      Quality=Basic
   End

EndEngine

eor
```

14.8 Restarts

14.8.1 Example: Restart analytical Hessian calculation

Download HessianRestart.run

```
#!/bin/sh

# This example demonstrates how one can complete an analytical Hessian calculation
# in three steps without re-doing CPKS for all atoms and the subsequent steps.
# We will define regions for the first and second steps right away.
# Note that the region for the second step includes atoms of the first one.
# This is necessary because we want the resulting Hessian in the subsequent steps
# to include the atoms done in the previous calculations.
# First, we calculate the Hessian sub-matrix corresponding to the C atoms.

AMS_JOBNAME=step1 "$AMSBIN/ams" << eor

Task SinglePoint
Properties
    NormalModes Yes
    SelectedRegionForHessian step1
End

System
    Atoms
        C -2.711485799381603 -2.593393872048840 -0.00016442137955 region=step2,step1
        C -1.846368848888818 -1.340467404429865  0.00016453093935 region=step2,step1
        H -2.203636997435432 -3.429266061978078 -0.51549899454129 region=step2
        H -3.673159561801013 -2.414816382275922 -0.51564516686341 region=step2
        H -2.938101685663578 -2.921829124623212  1.03094752484358 region=step2
        H -0.884694282363284 -1.519047266416641  0.51564280107136
        H -2.354216233206031 -0.504595127574485  0.51550071600252
        H -1.619753333793463 -1.012031055604717 -1.03094699007257
    End
    BondOrders
         1 2 1.0
         1 3 1.0
         1 4 1.0
         1 5 1.0
         2 6 1.0
         2 7 1.0
         2 8 1.0
    End
End

Engine ADF
 save TAPE13
EndEngine
eor

# In the second calculation, we add the Hessian sub-matrix corresponding to H atoms
# of the first methyl group. We use as much data as possible from the first one.
# This includes the part of the analytical Hessian computed for atoms of the "step1"
# region in the previous calculation.
# We use the adf.rkf file for restart but if the 1st calculation had crashed or
# stopped uncontrollably, we would have to use the TAPE13 file instead.
```

(continues on next page)

(continued from previous page)

```
#

AMS_JOBNAME=step2 "$AMSBIN/ams" << eor

Task SinglePoint

Properties
    NormalModes Yes
    SelectedRegionForHessian step2
End

LoadSystem
    File step1.results/ams.rkf
End

LoadEngine step1.results/adf.rkf
EngineRestart step1.results/adf.rkf
eor

# In the third calculation, we do not use the SelectedRegionForHessian key, because
# we want the full Hessian. In this calculation we use data from the second step.

AMS_JOBNAME=step3 "$AMSBIN/ams" << eor

Task SinglePoint

Properties
    NormalModes Yes
End

LoadSystem
    File step2.results/ams.rkf
End

LoadEngine step2.results/adf.rkf
EngineRestart step2.results/adf.rkf
eor
```

14.9 Scripting

14.9.1 Prepare an AMS job and generate a report

Example: Single point for multiple xyz files: Bakerset

Download BakersetSP.run

```
#! /bin/sh


# In this example you will find how to use amsprep to run a particular job (a
# single point calculation in this case) for all molecules in the Baker set. The
# molecules are simply xyz files and contain no ADF specific information.
# amsreport is used to collect the resulting bonding energies.
```

(continues on next page)

(continued from previous page)

```
rm -f runset
for f in $AMSHOME/examples/adf/BakersetSP/Bakerset/*.xyz
do
    "$AMSBIN/amsprep" -t SP -i 2.5 -b DZ -c Large -r None -m "$f" -j `basename $f .
↪xyz`>> runset
done

chmod +x runset
./runset

echo Results
ls -t -1 *.results/adf.rkf | while read f
do
    "$AMSBIN/amsreport" "$f" BondingEnergy
done
echo Ready
```

Example: Basis set and integration accuracy convergence test: Methane

Download ConvergenceTestCH4.run

```
#! /bin/sh


# In this example you will find how to use amsprep to test convergence of the
# bonding energy with respect to basis set and integration accuracy. amsreport
# is used to collect the resulting bonding energies.

rm -f runset
for b in SZ DZ DZP TZP TZ2P QZ4P
do
    "$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b $b␣
↪-j methane.$b>> runset
done

chmod +x runset
./runset

echo Results
echo Basis set convergence of Bonding Energy, SZ DZ DZP TZP TZ2P QZ4P
for b in SZ DZ DZP TZP TZ2P QZ4P
do
    "$AMSBIN/amsreport" "methane.$b.results/adf.rkf" BondingEnergy
done

rm -f runset
for i in 2 3 4 5
do
    "$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b␣
↪DZP -i $i -j methane.$i>> runset
done
"$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b DZP -i␣
↪Basic -j methane.bb>> runset
```

(continues on next page)

(continued from previous page)

```
"$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b DZP -i␣
↪Normal -j methane.bn>> runset
"$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b DZP -i␣
↪Good -j methane.bg>> runset
"$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b DZP -i␣
↪VeryGood -j methane.bv>> runset
"$AMSBIN/amsprep" -t "$AMSHOME/examples/adf/ConvergenceTestCH4/methane.ams" -b DZP -i␣
↪Excellent -j methane.be>> runset

chmod +x runset
./runset

echo Integration convergence of Bonding Energy, 2 3 4 5
for i in 2 3 4 5
do
    "$AMSBIN/amsreport" "methane.$i.results/adf.rkf" BondingEnergy
done

echo Integration Becke convergence of Bonding Energy, Basic, Normal, Good, VeryGood,␣
↪Excellent
"$AMSBIN/amsreport" "methane.bb.results/adf.rkf" BondingEnergy
"$AMSBIN/amsreport" "methane.bn.results/adf.rkf" BondingEnergy
"$AMSBIN/amsreport" "methane.bg.results/adf.rkf" BondingEnergy
"$AMSBIN/amsreport" "methane.bv.results/adf.rkf" BondingEnergy
"$AMSBIN/amsreport" "methane.be.results/adf.rkf" BondingEnergy

echo Ready
```

Example: amsprep: Replace atom with ligand

Download LoopAtomsLigands.run

```
#! /bin/sh

# generate a run file for methane, with 1 H replaced by some ligand (run amsprep
# -liststructures to get full list of available structures) note the -a flag
# makes amsprep produce an .ams file as well, which can be opened by AMSinput to
# check your system

for ligand in CN CO CO3 NC NH2 NH2CH3 NH3 OC OCH3 OH PH3 Pyridine; do
    "$AMSBIN/amsprep" -t GO -m "$AMSHOME/examples/adf/LoopAtomsLigands/methane.xyz" -
↪structure "2 Ligands/$ligand.ams" -a methane.$ligand.ams
done

# similar, replace the C atom by other atoms

for atom in C Si Ge Sn Pb; do
    "$AMSBIN/amsprep" -t GO -m "$AMSHOME/examples/adf/LoopAtomsLigands/methane.xyz" -
↪atomtype "1 $atom" -a methane.$atom.ams
done
```

14.10 List of Examples

- *FQQM_FQFMU_response* (page 635)
- *FQQM_FQFMU_scf* (page 583)
- *FQQM_FQFMU_td* (page 661)
- *FQQM_FDE* (page 687)
- *Frags_NiCO4* (page 986)
- *Frags_PtCl4H2* (page 989)
- *FranckCondon_NO2* (page 820)
- *Freq_NH3* (page 742)
- *Freq_NH3_RAMAN* (page 753)
- *Freq_NH3_Scan* (page 1078)
- *Freq_UF6* (page 745)
- *G3W2_H2O* (page 918)
- *GO_FDE_H2O-Li* (page 483)
- *GO_H2O* (page 702)
- *GO_LiF_Efield* (page 711)
- *GO_constraints* (page 705)
- *GO_restraint* (page 703)
- *green_Al* (page 964)
- *green_Au* (page 967)
- *green_BDT* (page 979)
- *green_WBL* (page 973)
- *GW_CH2O_Embed* (page 919)
- *GW_H2O* (page 912)
- *GW_O3* (page 913)
- *GW_Phenol* (page 914)
- *H2O_ADFNBO* (page 1052)
- *H2O_HF_freq* (page 747)
- *H2O_MCD* (page 797)
- *H2O_MCD_ZFS* (page 798)
- *H2O_SO_HDA* (page 802)
- *H2O_TD_magnet* (page 856)
- *H2O_Verdet* (page 831)
- *H2O_magnet* (page 855)
- *H2PO_B3LYP* (page 415)
- *HBr* (page 871)
- *HCN* (page 712)
- *HessianRestart* (page 1081)
- *HF_ResonanceRaman* (page 760)
- *HgI2* (page 442)
- *Hg_Mossbauer_X2C* (page 909)
- *HgMeBr_pnr* (page 874)
- *HgMeBr_psc* (page 874)
- *HgMeBr_zso* (page 874)
- *HI_EFG* (page 412)
- *HI_Raman* (page 756)
- *HI_SecDer_ZORA* (page 750)
- *Hplus_CO_etsnocv* (page 1034)
- *Hyperpol* (page 852)
- *I2_SO_MP2* (page 444)
- *InitHess* (page 708)
- *IQA* (page 1045)
- *LCY-BP_H2O* (page 416)
- *LiH_DampedBeta* (page 826)
- *LiH_DampedGamma* (page 828)
- *LoCOSMO_H2O-NH3* (page 506)
- *LoCOSMO_H2O-NH3_TDDFT* (page 508)
- *LocMO_Resp* (page 1026)
- *LoopAtomsLigands* (page 1084)
- *MBH_CH4* (page 752)
- *MBH_Ethanol* (page 751)
- *methane_dimer_dispersion* (page 430)
- *MM_Dispersion* (page 426)
- *ModStPot_N2+* (page 729)
- *Mossbauer* (page 906)
- *MP2_Li* (page 424)
- *Multiresolution_H2O* (page 1079)
- *N2_TDHF* (page 772)
- *NaCl_ionicbonding* (page 1003)
- *NaCl_pairbonding* (page 1005)
- *Naphthalene_VST* (page 822)
- *Ne_CoreExci* (page 787)
- *Ne_exciso* (page 787)
- *NH_ZFS* (page 904)
- *NH3_HDA* (page 801)
- *NH3_POLTDDFT* (page 859)
- *NMR_B3LYP* (page 879)
- *NMR_NICS* (page 877)
- *NOCV_CrCO5-CH2* (page 1036)
- *O2_ROKS* (page 740)
- *O2_pNMR* (page 882)
- *OH_MetaGGA* (page 405)
- *PCCP_Unr_BondEnergy* (page 997)
- *PbF_excisou* (page 792)
- *PbH4_finitenuc* (page 889)
- *pdb2adf* (page 510)
- *PdH_ESR_X2C* (page 901)
- *Phenylnitrene_ZFS* (page 905)
- *Pr_LFDFT* (page 863)
- *PyFrag* (page 1024)
- *Pyridine_CAM-B3LYP* (page 775)
- *QM/MM* (page 510)
- *qsGW_Br2_frozenCore* (page 917)
- *qsGW_N2* (page 916)
- *Quild* (page 510)
- *r2SCAN-3c* (page 403)
- *RPA_H2O* (page 424)
- *RPA-SOSSX_H2O* (page 425)
- *SCF_Ti2O4* (page 1070)
- *SD_CrNH3_6* (page 722)
- *SF_ROKS* (page 740)
- *SiH2_spinflip* (page 770)
- *SO_ZORA_Bi2* (page 433)
- *Solv_HCl* (page 445)
- *SUBEXCI_dimer* (page 493)
- *TCNE-Benzene_wB97* (page 809)
- *TD-DFTB_betacarotene* (page 803)
- *TiCl4_CoreExci* (page 778)
- *TiCl4_XASXES* (page 784)

CHAPTER FIFTEEN

REQUIRED CITATIONS

When you publish results in the scientific literature that were obtained with programs of the ADF package, you are required to include references to the program package with the appropriate release number, and a few key publications.

In addition to these general references, references to special features are mandatory, in case you have used them.

15.1 General References

For calculations with the molecular ADF program, version 2024.1: G. te Velde, F.M. Bickelhaupt, E.J. Baerends, C. Fonseca Guerra, S.J.A. van Gisbergen, J.G. Snijders and T. Ziegler, *Chemistry with ADF*, Journal of Computational Chemistry 22, 931 (2001) (https://doi.org/10.1002/jcc.1056) (`bibtex`)

ADF 2024.1, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, http://www.scm.com. Optionally, you may add the following list of authors and contributors: E.J. Baerends, T. Ziegler, A.J. Atkins, J. Autschbach, O. Baseggio, D. Bashford, A. Bérces, F.M. Bickelhaupt, C. Bo, P.M. Boerrigter, C. Cappelli, L. Cavallo, C. Daul, D.P. Chong, D.V. Chulhai, L. Deng, R.M. Dickson, J.M. Dieterich, F. Egidi, D.E. Ellis, M. van Faassen, L. Fan, T.H. Fischer, A. Förster, C. Fonseca Guerra, M. Franchini, A. Ghysels, A. Giammona, S.J.A. van Gisbergen, A. Goez, A.W. Götz, J.A. Groeneveld, O.V. Gritsenko, M. Grüning, S. Gusarov, F.E. Harris, P. van den Hoek, Z. Hu, C.R. Jacob, H. Jacobsen, L. Jensen, L. Joubert, J.W. Kaminski, G. van Kessel, C. König, F. Kootstra, A. Kovalenko, M.V. Krykunov, P. Lafiosca, E. van Lenthe, D.A. McCormack, M. Medves, A. Michalak, M. Mitoraj, S.M. Morton, J. Neugebauer, V.P. Nicu, L. Noodleman, V.P. Osinga, S. Patchkovskii, M. Pavanello, C.A. Peeples, P.H.T. Philipsen, D. Post, C.C. Pye, H. Ramanantoanina, P. Ramos, W. Ravenek, M. Reimann, J.I. Rodríguez, P. Ros, R. Rüger, P.R.T. Schipper, D. Schlüns, H. van Schoot, G. Schreckenbach, J.S. Seldenthuis, M. Seth, J.G. Snijders, M. Solà, M. Stener, M. Swart, D. Swerhone, V. Tognetti, G. te Velde, P. Vernooijs, L. Versluis, L. Visscher, O. Visser, F. Wang, T.A. Wesolowski, E.M. van Wezenbeek, G. Wiesenekker, S.K. Wolff, T.K. Woo, A.L. Yakovlev

Note: if you have used a modified (by yourself, for instance) version of the code, you should mention in the citation that a modified version has been used.

15.2 Feature References

When you have used special features, you should include one (or more, as the case may be) lead reference(s) to the implementation. Additional references to related publications are suggested.

15.2.1 Coordinates, basis sets, fragments

Basis Sets E. van Lenthe and E.J. Baerends, *Optimized Slater-type basis sets for the elements 1-118*, Journal of Computational Chemistry 24, 1142 (2003) (https://doi.org/10.1002/jcc.10255)

Correlation-consistent basis sets Arno Förster, Lucas Visscher, *GW100: A Slater-Type Orbital Perspective*, Journal of Chemical Theory and Computation 17 (8), 5080–5097 (2021) (https://doi.org/10.1021/acs.jctc.1c00308)

Spherical Gaussian nuclear charge distribution model J. Autschbach, *Magnitude of Finite-Nucleus-Size Effects in Relativistic Density Functional Computations of Indirect NMR Nuclear Spin-Spin Coupling Constants*, ChemPhysChem 10, 2274 (2009) (https://doi.org/10.1002/cphc.200900271)

15.2.2 Geometry optimizations, transition states, and reaction paths

Transition State search L. Versluis and T. Ziegler, *The determination of Molecular Structure by Density Functional Theory*, Journal of Chemical Physics 88, 322 (1988) (https://doi.org/10.1063/1.454603)

L. Fan and T. Ziegler, *Nonlocal density functional theory as a practical tool in calculations on transition states and activation energies*, Journal of the American Chemical Society 114, 10890 (1992) (https://doi.org/10.1021/ja00053a027)

IRC L. Deng, T. Ziegler and L. Fan, *A combined density functional and intrinsic reaction coordinate study on the ground state energy surface of H2 CO*, Journal of Chemical Physics 99, 3823 (1993) (https://doi.org/10.1063/1.466129)

L. Deng and T. Ziegler, *The determination of Intrinsic Reaction Coordinates by density functional theory*, International Journal of Quantum Chemistry 52, 731 (1994) (https://doi.org/10.1002/qua.560520406)

Nudged Elastic Band G. Henkelman, B.P. Uberuaga and H. Jónsson, *A climbing image nudged elastic band method for finding saddle points and minimum energy paths*, Journal of Chemical Physics 113, 9901 (2000) (https://doi.org/10.1063/1.1329672)

15.2.3 Model Hamiltonians

Range Separated Functionals M. Seth and T. Ziegler, *Range-Separated Exchange Functionals with Slater-Type Functions*, Journal of Chemical Theory and Computation 8, 901 (2012) (https://doi.org/10.1021/ct300006h)

MP2, double hybrids A. Förster, M. Franchini, E. van Lenthe, L. Visscher, *A Quadratic Pair Atomic Resolution of the Identity Based SOS-AO-MP2 Algorithm Using Slater Type Orbitals*, Journal of Chemical Theory and Computation 16, 875 (2020) (https://doi.org/10.1021/acs.jctc.9b00854)

double hybrids A. Förster, L. Visscher, *Double hybrid DFT calculations with Slater type orbitals*, Journal of Computational Chemistry 41 1660–1684 (2020) (https://doi.org/10.1002/jcc.26209)

RPA+SOSSX A. Förster, L. Visscher, *Exploring the statically screened $G3W2$ correction to the GW self-energy : Charged excitations and total energies of finite systems*, Physical Review B 105, 125121 (2022) (https://journals.aps.org/prb/abstract/10.1103/PhysRevB.105.125121)

RPA, RPA+SOSEX, RPA+SOSSX A. Förster, *Assessment of the Second-Order Statically Screened Exchange Correction to the Random Phase Approximation for Correlation Energies*, Journal of Chemical Theory and Computation 18, 5948 (2022) (https://doi.org/10.1021/acs.jctc.2c00366)

15.2.4 Relativistic Effects

ZORA E. van Lenthe, E.J. Baerends and J.G. Snijders, *Relativistic regular two-component Hamiltonians*, Journal of Chemical Physics 99, 4597 (1993) (https://doi.org/10.1063/1.466059)

E. van Lenthe, E.J. Baerends and J.G. Snijders, *Relativistic total energy using regular approximations*, Journal of Chemical Physics 101, 9783 (1994) (https://doi.org/10.1063/1.467943)

E. van Lenthe, A.E. Ehlers and E.J. Baerends, *Geometry optimization in the Zero Order Regular Approximation for relativistic effects*, Journal of Chemical Physics 110, 8943 (1999) (https://doi.org/10.1063/1.478813)

ZORA suggested E. van Lenthe, J.G. Snijders and E.J. Baerends, *The zero-order regular approximation for relativistic effects: The effect of spin.orbit coupling in closed shell molecules*, Journal of Chemical Physics 105, 6505 (1996) (https://doi.org/10.1063/1.472460)

E. van Lenthe, R. van Leeuwen, E.J. Baerends and J.G. Snijders, *Relativistic regular two-component Hamiltonians*, International Journal of Quantum Chemistry 57, 281 (1996) (https://doi.org/10.1002/(SICI)1097-461X(1996)57:3%3C281::AID-QUA2%3E3.0.CO;2-U)

Pauli J.G. Snijders, E.J. Baerends and P. Ros, *A perturbation theory approach to relativistic calculations. II. Molecules*, Molecular Physics 38, 1909 (1979) (https://doi.org/10.1080/00268977900102941)

P.M. Boerrigter, E.J. Baerends and J.G. Snijders, *A relativistic LCAO Hartree-Fock-Slater investigation of the electronic structure of the actinocenes M(COT)2 , M=Th, Pa, U, Np and Pu*, Chemical Physics 122, 357 (1988) (https://doi.org/10.1016/0301-0104(88)80018-1)

T. Ziegler, V. Tschinke, E.J. Baerends, J.G. Snijders and W. Ravenek, *Calculation of bond energies in compounds of heavy elements by a quasi-relativistic approach*, Journal of Physical Chemistry 93, 3050 (1989) (https://doi.org/10.1021/j100345a036)

15.2.5 Solvents and other environments

COSMO: Conductor like Screening Model C.C. Pye and T. Ziegler, *An implementation of the conductor-like screening model of solvation within the Amsterdam density functional package*, Theoretical Chemistry Accounts 101, 396 (1999) (https://doi.org/10.1007/s002140050457)

SM12: Solvation Model 12 C.A. Peeples and G. Schreckenbach, *Implementation of the SM12 Solvation Model into ADF and Comparison with COSMO*, Journal of Chemical Theory and Computation 12, 4033 (2016) (https://doi.org/10.1021/acs.jctc.6b00410)

QM/MM: Quantum mechanical and Molecular Mechanics model T. K. Woo, L. Cavallo and T. Ziegler, *Implementation of the IMOMM methodology for performing combined QM/MM molecular dynamics simulations and frequency calculations*, Theoretical Chemistry Accounts 100, 307 (1998) (https://doi.org/10.1007/s002140050391)

QM/MM suggested T. K. Woo, S. Patchkovskii, and T. Ziegler, *Atomic Scale Modeling of Polymerization Catalysts*, Computing in Science & Engineering, 2, 28-37 (2000) (https://doi.org/10.1109/5992.881705)

QM/MM AddRemove model M. Swart, *AddRemove: A new link model for use in QM/MM studies*, International Journal of Quantum Chemistry 91, 177 (2003) (https://doi.org/10.1002/qua.10463)

FDE: Frozen Density Embedding T.A. Wesolowski and A. Warshel, *Frozen Density Functional Approach for ab-initio Calculations of Solvated Molecules*, Journal of Physical Chemistry 97, 8050 (1993) (https://doi.org/10.1021/j100132a040)

J. Neugebauer, C.R. Jacob, T.A. Wesolowski and E.J. Baerends, *An Explicit Quantum Chemical Method for Modeling Large Solvation Shells Applied to Aminocoumarin C151* Journal of Physical Chemistry A 109, 7805 (2005) (https://doi.org/10.1021/jp0528764)

C.R. Jacob, J. Neugebauer and L. Visscher, *A flexible implementation of frozen-density embedding for use in multilevel simulations*, Journal of Computational Chemistry 29, 1011 (2008) (https://doi.org/10.1002/jcc.20861)

DIM/QM: Discrete Interaction Model/Quantum Mechanics J.L. Payton, S.M. Morton, Justin E. Moore, and Lasse Jensen, *A discrete interaction model/quantum mechanical method for simulating surface-enhanced Raman spectroscopy*, Journal of Chemical Physics 136, 214103 (2012) (https://doi.org/10.1063/1.4722755)

DRF: Discrete Solvent Reaction Field model L. Jensen, P.T. van Duijnen and J.G. Snijders, *A discrete solvent reaction field model within density functional theory* Journal of Chemical Physics 118, 514 (2003) (https://doi.org/10.1063/1.1527010)

QM/FQ: Quantum Mechanics/Fluctuating Charges T. Giovannini, F. Egidi and C. Cappelli, *Molecular spectroscopy of aqueous solutions: a theoretical perspective*, Chemical Society Reviews, 49, 5664 (2020) (https://doi.org/10.1039/C9CS00464E)

3D-RISM: Three-Dimensional Reference Interaction Site Model S. Gusarov, T. Ziegler, and A. Kovalenko, *Self-Consistent Combination of the Three-Dimensional RISM Theory of Molecular Solvation with Analytical Gradients and the Amsterdam Density Functional Package*, Journal of Physical Chemistry A 110, 6083 (2006) (https://doi.org/10.1021/jp054344t)

3D-RISM suggested A. Kovalenko and F. Hirata, *Self-consistent description of a metal-water interface by the Kohn-Sham density functional theory and the three-dimensional reference interaction site model*, Journal of Chemical Physics 110, 10095 (1999) (https://doi.org/10.1063/1.478883)

A. Kovalenko, *Three-dimensional RISM theory for molecular liquids and solid-liquid interfaces*, In Molecular Theory of Solvation; Hirata, Fumio, Ed.; Understanding Chemical Reactivity (series); Mezey, Paul G., Series Ed.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2003; Vol. 24, pp 169-275.

15.2.6 Frequencies, IR Intensities, Raman, VCD

Numerical Differentiation of Gradients L. Fan and T. Ziegler, *Application of density functional theory to infrared absorption intensity calculations on main group molecules*, Journal of Chemical Physics 96, 9005 (1992) (https://doi.org/10.1063/1.462258)

L. Fan and T. Ziegler, *Application of density functional theory to infrared absorption intensity calculations on transition-metal carbonyls*, Journal of Physical Chemistry 96, 6937 (1992) (https://doi.org/10.1021/j100196a016)

Analytical Second Derivatives A. Bérces, R. M. Dickson, L. Fan, H. Jacobsen, D. Swerhone and T. Ziegler, *An implementation of the coupled perturbed Kohn-Sham equations: perturbation due to nuclear displacements*, Computer Physics Communications 100, 247 (1997) (https://doi.org/10.1016/S0010-4655(96)00120-8)

H. Jacobsen, A. Bérces, D. Swerhone and T. Ziegler, *Analytic second derivatives of molecular energies: a density functional implementation*, Computer Physics Communications 100, 263 (1997) (https://doi.org/10.1016/S0010-4655(96)00119-1)

S. K. Wolff, *Analytical second derivatives in the Amsterdam density functional package*, International Journal of Quantum Chemistry 104, 645 (2005) (https://doi.org/10.1002/qua.20653)

Mobile Block Hessian (MBH) A. Ghysels, D. Van Neck, V. Van Speybroeck, T. Verstraelen and M. Waroquier, *Vibrational Modes in partially optimized molecular systems* Journal of Chemical Physics 126, 224102 (2007) (https://doi.org/10.1063/1.2737444)

MBH suggested A. Ghysels, D. Van Neck and M. Waroquier, *Cartesian formulation of the Mobile Block Hessian Approach to vibrational analysis in partially optimized systems* Journal of Chemical Physics 127, 164108 (2007) (https://doi.org/10.1063/1.2789429)

Raman scattering S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Application of time-dependent density functional response theory to Raman scattering*, Chemical Physics Letters 259, 599 (1996) (https://doi.org/10.1016/0009-2614(96)00858-5)

S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Implementation of time-dependent density functional response*

equations, Computer Physics Communications 118, 119 (1999) (https://doi.org/10.1016/S0010-4655(99)00187-3)

Resonance Raman: excited-state finite lifetime L. Jensen, L. Zhao, J. Autschbach and G.C. Schatz, *Theory and method for calculating resonance Raman scattering from resonance polarizability derivatives*, Journal of Chemical Physics 123, 174110 (2005) (https://doi.org/10.1063/1.2046670)

Resonance Raman: excited-state gradient J. Neugebauer, E.J. Baerends, E. Efremov, F. Ariese and C. Gooijer, *Combined Theoretical and Experimental Deep-UV Resonance Raman Studies of Substituted Pyrenes*, Journal of Physical Chemistry A 109, 2100 (2005) (https://doi.org/10.1021/jp045360d)

VROA: (Resonance) vibrational Raman optical activity L. Jensen, J. Autschbach, M. Krykunov, and G.C. Schatz, *Resonance vibrational Raman optical activity: A time-dependent density functional theory approach*, Journal of Chemical Physics 127, 134101 (2007) (https://doi.org/10.1063/1.2768533)

Vibrational Circular Dichroism (VCD) V.P. Nicu J. Neugebauer S.K. Wolff and E.J. Baerends, *A vibrational circular dichroism implementation within a Slater-type-orbital based density functional framework and its application to hexa- and hepta-helicenes*, Theoretical Chemical Accounts 119, 245 (2008) (https://doi.org/10.1007/s00214-006-0234-x)

VCD analysis: VCDtools V.P. Nicu, J. Neugebauer and E.J. Baerends, *Effects of Complex Formation on Vibrational Circular Dichroism Spectra*, Journal of Physical Chemistry A 112, 6978 (2008) (https://doi.org/10.1021/jp710201q)

M.A.J. Koenis, O. Visser, L. Visscher, W.J. Buma, V.P. Nicu, *GUI Implementation of VCDtools, A Program to Analyze Computed Vibrational Circular Dichroism Spectra*, J. Chem. Inf. Model 60, 259 (2020) (https://pubs.acs.org/doi/abs/10.1021/acs.jcim.9b00956)

V.P. Nicu, *Revisiting an old concept: the coupled oscillator model for VCD. Part 1: the generalised coupled oscillator mechanism and its intrinsic connection to the strength of VCD signals*, Physical Chemistry Chemical Physics 18, 21202 (2016) (https://doi.org/10.1039/C6CP01282E).

Franck-Condon factors J.S. Seldenthuis, H.S.J. van der Zant, M.A. Ratner and J.M. Thijssen, *Vibrational Excitations in Weakly Coupled Single-Molecule Junctions: A Computational Analysis*, ACS Nano 2, 1445 (2008) (https://doi.org/10.1021/nn800170h)

15.2.7 Time-Dependent DFT

For all Time-Dependent DFT features (Excitation Energies, (Hyper) Polarizabilities, Dispersion Coefficients, Raman Scattering, S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Implementation of time-dependent density functional response equations*, Computer Physics Communications 118, 119 (1999) (https://doi.org/10.1016/S0010-4655(99)00187-3)

Excitation Energies and Oscillator Strengths

Lead reference S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Implementation of time-dependent density functional response equations*, Computer Physics Communications 118, 119 (1999) (https://doi.org/10.1016/S0010-4655(99)00187-3)

Suggested (when ZORA relativistic results are used) A. Rosa, E.J. Baerends, S.J.A. van Gisbergen, E. van Lenthe, J.A. Groeneveld and J. G. Snijders, *Article Electronic Spectra of M(CO)6 (M = Cr, Mo, W) Revisited by a Relativistic TDDFT Approach*, Journal of the American Chemical Society 121, 10356 (1999) (https://doi.org/10.1021/ja990747t)

Open Shell ground state F. Wang and T. Ziegler, Mol. Phys.102, 2585 (2004)

Spin-flip transitions F. Wang and T. Ziegler, *Time-dependent density functional theory based on a noncollinear formulation of the exchange-correlation potential*, Journal of Chemical Physics 121, 12191 (2004) (https://doi.org/10.1063/1.1821494)

F. Wang and T. Ziegler, *The performance of time-dependent density functional theory based on a noncollinear exchange-correlation potential in the calculations of excitation energies*, Journal of Chemical Physics 122, 74109 (2005) (https://doi.org/10.1063/1.1844299)

Core excitations M. Stener, G. Fronzoni and M. de Simone, *Time dependent density functional theory of core electrons excitations*, Chemical Physics Letters 373, 115 (2003) (https://doi.org/10.1016/S0009-2614(03)00543-8)

Quadrupole intensities S. Bernadotte, A.J. Atkins, Ch.R. Jacob, *Origin-independent calculation of quadrupole intensities in X-ray spectroscopy*, Journal of Chemical Physics 137, 204106 (2012) (https://doi.org/10.1063/1.4766359)

XES: X-ray emission spectroscopy A.J. Atkins, M. Bauer, and Ch.R. Jacob, *The chemical sensitivity of X-ray spectroscopy: high energy resolution XANES versus X-ray emission spectroscopy of substituted ferrocenes*, Physical Chemistry Chemical Physics 15, 8095 (2013) (https://doi.org/10.1039/C3CP50999K)

Excitations including spin-orbit coupling F. Wang, T. Ziegler, E. van Lenthe, S.J.A. vand Gisbergen and E.J. Baerends, *The calculation of excitation energies based on the relativistic two-component zeroth-order regular approximation and time-dependent density-functional with full use of symmetry*, Journal of Chemical Physics 122, 204103 (2005) (https://doi.org/10.1063/1.1899143)

Perturbative approach to include spin-orbit coupling F. Wang and T. Ziegler, *A simplified relativistic time-dependent density-functional theory formalism for the calculations of excitation energies including spin-orbit coupling effect*, Journal of Chemical Physics 123, 154102 (2005) (https://doi.org/10.1063/1.2061187)

CV(n)-DFT: Constricted Variational DFT J. Cullen, M. Krykunov, and T. Ziegler, *The formulation of a self-consistent constricted variational density functional theory for the description of excited states*, Chemical Physics 391, 11 (2011) (https://doi.org/10.1016/j.chemphys.2011.05.021)

M. Krykunov and T. Ziegler, *Self-consistent Formulation of Constricted Variational Density Functional Theory with Orbital Relaxation. Implementation and Applications*, Journal of Chemical Theory and Computation 9, 2761 (2013) (https://doi.org/10.1021/ct300891k)

CV(n)-DFT: triplet states Y.C. Park, F. Senn, M. Krykunov, and T. Ziegler, *Self-Consistent Constricted Variational Theory RSCF-CV(∞)-DFT and Its Restrictions To Obtain a Numerically Stable Δ SCF-DFT-like Method: Theory and Calculations for Triplet States*, Journal of Chemical Theory and Computation 12, 5438 (2016) (https://doi.org/10.1021/acs.jctc.6b00333)

TD-DFT+TB: Tight-Binding approximations to time-dependent Density Functional Theory R. Rüger, E. van Lenthe, T. Heine, L. Visscher, *Tight-Binding Approximations to Time-Dependent Density Functional Theory - a fast approach for the calculation of electronically excited states*, Journal of Chemical Physics 144, 184103 (2016) (https://doi.org/10.1063/1.4948647)

GW-BSE: GW-Bethe-Salpeter Equation Arno Förster, Lucas Visscher, *Quasiparticle Self-Consistent GW-Bethe-Salpeter equation calculations for large chromophoric systems*, Journal of Chemical Theory and Computation 18 (11) 6779–6793 (2022) (https://doi.org/10.1021/acs.jctc.2c00531)

Excited state gradients

General M. Seth, G. Mazur, and T. Ziegler, *Time-dependent density functional theory gradients in the Amsterdam density functional package: geometry optimizations of spin-flip excitations*, Theoretical Chemistry Accounts 129, 331 (2011) (https://doi.org/10.1007/s00214-010-0819-2)

TD-DFT+TB excited state gradients S. Havenridge, R. Rüger, C.M. Aikens, *Analytical excited state gradients for time-dependent density functional theory plus tight binding (TDDFT + TB)* The Journal of Chemical Physics 158, 224103 (2023) (https://doi.org/10.1063/5.0142240).

Polarizabilities

General S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *A Density Functional Theory study of frequency-dependent polarizabilities and van der Waals dispersion coefficients for polyatomic molecules*, Journal of Chemical Physics 103, 9347 (1995) (https://doi.org/10.1063/1.469994)

Suggested V.P. Osinga, S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Density functional results for isotropic and anisotropic multipole polarizabilities and C6 , C7 , and C8 Van der Waals dispersion coefficients for molecules*, Journal of Chemical Physics 106, 5091 (1997) (https://doi.org/10.1063/1.473555)

Polarizabilities including spin-orbit coupling A. Devarajan, A. Gaenko, and J. Autschbach, *Two-component relativistic density functional method for computing nonsingular complex linear response of molecules based on the zeroth order regular approximation*, Journal of Chemical Physics 130, 194102 (2009) (https://doi.org/10.1063/1.3123765)

PolTDDFT: Damped complex polarizabilities O. Baseggio, G. Fronzoni, and M. Stener, *A new time dependent density functional algorithm for large systems and plasmons in metal clusters*, Journal of Chemical Physics 143, 024106 (2015) (https://doi.org/10.1063/1.4923368)

Hyperpolarizabilities

Hyperpolarizabilities with RESPONSE S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Calculating frequency-dependent hyperpolarizabilities using time-dependent density functional theory*, Journal of Chemical Physics 109, 10644 (1998) (https://doi.org/10.1063/1.477762)

Suggested: S.J.A. van Gisbergen, J.G. Snijders, and E.J. Baerends, *Time-dependent Density Functional Results for the Dynamic Hyperpolarizability of C60*, Physical Review Letters 78, 3097 (1997) (https://doi.org/10.1103/PhysRevLett.78.3097)

First hyperpolarizabilities with AORESPONSE Z. Hu, J. Autschbach, and L. Jensen, *Simulation of resonance hyper-Rayleigh scattering of molecules and metal clusters using a time-dependent density functional theory approach*, Journal of Chemical Physics 141, 124305 (2014) (https://doi.org/10.1063/1.4895971)

Second hyperpolarizabilities with AORESPONSE Z. Hu, J. Autschbach, and L. Jensen, *Simulating Third-Order Nonlinear Optical Properties Using Damped Cubic Response Theory within Time-Dependent Density Functional Theory*, Journal of Chemical Theory and Computation 12, 1294 (2016) (https://doi.org/10.1021/acs.jctc.5b01060)

CD, ORD, magnetizability, MCD, Verdet constants

Circular Dichroism (CD) J. Autschbach and T. Ziegler, *Calculating molecular electric and magnetic properties from time-dependent density functional response theory*, Journal of Chemical Physics 116, 891 (2002) (https://doi.org/10.1063/1.1420401)

J. Autschbach, T. Ziegler, S.J.A. van Gisbergen and E.J. Baerends, *Chiroptical properties from time-dependent density functional theory. I. Circular dichroism spectra of organic molecules*, Journal of Chemical Physics 116, 6930 (2002) (https://doi.org/10.1063/1.1436466)

Optical Rotation (OR), Optical Rotation Dispersion (ORD) J. Autschbach and T. Ziegler, *Calculating molecular electric and magnetic properties from time-dependent density functional response theory*, Journal of Chemical Physics 116, 891 (2002) (https://doi.org/10.1063/1.1420401)

J. Autschbach, S. Patchkovskii, T. Ziegler, S.J.A. van Gisbergen and E.J. Baerends, *Chiroptical properties from time-dependent density functional theory. II. Optical rotations of small to medium sized organic molecules*, Journal of Chemical Physics 117, 581 (2002) (https://doi.org/10.1063/1.1477925)

Magnetizability M. Krykunov and J. Autschbach, *Calculation of static and dynamic linear magnetic response in approximate time-dependent density functional theory*, Journal of Chemical Physics 126, 24101 (2007) (https://doi.org/10.1063/1.2423007)

Magnetic Circular Dichroism (MCD) M. Seth, M. Krykunov, T. Ziegler, J. Autschbach and A. Banerjee, *Application of magnetically perturbed time-dependent density functional theory to magnetic circular dichroism: Calculation of B terms*, Journal of Chemical Physics 128, 144105 (2008) (https://doi.org/10.1063/1.2901967)

M. Seth, M. Krykunov, T. Ziegler and J. Autschbach, *Application of magnetically perturbed time-dependent density functional theory to magnetic circular dichroism. II. Calculation of A terms*, Journal of Chemical Physics 128, 234102 (2008) (https://doi.org/10.1063/1.2933550)

M. Seth, T. Ziegler and J. Autschbach, *Application of magnetically perturbed time-dependent density functional theory to magnetic circular dichroism. III. Temperature-dependent magnetic circular dichroism induced by spin-orbit coupling*, Journal of Chemical Physics 129, 104105 (2008) (https://doi.org/10.1063/1.2976568)

Verdet constant and Faraday term M. Krykunov, A. Banerjee, T. Ziegler and J. Autschbach, *Calculation of Verdet constants with time-dependent density functional theory. Implementation and results for small molecules*, Journal of Chemical Physics 122, 074105 (2005) (https://doi.org/10.1063/1.1850919)

M. Krykunov, M. Seth, T. Ziegler and J. Autschbach, *Calculation of the magnetic circular dichroism B term from the imaginary part of the Verdet constant using damped time-dependent density functional theory*, Journal of Chemical Physics 127, 244102 (2007) (https://doi.org/10.1063/1.2806990)

Dispersion Coefficients

Lead reference V.P. Osinga, S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *Density functional results for isotropic and anisotropic multipole polarizabilities and C6 , C7 , and C8 Van der Waals dispersion coefficients for molecules*, Journal of Chemical Physics 106, 5091 (1997) (https://doi.org/10.1063/1.473555)

Suggested S.J.A. van Gisbergen, J.G. Snijders and E.J. Baerends, *A Density Functional Theory study of frequency-dependent polarizabilities and van der Waals dispersion coefficients for polyatomic molecules*, Journal of Chemical Physics 103, 9347 (1995) (https://doi.org/10.1063/1.469994)

15.2.8 LFDFT

Lead reference M. Atanasov, C.A. Daul, C. Rauzy, *A DFT Based Ligand Field Theory*, Structure & Bonding 106, 97 (2004) (https://doi.org/10.1007/b11308)

LFDFT for two-open-shell configuration H. Ramanantoanina, W. Urland, F. Cimpoesu, and C. Daul, *Ligand field density functional theory calculation of the* $4f^2 \rightarrow 4f^1\ 5d^1$ *transitions in the quantum cutter* $Cs_2\ KYF_6\ Pr^{3+}$, Physical Chemistry Chemical Physics 15, 13902 (2013) (https://doi.org/10.1039/C3CP51344K).

15.2.9 NMR

NMR Chemical Shifts

Lead reference G. Schreckenbach and T. Ziegler, *The calculation of NMR shielding tensors using GIAO's and modern density functional theory*, Journal of Physical Chemistry 99, 606 (1995) (https://doi.org/10.1021/j100002a024)

NMR chemical shifts with hybrid functionals M. Krykunov, T. Ziegler and E. van Lenthe, *Hybrid density functional calculations of nuclear magnetic shieldings using Slater-type orbitals and the zeroth-order regular approximation*, International Journal of Quantum Chemistry 109, 1676 (2009) (https://doi.org/10.1002/qua.21985)

NMR chemical shifts with NBO analysis J. Autschbach and S. Zheng, *Analyzing Pt chemical shifts calculated from relativistic density functional theory using localized orbitals: The role of Pt 5d lone pairs*, Magnetic Resonance in Chemistry 46, S45 (2008) (https://doi.org/10.1002/mrc.2289)

J. Autschbach, *Analyzing NMR shielding tensors calculated with two-component relativistic methods using spin-free localized molecular orbitals*, Journal of Chemical Physics 128, 164112 (2008) (https://doi.org/10.1063/1.2905235)

Paramagnetic NMR chemical shifts J. Autschbach, S. Patchkovskii, and B. Pritchard, *Calculation of Hyperfine Tensors and Paramagnetic NMR Shifts Using the Relativistic Zeroth-Order Regular Approximation and Density Functional Theory*, Journal of Chemical Theory and Computation 7, 2175 (2011) (https://doi.org/10.1021/ct200143w)

Suggested G. Schreckenbach and T. Ziegler, *The calculation of NMR shielding tensors based on density functional theory and the frozen-core approximation*, International Journal of Quantum Chemistry 60, 753 (1996) (https://doi.org/10.1002/(SICI)1097-461X(1996)60:3%3C753::AID-QUA4%3E3.0.CO;2-W)

G. Schreckenbach and T. Ziegler, *Calculation of NMR shielding tensors based on density functional theory and a scalar relativistic Pauli-type Hamiltonian. The application to transition metal complexes*, International Journal of Quantum Chemistry 61, 899 (1997) (https://doi.org/10.1002/(SICI)1097-461X(1997)61:6%3C899::AID-QUA3%3E3.0.CO;2-R)

S.K. Wolff and T. Ziegler, *Calculation of DFT-GIAO NMR shifts with inclusion of spin-orbit coupling*, Journal of Chemical Physics 109, 895 (1998) (https://doi.org/10.1063/1.476630)

S.K. Wolff, T. Ziegler, E. van Lenthe and E.J. Baerends, *Density functional calculations of nuclear magnetic shieldings using the zeroth-order regular approximation (ZORA) for relativistic effects: ZORA nuclear magnetic resonance*, Journal of Chemical Physics 110, 7689 (1999) (https://doi.org/10.1063/1.478680)

NMR spin-spin coupling

Lead reference J. Autschbach and T. Ziegler, *Nuclear spin-spin coupling constants from regular approximate density functional calculations. I. Formalism and scalar relativistic results for heavy metal compounds*, Journal of Chemical Physics 113, 936 (2000) (https://doi.org/10.1063/1.481874)

J. Autschbach, and T. Ziegler, *Nuclear spin-spin coupling constants from regular approximate relativistic density functional calculations. II. Spin-orbit coupling effects and anisotropies*, Journal of Chemical Physics 113, 9410 (2000) (https://doi.org/10.1063/1.1321310)

NMR spin-spin couplings with PBE0 J. Autschbach, *Two-component relativistic hybrid density functional computations of nuclear spin-spin coupling tensors using Slater-type basis sets and density-fitting techniques*, Journal of Chemical Physics 129, 094105 (2008) (https://doi.org/10.1063/1.2969100), Erratum: Journal of Chemical Physics 130, 209901 (2009) (https://doi.org/10.1063/1.3131724)

NMR spin-spin couplings with NBO analysis J. Autschbach, *Analyzing molecular properties calculated with two-component relativistic methods using spin-free Natural Bond Orbitals: NMR spin-spin coupling constants* Journal of Chemical Physics 127, 124106 (2007) (https://doi.org/10.1063/1.2768363)

Suggested R.M. Dickson and T. Ziegler, *NMR Spin.Spin Coupling Constants from Density Functional Theory with Slater-Type Basis Functions*, Journal of Physical Chemistry 100, 5286 (1996) (https://doi.org/10.1021/jp951930l)

J. Khandogin and T. Ziegler, *A density functional study of nuclear magnetic resonance spin.spin coupling constants in transition-metal systems*, Spectrochimica Acta 55, 607 (1999) (https://doi.org/10.1016/S1386-1425(98)00265-0)

J. Autschbach and T. Ziegler, *Solvent Effects on Heavy Atom Nuclear Spin.Spin Coupling Constants: A Theoretical Study of Hg.C and Pt.P Couplings*, Journal of the American Chemical Society 123, 3341 (2001) (https://doi.org/10.1021/ja003481v)

J. Autschbach and T. Ziegler, *A Theoretical Investigation of the Remarkable Nuclear Spin.Spin Coupling Pattern in [(NC)5 Pt-Tl(CN)]-*, Journal of the American Chemical Society 123, 5320 (2001) (https://doi.org/10.1021/ja003866d)

Suggested book reference J. Autschbach, T. Ziegler, in *Encyclopedia of Nuclear Magnetic Resonance*, Eds. D.M. Grant, R. K. Harris, John Wiley and Sons, Chichester, 2002, Vol. 9 *Advances in NMR*.

15.2.10 ESR/EPR

G-tensor: Zeeman interaction

G-tensor: self-consistent spin-orbit coupling E. van Lenthe, A. van der Avoird and P.E.S. Wormer, *Density functional calculations of molecular g-tensors in the zero order regular approximation for relativistic effects*, Journal of Chemical Physics 107, 2488 (1997) (https://doi.org/10.1063/1.474590)

G-tensor: perturbative inclusion spin-orbit coupling J. Autschbach and B. Pritchard, *Calculation of molecular g-tensors using the zeroth-order regular approximation and density functional theory: expectation value versus linear response approaches*, Theoretical Chemistry Accounts 129, 453 (2011) (https://doi.org/10.1007/s00214-010-0880-x)

G-tensor: perturbative inclusion spin-orbit coupling with EPR/NMR program G. Schreckenbach and T. Ziegler, *Calculation of the G-tensor of electron paramagnetic resonance spectroscopy using Gauge-Including Atomic Orbitals and Density Functional Theory*, Journal of Physical Chemistry A 101, 3388 (1997) (https://doi.org/10.1021/jp963060t) (for ESR/EPR g-tensor)

S. Patchkovskii and T. Ziegler, *Calculation of the EPR g-Tensors of High-Spin Radicals with Density Functional Theory*, Journal of Physical Chemistry A 105, 5490 (2001) (https://doi.org/10.1021/jp010457a) (for high-spin ESR/EPR g-tensor)

A-tensor: Nuclear magnetic dipole hyperfine interaction

A-tensor E. van Lenthe, A. van der Avoird and P.E.S. Wormer, *Density functional calculations of molecular hyperfine interactions in the zero order regular approximation for relativistic effects*, Journal of Chemical Physics 108, 4783 (1998) (https://doi.org/10.1063/1.475889)

A-tensor: perturbative inclusion spin-orbit coupling J. Autschbach, S. Patchkovskii, and B. Pritchard, *Calculation of Hyperfine Tensors and Paramagnetic NMR Shifts Using the Relativistic Zeroth-Order Regular Approximation and Density Functional Theory*, Journal of Chemical Theory and Computation 7, 2175 (2011) (https://doi.org/10.1021/ct200143w)

15.2.11 Nuclear Quadrupole Interaction (EFG)

EFG, NQCC (in ESR called Q-tensor: Nuclear electric quadrupole hyperfine interaction) E. van Lenthe and E.J. Baerends, *Density functional calculations of nuclear quadrupole coupling constants in the zero-order regular approximation for relativistic effects*, Journal of Chemical Physics 112, 8279 (2000) (https://doi.org/10.1063/1.481433)

EFG with NBO analysis A.J. Rossini, R.W. Mills, G.A. Briscoe, E.L. Norton, S.J. Geier, I. Hung, S. Zheng, J. Autschbach, and R.W. Schurko, *Solid-State Chlorine NMR of Group IV Transition Metal Organometallic Complexes*, Journal of the American Chemical Society 131, 3317 (2009) (https://doi.org/10.1021/ja808390a)

J. Autschbach, S. Zheng, and R.W. Schurko, *Analysis of Electric Field Gradient Tensors at Quadrupolar Nuclei in Common Structural Motifs*, Concepts in Magnetic Resonance Part A 36A, 84 (2010) (https://doi.org/10.1002/cmr.a.20155)

15.2.12 GW

Lead reference Arno Förster, Lucas Visscher, *Low-order scaling G0W0 by pair atomic density fitting*, Journal of Chemical Theory and Computation 16 (12), 7381–7399 (2020) (https://doi.org/10.1021/acs.jctc.0c00693)

Suggested Arno Förster, Lucas Visscher, *GW100: A Slater-Type Orbital Perspective*, Journal of Chemical Theory and Computation 17 (8), 5080–5097 (2021) (https://doi.org/10.1021/acs.jctc.1c00308)

Quasiparticle self-consistent GW Arno Förster, Lucas Visscher, *Low-Order Scaling Quasiparticle Self-Consistent GW for Molecules*, Frontiers in Chemistry 9, 736591 (2021) (https://doi.org/10.3389/fchem.2021.736591)

G3W2 correction Arno Förster, Lucas Visscher, *Exploring the statically screened G3W2 correction to the GW self-energy : Charged excitations and total energies of finite systems*, Physical Review B 105, 125121 (2022) (https://journals.aps.org/prb/abstract/10.1103/PhysRevB.105.125121)

15.2.13 Transport properties: Non-self-consistent Green's function

Lead reference Chapter 2 and appendix C of J.S. Seldenthuis, Electrical and mechanical effects in single-molecule junctions (http://downloads.scm.com/Doc/Seldenthuis2011.pdf), PhD thesis, Delft University of Technology, 2011

Wide-band limit C.J.O. Verzijl, J.S. Seldenthuis, and J.M. Thijssen, *Applicability of the wide-band limit in DFT-based molecular transport calculations*, Journal of Chemical Physics 138, 094102 (2013) (https://doi.org/10.1063/1.4793259)

15.2.14 Analysis

Bond Energy Analysis T. Ziegler and A. Rauk, *A theoretical study of the ethylene-metal bond in complexes between* Cu^+, Ag^+, Au^+, Pt^0 or Pt^{2+} *and ethylene, based on the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1558 (1979) (https://doi.org/10.1021/ic50196a034)

T. Ziegler and A. Rauk, *Carbon monoxide, carbon monosulfide, molecular nitrogen, phosphorus trifluoride, and methyl isocyanide as sigma donors and pi acceptors. A theoretical study by the Hartree-Fock-Slater transition-state method*, Inorganic Chemistry 18, 1755 (1979) (https://doi.org/10.1021/ic50197a006)

F.M. Bickelhaupt and E.J. Baerends, In: Rev. Comput. Chem.; K.B. Lipkowitz and D.B. Boyd, Eds.; Wiley, New York, 2000, Vol. 15, p.1-86

ETS-NOCV M. Mitoraj, A. Michalak and T. Ziegler, *A Combined Charge and Energy Decomposition Scheme for Bond Analysis*, Journal of Chemical Theory and Computation 5, 962 (2009) (https://doi.org/10.1021/ct800503d)

QTAIM, Bader analysis, grid-based algorithm J.I. Rodríguez, R.F.W. Bader, P.W. Ayers, C. Michel, A.W. Götz and C. Bo, *A high performance grid-based algorithm for computing QTAIM properties*, Chemical Physics Letters 472, 149 (2009) (https://doi.org/10.1016/j.cplett.2009.02.081)

J.I. Rodríguez, *An Efficient Method for Computing the QTAIM Topology of a Scalar Field: The Electron Density Case*, Journal of Computational Chemistry 34, 681 (2013) (https://doi.org/10.1002/jcc.23180)

Localized molecular orbitals J. Autschbach and H.F. King, *Analyzing molecular static linear response properties with perturbed localized orbitals*, Journal of Chemical Physics 133, 044109 (2010) (https://doi.org/10.1063/1.3455709)

15.2.15 Accuracy and efficiency

Linear scaling methods C. Fonseca Guerra, J.G. Snijders, G. te Velde and E.J. Baerends, *Towards an order-N DFT method*, Theoretical Chemistry Accounts 99, 391 (1998) (https://doi.org/10.1007/s002140050353)

Numerical integration M. Franchini, P.H.T. Philipsen, L. Visscher, *The Becke Fuzzy Cells Integration Scheme in the Amsterdam Density Functional Program Suite*, Journal of Computational Chemistry 34, 1818 (2013) (https://doi.org/10.1002/jcc.23323).

Density fitting M. Franchini, P.H.T. Philipsen, E. van Lenthe, L. Visscher, *Accurate Coulomb Potentials for Periodic and Molecular Systems through Density Fitting*, Journal of Chemical Theory and Computation 10, 1994 (2014) (https://doi.org/10.1021/ct500172n)

15.3 External programs and Libraries

Click here for the list of programs and/or libraries used in the ADF package. On some platforms optimized libraries have been used and/or vendor specific MPI implementations.

CHAPTER SIXTEEN

KEYWORDS

16.1 Links to manual entries

- *A1FIT* (page 325)
- *ADDDIFFUSEFIT* (page 328)
- *ALLOW* (page 337)
- *ALLPOINTS* (page 340)
- *ANALYTICALFREQ* (page 152)
- *AORESPONSE* (page 191)
- *ARH* (page 319)
- *AROMATICITY* (page 269)
- *ATOMS* (page 35)
- *BASIS* (page 42)
- *BECKEGRID* (page 321)
- *BONDORDER* (page 259)
- *CDFT* (page 68)
- *CDSPECTRUM* (page 179)
- *CM5* (page 258)
- *CONCEPTUALDFT* (page 270)
- *COREPOTENTIALS* (page 62)
- *cpl* (page 217)
- *cpl GGA* (page 219)
- *cpl HYPERFINE* (page 223)
- *cpl NMRCOUPLING* (page 217)
- *CREATE* (page 47)
- *CURRENTRESPONSE* (page 158)
- *CVNDFT* (page 174)
- *DEBUG* (page 277)
- *densf* (page 291)
- *DEPENDENCY* (page 335)
- *DIMPAR* (page 118)
- *DIMQM* (page 114)
- *disper* (page 199)
- *dos* (page 309)
- *DRF* (page 114)
- *ELECTRONTRANSFER* (page 242)
- *ELECTROSTATICEMBEDDING* (page 144)
- *ENERGYFRAG* (page 96)
- *EPRINT* (page 277)
- *ESR* (page 222)
- *ETSNOCV* (page 261)
- *EXACTDENSITY* (page 326)
- *EXCITATIONS* (page 160)
- *EXCITEDGO* (page 186)
- *EXTENDEDPOPAN* (page 259)
- *EXTERNALS* (page 118)
- *FDE* (page 128)
- *FORCEALDA* (page 166)
- *QMFQ* (page 124)
- *FRAGMENTS* (page 52)
- *FRAGMETAGGATOTEN* (page 81)
- *FRAGOCCUPATIONS* (page 66)
- *FULLFOCK* (page 340)
- *green* (page 246)
- *GW* (page 233)
- *HARTREEFOCK* (page 96)
- *HFATOMSPERPASS* (page 328)
- *HFMAXMEMORY* (page 328)
- *HYPERPOL* (page 190)
- *INTEGRATION* (page 322)
- *IQA* (page 254)
- *IRREPOCCUPATIONS* (page 61)
- *lfdft* (page 202)
- *lfdft_tdm* (page 204)
- *LINEARSCALING* (page 339)
- *LOCORB* (page 255)
- *MBPT* (page 331)
- *MCD* (page 180)
- *METAGGA* (page 96)
- *MODIFYEXCITATION* (page 168)
- *MODIFYSTARTPOTENTIAL* (page 64)
- *nmr* (page 207)
- *nmr GFACTORS* (page 223)
- *nmr NMR* (page 207)
- *NOBECKEGRID* (page 321)
- *NOPRINT* (page 278)
- *NOSHAREDARRAYS* (page 338)
- *NUCLEARMODEL* (page 49)
- *NUMERICALQUALITY* (page 315)
- *OCCUPATIONS* (page 59)
- *OPTICALROTATION* (page 191)
- *POLTDDFT* (page 196)
- *PRINT* (page 274)
- *QTAIM* (page 268)
- *QTENS* (page 226)
- *RADIALCOREGRID* (page 324)
- *RELATIVITY* (page 97)
- *REMOVEALLFRAGVIRTUALS* (page 67)
- *REMOVEFRAGORBITALS* (page 67)
- *RESPONSE* (page 189)
- *RESTART* (page 344)
- *RIHARTREEFOCK* (page 326)
- *RISM* (page 139)
- *SAVE* (page 340)
- *SCF* (page 316)
- *SELECTEXCITATION* (page 167)
- *SFTDDFT* (page 166)
- *SINGLEORBTRANS* (page 169)
- *SKIP* (page 336)
- *SLATERDETERMINANTS*

(page 370)
- *SM12* (page 109)
- *SOLVATION* (page 102)
- *SOMCD* (page 180)
- *SOPERT* (page 171)
- *SPINFLIP* (page 63)
- *SPINPOLARIZATION* (page 55)
- *STCONTRIB* (page 173)
- *STOFIT* (page 325)
- *STOPAFTER* (page 336)
- *SUBEXCI* (page 136)
- *SYMMETRY* (page 37)
- *SYMMETRYTOLERANCE* (page 37)
- *TAILS* (page 338)
- *TDA* (page 162)
- *TOTALENERGY* (page 252)
- *TRANSFERINTEGRALS* (page 240)
- *UNRESTRICTED* (page 55)
- *VANDERWAALS* (page 198)
- *VECTORLENGTH* (page 338)
- *XC* (page 72)
- *XES* (page 170)
- *ZFS* (page 225)
- *ZLMFIT* (page 325)

16.2 Summary of all keywords

16.2.1 Engine ADF

`A1Fit`

Type Float

Default value 10.0

Unit Angstrom

GUI name Symmetric fit for distance >

Description STO-Fit keyword: distance between atoms, in Angstrom. The symmetric fit approximation is applied only for atoms farther apart.

`AccurateGradients`

Type Bool

Default value No

Description Print the nuclear gradients with more digits than usual.

`AddDiffuseFit`

Type Bool

Default value No

GUI name Add diffuse functions in fit: Yes

Description STO-Fit keyword: One can get more diffuse fit functions by setting this to True.

`AllDipMat`

Type Bool

Default value No

Description Print all dipole matrix elements between occupied and virtual Kohn-Sham orbitals.

`Allow`

Type String

Recurring True

Description Controlled aborts can in some cases be overruled. Of course, the checks have been inserted for good reasons and one should realize that ignoring them probably produces incorrect results or may lead to a program-crash.

`AllPoints`

Type Bool

Default value No

GUI name Force use of all points

Description ADF makes use of symmetry in the numerical integrations. Points are generated for the irreducible wedge, a symmetry unique sub region of space. Optionally the symmetry equivalent points are also used. This is achieved by setting this key to True.

`AnalyticalFreq`

Type Block

Description Define options for analytical frequencies.

`B1Size`

Type Float

Description Sparse grid max memory size

`B1Thresh`

Type Float

Default value 1e-10

Description MMGF_DENB1 and MMGF_GRADB1 cutoff values

`Check_CPKS_From_Iteration`

Type Integer

Default value 1

Description Solution of the CPKS equations is an iterative process, and convergence is achieved if the difference between U1 matrix of successive iterations falls below a certain threshold. This key can be used to determine at which iteration the checking should start taking place.

`Debug`

Type String

Description For debugging purposes. Options: fit, hessian, b1, densities, numbers, symmetry, all.

`Hessian`

Type Multiple Choice

Default value reflect

Options [reflect, average]

Description Whether the final Hessian is obtained by reflecting or averaging?

`Max_CPKS_Iterations`

Type Integer

Default value 20

Description Calculating the analytical frequencies requires the solution of the Coupled Perturbed Kohn-Sham (CPKS) equations, which is an iterative process. If convergence is not achieved (a warning will be printed in the output if this is the case) then this subkey can be used to increase the number of iterations, although convergence is not guaranteed. The user required accuracy of the U1 matrix, as well as the ADF integration accuracy, can effect the rates of convergence.

`Print`

Type String

Description Primarily for debugging purposes. Options: eigs, u1, parts. Choosing EIGS results in the print out of the MO eigenvectors, while U1 results in the print out of the U1 matrices. Except for small molecules this will result in a lot of data being output, and so they are not recommended. Choosing PARTS results in the print out of various sub-hessians that add up to give the final analytical hessian.

`PrintNormalModeAnalysis`

Type Bool

Default value No

Description Request ADF to print analysis of the normal modes independently of AMS.

`U1_Accuracy`

Type Float

Default value 5.0

Description Solution of the CPKS equations is an iterative process, and convergence is achieved if the difference between U1 matrix of successive iterations falls below a certain threshold. This subkey can be used to set the threshold. The accuracy of the U1 will be 10**(-x). So, the higher the number the more accurate the U1 will be. While this parameter effects the accuracy of the frequencies, other factors also effect the accuracy of the frequencies, especially the ADF integration accuracy.

`AOMat2File`

Type Bool

Default value No

Description Write PMatrix, Fock matrix, and overlap matrix on AO basis to file for future analysis purposes

`AOResponse`

Type Block

Description If the block key AORESPONSE is used, by default, the polarizability is calculated. Note that if the molecule has symmetry the key ALLPOINTS should be included

`ALDA`

Type Bool

Default value No

Description Use ALDA only

`Alpha`

Type Bool

Default value No

Description Calculate linear response

`Beta`

Type Bool

Default value No

Description Will use 2n+1 rule to calculate beta.

`CALCTRANSFORMPROP`

Type String

Description Transformation Properties of Polarizabilities

`Components`

Type String

Description Limit the tensor components to the specified ones. Using this option may save the computation time. Options: XX, XY, XZ, YX, YY, YZ, ZX, ZY, ZZ

`Cubic`

Type Bool

Default value No

Description Calculate cubic response

`Damp`

Type Float

Default value 0.4

Description Specify damping for non-acceleration iteration

`Debug`

Type Integer

Default value 0

Description Debug level for AOResponse.

`DoNothing`

Type Bool

Default value No

Description Do nothing.

`EFG`

Type Block

Description Perform a Mulliken type analysis of the EFG principal components, and an analysis in terms of canonical MOs.

`Atom`

Type Integer

Default value 1

Description The number of the nucleus at which the EFG is to be analyzed (ADF input ordering).

`NBO`

Type Bool

Default value No

Description Perform an NBO/NLMO analysis of the EFG. Requires a series of calculations. See documentation.

Nuc

Type Integer

Default value 1

Description The number of the nucleus at which the EFG is to be analyzed (ADF internal atom ordering).

Thresh

Type Float

Default value 0.05

Description The threshold for printing the EFG-NBO contributions. The default is 0.05, which means that only orbitals with absolute value contribution larger than 5% of the total EFG are printed. To increase the number of contributions printed, specify a smaller threshold.

EFIOR

Type Bool

Default value No

Description

EFISHG

Type Bool

Default value No

Description

EFPLOT

Type Bool

Default value No

Description

EL_DIPOLE_EL_DIPOLE

Type String

Description

EL_DIPOLE_EL_OCTUPOLE

Type String

Description

EL_DIPOLE_EL_QUADRUPOLE

Type String

Description

EL_DIPOLE_MAG_DIPOLE

Type String

Description

`EL_DIPOLE_MAG_QUADRUPOLE`

Type String

Description

`EL_QUADRUPOLE_EL_QUADRUPOLE`

Type String

Description

`EL_QUADRUPOLE_MAG_DIPOLE`

Type String

Description

`EOPE`

Type Bool

Default value No

Description

`FitAODeriv`

Type Bool

Default value No

Description Use FITAODERIV for Coulomb potential

`Frequencies`

Type Float List

Default value [0.0]

Unit eV

Description List of frequencies of incident light, the perturbing field, at which the time-dependent properties will be calculated.

`GIAO`

Type Bool

Default value No

Description Use gauge-included atomic orbitals

`Gamma`

Type Bool

Default value No

Description Will use 2n+1 rule to calculate gamma.

`HirshPol`

Type Bool

Default value No

Description Hirshfeld Polarizability of fragments

`IDRI`

Type Bool

Default value No

Description

`LifeTime`

Type Float

Unit Hartree

Description Specify the resonance peak width (damping) in Hartree units. Typically the lifetime of the excited states is approximated with a common phenomenological damping parameter. Values are best obtained by fitting absorption data for the molecule, however, the values do not vary a lot between similar molecules, so it is not hard to estimate values. A value of 0.004 Hartree was used in Ref. [266].

`MAG_DIPOLE_MAG_DIPOLE`

Type String

Description

`MagOptRot`

Type Bool

Default value No

Description Calculate magneto-optical rotation

`MagneticPert`

Type Bool

Default value No

Description Use magnetic field as a perturbation

`NBO`

Type Bool

Default value No

Description Perform NBO analysis

`NoCore`

Type Bool

Default value No

Description if NOCORE is set we skip the core potential in diamagnetic term and/or in the unperturbed density of the CPKS solvers

`OKE`

Type Bool

Default value No

Description

`OPTICALR`

Type Bool

Default value No

Description

`OpticalRotation`

Type Bool

Default value No

Description Calculate optical rotation

`QuadBeta`

Type Bool

Default value No

Description Quadrupole operators with beta tensor

`QuadPert`

Type Bool

Default value No

Description Calculate quadrupole-quadrupole polarizability

`Quadratic`

Type Bool

Default value No

Description Calculate quadratic response

`Quadrupole`

Type Bool

Default value No

Description Calculate dipole-quadrupole polarizability

`Raman`

Type Bool

Default value No

Description

`SCF`

Type String

Description Specify CPKS parameters such as the degree of convergence and the maximum number of iterations: NOCYC - disable self-consistence altogetherNOACCEL - disable convergence accelerationCONV - convergence criterion for CPKS. The default value is 10-6 . The value is relative to the uncoupled result (i.e. to the value without self-consistence).ITER - maximum number of CPKS iterations, 50 by default.Specifying ITER=0 has the same effect as specifying NOCYC.

`SHG`

Type Bool

Default value No

Description

`STATIC`

Type Bool

Default value No

Description

THG

Type Bool

Default value No

Description

TPA

Type Bool

Default value No

Description

Traceless

Type Bool

Default value No

Description Traceless quadrupole tensors

VROA

Type Bool

Default value No

Description Calculate Vibrational Raman Optical Activity.

VelocityOrd

Type Bool

Default value No

Description Use VelocityOrd without GIAOs

XAlpha

Type Bool

Default value No

Description Xalpha potential

Aromaticity

Type Non-standard block

Description Calculate aromaticity indicators, i.e. the matrix of localization/delocalization indices (LI-DI), Iring (ring index) and MCI (multi center index) aromaticity indices.

AtomicChargesTypeForAMS

Type Multiple Choice

Default value Mulliken

Options [Mulliken, Hirshfeld, CM5, Voronoi, MDC-M, MDC-D, MDC-Q, QTAIM]

GUI name Atomic charges for AMS

Description Type of atomic charges to be used by AMS.

Note that some of these atomic charges are computed and printed by default in ADF.

Hirshfeld charges are available only for default atomic fragments.

`Balance`

Type Bool

Default value No

Description Measure the actual speed of the nodes in the parallel machine

`Basis`

Type Block

Description Definition of the basis set

`Core`

Type Multiple Choice

Default value Large

Options [None, Small, Large]

GUI name Frozen core

Description Select the size of the frozen core you want to use.

Small and Large will be interpreted within the basis sets available (of the selected quality), and might refer to the same core in some cases.

If you specify 'None' you are guaranteed to have an all-electron basis set.

`CreateOutput`

Type Bool

Default value No

Description If true, the output of the atomic create runs will be printed to standard output. If false, it will be saved to the file CreateAtoms.out in the AMS results folder.

`FitType`

Type Multiple Choice

Default value Auto

Options [Auto, SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE]

GUI name STO fit set

Description Expert option.

Select the auxiliary fit to be used for STOfit or old Hartree-Fock RI scheme.

The fit set for a given atom is taken from the all-electron basis set file for the specified choice, for the same element as the atom.

By default (Auto) the fit set is taken from the original basis set file.

`Path`

Type String

Description The name of an alternative directory with basis sets to use. ADF looks for appropriate basis sets only within this directory. Default $AMSRESOURCES/ADF.

`PerAtomType`

Type Block

Recurring True

Description Defines the basis set for all atoms of a particular type.

`Core`

Type Multiple Choice

Options [None, Small, Large]

Description Size of the frozen core.

`File`

Type String

Description The path of the basis set file (the path can either absolute or relative to $AMSRESOURCES/ADF). Note that one should include ZORA in the path for relativistic calculations, for example 'ZORA/QZ4P/Au'. Specifying the path to the basis file explicitly overrides the automatic basis file selection via the Type and Core subkeys.

`Symbol`

Type String

Description The symbol for which to define the basis set.

`Type`

Type Multiple Choice

Options [SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, mTZ2P, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, Corr/TZ3P, Corr/QZ6P, Corr/ATZ3P, Corr/AQZ6P, POLTDDFT/DZ, POLTDDFT/DZP, POLTDDFT/TZP]

Description The basis sets to be used.

`PerRegion`

Type Block

Recurring True

Description Defines the basis set for all atoms in a region. If specified, this overwrites the values set with the Basis%Type and Basis%PerAtomType keywords for atoms in that region. Note that if this keyword is used multiple times, the chosen regions may not overlap.

`Core`

Type Multiple Choice

Default value Large

Options [None, Small, Large]

Description Size of the frozen core.

`Region`

Type String

Description The identifier of the region for which to define the basis set. Note that this may also be a region expression, e.g. 'myregion+myotherregion' (the union of two regions).

`Type`

Type Multiple Choice

Default value DZ

Options [SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, mTZ2P, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, Corr/TZ3P, Corr/QZ6P, Corr/ATZ3P, Corr/AQZ6P, POLTDDFT/DZ, POLTDDFT/DZP, POLTDDFT/TZP]

Description The basis sets to be used.

`Type`

Type Multiple Choice

Default value DZ

Options [SZ, DZ, DZP, TZP, TZ2P, QZ4P, TZ2P-J, QZ4P-J, mTZ2P, AUG/ASZ, AUG/ADZ, AUG/ADZP, AUG/ATZP, AUG/ATZ2P, ET/ET-pVQZ, ET/ET-QZ3P, ET/ET-QZ3P-1DIFFUSE, ET/ET-QZ3P-2DIFFUSE, ET/ET-QZ3P-3DIFFUSE, Corr/TZ3P, Corr/QZ6P, Corr/ATZ3P, Corr/AQZ6P, POLTDDFT/DZ, POLTDDFT/DZP, POLTDDFT/TZP]

GUI name Basis set

Description Select the basis set to use.

SZ: Single Z DZ: Double Z DZP: Double Z, 1 polarization function TZP: Triple Z, 1 polarization function TZ2P: Triple Z, 2 polarization functions QZ4P: Quad Z, 4 pol functions, all-electron AUG: Augmented (extra diffuse functions) ET: Even tempered all electron basis sets J: Extra tight functions

These descriptions are meant to give an indication of the quality, but remember that ADF uses Slater type functions.

For standard calculations (energies, geometries, etc.) the relative quality is:

SZ < DZ < DZP < TZP < TZ2P < ET-pVQZ < QZ4P

The basis set chosen will apply to all atom types in your molecule. If no matching basis set is found, ADF will try to use a basis set of better quality.

For TDDFT applications and small negatively charged atoms or molecules, use basis sets with extra diffuse functions.

J: TZ2P-J, QZ4P-J: for use in ESR hyperfine or NMR spin-spin couplings.

Use the Basis panel to select a basis set per atom type, and to see what basis set actually will be used.

`BeckeGrid`

Type Block

Description Options for the numerical integration grid.

`AllowAngularBoost`

Type Bool

Default value Yes

Description Allow automatic augmentation of the Lebedev spherical grid for highly coordinated atoms.

`InnerShellsPruning`

Type Bool

Default value Yes

Description Allow automatic pruning of the Lebedev spherical grid for shells close to the nuclei.

`PartitionFunPruning`

Type Bool

Default value Yes

Description Allow pruning of integration points based on the value of the partition function.

`QPNear`

Type Float

Unit Angstrom

Description Only relevant if you have specified point charges in the input file. ADF generates grids only about those point charges that are close to any real atoms. The criterion, input with the qpnear subkey, is the closest distance between the point charge at hand and any real atom.

`Quality`

Type Multiple Choice

Default value Auto

Options [Auto, Basic, Normal, Good, VeryGood, Excellent]

Description Quality of the integration grid. For a description of the various qualities and the associated numerical accuracy see reference. If 'Auto', the quality defined in the 'NumericalQuality' will be used.

`QualityPerRegion`

Type Block

Recurring True

Description Sets the grid quality for all atoms in a region. If specified, this overwrites the globally set quality.

`Quality`

Type Multiple Choice

Options [Basic, Normal, Good, VeryGood, Excellent]

Description The region's integration grid quality.

`Region`

Type String

Description The identifier of the region for which to set the quality.

`RadialGridBoost`

Type Float

Description The number of radial grid points will be boosted by this factor. Some XC functionals require very accurate radial integration grids, so ADF will automatically boost the radial grid by a factor 3 for the following numerically sensitive functionals: LibXC M05, LibXC M05-2X, LibXC M06-2X, LibXC M06-HF, LibXC M06-L, LibXC M08-HX, LibXC M08-SO, LibXC M11-L, LibXC MS0, LibXC MS1, LibXC MS2, LibXC MS2H, LibXC MVS, LibXC MVSH, LibXC N12, LibXC N12-SX, LibXC SOGGA11, LibXC SOGGA11-X, LibXC TH1, LibXC TH2, LibXC WB97, LibXC WB97X, MetaGGA M06L, MetaHybrid M06-2X, MetaHybrid M06-HF, MetaGGA MVS.

`BondOrders`

Type Block

Description Options for the calculation of bond orders. Note: the calculation of bond orders should be requested via the Properties%BondOrders input option in the AMS driver input.

`PrintAll`

Type Bool

Default value No

Description If 'Yes', all five types of bond orders (i.e. Nalewajski-Mrozek-1,2 & 3, Mayer and Gopinathan-Jug) will be printed to the output. Otherwise only the Nalewajski-Mrozek-3 and the type requested in BondOrders%TypeForAMS will be printed.

`PrintTolerance`

Type Float

Default value 0.2

Description Only bond orders larger than this threshold will be printed in the output (this treshold applies only to the printing in the 'BOND-ORDER ANALYSIS' section of the ADF output.

`TypeForAMS`

Type Multiple Choice

Default value Nalewajski-Mrozek-3

Options [Nalewajski-Mrozek-1, Nalewajski-Mrozek-2, Nalewajski-Mrozek-3, Mayer, Gopinathan-Jug]

GUI name Bond order type for AMS

Description The type of bond order that will be saved, printed and used by AMS.

Nalewajski-Mrozek-1,2: bond orders calculated from two-electron valence indices based on partitioning of tr(Delta_P^2) using 3-index set or 4-index set respectively. Nalewajski-Mrozek-3: bond-orders calculated from valence indices based on partitioning of tr(P*Delta_P). Inter-atomic bond orders are not defined with non-atomic fragments.

`CalcOverlapOnly`

Type Bool

Default value No

Description Calculate overlaps of primitive basis and stops after computing them.

`CDFT`

Type Block

Description CDFT is a tool for carrying out DFT calculations in the presence of a constraint.

`AllAtoms`

Type Bool

Default value No

Description If AllAtoms is true, then TheAtoms is overridden and all the atoms in the active fragment are included in the set.

`AnalyticalHessian`

Type Integer

Default value 0

Description This will calculate the analytical derivative of the energy w.r.t. the Lagrange multiplier up to the specified SCF iteration. This key is not recommended due to the high computational cost that comes with it. The calculation is equivalent to a ground state Hessian, and it is carried out with the full sum-over-states formula.

`ChargeAndSpin`

Type Bool

Default value No

Description will constrain both the charge and the spin

`Constraints`

Type Float List

Description The values of the constraints. If CHARGEANDSPIN, constraints to the alpha and beta electrons need to be specified sequentially. One more electron => CONSTRAINTS -1.0. One less electron => CONSTRAINTS 1.0. If the CDFT type is EXCITEDCDFT, CONSTRAINTS=1.0 is recommended. Other values are technically possible but have not been tested yet.

`DoNotOptimize`

Type Bool

Default value No

Description If true, the multipliers chosen in INITIALMULTIPLIERS will not be optimized and will be constant throughout the entire SCF procedure.

`ExcitedCDFT`

Type Bool

Default value No

Description will generate an excited state with CONSTRAINTS number of ALPHA electrons constrained to occupy the virtual space of a ground state reference calculation. This is the essence of the eXcited Constrained DFT (XCDFT) method(P. Ramos, M. Pavanello, Low-lying excited states by constrained DFT, Journal of Chemical Physics 148, 144103 (2018) https://doi.org/10.1063/1.5018615) for the calculation of low-lying single excitations. XCDFT is found to correctly reproduce the energy surface topology at conical intersections between the ground state and the first singly excited state and can also accounts for the condensed phase effects in solvated chromophores where typical Delta SCF methods variationally collapse.

`InitialMultipliers`

Type Float List

Description If available, a guess for the Lagrange multipliers can be entered.

MaxIter

Type Integer

Default value 200

Description Maximum number of CDFT iterations. CDFT carries out a loop nested inside the SCF cycle.

Metric

Type Bool

Default value No

Description Relevant for XCDFT. In the XCDFT method orthogonality is not imposed between the KS-orbitals of ground and excited states. If METRIC is specified, the degree of mixing of the single excited state with the ground state or high-order excitations is calculated. Three parameters are calculated: p, m and d. The parameters p and m will give information about the amount of mixing with the ground state, while parameter d will determine the mixing with high order excitations. Additional information about the origin of these parameters can be found in the literature (P. Ramos, M. Pavanello, Low-lying excited states by constrained DFT, Journal of Chemical Physics 148, 144103 (2018) https://doi.org/10.1063/1.5018615)

NAtomsPerSet

Type Integer List

Description The number of atoms in each moiety (set).

NConstraints

Type Integer

Default value 1

Description This specifies the number of sets of atoms to be considered. For example, if the user wishes to constrain a positive charge on one part of the system, and a negative charge on another part, NCONSTRAINTS should be set to two. There is no limit on the number of constraints. However, SCF convergence becomes an issue with more than 2 constraints. Note: NCONSTRAINTS>1 is untested.

OnlyCharge

Type Bool

Default value Yes

Description Will constrain only the charge, letting spin relax (and potentially delocalize)

OnlySpin

Type Bool

Default value No

Description Will constrain only the spin

PopType

Type Multiple Choice

Default value yukawalike

Options [yukawalike, fuzzyvoronoibecke, fuzzyvoronoifermi]

Description The population analysis chosen for determining the constraint.

Print

Type Multiple Choice

Default value low

Options [low, medium, high]

Description Print level and debugging.

SelfConsistent

Type Bool

Default value No

Description Self-Consistent CDFT

StepSize

Type Float

Default value 0.5

Description The amount of the Lagrange multipliers step taken in each CDFT iteration

TheAtoms

Type Integer List

Description The atom numbers of the moieties in the input geometry order. If NCONSTRAINTS is larger than 1, the sets of atoms are entered as a single list.

Threshold

Type Float

Default value 1e-10

Description The threshold for convergence of the CDFT constraints. The tighter the SCF convergence criteria, the tighter the THRESHOLD should be.

CM5

Type Bool

Default value No

GUI name : CM5 charges

Description Calculate the charge model 5 (CM5) analysis.

comment

Type Non-standard block

Description The content of this block will be copied to the output header as a comment to the calculation.

ConceptualDFT

Type Block

Description Conceptual DFT Properties

AnalysisLevel

Type Multiple Choice

Default value Normal

Options [Normal, Extended, Full]

Description Set the level of the ConceptualDFT analysis:

Normal - global descriptors only,

Extended - both global and condensed (QTAIM) local descriptors,

Full - all descriptors including non local ones.

`AtomsToDo`

Type Integer List

GUI name Include atoms

Description Define a subset of atoms for which properties are calculated. If the [Domains] block is present then this list specifies which atoms are used to define the domains bounding box.

`Domains`

Type Block

Description Calculate integrated properties for the domains (same sign) of the dual descriptor.

`Border`

Type Float

Default value 7.0

Unit Bohr

Description Set the extent of the Cartesian grid. Extent is the distance between a face of the grid's bounding box and the most outlying atom in the corresponding direction. If the [AtomsToDo] key is present, the bounding box is created around the specified atoms.

`Display`

Type Float

Default value 0.005

Description Domains for which the integrated DD value is smaller (in magnitude) than the specified value are omitted from the printed output.

`Enabled`

Type Bool

Default value No

GUI name Properties of reactivity domains

Description Calculate properties of reactivity domains.

`Ensemble`

Type Multiple Choice

Default value Canonical

Options [Canonical, GrandCanonical]

Description Statistical ensemble for DD domains. Canonical: DD values are calculated using the statistical canonical ensemble. GrandCanonical: DD values are calculated using the statistical grand canonical ensemble. The grand canonical DD corresponds to (S^2 f(2) -

(gamma/eta^3) f^0), where f(2) is the canonical DD, gamma and eta - the hyper-hardness and hardness of the chemical system, respectively, and f^0 is the mean Fukui function. This statistical ensemble is a natural choice when comparing two chemical systems with a different number of electrons.

`Radius`

Type Float

Default value 0.0

Description This option adds a sphere around each nucleus, excluding all points inside it. This can help to separate domains around an atom or to exclude core electrons. Be careful when using this option. In particular, the radius of the sphere should exceed two or three times the [Spacing] value to be effective. By default, no spheres are added.

`Spacing`

Type Float

Default value 0.1

Unit Bohr

Description Specifies spacing (distance between neighboring points) of the rectangular Cartesian grid used when searching for DD domains. It may be useful to specify a smaller value (or increase the size of the grid, see [Border] key) if a substantial part of the electronic density is accounted for.

`Threshold`

Type Float

Default value 0.001

Description Arbitrary value of dual descriptor used to separate DD domains (values below this threshold are ignored).

`Electronegativity`

Type Bool

Default value No

GUI name Atomic electronegativities

Description Calculate atomic electronegativities. Requires an all-electron calculation (no frozen core), triggers the TotalEnergy and increases the [AnalysisLevel] to at least Extended.

`Enabled`

Type Bool

Default value No

GUI name Conceptual DFT (FMO): Calculate

Description Calculate Conceptual DFT properties.

`ConstructPot`

Type Block

Description Reads a density from a TAPE41 file and constructs numerically the corresponding potential to it

`CPBasis`

Type Bool

Default value Yes

Description

`CPGrid`

Type Bool

Default value No

Description

`Converge`

Type Float

Default value 1e-06

Description

`CutNegativeDens`

Type Float

Default value 0.0001

Description

`Damp`

Type Float

Default value 1.0

Description

`DensConv`

Type Float

Description

`EigenShift`

Type Float

Default value 0.01

Description

`FitBas`

Type Bool

Default value Yes

Description

`FixedLambda`

Type Bool

Default value No

Description

`ImportDens`

Type String

Description Filename of density…

`Lambda`

Type Float

Default value 0.01

Description

`PotBas`

Type String

Description Filename…

`PotProj`

Type String

Description

`ProjChange`

Type Float

Default value -1.0

Description

`ProjSmallDens`

Type Float

Default value 1e-50

Description

`QPiterations`

Type Integer

Default value 1000

Description

`SVD`

Type Bool

Default value No

Description

`SmallEigThresh`

Type Float

Default value 0.0001

Description

`StartPot`

Type String

Description Filename of potential…

`StepSize`

Type Float

Default value 1.0

Description

`TIKH`

Type Float

Default value 0.0

Description

`CorePotentials`

Type Non-standard block

Description With the key COREPOTENTIALS you specify the core file and (optionally) which sections pertain to the distinct atom types in the molecule.

`Create`

Type String

Description Keywords for create run. {Atomtype Datafile}

`CurrentResponse`

Type Block

Description

`CDSpec`

Type Bool

Default value No

Description

`Damping`

Type Float

Default value 0.0

Description

`GTensor`

Type Bool

Default value No

Description

`Magnet`

Type Bool

Default value No

Description

`NCT`

Type Float

Default value 0.0

Description

NMRShielding

Type Bool

Default value No

Description

NoVK

Type Bool

Default value No

Description

PARTVK

Type Float

Default value 1.0

Description

Parabolic

Type Float

Default value 0.0

Description

QIANVignale

Type Bool

Default value No

Description

Static

Type Bool

Default value No

Description

CVNDFT

Type Block

Description The CVNDFT block key regulates the execution of the CV(n)-DFT code, which calculates the singlet or triplet electronic excitations for the closed shell molecules.

CV_DFT

Type Block

Description The simplest case: the TDDFT transition density U-vector is substituted into the infinite order CV(infinity)-DFT excitation energy

InitGuess

Type Multiple Choice

Default value TDDFT

Options [TDDFT, SOR]

Description Initial guess

DSCF_CV_DFT

Type Block

Description The simplest case: the TDDFT transition density U-vector is substituted into the infinite order CV(infinity)-DFT excitation energy

DampOrbRelax

Type Float

Default value 0.2

Description The mix_relax parameter defines the relative weight of the new relaxation vector that is added to the one from the previous iteration.

DampVariable

Type Bool

Default value No

Description Damping condition

Damping

Type Float

Default value 0.2

Description Damping

InitGuess

Type Multiple Choice

Default value SOR

Options [TDDFT, SOR]

Description Initial guess

Optimize

Type Multiple Choice

Default value SVD

Options [SVD, SOR, COL]

Description Gradient optimization method

RelaxAlpha

Type Integer

Default value 1

Description The SCF cycle number at which the relaxation of alpha orbitals starts.

RelaxBeta

Type Integer

Default value 1

Description The SCF cycle number at which the relaxation of beta orbitals starts.

Iteration

Type Integer

Default value 50

Description The maximum number of iterations

RSCF_CV_DFT

Type Block

Description The simplest case: the TDDFT transition density U-vector is substituted into the infinite order CV(infinity)-DFT excitation energy

DampOrbRelax

Type Float

Default value 0.2

Description The mix_relax parameter defines the relative weight of the new relaxation vector that is added to the one from the previous iteration.

DampVariable

Type Bool

Default value No

Description Damping condition

Damping

Type Float

Default value 0.2

Description Damping

InitGuess

Type Multiple Choice

Default value TDDFT

Options [TDDFT, SOR]

Description Initial guess

RelaxAlpha

Type Integer

Default value 1

Description The SCF cycle number at which the relaxation of alpha orbitals starts.

RelaxBeta

Type Integer

Default value 1

Description The SCF cycle number at which the relaxation of beta orbitals starts.

R_CV_DFT

Type Block

Description The simplest case: the TDDFT transition density U-vector is substituted into the infinite order CV(infinity)-DFT excitation energy

DampOrbRelax

Type Float

Default value 0.2

Description The mix_relax parameter defines the relative weight of the new relaxation vector that is added to the one from the previous iteration.

`DampVariable`

Type Bool

Default value No

Description Damping condition

`InitGuess`

Type Multiple Choice

Default value TDDFT

Options [TDDFT, SOR]

Description Initial guess

`RelaxAlpha`

Type Integer

Default value 1

Description The SCF cycle number at which the relaxation of alpha orbitals starts.

`RelaxBeta`

Type Integer

Default value 1

Description The SCF cycle number at which the relaxation of beta orbitals starts.

`SCF_CV_DFT`

Type Block

Description The simplest case: the TDDFT transition density U-vector is substituted into the infinite order CV(infinity)-DFT excitation energy

`DampVariable`

Type Bool

Default value No

Description Damping condition

`Damping`

Type Float

Default value 0.2

Description Damping

`InitGuess`

Type Multiple Choice

Default value TDDFT

Options [TDDFT, SOR]

Description Initial guess

`Tolerance`

Type Float

Default value 0.0001

Description The convergence criterion, i.e. the SCF-CV(infinity)-DFT procedure stops when the given accuracy is achieved.

`Debug`

Type String

Recurring True

Description The amount of printed output is regulated with the keys Print, NoPrint, EPrint and Debug.

`Dependency`

Type Block

Description

`Enabled`

Type Bool

Default value No

GUI name Fix dependencies

Description Used to make the basis or fit set linearly independent, up to the threshold specified below. This is typically important when you have many diffuse functions in your basis or fit set.

`bas`

Type Float

Default value 0.0001

GUI name Threshold for basis

Description A criterion applied to the overlap matrix of unoccupied normalized SFOs. Eigenvectors corresponding to smaller eigenvalues are eliminated from the valence space.

Note: if you choose a very coarse value, you will remove too many degrees of freedom in the basis set, while if you choose it too strict, the numerical problems may not be countered adequately.

`eig`

Type Float

Default value 100000000.0

Description Merely a technical parameter. When the DEPENDENCY key is activated, any rejected basis functions (i.e.: linear combinations that correspond with small eigenvalues in the virtual SFOs overlap matrix) are normally processed until diagonalization of the Fock matrix takes place. At that point, all matrix elements corresponding to rejected functions are set to zero (off-diagonal) and BigEig (diagonal).

In AMSinput you must check the Fix Linear dependency check box for this option to be used.

`fit`

Type Float

Default value 1e-10

GUI name Threshold for fit

Description Similar to Dependency%bas.

The criterion is now applied to the overlap matrix of fit functions.

The fit coefficients, which give the approximate expansion of the charge density in terms of the fit functions (for the evaluation of the coulomb potential) are set to zero for fit functions (i.e.: combinations of) corresponding to small-eigenvalue eigenvectors of the fit overlap matrix.

`Diffuse`

Type Bool

Default value No

Description Adding diffuse integration points in case of the old Voronoi numerical integration grid.

`DIMPAR`

Type Non-standard block

Description In this block, the parameters for the DIM atoms are defined in DIM/QM calculations.

`DIMQM`

Type Non-standard block

Description Input for DIM/QM

`DipoleLength`

Type Bool

Default value No

Description Use dipole-length elements for perturbing (external) integrals in CURRENT response

`DipoleResponse`

Type Bool

Default value No

Description

`DumpBasisOnly`

Type Bool

Default value No

Description Dump basis and fit set files use for each atom.

`ElectronTransfer`

Type Block

Description Block key for charge transfer integrals with FDE.

`CDFT`

Type Bool

Default value No

Description

`Debug`

Type Bool

Default value No

Description

`Disjoint`

Type Bool

Description

`FDE`

Type Bool

Default value No

Description

`InvThr`

Type Float

Default value 0.001

Description

`Joint`

Type Bool

Description

`KNADD`

Type Bool

Default value No

Description

`NonCT`

Type Bool

Default value No

Description

`NumFrag`

Type Integer

Description

`Print`

Type String

Description

`EnergyFrag`

Type Non-standard block

Description

`EPrint`

Type Block

Description Print switches that require more specification than just off or on

`AtomPop`

Type String

Description Mulliken population analysis on a per-atom basis

`BASPop`

Type String

Description Mulliken population analysis on a per-bas-function basis

`Eigval`

Type String

Description One-electron orbital energies

`Fit`

Type String

Description Fit functions and fit coefficients

`Frag`

Type String

Description Building of the molecule from fragments

`FragPop`

Type String

Description Mulliken population analysis on a per fragment basis

`Freq`

Type String

Description Intermediate results in the computation of frequencies (see debug: freq).

`GeoStep`

Type String

Description Geometry updates (Optimization, Transition State, …)

`NumInt`

Type String

Description Numerical Integration

`OrbPop`

Type Non-standard block

Description (Mulliken type) population analysis for individual MOs

`OrbPopEr`

Type String

Description Energy Range (ER) in hartree units for the OrbPop subkey

`Repeat`

Type String

Description Repetition of output in Geometry iterations (SCF, optimization, …)

SCF

Type String

Description Self Consistent Field procedure

SFO

Type String

Description Information related to the Symmetrized Fragment Orbitals and the analysis

TF

Type String

Description Transition Field method

ESR

Type Block

Description

Enabled

Type Bool

Default value No

Description Calculate ESR (g- and/or A tensors)

PARANMR

Type Bool

Default value No

Description Paramagnetic part NMR shielding.

ETSNOCV

Type Block

Description Perform ETS-NOCV analysis.

DEBUGTV

Type Bool

Default value No

Description For T/V debugging

EKMin

Type Float

Default value 2.0

Unit kcal/mol

GUI name Energy threshold

Description The threshold for orbital interaction energy contributions corresponding to deformation density components originating from each NOCV-pairs

ENOCV

Type Float

Default value 0.05

GUI name NOCVs with ev larger than

Description The threshold for NOCV-eigenvalues

`Enabled`

Type Bool

Default value No

Description Perform ETS-NOCV analysis.

`RhoKMin`

Type Float

Default value 0.01

GUI name Population threshold

Description The threshold for population analysis of each deformation density contribution in terms of individual SFOs.

`TVanalysis`

Type Bool

Default value No

GUI name T/V analysis

Description Perform T/V decomposition

`ExactDensity`

Type Bool

Default value No

Description Use the exact density (as opposed to the fitted density) for the computation of the exchange-correlation potential

`Excitations`

Type Block

Description Excitation energies: UV/Vis

`ALLXASMOMENTS`

Type Bool

Default value No

Description To be used in combination with XAS. This will print out all the individual transition moments used within the calculation of the total oscillator strength

`ALLXASQUADRUPOLE`

Type Bool

Default value No

Description To be used in combination with XAS.This will print out the individual oscillator strength components to the total oscillator strength.

`Allowed`

Type Bool

Default value No

Description Treat only those irreducible representations for which the oscillator strengths will be nonzero (as opposed to all)

`AlsoRestricted`

Type Bool

Description Include also excitation energies in which a spin-restricted exchange-correlation kernel is used

`Analytical`

Type Bool

Default value No

Description The required integrals for the CD spectrum are calculated analytically, instead of numerically. Only used in case of CD spectrum

`AsympCor`

Type Float

Default value 500.0

Description

`BSE`

Type Bool

Default value No

Description Solve the static Bethe-Salpeter equation based on a GW calculation

`CDSpectrum`

Type Bool

Default value No

Description Compute the rotatory strengths for the calculated excitations, in order to simulate Circular Dichroism (CD) spectra

`DTensor`

Type String

Description MCD gtensor

`Davidson`

Type Non-standard block

Description Use the Davidson procedure

`Descriptors`

Type Bool

Default value No

Description Compute charge-transfer descriptors and SFO analysis

Descriptors_CT_AT_Rab

Type Float

Default value 2.0

Description Atomic distance criterion used for the calculation of CT_AT descriptors

ESESTDM

Type Bool

Default value No

Description Compute transition dipole moments between excited states

Exact

Type Non-standard block

Description The most straightforward procedure is a direct diagonalization of the matrix from which the excitation energies and oscillator strengths are obtained. Since the matrix may become very large, this option is possible only for very small molecules

FullKernel

Type Bool

Default value No

Description Use the non-ALDA kernel (with XCFUN)

GTensor

Type String

Description MCD gtensor

HDA

Type Bool

Default value No

GUI name Hybrid diagonal approximation

Description Activate the diagonal HF exchange approximation.

This is only relevant if a (meta-)hybrid is used in the SCF.

HDA_CutOff

Type Float

Default value 10000000.0

Unit eV

GUI name HDA cutoff

Description This is cutoff based on differences in energy between eps_virt-eps_occ, to reduce number of diagonal HF exchange integrals.

Iterations

Type Integer

Default value 200

Description The maximum number of attempts within which the Davidson algorithm has to converge

`KFWrite`

Type Integer

Default value 3

Description If kfwrite is 0 then do not write contributions, transition densities, and restart vectors to TAPE21, since this can lead to a huge TAPE21, especially if many excitations are calculated. 3 means that contributions, transition densities, and restart vectors are written to TAPE21.

`Lowest`

Type Integer List

Default value [10]

GUI name Number of excitations

Description Number of lowest excitations to compute

`MCD`

Type String

Description TODO: Magnetic Circular Dichroism

`NTO`

Type Bool

Default value No

Description Compute the Natural Transition Orbitals

`N_HDA_integral`

Type Integer

Default value 1000000000

Description Maximum number of HDA integrals

`N_SFO`

Type Integer

Default value 40

Description Number of SFO analyzed and printed

`OnlySing`

Type Bool

Description Compute only singlet-singlet excitations

`OnlyTrip`

Type Bool

Description Compute only singlet-triplet excitations

`Orthonormality`

Type Float

Default value 1e-06

Description The Davidson algorithm orthonormalizes its trial vectors. Increasing the default orthonormality criterion increases the CPU time somewhat, but is another useful check on the reliability of the results.

`ROSCFType`

Type Multiple Choice

Default value S-TDA

Options [None, R-TDA, S-TDA, X-TDA, SF-TDA]

Description Specifies the type of method to be used in case of ROSCF.

`Residu`

Type Float

Default value 1e-06

Unit Hartree

Description

`SFOAnalysis`

Type Bool

Default value No

Description Do SFO analysis

`SOSFreq`

Type Float

Description

`STDA`

Type Bool

Default value No

Description Simplified Tamm-Dancoff approach

`STDDFT`

Type Bool

Default value No

Description Simplified time-dependent DFT

`ScaleCoul`

Type Float

Default value 1.0

Description Scaling of Coulomb kernel with scale parameter

`ScaleHF`

Type Float

Default value 1.0

Description Scaling of the HF part of the kernel with scale parameter

`ScaleXC`

Type Float

Default value 1.0

Description Scaling of the XC-kernel (excluding a possible HF-part) with scale parameter

`Select`

Type String

Description Rather than selecting the first nmcdterm transitions for consideration individual transitions can be selected through the SELECT keyword

`SingleOrbTrans`

Type Bool

Default value No

Description keyword to use only orbital energy differences

`TD-DFTB`

Type Bool

Default value No

Description Use the molecular orbitals from a DFT ground state calculation as input to an excited state calculation with TD-DFTB coupling matrices

`TDA-DFTB`

Type Bool

Default value No

Description Use the molecular orbitals from a DFT ground state calculation as input to an excited state calculation with TDA-DFTB coupling matrices

`Tolerance`

Type Float

Default value 1e-06

Unit Hartree

Description

`Vectors`

Type Integer

Description The maximum number of trial vectors in the Davidson algorithm for which space is allocated. If this number is small less memory will be needed, but the trial vector space is smaller and has to be collapsed more often, at the expense of CPU time. The default if usually adequate.

`Velocity`

Type Bool

Default value No

GUI name Velocity representation

Description Calculates the dipole-velocity representation of the oscillator strength. If applicable, the dipole-velocity representation of the rotatory strength is calculated. Default the dipole-length representation of the oscillator strength and rotatory strength is calculated

XAS

Type Bool

Default value No

Description Calculation of the higher oder multipole moment integrals and the calculation of the quadrupole oscillator strengths. This will only print the total oscillator strength and the excitation energy.

ExcitedEDA

Type Block

Description Options for excited energy decomposition (EDA).

Calc

Type Multiple Choice

Default value None

Options [None, Electrostatic, Pauli]

Description None: No calculation of parts of excited EDA.

Electrostatic: calculate electrostatic part EDA excited state.

Pauli: calculate Pauli repulsion part of excited state.

ElectrostaticFile

Type String

Default value

Description Path to adf.rkf file from which ADF reads electrostatic part excited EDA.

PauliFile

Type String

Default value

Description Path to adf.rkf file from which ADF reads Pauli repulsion part excited EDA.

ExcitedGO

Type Block

Description Excited state geometry optimization

ALLGRADIENTS

Type Bool

Default value No

Description

CPKS

Type Block

Description Some control parameters for the CPKS(Z-vector) part of the TDDFT gradients calculation

Eps

Type Float

Default value 0.0001

Description Convergence requirement of the CPKS

`IterOut`

Type Integer

Default value 5

Description Details of the CPKS calculation are printed every iter iterations

`NoPreConiter`

Type Integer

Default value 200

Description maximum number of iterations allowed for the unpreconditioned solver.

`PreConiter`

Type Integer

Default value 30

Description maximum number of iterations allowed for the preconditioned solver

`EigenFollow`

Type Bool

Default value No

Description This key tries to follow the eigenvector in excited state geometry optimizations

`Output`

Type Integer

Default value 0

Description The amount of output printed. A higher value requests more detailed output

`SING_GRADS`

Type Non-standard block

Description

`Singlet`

Type Bool

Default value Yes

Description Singlet-singlet excitation is considered

`State`

Type String

Description Choose the excitation for which the gradient is to be evaluated: ‘State Irreplab nstate’. ‘Irreplab’ is the label from the TDDFT calculation. NOTE: the TDDFT module uses a different notation for some representation names, for example, A’ is used instead of AA. ‘nstate’: this value indicates that the nstate-th transition of symmetry Irreplab is to be evaluated. Default is the first fully symmetric transition.

`TRIP_GRADS`

Type Non-standard block

Description

`Triplet`

Type Bool

Default value No

Description Singlet-triplet excitation is considered

`ExtendedPopan`

Type Bool

Default value No

GUI name : Extended population analysis

Description Calculate the Mayer bond orders and Mulliken atom-atom populations per l-value

`Externals`

Type Non-standard block

Description Legacy support of the older DRF code.

`FDE`

Type Block

Description Frozen Density Embedding options

`AMOLFDE`

Type Bool

Default value No

Description placeholder

`CAPDENSCONV`

Type Float

Default value 0.0001

Description placeholder

`CAPPOTBASIS`

Type Bool

Default value No

Description placeholder

`CAPPOTLINESEARCH`

Type Bool

Default value No

Description placeholder

`CAPRADIUS`

Type Float

Default value 3.0

Description placeholder

`CJCORR`

Type Float

Default value 0.1

Description Option to switch on a long-distance correction

`Coulomb`

Type Bool

Description Neglecting completely vt[rhoA,rhoB] (vt[rhoA,rhoB] equals zero) together with the exchange-correlation component of the embedding potential introduced by Wesolowski and Warshel.

`Dipole`

Type Bool

Default value No

Description placeholder

`E00`

Type Bool

Description placeholder

`EIGENSHIFT`

Type Float

Default value 0.01

Description placeholder

`ENERGY`

Type Bool

Default value No

Description placeholder

`EXTERNALORTHO`

Type Float

Default value 1000000.0

Description Used to specify the use of external orthogonality (EO) in the FDE block

`EXTPRINTENERGY`

Type Bool

Default value No

Description placeholder

`FULLGRID`

Type Bool

Default value No

Description placeholder

FreezeAndThawCycles

Type Integer

Description This keyword duplicates RelaxCycles

FreezeAndThawDensType

Type String

Description placeholder

FreezeAndThawPostSCF

Type Bool

Description This keyword duplicates RelaxPostSCF

GGA97

Type Bool

Description placeholder

GGAPotCFD

Type String

Description The correlation approximant is used in the construction of the embedding potential. The same correlation approximants as in the XC key are available.

GGAPotXFD

Type String

Description The exchange approximant is used in the construction of the embedding potential. The same exchange approximants as in the XC key are available.

LAMBDATIKH

Type Float

Default value 0.1

Description placeholder

LBDAMP

Type Float

Default value 0.25

Description placeholder

LBMAXSTEP

Type Float

Default value 0.05

Description placeholder

LLP91

Type Bool

Description placeholder

LLP91S

Type Bool

Description placeholder

`NDSD`

Type String

Description placeholder

`NOCAPSEPCONV`

Type Bool

Description placeholder

`NOFDKERN`

Type Bool

Default value Yes

Description placeholder

`OL91A`

Type Bool

Description placeholder

`OL91B`

Type Bool

Description placeholder

`ONEGRID`

Type Bool

Default value No

Description placeholder

`P92`

Type Bool

Description placeholder

`PBE2`

Type Bool

Description placeholder

`PBE3`

Type Bool

Description placeholder

`PBE4`

Type Bool

Description placeholder

`PDFT`

Type Bool

Default value No

Description placeholder

`PRINTEACHCYCLE`

Type Bool

Default value No

Description placeholder

`PRINTRHO2`

Type Bool

Default value No

Description placeholder

`PW86K`

Type Bool

Description placeholder

`PW91K`

Type Bool

Description placeholder

`PW91Kscaled`

Type Bool

Description placeholder

`RHO1FITTED`

Type Bool

Default value No

Description placeholder

`RelaxCycles`

Type Integer

Default value 5

Description This gives the maximum number of freeze-and-thaw cycles that are performed for this fragment. If the maximum number given in the FDE block is smaller, or if convergence is reached earlier, then fewer cycles are performed.

`RelaxDensType`

Type String

Default value

Description placeholder

`RelaxPostSCF`

Type Bool

Default value No

Description this option is included, several post-SCF properties will be calculated after each freeze-and-thaw cycle. These are otherwise only calculated in the last cycle.

`SCFCONVTHRESH`

Type Float

Default value 0.001

Description placeholder

`SDFTEnergy`

Type Bool

Default value No

Description placeholder

`SHORTPRINTENERGY`

Type Bool

Default value No

Description placeholder

`SMALLEIGTHRESH`

Type Float

Default value 0.0001

Description placeholder

`TF9W`

Type Bool

Description placeholder

`THAKKAR92`

Type Bool

Description placeholder

`THOMASFERMI`

Type Bool

Description Local-density-approximation form of vt[rhoA,rhoB] derived from Thomas-Fermi expression for Ts[rho]

`TW02`

Type Bool

Description placeholder

`WEIZ`

Type Bool

Description placeholder

`XCFun`

Type Bool

Default value No

Description Use XCFUN for nonadditive functionals

XCNAdd

Type String

Description

FitExcit

Type Bool

Default value No

Description

ForceALDA

Type Bool

Default value No

Description In spin-flip TDDFT, the XC kernel can be calculated directly from the XC potential. To use the LDA potential for the XC kernel, which roughly corresponds to the ALDA in ordinary TDDFT, one must specify the key

Fragments

Type Non-standard block

Description Definitions of the fragment type/files: {FragmentName FragmentFile}. In the block header one can specify the directory where the fragment files are located

FragMetaGGAToten

Type Bool

Default value No

GUI name XC energy difference (for meta XCs): Use molecular grid

Description By setting this to true the difference in the metahybrid or metagga exchange-correlation energies between the molecule and its fragments will be calculated using the molecular integration grid, which is more accurate than the default, but is much more time consuming.

FragOccupations

Type Non-standard block

Description Simulation of unrestricted fragments with the key FRAGOCCUPATIONS. Fragments need to be calculated spin-restricted. One can specify occupation numbers as if these fragments are calculated spin-unrestricted. The sum of spin-alpha and spin-beta occupations must, for each fragment orbital in each irrep separately, be equal to the total spin-restricted occupation of that orbital in the fragment.

FullFock

Type Bool

Default value No

GUI name Full Fock matrix: Always

Description Calculate the full Fock matrix each SCF iteration (instead of the difference with the previous cycle).

FullTotEn

Type Bool

Default value No

Description

Fuzzy_BO

Type Bool

Default value No

Description

GPU

Type Block

Description Set GPU options

Enabled

Type Bool

Default value No

GUI name Use GPU

Description Use a CUDA-compatible GPU.

UseDevices

Type Integer List

GUI name Only use devices

Description Use only specified devices for this calculation. Multiple devices will be distributed evenly among MPI ranks.

GUIBonds

Type Non-standard block

Description The bonds used by the GUI (this does not affect the ADF calculation in any way)

GW

Type Block

Description Instruct ADF to perform a G0W0 calculation.

AdaptiveMixing

Type Float List

Description Requests to use adaptive mixing instead of DIIS and sets the staring mixing parameter for mixing of Green's function in case of self-consistency.

Adapative mixing is recommenened in case a qsGW calculation does not converge with DIIS.

It is ignored in non-selfconsistent calculation and overwritten by DIIS when DIIS is also present.

AnalyticalIntegration

Type Block

Description Use analytical integration to calculate the self-energy. Very slow, unless the system is very small but useful to check the accuracy of the frequency integration

Enabled

Type Bool

Default value No

GUI name analytical integration

Description Enable the calculation of the GW quasi-particle energies via analytical integration.

`Polarizability`

Type Multiple Choice

Default value RPA

Options [RPA, BSE]

Description Sets the expression for the Polarizability used in the GW calculation.

RPA is the Default and amounts to a standard GW calculation.

BSE denotes screening in the Bethe-Salpeter-equation formalism.

`PrintSpectralFunction`

Type Bool

Default value No

Description Plot the self-energy as a function of freuency. Aotumatically done in case of analytical continuation. However, this is expensive in the analytical integration formalism.

`SpectralFunctionResolution`

Type Integer

Default value 800

Description Number of points at which spectral function is evaluated.

`TDA`

Type Bool

Default value No

Description Solve the linear reponse equations in the Tamm-Dancoff approximation.

`eta`

Type Float

Default value 0.001

Description Artificial (positive) broadening parameter for evaluation of self-energy in analytical integration.

Ideally should be as small as possible but this might lead to convergence issues in partially self-consistent approaches.

In this case, a value of up to 0.1 is possible.

`Converge`

Type Block

Description Sets convergence criteria for the GW calculation in self-consistent case

`Density`

Type Float List

Default value [1e-08, 1e-05]

Description First Criterion for self-consistency procedure to terminate.

Criterion is the trace of the density matrix. Ignored in non-selfconsistent Calculation and in eigenvalue self-consistent GW

It is possible to run a qsGW calculation with an inner SCF loop which updates the static part of the elf-energy only. This can be useful to accelerate the convergence in case linear mixing is used. It is not recommended to use linear mixing, so it is also not recommened to use that inner loop as well. The second number in this list specifies the convergence criterion for the inner SCF loop.

HOMO

Type Float

Default value 0.003

Unit eV

GUI name HOMO energy convergence

Description Criterion for self-consistency procedure to terminate.

The self-consistent GW calculation terminates, when the difference between the HOMO QP energies between 2 consecutive iterations is below this number.

The LUMO energy converged faster than the HOMO energy so when the HOMO energy is converged according to this criterion, the LUMO energy will be converged as well.

In non-selfconsistent Calculation, this criterion is ignored.

DIIS

Type Integer

Default value 10

Description Requests to use DIIS. This is the Default. Number of expansion coefficients can be requested as well. Ignored in non-selfconsistent calculation

Enabled

Type Bool

Default value No

GUI name Calculate GW quasi-particle energies

Description Enable the calculation of the GW quasi-particle energies.

FixedGrids

Type Bool

Default value No

Description In a self-consistent GW calculation, do not recalculate Grids. Can be useful in case of convergence problems. Only relevant for qsGW and qsGW0. In case of evGW and evGW0, the grids are always kept fixed.

LinearMixing

Type Float List

Description Requests to use linear mixing instead of DIIS and sets the mixing parameter for linear mixing of Green's function in case of self-consistency.

It is ignored in non-selfconsistent calculation and overwritten by DIIS when DIIS is also present.

`LinearizeQPequations`

Type Bool

Default value No

Description Instead of solving the non-linear QP equations in a G0W0 (or evGW calculation) by bisection exacly, linearize them by first-order Taylor expansion.

This is not recommended since it does not save computational time when used together with analytical continuation (as implemented in AMS). It might however be useful for benchmarking or for validating results.

If the results os the linearization differ by a lot (for instance, more than 0.1 eV in frontier QP energies) from the non-linearized results, this might indicate that the GW calculation is not reliable.

`OffDiagonalEFermi`

Type Bool

Default value No

Description Analytically continue the off-diagonal elements of the KSF2 qsGW Hamiltonian at the Fermi-energy instead of omega=0. Typically leads to slightly lower QP energies, i.e. higher ionization potentials. The HOMO-LUMO gaps are typically not affected.

`PrintAllSolutions`

Type Bool

Default value No

Description Print out all solutions for all requested states. Detects multiple solutions of the QP equations.

`QPHamiltonian`

Type Multiple Choice

Default value KSF2

Options [KSF1, KSF2, SRG, LQSGW]

Description The quasi-particle Hamiltonian can be constructed in different ways.

KSF1 refers to the original construction by Kotani, Van Schilfgaarde anf Faleev (KSF) which is also implemented in TURBOMOLE.

KSF2 refers to an alternative construction by KSF.

KSF1 is not recommended since it is numerically less stable than KSF2. The results are typically very similar.

The QP energies at which the matrix elements are evaluated can be tweaked further, see the two subsequent keys: However, KSF2 is recommended since it typically leads to QP energies with the best agreement with experiment.

Ignored when not a quasi-particle self-consistent GW calculation is performed

`ScissorShift`

Type Bool

Default value No

Description Only calculate the HOMO and LUMO QP energies and shift the remaining QP energies by the same amount.

This is a rather crude approximaiton and not recommended.

It might again be useful for benchmarking purposes.

SelfConsistency

Type Multiple Choice

Default value G0W0

Options [G0W0, EVGW0, EVGW, QSGW0, QSGW]

Description Sets the level of self-consistency in a GW calculation.

G0W0 calculates a one-shot, perturbative correction to the KS eigenvalues.

In evGW and evGW0, the quasi-particle energies are updated until self-consistency is reached.

evGW0 requests that the Green's function is evaluated self-consistently but not the screened interaction.

In qsGW, the density is updated as well, however, the self-energy is mapped to a static effective potential and the Dyson equation is solved by diagonalization instead of inversion. The results of a qsGW are independent of the choice of the underlying exchange-correlation functional and are usually the most accurate ones.

The same is done in qsGW0, but the screened interaction is not updated.

SelfEnergy

Type Multiple Choice

Default value GW

Options [HF, GW, G3W2, SOSEX, GWGamma, G3W2dynamic]

Description Controls the form of the self-energy.

GW is the default and corresponds to the standard GW calculation.

G3W2 is a GW calculation plus a perturbative second-order statically screened exchange correction (second order expansion in the self-energy). Note, that there the self-energy is always static.

nIterations

Type Integer List

Default value [10]

GUI name Number of iterations

Description The maximum number of iterations within the (partially or fully) self-consistent GW calculation has to converge.

Ignored when Formalism is set to G0W0

nLowest

Type Integer

Default value 1

GUI name N Lowest

Description Number of lowest occupied QP levels to be evaluated, overwrites nStates'

`nStates`

Type Integer

Default value 5

GUI name N states

Description Number of Quasiparticle States to be printed to output.

The default is 5 states which in this case means that min(5, Number of particle states) occupied and min(5, Number of hole states) hole states are printed. The whole list of states can be printed by setting this parameter to -1'

`preconditionQSGW`

Type Bool

Default value No

Description If true, the QSGW equations are solved but prior to each diagonalization, i.e. a G0W0 calculation is performed to find the optimal QP energies at which to analytically continue the self-energy.

This is in principle a more consistent construction than KSF1 or KSF2 since the diagonal elements are consistent with G0W0.

In KSF1 and KSF2, the diagonal elements are evaluated at the QP energies from the previous iteration which is equivalent to a zeroth-order Taylor expansion of the diaognal elements around the previous QP energies.Enabling this option typically leads to slightly lower QP energies.

`GZip`

Type String

Description GZip the corresponding tape (possibly working only for TAPE21)

`HartreeFock`

Type Bool

Default value No

Description Compute hybrid meta-GGA energy functionals (if METAGGA key is True)

`HFAtomsPerPass`

Type Integer

Description Memory usage option for old HF scheme

`HFMaxMemory`

Type Integer

Description Memory usage option for old HF scheme

`hydrogenbonds`

Type Bool

Default value No

Description Option for SFO population analysys to print small numbers.

`IgnoreOverlap`

Type Bool

Default value No

Description Expert option. Ignore that atoms are close.

`ImportEmbPot`

Type String

Description File containing an external embedding potential (FDE calculations only)

`ImportGrid`

Type String

Description FDE option for importing numerical integration grid.

`Integration`

Type Non-standard block

Description Options for the obsolete Voronoi numerical integration scheme

`IQA`

Type Block

Description Total energy decomposition based on the interacting quantum atoms (IQA) approach and using QTAIM real-space partition.

`AtomsToDo`

Type Integer List

GUI name Include atoms

Description Define a subset of atoms for which the IQA atom-atom interactions are calculated (no intra-atomic terms).

If left empty, all atoms will be included (full IQA).

`Enabled`

Type Bool

Default value No

GUI name Calculate: Interacting Quantum Atoms (IQA)

Description Calculate the total energy decomposition using the interacting quantum atoms (IQA) approach and the QTAIM real-space partitioning.

`Print`

Type Multiple Choice

Default value verbose

Options [normal, verbose]

Description Minimal output (default) or verbose mode (detailed energy decomposition)

`IrrepOccupations`

Type Non-standard block

Description Explicit occupation numbers per irrep

`IsotopicShift`

Type String

Description Untested

`LinearScaling`

Type Block

Description

`Cutoff_Coulomb`

Type Float

Description determines the radii for the fit functions in the evaluation of the (short-range part of) the Coulomb potential.

`Cutoff_Fit`

Type Float

Description determines how many atom pairs are taken into account in the calculation of the fit integrals and the density fit procedure. If the value is too low, charge will not be conserved and the density fitting procedure will become unreliable. This parameter is relevant for the timings of the FITINT and RHOFIH routines of ADF.

`Cutoff_Multipoles`

Type Float

Description determines the cut-offs in the multipole (long-range) part of the Coulomb potential

`HF_Fit`

Type Float

Description Parameter for HF exchange

`Overlap_Int`

Type Float

Description determines the overlap criterion for pairs of AOs in the calculation of the Fock-matrix in a block of points. Indirectly it determines what the cut-off radii for AO's should be. The value of ovint has a strong influence on the timing for the evaluation of the Fock matrix, which is very important for the overall timings

`ProgConv`

Type Float

Description determines how the overall accuracy changes during the SCF procedure

`LocOrb`

Type Non-standard block

Description The computation of localized orbitals is controlled with this block-type key

`MBPT`

Type Block

Description Technical aspects of the MP2 algorithm.

`Dependency`

Type Bool

Default value Yes

Description If true, to improve numerical stability, almost linearly-dependent combination of basis functions are removed from the Green's function that are used in the MBPT equations. Disabeling this key is stringly discouraged. Its value can however be changed. The key to adjust this value is RiHartreeFock%DependencyThreshold

`ExcludeCore`

Type Bool

Description If active, excludes core states from the calculation of the optiomal imaginary time and frequency grids.

The core states are still included in all parts of the calculations.

In case a frozen care calculation is performed, this key is ignored.

For MP2 and double hybrid calculation, it defauls to false. For RPA and GW calculations, it defaults to true.

`FitSetQuality`

Type Multiple Choice

Default value Auto

Options [Auto, VeryBasic, Basic, Normal, Good, VeryGood]

Description Specifies the fit set to be used in the MBPT calculation.

'Normal' quality is generally sufficient for basis sets up to and including TZ2P.

For larger basis sets (or for benchmarking purposes) a 'VeryGood' fit set is recommended. Note that the FitSetQuality heavily influences the computational cost of the calculation.

If not specified or 'Auto', the RIHartreeFock%FitSetQuality is used.

`Formalism`

Type Multiple Choice

Default value Auto

Options [Auto, RI, LT, All]

Description Specifies the formalism for the calculation of the MP2 correlation energy.

'LT' means Laplace Transformed MP2 (also referred to as AO-PARI-MP2),

'RI' means that a conventional RI-MP2 is carried out.

If 'Auto', LT will be used in case of DOD double hybrids and SOS MP2, and RI will be used in all other cases.

'All' means that both RI and LT formalisms are used in the calculation.

For a RPA or GW calculation, the formalism is always LT, irrespective of the formalism specified with this key.

`FrequencyGridType`

Type Multiple Choice

Default value LeastSquare

Options [LeastSquare, GaussLegendre]

Description Use Gauss-legendre grid for imaginary frequency integration in RPA and GW calculations instead of the usually used Least-Square optimized ones. Has the advantage that it can be systematically converged and an arbitrary number of grid points can be used. Typically more grid points will be needed to get the same level of accuracy. However, the convergence of the results with the size of the grid can be more systematic. These grids can only be used when Formalism is set to RI.

`IntegrationQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood]

Description Specifies the integration quality to be used in the MBPT calculation. If not specified, the RIHartreeFock%IntegrationQuality is used.

`SigmaFunctionalParametrization`

Type Multiple Choice

Default value S1re

Options [W1, W2, S1, S2, S1re]

Description Only relevant if a sigma-functional calculation is performed. Possible choices for the parametrization of the sigma-functional. Not all options are supported for all functionals.

`ThresholdQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description Controls the distances between atomic centers for which the product of two basis functions is not fitted any more. Especially for spatially extended, large systems, 'VERYBASIC' and 'BASIC' can lead to large computational savings, but the fit is also more approximate. If not specified, the RIHartreeFock%ThresholdQuality is used.

`UseScaledZORA`

Type Bool

Default value Yes

Description If true, use the scaled ZORA orbital energies instead of the ZORA orbital energies in the MBPT equations.

`frozencore`

Type Bool

Default value No

Description Freeze core states in correlation part of MBPT calculation

`nCore`

Type Integer

Default value 0

GUI name Number of core levels

Description Number of core states which will be excluded from the correlated calculation.

Will be ignored if frozencore is false.

In case nothing is specified, the number of core levels will be determined automatically.

Needs to be smaller than the number of occupied states.

`nFrequency`

Type Integer

Default value 12

GUI name Number of frequency points

Description Number of imaginary frequency points. This key is only relevant for RPA and GW and will be ignored if used in an AO-PARI-MP2 calculation. 12 Points is the default for a RPA calculation. It is technically possible to use a different number of imaginary frequency points than for imaginary time. The maximum number of points which can be requested for imaginary frequency integration is 42. Important note: The computation time and memory requirements roughyl scale linearly with the number of imaginary frequency points. However, memory can be an issue for RPA and GW when the number of imaginary frequency points is high. In case a job crashes, it is advised to increase the number of nodes since the necessary memory distributes over all nodes.

`nFrequencyG3W2`

Type Integer

Default value 32

GUI name Number of frequency points for G3W2 integration

Description Number of imaginary frequency points for G3W2 integration

`nLambda`

Type Integer

Default value 1

GUI name Number of lambda points

Description Size of coupling constant integration grid for SOSEX variants in RPA. Default is 4 points

`nTime`

Type Integer

GUI name Number of time points

Description Number of imaginary time points (only relevant in case the Laplace Transformed (LT) formalism is used).

In the many-body-perturbation theory module in ADF, the polarizability (or Kohn-Sham density response function) is evaluated in imaginary time to exploit sparsity in the AO basis. For MP2, this is often referred to as a Laplace transform. For MP2, 9 points are the default. This is a safe choice, guaranteeing accuracies higher than 1 Kj/mol for most systems (For many simple organic systems, 6 points are sufficient for good accuracy).

Only for systems with a very small HOMO-LUMO gap or low-lying core states (heavy elements starting from the 4th row of the periodic table) more points might be necessary.

In principle, the same considerations apply for RPA and GW as well, however, the accuracy requirements are somewhat higher and 12 point are the default for RPA. In a GW calculation, the number of points is adjusted according to the numerical quality. Using less than 9 points is strongly discouraged except for the simplest molecules.

In ADF2019, it can happen that the algorithm determining the imaginary time grid does not converge. In this case, the usual reason is that the number of points is too small and more points need to be specified. Starting from AMS2020, this does not happen any more. In case the imaginary time grid does not converge, the number of points is automatically adjusted until it does.

The computation time of AO-PARI-MP2, RPA, and GW scales linearly with the number of imaginary time points.

`MetaGGA`

Type Bool

Default value No

Description

`ModifyExcitation`

Type Block

Description

`DipStrength`

Type Float

Description

`GRIMMEAEX`

Type Float

Description

`GRIMMEALPHA`

Type Float

Description

`GRIMMEBETA`

Type Float

Description

`GRIMMEDEMAX`

Type Float

Description

`GRIMMEPERTC`

Type Bool

Description

`GRIMMETPMIN`

Type Float

Description

`HighExcit`

Type Float

Description

`NOGRIMMEPERTC`

Type Bool

Description

`NOverlap`

Type Integer

Default value 0

Description

`OscStrength`

Type Float

Description Use only pairs of an occupied and virtual orbital as guess vectors, for which the oscillator strength of the single-orbital transition is larger than this value

`SetLargeEnergy`

Type Float

Default value 1000000.0

Unit Hartree

Description The orbital energies of the uninteresting occupied orbitals are changed to -epsbig Hartree, and the orbital energies of the uninteresting virtual orbitals are changed to epsbig Hartree

`SetOccEnergy`

Type Float

Description All occupied orbitals that have to be used will change their orbital energy to this value. In practice only useful if one has selected one occupied orbital energy, and one want to change this to another value. Default: the orbital energies of the occupied orbitals that are used are not changed.

`UseOccRange`

Type Float List

Unit Hartree

Description Use only occupied orbitals which have orbital energies between the two numbers.

`UseOccVirtNumbers`

Type Integer List

Description Use only pairs of an occupied and virtual orbital as guess vectors, for which in the sorted list of the orbital energy differences, the number of the single-orbital transition is between the two numbers.

`UseOccVirtRange`

Type Float List

Unit Hartree

Description Use only pairs of an occupied and virtual orbital, for which the orbital energy difference is between the two numbers

`UseOccupied`

Type Non-standard block

Description Use only the occupied orbitals which are specified

UseScaledZORA

Type Bool

Default value No

Description Use everywhere the scaled ZORA orbital energies instead of the ZORA orbital energies in the TDDFT equations. This can improve deep core excitation energies. Only valid if ZORA is used.

UseVirtRange

Type Float List

Unit Hartree

Description Use only virtual orbitals which have orbital energies between the two numbers

UseVirtual

Type Non-standard block

Description Use only the virtual orbitals which are specified

ModifyStartPotential

Type Non-standard block

Description Modify the starting spin-dependent potential for unrestricted calculations.

NoBeckeGrid

Type Bool

Default value No

Description If true ADF will use the Voronoi numerical integration grid.

NoFDEPot

Type Bool

Default value No

Description Expert FDE option.

NoPrint

Type String

Recurring True

Description The amount of printed output is regulated with the keys Print, NoPrint, EPrint and Debug.

NoSharedArrays

Type Bool

Default value No

Description To disable the use of shared memory.

NoSymFit

Type Bool

Default value No

Description Do not use only an A1 symmetric fit.

`NoTotEn`

Type Bool

Default value No

Description

`NuclearModel`

Type Multiple Choice

Default value PointCharge

Options [PointCharge, Gaussian]

Description Model for the nuclear charge distribution.

To see effects from your choice you will need to use a basis set with extra steep functions. For example you can find these in the ZORA/TZ2P-J basis directory.

`NumericalQuality`

Type Multiple Choice

Default value Normal

Options [Basic, Normal, Good, VeryGood, Excellent]

Description Set the quality of several important technical aspects of an ADF calculation (with the notable exception of the basis set). It sets the quality of: BeckeGrid (numerical integration) and ZlmFit (density fitting). Note: the quality defined in the block of a specific technical aspects supersedes the value defined in NumericalQuality (e.g. if I specify 'NumericalQuality Basic' and 'BeckeGrid%Quality Good', the quality of the BeckeGrid will be 'Good')

`Occupations`

Type String

Description Occupations options

`OPop_Analysis`

Type String

Description

`OrbitalsCoulombInteraction`

Type Integer List

Recurring True

Description Compute the Coulomb interaction energy between the density of two orbitals. After the key, specify the indices of the two orbitals for which you want to compute the Coulomb interaction energy. Can only be used for spin-restricted calculations. Cannot be used in case of Symmetry (use Symmetry NoSym).

`OrthFragPrep`

Type Bool

Default value No

Description Expert FDE option.

`PertLoc`

Type Block

Description Perturbed localized molecular orbitals, correct to first order in an applied field, can be calculated in case of AORESPONSE. Can be used if the applied field changes the density in first order.

`Alfa`

Type Bool

Default value No

Description Analyze the static or dynamic polarizability

`BField`

Type Bool

Default value No

Description The perturbation is a magnetic field. Should be consistent with AORESPONSE

`Beta`

Type Bool

Default value No

Description Analyze the optical rotation parameter beta. The relation to G' is beta = -G'/omega. The optical rotation parameter beta is calculated directly and has a well-defined static limit, i.e. omega can be zero or non-zero

`Diag`

Type Bool

Default value Yes

Description Only analyze the diagonal of the response tensor

`Dynamic`

Type Bool

Default value No

Description Should be used for a frequency dependent perturbation field.

`EField`

Type Bool

Default value Yes

Description The perturbation is an electric field

`Fulltens`

Type Bool

Default value No

Description The full tensor is analyzed

`GPrime`

Type Bool

Default value No

Description Analyze the G' (gyration) tensor, for optical rotation dispersion. Requires a frequency dependent perturbation field, with a frequency (omega) unequal to zero.

`Static`

Type Bool

Default value Yes

Description should be used for a static field

`PolTDDFT`

Type Block

Description POLTDDFT is a fast algorithm to solve the TDDFT equations in the space of the density fitting auxiliary basis set. The (real and imaginary part of the) diagonal of the polarizability tensor and rotatory strengths will be calculated, which can be used to calculate the photoabsorption and circular dichroism (CD) spectra.

`Analysis`

Type Bool

Default value No

Description An analysis of the absortion and CD spectrum in terms of single orbital transitions.

`CutOff`

Type Float

Default value 4.0

Unit eV

Description For a given point in the spectrum, only include pairs of an occupied and virtual orbital, for which the orbital energy difference is lower than the energy of the point in the spectrum plus cutoff.

`Enabled`

Type Bool

Default value No

GUI name UV/Vis and CD spectrum

Description Calculate UV/Vis and CD spectrum from the imaginary part of the polarizability tensor at any given photon energy. This avoids the bottleneck of Davidson diagonalization.

`FreqRange`

Type Float List

Default value [0.0, 5.0]

Unit eV

Description Specifies a frequency range of frequencies of incident light, the perturbing field, at which the complex dynamical polarizability will be calculated. 2 numbers: an upper and a lower bound. Use subkey NFreq to specify the number of frequencies.

`HDA_fitted`

Type Bool

Default value No

GUI name Fitted HDA

Description Use fit functions to calculate HDA (Hybrid diagonal approximation), only relevant for hybrids.

`Irrep`

Type Non-standard block

Description Subblock key for selecting which symmetry irreps of the excitations to calculate (all excitations by default). In the subkey data block list the symmetry irrep labels, like B1, for example

`KGrid`

Type Float

Default value 9.0

Unit eV

Description Keyword KGRID is used to discretize the energy scale for calculating the complex dynamical polarizability. Only pairs of an occupied and virtual orbital are included, for which the orbital energy difference is lower than this value. Use key NGRID to set the number of points within the energy grid.

`Lambda`

Type Float

Default value 1.0

Description Jacob's scaling factor for the study of plasmonic resonances.

This factor, 0<lambda<1, turns on the coupling matrix K.

`Lifetime`

Type Float

Default value 0.1

Unit eV

Description Specify the resonance peak width (damping).

Typically the lifetime of the excited states is approximated with a common phenomenological damping parameter. Values are best obtained by fitting absorption data for the molecule, however, the values do not vary a lot between similar molecules, so it is not hard to estimate values.

`NFreq`

Type Integer

Default value 100

Description NFreq is the number of frequencies of incident light, the perturbing field, at which the complex dynamical polarizability will be calculated. Use FreqRange to specify the frequency range.

`NGrid`

Type Integer

Default value 180

Description Ngrid is the number of points within the energy grid.

`N_FitOrb`

Type Integer

Default value 1000000000

Description The number of vectors containing the coefficients we use to expand the projection of each fitting function over the electron density (of a particular molecular orbital) as a linear combination of overlap matrices between fitting functions pair

`N_HDA_integral`

Type Integer

Default value 1000000000

Description

`N_SubMatricesAk`

Type Integer

Default value 1000000000

Description

`Print_Int_Time`

Type Integer

Default value 0

Description Print detailed timing during calculation of integrals of Tape63 and Tape64

`RegionsForAnalysis`

Type String

Description Names of regions for analysis per region using the fragment projection analysis approach. Will split the absorption and CD spectrum in region_i -> region_j terms.

`Velocity`

Type Bool

Default value No

GUI name Velocity representation

Description If True, ADF calculates the dipole moment in velocity gauge.

If false: dipole-length representation is used

`Print`

Type String

Recurring True

Description The amount of printed output is regulated with the keys Print, NoPrint, EPrint and Debug.

`QMFQ`

Type Block

Description Block input key for QM/FQ(FMu).

`AtomType`

Type Block

Recurring True

Description Definition of atomic types in MM environment

`Alpha`

Type Float

Description Polarizability of FQFMU atom

`Charge`

Type Float

Description MM fixed charge (non-polarizable only)

`Chi`

Type Float

Description Electronegativity of FQ atom

`Eta`

Type Float

Description Chemical Hardness of FQ atom

`Symbol`

Type String

Description Symbol associated with atom type

`Coords`

Type Non-standard block

Description Coordinates and fragment information (FQ only)

`FDERESP`

Type Bool

Default value No

Description In response calculations (TD), the polarization contribution of the FDE part is introduced at the FQ level [See F. Egidi et al. J. Chem. Phys. 2021, 154, 164107].

`Forcefield`

Type Multiple Choice

Default value FQ

Options [FQ, FQFMU]

Description Version of the FQ family of polarizable forcefields

`Frozen`

Type Bool

Default value No

Description Expert option. Do not introduce polarization effect in response calculations.

`Kernel`

Type Multiple Choice

Default value OHNO

Options [OHNO, COUL, GAUS]

Description Expert option. KERNEL can be used to choose the functional form of the charge-charge interaction kernel between MM atoms. Recommended is to use the default OHNO. The COUL screening is the standard Coulomb interaction 1/r. The OHNO choice introduce the Ohno functional (see [K. Ohno, Theoret. Chim. Acta 2, 219 (1964)]), which depends on a parameter n that is set equal to 2. Finally, the GAUS screening models each FQ charge by means of a spherical Gaussian-type distribution, and the interaction kernel is obtained accordingly. For QM/FQFMU only GAUS SCREEN is implemented.

`MolCharge`

Type Float

Default value 0.0

Description Total charge of each fragment (FQ only)

`QMSCREEN`

Type Multiple Choice

Default value GAUS

Options [ERF, EXP, GAUS, NONE]

Description Expert option. QMSCREEN can be used to choose the functional form of the charge-charge interaction kernel between MM atoms and the QM density. The screening types available are ERF (error function), EXP (exponential), GAUS (Gaussian), or NONE. The default is GAUS.

`QMSCREENFACTOR`

Type Float

Default value 0.2

Description Expert option. Sets the QM/MM interaction kernel screening length. Recommended is to use the default value 0.2 with the GAUS QM/MM screening function.

`QTAIM`

Type Block

Description This block is used to request a topological analysis of the gradient field of the electron density, also known as the Bader's analysis. If this block is specified without any sub-key, only local properties are calculated.

`AnalysisLevel`

Type Multiple Choice

Default value Normal

Options [Normal, Extended, Full]

Description Set the level of the QTAIM analysis:

Normal - topology analysis and properties at the density critical points,

Extended - same as Normal plus condensed atomic descriptors,

Full - same as Extended plus non-local descriptors.

`AtomsToDo`

Type Integer List

GUI name Include atoms

Description List of atoms for which condensed descriptors are to be calculated. By default all atoms are included.

`Enabled`

Type Bool

Default value No

GUI name Perform QTAIM analysis

Description Calculate QTAIM (also known as Bader) properties.

`Energy`

Type Bool

Default value No

GUI name Atomic energies

Description Calculate atomic energies. Requires an all-electron calculation (no frozen core), triggers the TotalEnergy and increases the [AnalysisLevel] to at least Extended.

`Source`

Type Bool

Default value No

GUI name Source Function

Description Calculate the Source Function at BCPs and RCPs.

`Spacing`

Type Float

Default value 0.5

Unit Bohr

Description Specifies spacing of the initial Cartesian grid when searching for critical points. It may be useful to specify a smaller value than the default if some critical points are missed. This will result in a more accurate but slower calculation.

`QTens`

Type Bool

Default value No

Description Calculate the the Nuclear Electric Quadrupole Hyperfine interaction (Q-tensor, NQCC, NQI), related to the Electric Field Gradient (EFG).

`RadialCoreGrid`

Type Block

Description For each atom the charge densities and the coulomb potentials of frozen core and valence electrons are computed in a radial grid. The radial grid consists of a sequence of r-values, defined by a smallest value, a constant multiplication factor to obtain each successive r-value, and the total number of points. Equivalently it can be characterized by the smallest r-value, the largest r-value, and the number of points; from these data the program computes then the constant multiplication factor.

`NRad`

Type Integer

Default value 5000

Description The number of radial grid points

`RMax`

Type Float

Default value 100.0

Unit Angstrom

Description The largest distance in the radial grid

`RMin`

Type Float

Default value 1e-06

Unit Angstrom

Description The shortest distance used in the radial grid

`Relativity`

Type Block

Description Options for relativistic effects.

`Formalism`

Type Multiple Choice

Default value ZORA

Options [Pauli, ZORA, X2C, RA-X2C]

Description Note that if Level is None, no relativistic effects are taken into account, irrespective of the chosen formalism.

Pauli stands for the Pauli Hamiltonian.

ZORA means the Zero Order Regular Approximated Hamiltonian, recommended.

X2C and RA-X2C both stand for an exact transformation of the 4-component Dirac equation to 2-components.

X2C is the modified Dirac equation by Dyall.

RA-X2C is the regular approach to the modified Dirac equation.

`Level`

Type Multiple Choice

Default value Scalar

Options [None, Scalar, Spin-Orbit]

GUI name Relativity

Description None: No relativistic effects.

Scalar: Scalar relativistic. This option comes at very little cost.

Spin-Orbit: Spin-orbit coupled. This is the best level of theory, but it is (4-8 times) more expensive than a normal calculation. Spin-orbit effects are generally quite small, unless there are very heavy atoms in your system, especially with p valence electrons (like Pb).

See also the SpinOrbitMagnetization subkey.

`Potential`

Type Multiple Choice

Default value MAPA

Options [MAPA, SAPA]

Description Starting from ADF2017 instead of SAPA (the Sum of neutral Atomical potential Approximation) MAPA is used by default for ZORA. The MAPA (the Minumium of neutral Atomical potential Approximation) at a point is the minimum of the neutral Atomical potentials at that point. Advantage of MAPA over SAPA is that the gauge dependence of ZORA is reduced. The ZORA gauge dependency is small for almost all properties, except for the electron density very close to a heavy nucleus. The electron density very close to a heavy nucleus can be used for the interpretation of isomer shifts in Mossbauer spectroscopy.

`SpinOrbitMagnetization`

Type Multiple Choice

Default value CollinearZ

Options [NonCollinear, Collinear, CollinearX, CollinearY, CollinearZ]

Description Relevant only for spin-orbit coupling and if unrestricted key has been activated.

Most XC functionals have as one ingredient the spin polarization in case of unrestricted calculations. Normally the direction of the spin quantization axis is arbitrary and conveniently chosen to be the z-axis.

However, in a spin-orbit calculation the direction matters, and it is arbitrary to put the z-component of the magnetization vector into the XC functional. There is also the exotic option to choose the quantization axis along the x or y axis.

It is also possible to plug the size of the magnetization vector into the XC functional. This is called the non-collinear approach.

- NonCollinear: the non-collinear method. - CollinearXYZ: use the x, y, or z component as spin polarization for the XC functional. - Collinear: the same as CollinearZ.

`RemoveAllFragVirtuals`

Type Bool

Default value No

Description Remove all virtual fragment orbitals.

`RemoveFragOrbitals`

Type Non-standard block

Description Block key to remove selected virtual fragment orbitals.

`RemoveOtherFragVirtuals`

Type Bool

Default value No

Description Remove all virtual fragment orbitals, except on first fragment.

`Response`

Type Block

Description The calculation of frequency-dependent (hyper)polarizabilities and related properties (Raman, ORD)

`ALLCOMPONENTS`

Type Bool

Description

`ALLHYPER`

Type Bool

Description

`ALPHAINANG`

Type Bool

Description

`ANALYTIC`

Type Bool

Description

`AllCycles`

Type Bool

Default value No

Description Convergence printout

`AllTensor`

Type Bool

Default value No

Description Higher dispersion coefficients are also calculated

`C8`

Type Bool

Description

`CUTTAILS`

Type Bool

Description

`DYNAHYP`

Type Bool

Description

`Dipole`

Type Bool

Description

`DmpDII`

Type Float

Default value 0.8

Description

`DmpRsp`

Type Float

Default value 0.9

Description

`ERABSX`

Type Float

Default value 1e-06

Description

`ERRALF`

Type Float

Default value 1e-05

Description

`ERRTMX`

Type Float

Default value 1e-06

Description

`EpsRho`

Type Float

Description Rho threshold

`FXCALPHA`

Type Float

Description

`FXCDRCONV`

Type Bool

Description

`FXCLB`

Type Bool

Description

`Frequencies`

Type Float List

Default value [0.0]

Unit eV

Description List of frequencies of incident light, the perturbing field, at which the time-dependent properties will be calculated.

`GXCALPHA`

Type Float

Description

`HyperPol`

Type Float

Default value 0.0

Unit Hartree

Description

`IFILES`

Type Integer

Default value 0

Description Integration run including external files. Used for Van der Waals dispersion coefficients calculations.

`IPRESP`

Type Integer

Default value 1

Description

`IReal`

Type Integer

Default value 1

Description

`KSORBRUN`

Type Bool

Description

`MAGNETICPERT`

Type Bool

Description

`MAXWAALS`

Type Integer

Default value 8

Description

`NCycMx`

Type Integer

Default value 30

Description

`NOFXCDR`

Type Bool

Description

`NUMERIC`

Type Bool

Description

`OPTICALROTATION`

Type Bool

Description

`Octupole`

Type Bool

Description

`Quadrupole`

Type Bool

Description

`Raman`

Type Bool

Description

`STARTREALGR`

Type Bool

Description

`SYMRUN`

Type Bool

Description

`Temperature`

Type Float

Default value 300.0

Unit Kelvin

Description Wavelength of incoming light is equal to the wavelength at which the calculation is performed and temperature is equal to room temperature (300K) Total Raman band is default, not the Q-branch of diatomic. (Relevant for Raman scattering cross section)

`VANDERWAALS`

Type Integer

Description

`VERDET`

Type Float

Default value 0.01

Description For numerical differentiation d alfa(omega) /d omega, needed for Verdet constant, the default frequencies are omega + dverdt and omega - dverdt

`ResponseFormalism`

Type Multiple Choice

Default value Auto

Options [Auto, Response, AOResponse]

Description Set to RESPONSE or AORESPONSE.

`Restart`

Type Block

Description Options for restarts

`NoOrb`

Type Bool

Default value No

GUI name Ignore orbitals

Description Do not use orbitals from the restart file

`NoSCF`

Type Bool

Default value No

GUI name Ignore SCF fit coefficients

Description Do not use any fit coefficients from the restart file as a first approximation to the (fitted) SCF density for the new calculation. Instead, the sum-of-fragments density will be used, as in a non-restart run. Note, typically noSCF should be used in combination with noORB.

`NoSmear`

Type Bool

Default value No

GUI name Ignore smearing

Description Do not use any electron smearing data from the restart file.

`SpinFlip`

Type Integer List

GUI name Spin flip on restart for

Description Select the atoms for which the spin is to be flipped upon restart.

`RESTOCC`

Type Bool

Default value No

Description

`RIHartreeFock`

Type Block

Description

`DependencyCoreRange`

Type Float

Description Basis functions may be given a core character based on the range. For now only active in Band and only if present in the input

`DependencyThreshold`

Type Float

Default value 0.001

Description To improve numerical stability, almost linearly-dependent combination of basis functions are removed from the Hartree-Fock exchange matrix.

If you obtain unphysically large bond energy in an Hybrid calculation, or an unphysically low correlation energy in an RPA, MP2, or double hybrid calculation, you might try setting the DependencyThreshold to a larger value (e.g. 3.0E-3)

Note, that in GW calculations and GW-BSE calculations the default for this key is 5.0e-3.

`FitGenerationDetails`

Type Block

Description Technical details about how the RI Hartree-Fock fit functions are generated.

`BoostL`

Type Bool

Default value No

Description Add extra max(l)+1 diffuse function

When l denotes the highest angular momentum present in the primary basis,

FromBasisProducts will generate auxiliary fit functions with up to 2l angular momentum.

When this key is set to true, the maximum angular momentum in the auxiliary fit set becomes 2l+2.

Typically, this option is not needed and when precision issues arise, it is rather advised to adjust the OneCenterDependencyThreshold key to a smaller value.

`LapackWorkAround`

Type Bool

Default value No

Description GetFitFunctionsForAtomType diagonalization done with Lapack instead of Scalapack

`Method`

Type Multiple Choice

Default value Auto

Options [Auto, FromBasisProducts]

Description The way in which fit functions are generated. The main distinction is whether it depends on the basis functions used.

When FromBasisProducts is used, the auxiliary basis is generated directly from the products of primary basis functions.

This has the advantage that the auxiliary fit adapts automatcally to the basis set size.

Especially for basis sets of QZ quality or larger, this is often necessary to otbain highly precise correlation energies using RPA or double hybrids

FromBasisProducts option is also useful for GW or BSE calculations with basis sets of QZ quality or larger.

`OneCenterDependencyThreshold`

Type Float

Default value 1e-08

Description This key is only active when FromBasisProducts is chosen as method to generate the auxiliary basis.

This threshold controls the size, and at the samw time, the precision of the auxiliary basis set. A smaller number leads to a larger auxiliary fit set.

The default value of 1e-8 is typically sufficient to converge correlation energies and QP energies to a very high precision.

It corresponds to an auxiliary basis which is typically 8-9 times larger than the primary basis.

`UseBandRadialGrid`

Type Bool

Default value Yes

Description Only applies to band. The band logarithmic grid ranges (by default) from 1e-6 to 100 with 3000 points. Otherwise 300 points will be used.

For 0-periodicity (molecules) it is advaisable to set this key to false since lots of memory is needed to evaluate all necessary integrals.

`FitSetQuality`

Type Multiple Choice

Default value Auto

Options [Auto, VeryBasic, Basic, Normal, Good, VeryGood, Excellent, FromBasisProducts]

Description The quality of auxiliary fit set employed in the RI scheme.

If ‘Auto’, the value of the RIHartreeFock Quality option will be used.

Normal quality is generally sufficient for basis sets up to and including TZ2P.

For larger basis sets (or for benchmarking purposes) a VeryGood fit set is recommended.

Note that the FitSetQuality heavily influences the computational cost of the calculation.

`IntegrationQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description Quality of the numerical integration for evaluating the integrals between basis functions and fit functions. If IntegrationQuality is not defined in input, the value defined in RIHartreeFock%Quality will be used.

`Quality`

Type Multiple Choice

Default value Auto

Options [Auto, VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description Numerical accuracy of the RI procedure. If 'Auto', the quality specified in the 'NumericalQuality' will be used.

`QualityPerRegion`

Type Block

Recurring True

Description Sets the fit-set quality for all atoms in a region. If specified, this overwrites the globally set quality.

`Quality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description This region's quality of the auxiliary fit set employed in the RI scheme.

`Region`

Type String

Description The identifier of the region for which to set the quality.

`ResponseQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description Numerical accuracy of the RI procedure for the Response module.

`ThresholdQuality`

Type Multiple Choice

Options [VeryBasic, Basic, Normal, Good, VeryGood, Excellent]

Description Linear scaling thresholds (also used for determining at what range the multiple approximation is used). To disable all linear scaling thresholds set this to Excellent.

`UseMe`

Type Bool

Default value Yes

Description Set to False if you want to use the old RI scheme (ADF only)

`RISM`

Type Non-standard block

Description 3D-RISM-related input keys.

`Save`

Type String

Recurring True

Description A sequence of file names separated by blanks or commas. Possible file names are TAPE10, TAPE13, TAPE14.

`scaledkinfunctionals`

Type Bool

Default value No

Description FDE option.

`SCF`

Type Block

Description Control aspects of the Self Consistent Field procedure

`AccelerationMethod`

Type Multiple Choice

Default value ADIIS

Options [ADIIS, fDIIS, LISTb, LISTf, LISTi, MESA, SDIIS]

Description SCF acceleration method.

The default method is ADIIS, which is actually a mix of A-DIIS and SDIIS: A-DIIS is used at the start of the SCF and SDIIS is used closer to convergence, with a smooth switching function.

The other methods are from the LIST family developed by Alex Wang and co-workers. They may perform better than the default in some situations.

Setting AccelerationMethod to SDIIS effectively disables A-DIIS and is equivalent to the legacy mixing+DIIS method.

`Converge`

Type Float List

Default value [1e-06, 0.001]

Description The criterion to stop the SCF updates.

The tested error is the commutator of the Fock matrix and the P-matrix (=density matrix in the representation of the basis functions) from which the F-matrix was obtained. This commutator is zero when absolute self-consistency is reached. Convergence is considered reached when the maximum element falls below SCFcnv and the norm of the matrix below 10*SCFcnv. The default is fairly strict.

A second criterion which plays a role when the SCF procedure has difficulty converging. When in any SCF procedure the currently applicable criterion does not seem to be achievable, the program stops the SCF. When the secondary criterion (sconv2) has been met, only a warning is issued and the program continues normally.

`DIIS`

Type Block

Description The maximum number of SCF cycles allowed.

`BFac`

Type Float

Default value 0.0

GUI name Bias DIIS towards latest vector with

Description By default, the latest vector is not favored in the DIIS algorithm (value 0.0). A sensible value would be 0.2.

`CX`

Type Float

Default value 5.0

GUI name Reduce DIIS space when coefs >

Description The DIIS space is reduced when very large DIIS coefficients appear. The value is the threshold.

`CXX`

Type Float

Default value 25.0

GUI name No DIIS (but damping) when coefs >

Description When very large DIIS coefficients appear, switch to traditional damping. The value is the threshold.

`Cyc`

Type Integer

Default value 5

GUI name Start DIIS anyway at cycle

Description When A-DIIS is disabled, the Pulay DIIS will start at this iteration irrespective of the DIIS OK value.

`N`

Type Integer

Default value 10

GUI name Size of DIIS space

Description The number of expansion vectors used for accelerating the SCF. The number of previous cycles taken into the linear combination is then n-1 (the new computed potential is also involved in the linear combination)

`Ok`

Type Float

Default value 0.5

GUI name Start DIIS when max [F,P] <

Description The Pulay DIIS starting criterion, when A-DIIS is disabled,

`Iterations`

Type Integer

Default value 300

GUI name Maximum number of SCF cycles

Description The maximum number of SCF cycles allowed.

`LShift`

Type Float

Default value 0.0

Unit Hartree

GUI name Level shift

Description The level shifting parameter.

The diagonal elements of the Fock matrix, in the representation of the orbitals of the previous iteration, are raised by vshift hartree energy units for the virtual orbitals. This may help to solve convergence problems when during the SCF iterations charge is sloshing back and forth between different orbitals that are close in energy and all located around the Fermi level.

Level shifting is not supported in the case of Spin-Orbit coupling.

At the moment properties that use virtuals, like excitation energies, response properties, NMR calculations, will give incorrect results if level shifting is applied.

`LShift_cyc`

Type Integer

Default value 1

Description Specifies that level shifting is not turned on before the given SCF cycle number (for the start-up geometry).

`LShift_err`

Type Float

Default value 0.0

Description Specifies that level shifting will be turned off by the program as soon as the SCF error drops below a threshold.

`MESA`

Type String

Description

`Mixing`

Type Float

Default value 0.2

GUI name Mixing (% new vector included)

Description When none of the SCF acceleration methods is active, the next Fock matrix is determined F = mixing * F_n + (1-mixing)F_(n-1).

`Mixing1`

Type Float

Default value 0.2

GUI name Mixing 1st SCF cycle

Description The mixing parameter at the 1st SCF cycle.

`OldSCF`

Type Bool

Default value No

Description Disable the default SCF algorithm and use the old SCF algorithm.

The default SCF improves performance for big systems on big machines (when your calculation uses many tasks).

It is also recommended for machines with slow disk I/O as it writes less data to disk.

The default convergence method supported is A-DIIS, but LISTi is also supported.

`ROSCF`

Type Block

Description Settings for the ROSCF method.

`Alpha`

Type Float List

Default value [0.5, 0.5, 0.5]

Description Coefficients to build the alpha-spin orbital contribution to the diagonal closed-, open-, and virtual-shell blocks of the Fock matrix. The beta-spin orbital contributions are 1.0 minus the alpha ones.

`SCRF`

Type Non-standard block

Description SCRF is no longer supported. Use AMS2023 or earlier.

`SelectExcitation`

Type Block

Description

`DipStrength`

Type Float

Description

`GRIMMEAEX`

Type Float

Description

`GRIMMEALPHA`

Type Float

Description

`GRIMMEBETA`

Type Float

Description

`GRIMMEDEMAX`

Type Float

Description

`GRIMMEPERTC`

Type Bool

Description

`GRIMMETPMIN`

Type Float

Description

`HighExcit`

Type Float

Description

`NOGRIMMEPERTC`

Type Bool

Description

`NOverlap`

Type Integer

Default value 0

Description

`OscStrength`

Type Float

Description Use only pairs of an occupied and virtual orbital as guess vectors, for which the oscillator strength of the single-orbital transition is larger than this value

`SetLargeEnergy`

Type Float

Default value 1000000.0

Unit Hartree

Description The orbital energies of the uninteresting occupied orbitals are changed to -epsbig Hartree, and the orbital energies of the uninteresting virtual orbitals are changed to epsbig Hartree

`SetOccEnergy`

Type Float

Description All occupied orbitals that have to be used will change their orbital energy to this value. In practice only useful if one has selected one occupied orbital energy, and one want to change this to another value. Default: the orbital energies of the occupied orbitals that are used are not changed.

`UseOccRange`

Type Float List

Unit Hartree

Description Use only occupied orbitals which have orbital energies between the two numbers.

`UseOccVirtNumbers`

Type Integer List

Description Use only pairs of an occupied and virtual orbital as guess vectors, for which in the sorted list of the orbital energy differences, the number of the single-orbital transition is between the two numbers.

`UseOccVirtRange`

Type Float List

Unit Hartree

Description Use only pairs of an occupied and virtual orbital, for which the orbital energy difference is between the two numbers

`UseOccupied`

Type Non-standard block

Description Use only the occupied orbitals which are specified

`UseScaledZORA`

Type Bool

Default value No

Description Use everywhere the scaled ZORA orbital energies instead of the ZORA orbital energies in the TDDFT equations. This can improve deep core excitation energies. Only valid if ZORA is used.

`UseVirtRange`

Type Float List

Unit Hartree

Description Use only virtual orbitals which have orbital energies between the two numbers

`UseVirtual`

Type Non-standard block

Description Use only the virtual orbitals which are specified

`SFTDDFT`

Type Bool

Default value No

GUI name Spin-flip excitations

Description Calculate spin-flip excitation energies (requires TDA and FORCEALDA keys).

`SharcOverlap`

Type Bool

Default value No

Description

`Skip`

Type String

Recurring True

Description Expert key. To restrict which parts of the program are actually executed.

`SlaterDeterminants`

Type Non-standard block

Description The calculation of the one-determinant states based on the AOC reference state is controlled with this key.

`Solvation`

Type Block

Description

`ARO`

Type Float

Description

`Acid`

Type Float

Description

`Ass`

Type Bool

Description

`Base`

Type Float

Description

`BornC`

Type Float

Description Coulomb constant for Born

`C-Mat`

Type String

Description

`COSKFAtoms`

Type Integer List

Recurring True

Description This subkey COSKFATOMS specifies for which nuclei the segments in the COSMO section of the COSKF file should be used. Default all nuclei should be used, i.e. as for omitting the subkey COSKFATOMS. The numbers refer to the input ordering in the ADF calculation.

`Charged`

Type String

Description

`Chgal`

Type Float

Description

`CsmRsp`

Type Bool

Description

`Cust`

Type String

Description

`Debug`

Type String

Description

`Disc`

Type String

Description

`Div`

Type String

Description

`EPS`

Type Float

Description

`ForceCosmo`

Type String

Description

`HALO`

Type Float

Description

`Lpr`

Type Bool

Description

`NoAss`

Type Bool

Description

`NoCsmRsp`

Type Bool

Description

`NoLpr`

Type Bool

Description

`NoPVec`

Type Bool

Description

`PVec`

Type Bool

Description

`PrintSM12`

Type Bool

Description

`RADII`

Type Non-standard block

Description

`RadSolv`

Type Float

Description

`Ref`

Type Float

Description

`SCF`

Type String

Description

`Solv`

Type String

Description Solvent details

`Surf`

Type Multiple Choice

Default value delley

Options [wsurf, asurf, esurf, klamt, delley, wsurf nokeep, asurf nokeep, esurf nokeep, klamt nokeep, delley nokeep]

Description Defines the type of cavity to be used.

`Tens`

Type Float

Description

`SOMCD`

Type Bool

Default value No

Description MCD option. Required for a calculation of MCD temperature-dependent C terms. The calculation must be an unrestricted and scalar relativistic ZORA.

`SOPert`

Type Block

Description Key for perturbative inclusion of spin-orbit coupling.

`EShift`

Type Float

Default value 0.2

Description The actually calculated eigenvalues are calculated up to the maximum singlet-singlet or singlet-triplet scalar relativistic excitation energy plus eshift (in Hartree).

`GSCorr`

Type Bool

Default value Yes

GUI name Include GS

Description The singlet ground state is included, which means that spin-orbit coupling can also have some effect on energy of the ground state. The spin-orbit matrix in this case is on basis of the ground state and the singlet and triplet excited states.

`NCalc`

Type Integer

Description Number of spin-orbit coupled excitation energies to be calculated. Default (and maximum) value: 4 times the number of scalar relativistic singlet-singlet excitations.

`sozero`

Type Bool

Default value No

Description Debug option to set spin-orbit matrix to zero.

`SpinPolarization`

Type Float

Description The spin polarization of the system, which is the number of spin-alpha electrons in excess of spin-beta electrons. Specification is only meaningful in a spin-unrestricted calculation. However, specification is not meaningful in an unrestricted Spin-Orbit coupled calculation using the (non-)collinear approximation.

`STContrib`

Type Bool

Default value No

Description For an analysis of spin-orbit coupled excitations in terms of scalar relativistic singlet and triplet excitations. In order to get this analysis one needs to perform a scalar relativistic TDDFT calculation of excitation energies on the closed shell molecule first, and use the resulting adf.rkf as a fragment in the spin-orbit coupled TDDFT calculation of excitation energies, including this keyword STCONTRIB.

`STOFit`

Type Bool

Default value No

Description Computation of the Coulomb potential with the pair fit method.

`StopAfter`

Type String

Description

`SubExci`

Type Block

Description Subsystem TDDFT (FDE)

`CICoupl`

Type Bool

Default value No

Description Within the Tamm-Dancoff Approximation, the couplings between localized excited states on different subsystems correspond directly to so-called exciton couplings. The CICOUPL keyword, in conjunction with TDA, prints these exciton couplings. It is also possible to use CICOUPL with full FDEc-TDDFT. In that case, the excitonic couplings between monomers are reconstructed from an effective 2x2 CIS-like eigenvalue problem.

`COULKERNEL`

Type Bool

Default value Yes

Description

`COUPLBLOCK`

Type Bool

Default value No

Description

`COUPLSYS`

Type Integer List

Description

`CPLTAPE`

Type String

Description

`CThres`

Type Float

Default value 30.0

Unit eV

Description all excitations of all subsystems (present on the fragment TAPE21 files) with an excitation energy that differs by less than coupling_threshold. From one of the reference states are selected to be included in the coupling. Note that additional excited states of system 1 may be included here.

`DIPVEL`

Type Bool

Default value No

Description

`DiagType`

Type Multiple Choice

Default value EXACT

Options [EXACT]

Description

`EIGPRINT`

Type Integer

Default value 100

Description

`ETHRES`

Type Float

Default value 0.0

Unit eV

Description Threshold for effective coupling

`FULLGRID`

Type Bool

Default value No

Description

`InvGuess`

Type Multiple Choice

Default value EigVal-OrbDiff

Options [EigVal-OrbDiff, OrbDiff-OrbDiff, Exact]

Description Type of states to be coupled

`LOCALFXCK`

Type Bool

Default value No

Description

`Lowest`

Type Integer

Default value 10

Description The selection of the excited states to be coupled consists of two steps

`NITER`

Type Integer

Default value 1

Description

`NOINTERSOLV`

Type Bool

Default value No

Description

`NOSOLVCCHECK`

Type Bool

Default value No

Description

`ONEGRID`

Type Bool

Default value No

Description

`OptStates`

Type Integer List

Description If the keyword OPTSTATES is given, only those excited states of the first subsystem are considered as reference states that are given in this list.

`PFRAGOUT`

Type Bool

Default value No

Description

`PTHRES`

Type Float

Default value 1.0

Description

`SETDIAG`

Type Float

Description

`SFThres`

Type Float

Default value 1e-05

Description To reduce the computational effort, it is possible to ignore the effect of orbital pairs with coefficients less than solutionfactor_threshold in the solution factors (TDDFT eigenvectors) of the underlying uncoupled calculation in the construction of the exact trial densities during the calculation of the coupling matrix elements. These orbital pair contributions are not ignored in the subsequent calculation of transition moments, oscillator, and rotational strengths.

`SMARTGRID`

Type Bool

Default value No

Description

`Stat2CPL`

Type Multiple Choice

Default value OnlyKnown

Options [OnlyKnown]

Description Type of states to be coupled

`TCOMEGA`

Type Bool

Default value No

Description Transpose construction of Omega matrix

`TDA`

Type Bool

Default value No

Description TDA specifies the use of the Tamm-Dancoff-Approximation (Tamm-Dancoff approximation) in the underlying uncoupled FDE-TDDFT calculations. Contrary to the full SUBEXCI-TDDFT variant, SUBEXCI-TDA allows for the usage of hybrid functionals in the underlying uncoupled FDE-TDDFT calculations.

`TKINKERNEL`

Type Bool

Default value Yes

Description

`XCKERNEL`

Type Bool

Default value Yes

Description

`Symmetry`

Type Multiple Choice

Default value AUTO

Options [AUTO, NOSYM, ATOM, C(LIN), D(LIN), C(I), C(S), C(2), C(2V), C(3V), C(4V), C(5V), C(6V), C(7V), C(8V), C(2H), D(2), D(3), D(4), D(5), D(6), D(7), D(8), D(2D), D(3D), D(4D), D(5D), D(6D), D(7D), D(8D), D(2H), D(3H), D(4H), D(5H), D(6H), D(7H), D(8H), O(H), T(D)]

Description Use (sub)symmetry with this Schoenflies symbol. Can only be used for molecules. Orientation should be correct for the (sub)symmetry. Coordinates must be symmetric within SymmetryTolerance.

`SymmetryTolerance`

Type Float

Default value 1e-07

Description Tolerance used to detect symmetry in the system. If symmetry Schoenflies symbol is specified, the coordinates must be symmetric within this tolerance.

`Tails`

Type Block

Description Obsolete option for linear scaling and distance effects. We recommend using the LinearScaling key instead.

`Bas`

Type Float

Description Parameter related to the threshold for the calculation of basis functions on a block of integration points. A higher value implies higher precision. The default depends on the Integration numerical quality.

`Fit`

Type Float

Description Parameter related to the threshold for the calculation of fit functions on a block of integration points. A higher value implies higher precision. The default depends on the Integration numerical quality.

`TDA`

Type Bool

Default value No

Description Use the Tamm-Dancoff approximation (TDA) (requires the EXCITATION block key)

`TDDFTSO`

Type Bool

Default value No

Description

`TIDegeneracyThreshold`

Type Float

Default value 0.1

Unit eV

Description If the orbital energy of the fragment MO is within this threshold with fragment HOMO or LUMO energy, then this fragment MO is included in the calculation of the transfer integrals. Relevant in case there is (near) degeneracy.

`Title`

Type String

Default value *** (NO TITLE) ***

Description Title of the calculation.

`TotalEnergy`

Type Bool

Default value No

GUI name Print: Total Energy

Description Calculate the total energy.

Normally only the bonding energy with respect to the fragments is calculated.

The total energy will be less accurate then the bonding energy (about two decimal places), and is not compatible with some options.

In most cases the total energy will not be needed.

`TransferIntegrals`

Type Bool

Default value No

GUI name : Charge transfer integrals (for transport properties)

Description Calculate the charge transfer integrals, spatial overlap integrals and site energies.

Charge transfer integrals can be used in models that calculate transport properties.

`Unrestricted`

Type Bool

Default value No

Description By default, a spin-restricted calculation is performed where the spin alpha and spin beta orbitals are spatially the same.

`UnrestrictedFragments`

Type Bool

Default value No

Description Use fragments calculated a spin-unrestricted calculation: the spin alpha and spin beta orbitals may be spatially different.

The total spin polarization of your fragments must match the spin polarization of your final molecule.

`UseSPCode`

Type Bool

Default value No

Description Use Patchkovskii routines for PBE

`VectorLength`

Type Integer

GUI name Vectorlength (blocksize)

Description Specify a different batch size for the integration points here (default: 128 on most machines and 2047 on vector machines).

VSCRF

Type Non-standard block

Description VSCRF is no longer supported. Use AMS2023 or earlier.

XC

Type Block

Description Definition of the XC.

Dispersion

Type String

Description Dispersion corrections.

DoubleHybrid

Type String

Description Specifies the double hybrid functional that should be used during the SCF.

EmpiricalScaling

Type Multiple Choice

Default value None

Options [None, SOS, SCS, SCSMI]

Description Calculate the (SOS/SCS/SCSMI)-MP2 correlation energy.

GCPparameters

Type String

Description Applying parameters for the geometrical counter poise correction.

GGA

Type String

Description Specifies the GGA part of the XC Functional

HartreeFock

Type Bool

Default value No

Description Use the Hartree-Fock exchange should be used during the SCF.

Hybrid

Type String

Description Specifies the hybrid functional that should be used during the SCF.

LDA

Type String

Description Defines the LDA part of the XC functional

LibXC

Type String

Description Use the LibXC library with the specified functional.

`MP2`

Type Bool

Default value No

Description Calculate the MP2 correlation energy after the HF SCF is completed.

`MetaGGA`

Type String

Description Specifies that a meta-GGA should be used during the SCF

`MetaHybrid`

Type String

Description Specifies the meta-hybrid functional that should be used during the SCF.

`Model`

Type String

Description Model potential to be used

`NoLibXC`

Type Bool

Default value No

Description Prevent the usage of the LibXC library

`OEP`

Type String

Description Defines the optimized effective potential expanded into a set of the fit functions

`RPA`

Type Multiple Choice

Default value None

Options [None, Direct, Sigma, SOSEX, SOSSX]

Description Specifies that RPA is used an possibly also a post-RPA method. By default, RPA is not used.

`RangeSep`

Type String

Description Range separated hybrids parameters

`XCFun`

Type Bool

Default value No

Description Use the XCFun library

`gCP`

Type String

Description Use the geometrical counter poise correction.

XES

Type Block

Description X-ray emission spectroscopy

AllXESMoments

Type Bool

Default value No

GUI name Print: All XES Moments

Description Print out all the individual transition moments used within the calculation of the total oscillator strength

AllXESQuadrupole

Type Bool

Default value No

GUI name : All XES Quadrupole

Description Print out the individual oscillator strength components to the total oscillator strength

CoreHole

Type String

GUI name Acceptor orbital

Description selection of the acceptor orbital for the calculation of the emission oscillator strengths. For example 'CoreHole A1 2' calculates oscillator strengths to the orbital 2 in irrep A1.

In AMSinput you may also use the notation 2A1 (so first the orbital number, next the symmetry)

Enabled

Type Bool

Default value No

GUI name Calculate XES

Description Calculate the X-ray emission energies to a core orbital.

By default it calculates the emission to the first orbital in the first symmetry.

ZExact

Type Bool

Default value No

Description Expert option in TDDFT excitations.

ZFS

Type String

Description Calculate the zero-field splitting (ZFS) of an open shell ground state. An unrestricted calculation is required and a spin larger than 1/2, and no no spatial degeneracy. Scalar relativistic ZORA is required.

ZlmFit

Type Block

Description Options for the density fitting scheme 'ZlmFit'.

`AllowBoost`

Type Bool

Default value Yes

Description Allow automatic atom-dependent tuning of maximum l of spherical harmonics expansion. Whether or not this boost is needed for a given atom is based on an heuristic estimate of how complex the density around that atom is.

`DensityThreshold`

Type Float

Default value 1e-07

Description Threshold below which the electron density is considered to be negligible.

`GridAngOrder`

Type Integer

Default value 21

Description

`GridRadialFactor`

Type Float

Default value 1.0

Description

`PartitionFunThreshold`

Type Float

Default value 0.0

Description

`PotentialThreshold`

Type Float

Default value 1e-07

Description

`Pruning`

Type Bool

Default value Yes

Description

`Quality`

Type Multiple Choice

Default value Auto

Options [Auto, Basic, Normal, Good, VeryGood, Excellent]

Description Quality of the density-fitting approximation. For a description of the various qualities and the associated numerical accuracy see reference. If 'Auto', the quality defined in the 'NumericalQuality' will be used.

`QualityPerRegion`

Type Block

Recurring True

Description Sets the ZlmFit quality for all atoms in a region. If specified, this overwrites the globally set quality.

`Quality`

Type Multiple Choice

Options [Basic, Normal, Good, VeryGood, Excellent]

Description The region's quality of the ZlmFit.

`Region`

Type String

Description The identifier of the region for which to set the quality.

`lExpansion`

Type Integer

Default value 4

Description

`lMargin`

Type Integer

Default value 4

Description

16.2.2 adfnbo

`ADFFile`

Type String

Default value TAPE21

Description Path to TAPE21 file from which adfnbo reads data and to which adfnbo possibly writes data

`Choose`

Type Non-standard block

Description

`Copy`

Type Bool

Default value No

Description

`Fock`

Type Bool

Default value No

Description

`NBOKeyList`

Type String

Default value BNDIDX NBONLMO=W AONBO=W AONLMO=W NLMOMO=W STERIC DIST

Description $NBO keylist

`Read`

Type Bool

Default value No

Description

`Spherical`

Type Bool

Default value No

Description

`TAPE15File`

Type String

Default value TAPE15

Description Path to the TAPE15 file from which adfnbo reads data

`TestJob`

Type Bool

Default value No

Description include extra options in FILE47, such as NRT (natural resonance theory) which is expensive for large molecules

`Write`

Type Bool

Default value No

Description

16.2.3 cpl

`ADFFile`

Type String

Default value TAPE21

Description Path to TAPE21 file from which cpl reads data and to which cpl writes data

`CALCV2007`

Type Bool

Default value No

Description compatibility with older versions of CPL that did not use the SAPA approximation but always calculated the potential during the CPL run, which is inconsistent with SAPA settings in ADF

`Fractional`

Type Bool

Default value No

Description Allow Fractional occupations

`GGA`

Type Bool

Default value No

Description Use first-order GGA potential instead of the first-order VWN potential

`Hyperfine`

Type Block

Description control the computation of the NSSCCs

`ADFGUI`

Type Bool

Description

`Atoms`

Type Integer List

Recurring True

Description

`Enabled`

Type Bool

Default value No

Description

`FC`

Type Bool

Default value No

Description

`NOFC`

Type Bool

Default value No

Description

`NOPSOSO`

Type Bool

Default value No

Description

`NOSD`

Type Bool

Default value No

Description

`Nuclei`

Type Integer List

Recurring True

Description

`PSOSO`

Type Bool

Default value No

Description

`SCF`

Type Block

Description

`Converge`

Type Float

Default value 0.0001

Description maximum number of iterations

`Iterations`

Type Integer

Default value 25

Description maximum number of iterations

`NOCYC`

Type Bool

Default value No

Description

`SD`

Type Bool

Default value No

Description

`NMRCoupling`

Type Block

Description control the computation of the NSSCCs

`ADFGUI`

Type Bool

Description

`ALDA`

Type Bool

Default value No

Description

`AtomPert`

Type Integer List

Recurring True

Description

`AtomResp`

Type Integer List

Recurring True

Description

`Contributions`

Type String

Description Analyze orbital contributions

`DSO`

Type Bool

Default value No

Description

`FC`

Type Bool

Default value No

Description

`Gamma`

Type String

Recurring True

Description

`NOFC`

Type Bool

Default value No

Description

`NOSD`

Type Bool

Default value No

Description

`Nuclei`

Type Integer List

Recurring True

Description

`PSO`

Type Bool

Default value No

Description

`PertAllAtomsOfType`

Type String

Description Space separated list of type of perturbing nuclei (like H, C, P) for which the NMR spin-spin coupling should be calculated.

`RespAllAtomsOfType`

Type String

Description Space separated list of type of responding nuclei (like H, C, P) for which the NMR spin-spin coupling should be calculated.

`SCF`

Type Block

Description

`Converge`

Type Float

Default value 0.0001

Description maximum number of iterations

`Iterations`

Type Integer

Default value 25

Description maximum number of iterations

`NOCYC`

Type Bool

Default value No

Description

`SD`

Type Bool

Default value No

Description

`XAlpha`

Type Bool

Default value No

Description

`Save`

Type String

Recurring True

Description

`TAPE10File`

Type String

Default value TAPE10

Description Path to the TAPE10 file from which cpl reads data

16.2.4 densf

`ADFFile`

Type String

Default value TAPE21

Description Path to the TAPE21 file from which densf reads the input data

`AOResponse`

Type String

Description

`Convert`

Type Bool

Default value No

Description

`COSMO`

Type Bool

Default value No

Description

CubInput

Type String

Description If the CubInput keyword is present then the grid as specified in the file is used to calculate all requested quantities. Any volume data found in the cube file is also saved in the output file. NOTE: CUBINPUT option cannot be used with a pre-existing TAPE41 file because they both specify the grid, which may lead to a conflict.

CubOutput

Type String

Description Presence of the CubOutput keyword tells densf to save all computed quantities as cube files using file as filename prefix. The prefix can also contain a complete path including directories. For example, specifying the following in the densf input

DenGrad

Type String

Recurring True

Description

DenHess

Type String

Recurring True

Description

Density

Type String

Recurring True

Description

DualDescriptor

Type Bool

Default value No

Description

Extend

Type Float

Description Extend grid?

FOD

Type Bool

Default value No

Description

GenFit

Type Non-standard block

Description

Grid

Type Non-standard block

Description

`IrrepDensity`

Type Non-standard block

Description Select particular symmetry to compute the electron density for.

`KinDens`

Type String

Recurring True

Description

`Laplacian`

Type String

Recurring True

Description

`Line`

Type Non-standard block

Description

`NCI`

Type String

Description

`NEBImage`

Type Integer

Description

`NOCV`

Type Non-standard block

Description

`Orbitals`

Type Non-standard block

Recurring True

Description

`OutputFile`

Type String

Default value TAPE41

Description Path to the (possibly existing) TAPE41 file. If the file exists, densf will read grid specifications from it ignoring GRID keyword in the input. Computed quantities are saved in the file overwriting existing data with the same name, if any

`POLTDDFT`

Type Integer

Default value 0

Description Frequency point for transition density

`Potential`

Type String

Recurring True

Description

`QP`

Type Bool

Default value No

Description

`Ridge`

Type Bool

Default value No

Description

`RISM`

Type Bool

Default value No

Description

`SEDD`

Type Bool

Default value No

Description

`Spinor`

Type Non-standard block

Description

`StericInteraction`

Type Non-standard block

Description

`TAPE16File`

Type String

Default value TAPE16

Description Path to the TAPE16 file from which densf reads the input data

`TransitionDensity`

Type Non-standard block

Description Select particular excitations to calculate the transition density for. Format: SS|ST SymLabel Index

Units

Type Block

Description Definitions of the units.

length

Type Multiple Choice

Default value angstrom

Options [bohr, angstrom]

Description Units of length

VTKFile

Type String

Description Specifies path to a file in the format readable by VTK directly. This option exists primarily for better integration with AMS-GUI and the user should not specify it.

16.2.5 green

DOS

Type String

Description Enables the calculation of the density of states. The string specifies the TAPE21 file containing the result of an ADF calculation of the extended molecule (performed with SYMMETRY NOSYM)

Eps

Type String

Description mineps maxeps numeps: The energy range for which either the self-energy matrices or the DOS and transmission have to be calculated. The range consists of numeps (<=1) points running from mineps to maxeps inclusive.

ETA

Type Float

Default value 1e-06

Unit Hartree

Description The imaginary energy, or the distance from the real axis, in the calculation of the Green's function. The value needs to be a small positive number to prevent singularities in the calculation.

FermiLevel

Type String

Description

Left

Type Block

Description Specify the left self-energies used in a calculation of the DOS and transmission. If a filename is specified in the header, the self-energy matrices are read from that file.

ETA

Type Float

Default value 0.001

Unit Hartree

Description Magnitude of the coupling

`Fragment`

Type String

Description

`NoSave`

Type String

Description

`Right`

Type Block

Description Specify the right self-energies used in a calculation of the DOS and transmission. If a filename is specified in the header, the self-energy matrices are read from that file.

`ETA`

Type Float

Default value 0.001

Unit Hartree

Description Magnitude of the coupling

`Fragment`

Type String

Description

`SO`

Type Float List

Description

`Surface`

Type Block

Description Enables the calculation of the self-energy matrices. The filename in the header specifies the TAPE21 file resulting from an ADF calculation of the contacts

`Fragments`

Type String

Description The two principal layers between which the surface is defined

`Trans`

Type String

Description

16.2.6 lfdft

`ADFFile`

Type String

Default value TAPE21

Description Path to TAPE21 file from which lfdft reads data and to which lfdft writes data

`BField`

Type Float List

Default value [0.0, 0.0, 0.0]

Unit Tesla

Description Include a finite magnetic Field. For MCD calculations include a magnetic field in the z-direction. The DegeneracyThreshold should be small to see the splitting of levels due to the magnetic field.

`DegeneracyThreshold`

Type Float

Default value 0.001

Unit eV

Description Energy difference threshold to determine degenerate levels

`MOIND1`

Type Integer List

Default value [0, 0, 0, 0, 0, 0, 0]

Description The indices of the MOs that participate for shell 1.

`MOIND2`

Type Integer List

Default value [0, 0, 0, 0, 0, 0, 0]

Description The indices of the MOs that participate for shell 2.

`NLVAL1`

Type Integer List

Default value [0, 0]

Description n and l value of shell 1.

`NLVAL2`

Type Integer List

Default value [0, 0]

Description n and l value of shell 2.

`NSHELL`

Type Integer

Default value 1

Description number of shells

SOC

Type Float List

Default value [1.0, 1.0, 1.0, 1.0]

Description Include Spin-Orbit coupling for the shells, scaling it with the specicied factor(s).

SOCType

Type Block

Description Choose the type of Spin-Orbit coupling calculation used for the shells.

Shell1

Type Multiple Choice

Default value ZORA

Options [ZORA, Core]

Description Type of Spin-Orbit coupling for the first shell

Shell2

Type Multiple Choice

Default value ZORA

Options [ZORA, Core]

Description Type of Spin-Orbit coupling for the second shell

16.2.7 lfdft_tdm

STATE1

Type String

Default value NONE

Description NAME of the state1 file.

STATE2

Type String

Default value NONE

Description NAME of the state2 file.

16.2.8 nmr

ADFFile

Type String

Default value TAPE21

Description Path to TAPE21 file from which nmr reads data and to which nmr writes data

AllInOne

Type Bool

Description Tensor in one step

`Analysis`

Type Block

Description Block for analysis options.

`Components`

Type Bool

Default value No

Description The components keyword is optional and enables an analysis not only of the isotropic shielding but also of the diagonal Cartesian components of the tensor XX, YY, and ZZ). In order to analyze the principal shielding tensor components with canonical MOs you can calculate the shielding tensor first with the NMR code, rotate the molecule such that the principal axes system aligns with the Cartesian coordinate system, and then repeat the NMR calculation with the analysis features switched on.

`FakeSO`

Type Bool

Default value No

Description

`NoPrincipal`

Type Bool

Default value No

Description Do not transform to principal axes for analysis

`Print`

Type Float

Default value 0.001

Description The print keyword selects printout of contributions relative to the total diamagnetic, paramagnetic. For example in case of print 0.01 only contributions greater than 1% are printed. Set to zero to print ALL contributions.

`ZSOAO2007`

Type Bool

Default value No

Description

`canonical`

Type Bool

Default value No

Description It enables an analysis of the shielding in terms of the canonical MOs.

`nbo`

Type Bool

Default value No

Description

`FakeSO`

Type Bool

Default value No

Description

`Fractional`

Type Bool

Default value No

Description

`HFAtomsPerPass`

Type Integer

Description Memory usage option for old HF scheme

`HFMaxMemory`

Type Integer

Description Memory usage option for old HF scheme

`Logfile`

Type String

Default value Flush

Description

`NBO`

Type Bool

Description

`NMR`

Type Block

Description Main NMR options.

`ADFGUI`

Type Bool

Default value No

Description

`AllAtomsOfType`

Type String

Description Space separated list of type of nuclei (like H, C, P) for which the NMR shielding should be calculated. In addition to Nuc or Atoms.

`Analysis`

Type Integer

GUI name Number of MOs in analysis

Description This key controls the MO analysis. Its value should be an integer, which then specifies that the first so many MOs are to be analyzed.

Default no Analysis.

The value of this analysis subkey in the block key NMR is somewhat limited. The separate ANALYSIS block key can give more analysis of the NMR chemical shielding.

`Atoms`

Type Integer List

Recurring True

Description This subkey ATOMS specifies for which nuclei the NMR shielding is calculated. Default all nuclei are calculated, i.e. as for omitting the subkeys ATOMS and NUC. The numbers refer to the input ordering in the ADF calculation. Use the subkey NUC to specify the nuclei according to the internal NMR numbers of the atoms.

`Calc`

Type String

Default value All

Description The sub key Calc controls what is actually calculated. All: Implies all of the other options to this key. Para: The paramagnetic part, Dia: The diamagnetic part, FC: The Fermi-contact part in case of the Pauli Hamiltonian, SO: The Fermi-contact part in case of the ZORA Hamiltonian.

`GFactors`

Type Bool

Default value No

Description Calculate g-factors

`Ghosts`

Type Non-standard block

Description The subkey GHOSTS is a block type subkey. The format is Ghosts | xx1 yy1 zz1 | xx2 yy2 zz2 | … | SubEnd

`Nuc`

Type Integer List

Description This subkey NUC specifies for which nuclei the NMR shielding is calculated. Default all nuclei are calculated, i.e. as for omitting the subkeys ATOMS and NUC. Else you may use this options by simply typing Nuc in the NMR block (without any further data); this means: for no nuclei at all. Alternatively you may type the index of the atom(s) you want to see analyzed. Default all nuclei are calculated, i.e. as for omitting this subkey. The numbers refer to the internal numbering of the nuclei as it appears somewhere early in the general ADF output. This internal numbering is also the internal NMR numbering, but it is not necessarily the same as the input ordering. Use the subkey ATOMS to specify the nuclei according to this input ordering in the ADF calculation. Note that the number of nuclei has a significant consequence for the total CPU time.

`Out`

Type String

Default value ISO TENS

Description Controls printed output. Options: All: All the other options, ISO: Isotropic shielding constants, Tens: Shielding tensors, Eig: Eigenvectors, U1: The U1 Matrix, F1: The first order change in the Fock matrix, S1: The first order change in the Overlap matrix, AOP: The paramagnetic AO matrix (= the matrix in the representation of elementary atomic basis functions), AOD: The diamagnetic AO matrix, AOF: The Fermi-contact AO matrix, Refs: Literature references, INFO: General information.

`SCF`

Type Float

Default value 1e-06

Description Convergence threshold for CPKS cycle

`U1K`

Type String

Default value Best

Description Determines which terms are included in the calculation of the U1 matrix (first order changes in MO coefficients). Best: The best (recommended) options for each relativistic option are included for this subkey. Implies None for non-relativistic and scalar relativistic ZORA, SO + SOFULL for spin-orbit coupled ZORA, and MV + Dar for the Pauli Hamiltonian. None: Implies none of the other options to this key. All: Implies all the other options to this key. MV: The mass-velocity term. Dar: The Darwin term. ZMAN: The Spin-Zeeman term (can be included only in case of spin-orbit coupled Pauli Hamiltonian). SO: ZORA spin-orbit part. SOFULL: ZORA spin-orbit part.

`Use`

Type String

Description The subkey Use controls some optional options. FXC: Improves the exchange-correlation kernel used, as was implemented by J. Autschbach [http://dx.doi.org/10.1080/00268976.2013.796415]. Important only in case of spin-orbit coupled calculations. This may give some (small) gauge dependent results when using this. Important option that should be seriously considered and has been advocated in Ref [http://dx.doi.org/10.1080/00268976.2013.796415]. SCALED: Implies the scaled ZORA method, which gives (slightly) gauge dependent results. Note that in case of the ZORA Hamiltonian default the unscaled ZORA method is used. For chemical shifts, only compare results with the same options. SO1C: Before ADF2008.01 in the the spin-orbit term a 1-center approximation was used, which does not suffer from gauge dependence. This 1-center approximation can be used with USE SO1C.

`NoScale`

Type Bool

Description

`PNMRFile`

Type String

Default value

Description Path to file that contains pNMR data

`RecalculateTAPE10`

Type Bool

Default value No

Description

`Save`

Type String

Recurring True

Description

`Scaled`

Type Bool

Description

`TAPE10File`

Type String

Default value TAPE10

Description Path to the TAPE10 file from which nmr reads data

`Temperature`

Type Float

Default value 298.15

Description Temperature (Kelvin) for temperature dependent part shielding tensor.

CHAPTER

SEVENTEEN

KF OUTPUT FILES

Main binary output files

- *ams.rkf* (page 383)
- *adf.rkf* (page 383)

Special binary output files

- *TAPE10* (page 385)
- *TAPE13* (page 385)
- *TAPE15* (page 386)
- *TAPE41* (page 291)

17.1 Accessing KF files

KF files are Direct Access binary files. KF stands for Keyed File: KF files are keyword oriented, which makes them easy to process by simple procedures. Internally all the data on KF files is organized into sections containing variables, so each datum on the file can be identified by the combination of section and variable.

All KF files can be opened using the KFbrowser GUI program:

```
$AMSBIN/kfbrowser path/to/ams.rkf
```

By default KFbrowser shows a just a curated summary of the results on the file, but you can make it show the raw section and variable structure by switching it to expert mode. To do this, click on **File → Expert Mode** or press **ctrl/cmd + e**.

KF files can be opened and read with Command line tools.

For working with the data from KF files, it is often useful to be able to read them from Python. Using the AMS Python Stack, this can easily be done with the AKFReader class:

```
>>> from scm.akfreader import AKFReader
>>> kf = AKFReader("path/to/ams.rkf")
>>> "Molecule%Coords" in kf
True
>>> kf.description("Molecule%Coords")
{
    '_type': 'float_array',
    '_shape': [3, 'nAtoms'],
    '_comment': 'Coordinates of the nuclei (x,y,z)',
    '_unit': 'Bohr'
}
```

(continues on next page)

(continued from previous page)

```
>>> kf.read("Molecule%Coords")
array([[-11.7770694 ,  -4.19739597,   0.04934546],
       [ -9.37471321,  -2.63234227,  -0.13448698],
        ...
       [ 10.09508738,  -1.06191208,   1.45286913],
       [ 10.11689333,  -1.5080196 ,  -1.87916127]])
```

Tip: For a full overview of the available methods in AKFReader, see the AKFReader API documentation.

17.2 Sections and variables on adf.rkf

adf.rkf is the general result file of an ADF calculation. In ADF<=2019 it was known as TAPE21.

ActiveFrag **Section content:** Data related to the active fragment.

`ActiveFrag%alfbas`

Type float_array

Description Basis set: alpha values (the exponent in the Slater type-function) of function sets.

Shape [nbset]

`ActiveFrag%alfcor`

Type float_array

Description Core functions set: alpha values (the exponent in the Slater type-function) of function sets.

Shape [ncset]

`ActiveFrag%alffit`

Type float_array

Description Fit functions set: alpha values (the exponent in the Slater type-function) of function sets.

Shape [nfset]

`ActiveFrag%AOIndex`

Type int_array

Description ?

`ActiveFrag%atomIndices`

Type int_array

Description indices of the atoms in the fragments of this type in the ‘global’ geometry

Shape [nr of atoms]

`ActiveFrag%atomtype`

Type lchar_string_array

Description The names of the atom types.

Shape [nr of atomtypes]

`ActiveFrag%atomtype effective charge`

Type float_array

Description The effective charge of the atom types (i.e. the nuclear charge minus the number of electrons in the frozen core).

Shape [nr of atomtypes]

`ActiveFrag%atomtype isGhost`

Type bool_array

Description Whether this atom type is a ghost type.

Shape [nr of atomtypes]

`ActiveFrag%atomtype total charge`

Type float_array

Description The nuclear charge of the atom types.

Shape [nr of atomtypes]

`ActiveFrag%atomtypeIndices`

Type int_array

Description indices of the atom types in this fragment type in the 'global' geometry

Shape [nr of atomtypes]

`ActiveFrag%Bas-I*`

Type float_array

Description ?

`ActiveFrag%Bas-R*`

Type float_array

Description ?

`ActiveFrag%ccor`

Type float_array

Description Some core related stuf…?

`ActiveFrag%cofcom`

Type float_array

Description STO-fit: coefficients in atomic fit a1-comb's

Shape [na1cof]

`ActiveFrag%Coordinates`

Type float_array

Description The coordinates of the fragments.

Shape [3, nr of atoms, numFrag]

`ActiveFrag%cum nr of atoms`

Type int_array

Description Cumulative number of atoms, up to a certain atomtype.

Shape [nr of atomtypes+1]

`ActiveFrag%electrons`

Type float

Description Number of valence electrons in the fragment.

`ActiveFrag%faith`

Type float_array

Description The symmetry operator matrices.

Shape [3, 3, nogr]

`ActiveFrag%frgmap`

Type float_array

Description Affine transformation from the master fragments geometry to the actual fragments.

Shape [3, 4, numFrag]

`ActiveFrag%frgtyp`

Type string

Description The fragment type (name)

`ActiveFrag%grouplabel`

Type string

Description Schoenflies symbol of the symmetry group.

`ActiveFrag%igr`

Type int

Description Point group identification number. 1: atom, 10: c(lin), 20: d(lin), 30: t(d), 60: o(h), 100: c(n), 200: c(nh), 400: c(nv), 450: d(n), 500: d(nh), 600: d(nd), 700: c(i), 800: c(s), 999: nosym

`ActiveFrag%initialCoordinates`

Type float_array

Description Coordinates of the initial fragment. These are rotated and translated to obtain the 'true' geometry.

Shape [3, nr of atoms]

`ActiveFrag%ioprel`

Type int

Description Integer code for relativistic option used. 0: non-relativistic, 1: scalar Pauli + sum of frozen core pot., 3: scalar ZORA + MAPA, 4: scalar ZORA + full pot. (not supported anymore), 5: scalar ZORA + APA (Band), 6: scalar X2C + MAPA, 7: scalar X2C ZORA + MAPA, 11: spin-orbit Pauli + sum of frozen core pot., 13: spin-orbit ZORA + MAPA, 14: spin-orbit ZORA + full pot. (not supported anymore), 15: spin-orbit ZORA + APA (Band), 16: spin-orbit X2C + MAPA, 17: spin-orbit X2C ZORA + MAPA

`ActiveFrag%isfrozen`

Type bool

Description FDE (frozen density embedding): whether the fragment is frozen.

`ActiveFrag%iskf`

Type int_array

Description STO-fit: pointer array for atomic parts of fit a1-comb's.

Shape [4, niskf]

`ActiveFrag%ja1ok`

Type int_array

Description An array (1:npeq), with values 0 or 1. 1=the pair density can be fitted using A1 fit functions only. 0=all fit functions (on the involved atoms) are to be used. The value 1 may arise because of symmetry properties, or because the distance between the atoms is so large that the inaccuracy from using only A1 fit functions can be neglected.

Shape [npeq]

`ActiveFrag%jasym`

Type int_array

Description An array that runs over the npeq sets of equivalent atom pairs. Its value gives for the indicated the set the number of pairs in that set.

Shape [npeq]

`ActiveFrag%jsyml`

Type int_array

Description For each of the nsym representations: if it belongs to a one-dimensional irrep, the value is 1, otherwise: for the first subspecies in the irrep the value is the dimension of the irrep, for the other subspecies in the same irrep the value is 0.

Shape [nsym]

`ActiveFrag%lnosymfit`

Type bool

Description STO-fit: whether the symmetry should be used for density fitting or not.

`ActiveFrag%lqbas`

Type int_array

Description Basis set: l values (angular momentum) of function sets.

Shape [nbset]

`ActiveFrag%lqcor`

Type int_array

Description Core functions set: l values (angular momentum) of function sets.

Shape [ncset]

`ActiveFrag%lqfit`

Type int_array

Description Fit functions set: l values (angular momentum) of function sets.

Shape [nfset]

`ActiveFrag%lrl`

Type int

Description STO-fit: fitint dimentions…?

`ActiveFrag%mass`

Type float_array

Description Atomic masses of the various atom types.

Shape [nr of atomtypes]

`ActiveFrag%maxsf`

Type int

Description STO-fit: fitint dimentions…?

`ActiveFrag%na1cof`

Type int

Description STO-fit: Number of elements in numcom/cofcom.

`ActiveFrag%na1ptr`

Type int_array

Description STO-fit: index array (like nfptr, but only for the symmetric fit functions; per nucleus)

Shape [nnuc+1]

`ActiveFrag%naos`

Type int

Description Number of cartesian basis functions.

`ActiveFrag%naosx`

Type int

Description Number of elementes in a triangular matrix of size naos*naos, i.e. (naos*(naos + 1))/2.

`ActiveFrag%nbaspt`

Type int_array

Description Basis set: cumulative number of functions per atom type.

Shape [ntyp+1]

`ActiveFrag%nbset`

Type int

Description Basis set: total number of function sets (not counting spherical/cartesian componets).

`ActiveFrag%ncorpt`

Type int_array

Description Core functions set: cumulative number of functions per atom type.

Shape [ntyp+1]

ActiveFrag%ncset

Type int

Description Core functions set: total number of function sets (not counting spherical/cartesian componets).

ActiveFrag%nfcn

Type int_array

Description Number of primitive functions in the various irreps.

Shape [nsym]

ActiveFrag%nfitpt

Type int_array

Description Fit functions set: cumulative number of functions per atom type.

Shape [ntyp+1]

ActiveFrag%nfset

Type int

Description Fit functions set: total number of function sets (not counting spherical/cartesian componets).

ActiveFrag%ngr

Type int

Description One of the integer-code components that fix the symmetry group. See routine adf/maisya

ActiveFrag%niskf

Type int

Description STO-fit: counts the number of g functions.

ActiveFrag%nnuc

Type int

Description The total number of atoms.

ActiveFrag%noat

Type int_array

Description Map between normal list of atoms and symmetry sets.

Shape [nr of atoms]

ActiveFrag%nogr

Type int

Description The number of symmetry operators. NB, for the special cases of infinite symmetries, only the operators corresponding to finite elements are counted. Therefore, ATOM has nogr=1 (only the unit operator); C(LIN) has nogr=1, D(LIN) has nogr=2

ActiveFrag%norb

Type int_array

Description Number of orbitals for the various irreps.

Shape [nsym]

`ActiveFrag%notyps`

Type int_array

Description For each set of symmetry equivalent atoms, the atom type to which the set belongs.

Shape [nsetat]

`ActiveFrag%npeq`

Type int

Description The number of symmetry unique pairs of atoms.

`ActiveFrag%nqbas`

Type int_array

Description Basis set: n values (main quantum number) of function sets.

Shape [nbset]

`ActiveFrag%nqcor`

Type int_array

Description Core functions set: n values (main quantum number) of function sets.

Shape [ncset]

`ActiveFrag%nqfit`

Type int_array

Description Fit functions set: n values (main quantum number) of function sets.

Shape [nfset]

`ActiveFrag%nr of atoms`

Type int

Description The total number of atoms.

`ActiveFrag%nr of atomtypes`

Type int

Description The number of atom types in the fragment.

`ActiveFrag%nratst`

Type int_array

Description Number of atoms in each set of symmetry equivalent atoms.

Shape [nsetat]

`ActiveFrag%nrcorb`

Type int_array

Description Number of core orbitals…?

Shape [4, ntyp]

`ActiveFrag%nrcset`

Type int_array

Description Some core information…?

`ActiveFrag%nrorb`

Type int_array

Description Number number of orbitals per the various relativistic irreps.

Shape [nrsym]

`ActiveFrag%nrsym`

Type int

Description Number of double group irreps.

`ActiveFrag%nsetat`

Type int

Description Number of sets of symmetry equivalent atoms.

`ActiveFrag%nsfos`

Type int

Description STO-fit: total no. of fully symmetric (a1) fit functions.

`ActiveFrag%nspin`

Type int

Description nspin used in this fragment.

`ActiveFrag%nsym`

Type int

Description Number of irreps.

`ActiveFrag%ntyp`

Type int

Description The number atom types in the fragment

`ActiveFrag%numcom`

Type int_array

Description STO-fit: no's of functions in atomic fit a1-comb's.

Shape [na1cof]

`ActiveFrag%numFrag`

Type int

Description Number of times this fragment is used.

`ActiveFrag%Pmat_AO`

Type float_array

Description ?

`ActiveFrag%Pmat_AO_beta`

Type float_array

Description ?

ActiveFrag%Smat_AO

Type float_array

Description ?

ActiveFrag%symlab

Type lchar_string_array

Description Labels of the irreps.

Shape [nsym]

ActiveFrag%symlabr

Type lchar_string_array

Description Labels of the double group irreps.

Shape [nrsym]

ActiveFrag%usebasis

Type bool

Description FDE (frozen density embedding). ?

ActiveFrag%xyz

Type float_array

Description Coordinates of the atoms in the fragment… same as initialCoordinates?

Shape [3, nr of atoms]

ActiveFrag* **Section content:** ?

ActiveFrag*%Bas-I_A *

Type float_array

Description Spin-orbit data.

ActiveFrag*%Bas-R_A *

Type float_array

Description Spin-orbit data.

ActiveFrag*%Eigen-Bas_A

Type float_array

Description MO expansion coefficients in the BAS representation for all nmo_A orbitals (in unrestricted case for spin A). The coefficients run over all BAS functions indicated by npart.

Shape [nbas, nmo_A]

ActiveFrag*%Eigen-Bas_B

Type float_array

Description Unrestricted case: same as 'Eigen-Bas_A', but for spin B.

Shape [nbas, nmo_A]

ActiveFrag*%eps_A

Type float_array

Description The orbital energies for the nmo_A orbitals (in unrestricted case for spin A). When they result from a ZORA calculations, the non-scaled values are stored on file, see qscal how to scale.

Shape [nmo_A]

`ActiveFrag*%eps_B`

Type float_array

Description Unrestricted case: same as 'eps_A', but for spin B.

`ActiveFrag*%escale_A *`

Type float_array

Description Spin-orbit data.

`ActiveFrag*%froc`

Type float_array

Description The occupation numbers of the MOs in the irrep.

Shape [nmo_A]

`ActiveFrag*%froc_A`

Type float_array

Description The occupation numbers of the MOs in the irrep (in unrestricted case for spin A).

Shape [nmo_A]

`ActiveFrag*%froc_B`

Type float_array

Description Unrestricted case: the occupation numbers of the MOs in the irrep, for spin B

Shape [nmo_A]

`ActiveFrag*%frocr_A *`

Type float_array

Description Spin-orbit data.

`ActiveFrag*%nbas`

Type int

Description Number of primitive STOs in this symmetry group.

`ActiveFrag*%nmo_A`

Type int

Description Number of alpha molecular orbitals in this symmetry group (in unrestricted case for spin A).

`ActiveFrag*%nmo_B`

Type int

Description Unrestricted case: number of beta molecular orbitals in this symmetry group. Should be equal to nmo_A.

`ActiveFrag*%npart`

Type int_array

Description A list of indices of the BAS functions that are used in this irrep.

Shape [nbas]

`ActiveFrag*%qscal_A`

Type float_array

Description Used only for ZORA. Scaled eps is (eps/(1+qscal)).

Shape [nmo_A]

`ActiveFrag*%qscal_B`

Type float_array

Description Unrestricted case: same as ‘qscal_A’, but for spin B.

Shape [nmo_A]

`ActiveFrag*%qscalr_A *`

Type float_array

Description Spin-orbit data.

All excitations **Section content:** Section related to all excitations.

`All excitations%All Sing-Sing excitations`

Type float_array

Description Singlet-Singlet excitation energies.

Unit hartree

Shape [nr Sing-Sing excitations]

`All excitations%All Sing-Trip excitations`

Type float_array

Description Singlet-Triplet excitation energies.

Unit hartree

Shape [nr Sing-Trip excitations]

`All excitations%All Spin-Polar excitations`

Type float_array

Description Spin-orbit coupled spin-polarized excitation energies

Unit hartree

Shape [nr excitations]

`All excitations%All Spin-Restr excitations`

Type float_array

Description Spin-orbit coupled spin-restricted excitation energies

Unit hartree

Shape [nr excitations]

`All excitations%ES DIP`

Type float_array

Description Unrelaxed approximate singlet excited state dipole moments.

Unit bohr

Shape [3, nr Sing-Sing including deg]

`All excitations%ESES OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between singlet excited states.

Unit

Shape [nr Sing-Sing including deg, nr Sing-Sing including deg]

`All excitations%ESES TDM`

Type float_array

Description Unrelaxed approximate transition dipole moments between singlet excited states.

Unit bohr

Shape [3, nr Sing-Sing including deg, nr Sing-Sing including deg]

`All excitations%ET DIP`

Type float_array

Description Unrelaxed approximate triplet excited state dipole moments.

Unit bohr

Shape [3, nr Sing-Trip including deg]

`All excitations%ETET OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between triplet excited states.

Unit

Shape [nr Sing-Trip including deg, nr Sing-Trip including deg]

`All excitations%ETET TDM`

Type float_array

Description Unrelaxed approximate transition dipole moments between triplet excited states.

Unit bohr

Shape [3, nr Sing-Trip including deg, nr Sing-Trip including deg]

`All excitations%GSES TDM`

Type float_array

Description Transition dipole moments between singlet ground state and singlet excited state.

Unit bohr

Shape [3, nr Sing-Sing including deg]

`All excitations%nr excitations`

Type int

Description Number of excitations.

`All excitations%nr excitations including deg`

Type int

Description Number of excitations including subspecies in case of multidimensional irreps.

`All excitations%nr Sing-Sing excitations`

Type int

Description Number of Singlet-Singet excitations.

`All excitations%nr Sing-Sing including deg`

Type int

Description Number of Singlet-Singlet excitations including subspecies in case of multidimensional irreps.

`All excitations%nr Sing-Trip excitations`

Type int

Description Number of Singlet-Triplet excitations.

`All excitations%nr Sing-Trip including deg`

Type int

Description Number of Singlet-Triplet excitations including subspecies in case of multidimensional irreps.

`All excitations%Spin-Polar ES DIP`

Type float_array

Description Unrelaxed approximate excited state dipole moments. Spin-orbit coupled spin-polarized calculation.

Unit bohr

Shape [3, nr excitations including deg]

`All excitations%Spin-Polar ESES OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between excited states. Spin-orbit coupled spin-polarized calculation.

Unit

Shape [nr excitations including deg, nr excitations including deg]

`All excitations%Spin-Polar ESES TDM`

Type float_array

Description Unrelaxed approximate transition dipole moments between excited states. Real and imaginary part. Spin-orbit coupled spin-polarized calculation.

Unit bohr

Shape [3, 2, nr excitations including deg, nr excitations including deg]

`All excitations%Spin-Restr ES DIP`

Type float_array

Description Unrelaxed approximate excited state dipole moments. Spin-orbit coupled spin-restricted calculation.

Unit bohr

Shape [3, nr excitations including deg]

`All excitations%Spin-Restr ESES OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between excited states. Spin-orbit coupled spin-restricted calculation.

Unit

Shape [nr excitations including deg, nr excitations including deg]

`All excitations%Spin-Restr ESES TDM`

Type float_array

Description Unrelaxed approximate transition dipole moments between excited states. Real and imaginary part. Spin-orbit coupled spin-restricted calculation.

Unit bohr

Shape [3, 2, nr excitations including deg, nr excitations including deg]

`All excitations%SS energy order index`

Type int_array

Description Energy order index array Singlet-Singlet excitations.

Shape [nr Sing-Sing excitations]

`All excitations%ST energy order index`

Type int_array

Description Energy order index array Singlet-Triplet excitations.

Shape [nr Sing-Trip excitations]

All excitations * **Section content:** Section related to all excitations for a given localization.

`All excitations *%All Sing-Sing excitations`

Type float_array

Description Singlet-Singlet excitation energies.

Unit hartree

Shape [nr Sing-Sing excitations]

`All excitations *%All Sing-Trip excitations`

Type float_array

Description Singlet-Triplet excitation energies.

Unit hartree

Shape [nr Sing-Trip excitations]

`All excitations *%All Spin-Polar excitations`

Type float_array

Description Spin-orbit coupled spin-polarized excitation energies

Unit hartree

Shape [nr excitations]

`All excitations *%All Spin-Restr excitations`

Type float_array

Description Spin-orbit coupled spin-restricted excitation energies

Unit hartree

Shape [nr excitations]

`All excitations *%ES DIP`

Type float_array

Description Unrelaxed approximate singlet excited state dipole moments.

Unit bohr

Shape [3, nr Sing-Sing including deg]

`All excitations *%ESES OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between singlet excited states.

Unit

Shape [nr Sing-Sing including deg, nr Sing-Sing including deg]

`All excitations *%ESES TDM`

Type float_array

Description Unrelaxed approximate transition dipole moments between singlet excited states.

Unit bohr

Shape [3, nr Sing-Sing including deg, nr Sing-Sing including deg]

`All excitations *%ET DIP`

Type float_array

Description Unrelaxed approximate triplet excited state dipole moments.

Unit bohr

Shape [3, nr Sing-Trip including deg]

`All excitations *%ETET OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between triplet excited states.

Unit

Shape [nr Sing-Trip including deg, nr Sing-Trip including deg]

`All excitations *%ETET TDM`

Type float_array

Description Unrelaxed approximate transition dipole moments between triplet excited states.

Unit bohr

Shape [3, nr Sing-Trip including deg, nr Sing-Trip including deg]

`All excitations *%GSES TDM`

Type float_array

Description Transition dipole moments between singlet ground state and singlet excited state.

Unit bohr

Shape [3, nr Sing-Sing including deg]

`All excitations *%nr excitations`

Type int

Description Number of excitations.

`All excitations *%nr excitations including deg`

Type int

Description Number of excitations including subspecies in case of multidimensional irreps.

`All excitations *%nr Sing-Sing excitations`

Type int

Description Number of Singlet-Singet excitations.

`All excitations *%nr Sing-Sing including deg`

Type int

Description Number of Singlet-Singlet excitations including subspecies in case of multidimensional irreps.

`All excitations *%nr Sing-Trip excitations`

Type int

Description Number of Singlet-Triplet excitations.

`All excitations *%nr Sing-Trip including deg`

Type int

Description Number of Singlet-Triplet excitations including subspecies in case of multidimensional irreps.

`All excitations *%Spin-Polar ES DIP`

Type float_array

Description Unrelaxed approximate excited state dipole moments. Spin-orbit coupled spin-polarized calculation.

Unit bohr

Shape [3, nr excitations including deg]

`All excitations *%Spin-Polar ESES OSC`

Type float_array

Description Unrelaxed approximate oscillator strengths between excited states. Spin-orbit coupled spin-polarized calculation.

Unit

Shape [nr excitations including deg, nr excitations including deg]

All excitations *%Spin-Polar ESES TDM

Type float_array

Description Unrelaxed approximate transition dipole moments between excited states. Real and imaginary part. Spin-orbit coupled spin-polarized calculation.

Unit bohr

Shape [3, 2, nr excitations including deg, nr excitations including deg]

All excitations *%Spin-Restr ES DIP

Type float_array

Description Unrelaxed approximate excited state dipole moments. Spin-orbit coupled spin-restricted calculation.

Unit bohr

Shape [3, nr excitations including deg]

All excitations *%Spin-Restr ESES OSC

Type float_array

Description Unrelaxed approximate oscillator strengths between excited states. Spin-orbit coupled spin-restricted calculation.

Unit

Shape [nr excitations including deg, nr excitations including deg]

All excitations *%Spin-Restr ESES TDM

Type float_array

Description Unrelaxed approximate transition dipole moments between excited states. Real and imaginary part. Spin-orbit coupled spin-restricted calculation.

Unit bohr

Shape [3, 2, nr excitations including deg, nr excitations including deg]

All excitations *%SS energy order index

Type int_array

Description Energy order index array Singlet-Singlet excitations.

Shape [nr Sing-Sing excitations]

All excitations *%ST energy order index

Type int_array

Description Energy order index array Singlet-Triplet excitations.

Shape [nr Sing-Trip excitations]

AMSResults **Section content:** Generic results of the ADF evaluation.

AMSResults%AAT_Transpose

Type float_array

Description VCD atomic axial tensors (AATs).

Shape [3, 3, Molecule%nAtoms]

`AMSResults%Bonds`

Type subsection

Description Bond info

`AMSResults%Bonds%Atoms`

Type archived_int_array

Description ?

`AMSResults%Bonds%CellShifts`

Type archived_int_array

Description ?

`AMSResults%Bonds%description`

Type string

Description A string containing a description of how the bond orders were calculated / where they come from

`AMSResults%Bonds%hasCellShifts`

Type bool

Description Whether there are cell shifts (relevant only in case of periodic boundary conditions)

`AMSResults%Bonds%Index`

Type archived_int_array

Description index(i) points to the first element of Atoms, Orders, and CellShifts belonging to bonds from atom 'i'. Index(1) is always 1, Index(nAtoms+1) is always nBonds + 1

`AMSResults%Bonds%Orders`

Type archived_float_array

Description The bond orders.

`AMSResults%BulkModulus`

Type float

Description The Bulk modulus (conversion factor from hartree/bohr^3 to GPa: 29421.026)

Unit hartree/bohr^3

`AMSResults%Charges`

Type float_array

Description Net atomic charges as computed by the engine (for example, the Charges for a water molecule might be [-0.6, 0.3, 0.3]). The method used to compute these atomic charges depends on the engine.

Unit e

Shape [Molecule%nAtoms]

`AMSResults%DipoleGradients`

Type float_array

Description Derivative of the dipole moment with respect to nuclear displacements.

Shape [3, 3, Molecule%nAtoms]

`AMSResults%DipoleMoment`

Type float_array

Description Dipole moment vector (x,y,z)

Unit e*bohr

Shape [3]

`AMSResults%DipQuadPolarizability`

Type float_array

Description Electric dipole-quadrupole polarizability. Element order [X,Y,Z] [XX,XY,XZ,YX,YY,YZ,ZX,ZY,ZZ].

Unit a.u.

Shape [3, 9]

`AMSResults%DipQuadPolarizabilityDerivs`

Type float_array

Description (Cartesian or symmetry-adapted) Nuclear derivative electric dipole-quadrupole polarizability. Element order [X,Y,Z] [XX,XY,XZ,YX,YY,YZ,ZX,ZY,ZZ].

Unit a.u.

Shape [3, 9, :]

`AMSResults%DipQuadPolarizabilityDerivsImag`

Type float_array

Description (Cartesian or symmetry-adapted) Nuclear derivative imaginary part electric dipole-quadrupole polarizability. Element order [X,Y,Z] [XX,XY,XZ,YX,YY,YZ,ZX,ZY,ZZ].

Unit a.u.

Shape [3, 9, :]

`AMSResults%DipQuadPolarizabilityImag`

Type float_array

Description Imaginary part electric dipole-quadrupole polarizability. Element order [X,Y,Z] [XX,XY,XZ,YX,YY,YZ,ZX,ZY,ZZ].

Unit a.u.

Shape [3, 9]

`AMSResults%ElasticTensor`

Type float_array

Description The elastic tensor in Voigt notation (6x6 matrix for 3D periodic systems, 3x3 matrix for 2D periodic systems, 1x1 matrix for 1D periodic systems).

Unit hartree/bohr^nLatticeVectors

Shape [:, :]

`AMSResults%Energy`

Type float

Description The energy computed by the engine.

Unit hartree

`AMSResults%fractionalOccupation`

Type bool

Description Whether of not we have fractionally occupied orbitals (i.e. not all occupations are integer numbers).

`AMSResults%Gradients`

Type float_array

Description The nuclear gradients.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`AMSResults%GSESDerivs`

Type float_array

Description Thermally averaged ground-to-excited state electronic transition properties?

Shape [:, 3*Molecule%nAtoms]

`AMSResults%GSESEnergies`

Type float_array

Description Thermally averaged ground-to-excited state electronic transition properties?

Shape [:]

`AMSResults%GSESProperties`

Type float_array

Description Thermally averaged ground-to-excited state electronic transition properties?

Shape [:]

`AMSResults%Hessian`

Type float_array

Description The Hessian matrix

Unit hartree/bohr^2

Shape [3*Molecule%nAtoms, 3*Molecule%nAtoms]

`AMSResults%HOMOEnergy`

Type float_array

Description Molecular Orbital Info: energy of the HOMO.

Unit hartree

Shape [nSpin]

`AMSResults%HOMOIndex`

Type int_array

Description Molecular Orbital Info: index in the arrays orbitalEnergies and orbitalOccupations corresponding to the HOMO.

Shape [nSpin]

`AMSResults%HOMOLUMOGap`

Type float_array

Description Molecular Orbital Info: HOMO-LUMO gap per spin.

Unit hartree

Shape [nSpin]

`AMSResults%LUMOEnergy`

Type float_array

Description Molecular Orbital Info: energy of the LUMO.

Unit hartree

Shape [nSpin]

`AMSResults%LUMOIndex`

Type int_array

Description Molecular Orbital Info: index in the arrays orbitalEnergies and orbitalOccupations corresponding to the LUMO.

Shape [nSpin]

`AMSResults%Molecules`

Type subsection

Description Molecules

`AMSResults%Molecules%AtCount`

Type archived_int_array

Description shape=(nMolType), Summary: number of atoms per formula.

`AMSResults%Molecules%Atoms`

Type archived_int_array

Description shape=(nAtoms), atoms(index(i):index(i+1)-1) = atom indices of molecule i

`AMSResults%Molecules%Count`

Type archived_int_array

Description Mol count per formula.

`AMSResults%Molecules%Formulas`

Type string

Description Summary: unique molecule formulas

`AMSResults%Molecules%Index`

Type archived_int_array

Description shape=(nMol+1), index(i) = index of the first atom of molecule i in array atoms(:)

`AMSResults%Molecules%Type`

Type archived_int_array

Description shape=(nMol), type of the molecule, reference to the summary arrays below

`AMSResults%nOrbitals`

Type int

Description Molecular Orbital Info: number of orbitals.

`AMSResults%nSpin`

Type int

Description Molecular Orbital Info: number spins (1: spin-restricted or spin-orbit coupling, 2: spin unrestricted).

`AMSResults%OpticalRotation`

Type float_array

Description Optical rotation.

Unit a.u.

Shape [9]

`AMSResults%OpticalRotationDerivs`

Type float_array

Description (Cartesian or symmetry-adapted) Nuclear derivative optical rotation.

Unit a.u.

Shape [9, :]

`AMSResults%OpticalRotationDerivsImag`

Type float_array

Description (Cartesian or symmetry-adapted) Nuclear derivative imaginary part optical rotation.

Unit a.u.

Shape [9, :]

`AMSResults%OpticalRotationImag`

Type float_array

Description Imaginary part optical rotation.

Unit a.u.

Shape [9]

`AMSResults%orbitalEnergies`

Type float_array

Description Molecular Orbital Info: the orbital energies.

Unit hartree

Shape [nOrbitals, nSpin]

`AMSResults%orbitalOccupations`

Type float_array

Description Molecular Orbital Info: the orbital occupation numbers. For spin restricted calculations, the value will be between 0 and 2. For spin unrestricted or spin-orbit coupling the values will be between 0 and 1.

Shape [nOrbitals, nSpin]

`AMSResults%PESPointCharacter`

Type string

Description The character of a PES point.

Possible values [‘local minimum’, ‘transition state’, ‘stationary point with >1 negative frequencies’, ‘non-stationary point’]

`AMSResults%PoissonRatio`

Type float

Description The Poisson ratio

`AMSResults%Polarizability`

Type float_array

Description (electric dipole-dipole) Polarizability. Elements order: [XX,XY,YY,XZ,YZ,ZZ].

Unit a.u.

Shape [6]

`AMSResults%PolarizabilityDerivs`

Type float_array

Description (Cartesian or symmetry-adapted) Nuclear derivative (electric dipole-dipole) polarizability. Elements order: [XX,XY,YY,XZ,YZ,ZZ].

Unit a.u.

Shape [6, :]

`AMSResults%PolarizabilityDerivsImag`

Type float_array

Description (Cartesian or symmetry-adapted) Nuclear derivative imaginary part (electric dipole-dipole) polarizability. Elements order: [XX,XY,YY,XZ,YZ,ZZ].

Unit a.u.

Shape [6, :]

`AMSResults%PolarizabilityImag`

Type float_array

Description Imaginary part (electric dipole-dipole) polarizability. Elements order: [XX,XY,YY,XZ,YZ,ZZ].

Unit a.u.

Shape [6]

`AMSResults%ShearModulus`

Type float

Description The Shear modulus (conversion factor from hartree/bohr^3 to GPa: 29421.026)

Unit hartree/bohr^3

`AMSResults%SmallestHOMOLUMOGap`

Type float

Description Molecular Orbital Info: the smallest HOMO-LUMO gap irrespective of spin (i.e. min(LUMO) - max(HOMO)).

Unit hartree

`AMSResults%StressTensor`

Type float_array

Description The clamped-ion stress tensor in Cartesian notation.

Unit hartree/bohr^nLatticeVectors

Shape [:, :]

`AMSResults%YoungModulus`

Type float

Description The Young modulus (conversion factor from hartree/bohr^3 to GPa: 29421.026)

Unit hartree/bohr^3

AngularBoost **Section content:** Both the Becke grid and the Zlm fit grid may boost the angular grid for certain areas.

`AngularBoost%boost`

Type bool_array

Description Whether to use a booster grid per atom.

AOMatrices **Section content:** Some matrices on AO

AORESPONSE **Section content:** Results of AOResponse calculation

aoresponse_data **Section content:** Technical AOresponse data

atens **Section content:** Data for spin-orbit unrestricted calculation of ESR A-tensor

Atyp* **Section content:** The core (and possibly also valence) radial density and potential of one particular atom type. The radial densities and potentials may be represented as simple tables - a sequence of values for r, the distance to the nucleus, and the corresponding density or potential - or as a piecewise expansion in Chebyshev polynomials over a sequence of intervals (r1,r2). The core density and potential have been constructed from the Frozen Core orbitals, which are defined in the section Core.

`Atyp*%ccheb core`

Type float_array

Description Coefficients of the Chebyshev expansion for the core. All coefficients, for all intervals, are stored contiguously in one linear array. The parts pertaining to a particular interval are determined by using the arrays ncheb.

`Atyp*%ccheb val`

Type float_array

Description Coefficients of the Chebyshev expansion for the valence. All coefficients, for all intervals, are stored contiguously in one linear array. The parts pertaining to a particular interval are determined by using the arrays ncheb.

Atyp*%core den

Type float_array

Description The core density on the radial grid

Shape [nrad]

Atyp*%core den ext

Type float_array

Description The core density on the radial grid calculated with Dirac interpolated on standard grid

Shape [nrad]

Atyp*%core pot

Type float_array

Description The core Coulomb potential on the radial grid (including a nuclear term -Qcore/r).

Shape [nrad]

Atyp*%core pot ext

Type float_array

Description The core Coulomb potential on the radial grid (including a nuclear term -Qcore/r) calculated with Dirac interpolated on standard grid

Shape [nrad]

Atyp*%core totpot ext

Type float_array

Description Total KS potential calculated with Dirac interpolated on standard grid.

Shape [nrad]

Atyp*%ncheb core

Type int_array

Description Number of expansion coefficients for each interval (core).

Shape [nrint core]

Atyp*%ncheb val

Type int_array

Description Number of Chebyshev expansion coefficients for each interval.

Shape [nrint val]

Atyp*%nrad

Type int

Description Number of points used in the direct tabular representation of the atomic densities and potentials.

Atyp*%nrad12

Type int

Description equals nrad

Atyp*%nrint core

Type int

Description Number of intervals for piecewise expansion of the core density in Chebyshev polynomials.

`Atyp*%nrint val`

Type int

Description Number of intervals for piecewise expansion of the valence density in Chebyshev polynomials.

`Atyp*%qcore`

Type float

Description The number of electrons contained in the core density.

`Atyp*%qval`

Type float

Description The number of electrons contained in the valence density.

`Atyp*%rfac`

Type float

Description The multiplication factor of the radial grid.

`Atyp*%rmin`

Type float

Description The first r-value of the table: the radial grid is defined by a first value (rmin), a constant multiplication factor defining rk+1 w.r.t. rk (rfac, see next), and the total nr of points (nrad).

`Atyp*%rup core`

Type float_array

Description Upper bounds of the intervals (core). The lower bound of the first interval is zero.

Shape [nrint core]

`Atyp*%rup val`

Type float_array

Description Upper bounds of the intervals. The lower bound of the first interval is zero.

Shape [nrint val]

`Atyp*%rx core`

Type float

Description Maximum r-value for which the core density is non-negligible.

`Atyp*%rx val`

Type float

Description Maximum r-value for which the valence density is non-negligible.

`Atyp*%valence den`

Type float_array

Description The valence density on the radial grid.

Shape [nrad]

`Atyp*%valence pot`

Type float_array

Description The valence Coulomb potential on the radial grid (including a nuclear term -Qval/r).

Shape [nrad]

Basis **Section content:** Description of the (valence) basis set.

`Basis%alf`

Type float_array

Description Exponential decay factors of the nbos Cartesian STO basis functions.

Shape [nbos]

`Basis%alfbas`

Type float_array

Description The exponential decay parameters of the STO functions in the basis set.

Shape [nbset]

`Basis%basnrm`

Type float_array

Description Normalization coefficients for the basis sets.

Shape [nbset]

`Basis%bnorm`

Type float_array

Description Normalization factors for the nbos Cartesian STO basis functions.

Shape [nbos]

`Basis%kr`

Type int_array

Description Powers of r of the nbos Cartesian STO basis functions.

Shape [nbos]

`Basis%kx`

Type int_array

Description Powers of x of the nbos Cartesian STO basis functions.

Shape [nbos]

`Basis%ky`

Type int_array

Description Powers of y of the nbos Cartesian STO basis functions.

Shape [nbos]

`Basis%kz`

Type int_array

Description Powers of z of the nbos Cartesian STO basis functions.

Shape [nbos]

`Basis%lorde`

Type int_array

Description Maximum of the angular momentum quantum number (l) for all STO basis and fit functions corresponding to that atom type.

Shape [Geometry%nr of atomtypes]

`Basis%lqbas`

Type int_array

Description Angular momentum quantum number of each basis set. The current implementation of ADF supports only s, p, d, and f basis functions, so the allowed lqbas values are 0, 1, 2, and 3.

Shape [nbset]

`Basis%naos`

Type int

Description The total number of basis functions, counting all Cartesian polynomials and all copies of the functions on the atoms of the pertaining atom type.

`Basis%nbaspt`

Type int_array

Description Cumulative number of basis sets (see nbset variable, for ‘set’), on a per atom type basis.

`Basis%nbos`

Type int

Description The total number of Cartesian basis functions, not counting the copies of the functions on the different atoms of the atom type: the functions are defined per atom type and are (for nbos) counted only once. Essentially, this means counting all functions with distinct characteristics (apart from their geometrical center.

`Basis%nbptr`

Type int_array

Description Index array of the nbos functions, where the entries are the cumulative numbers of functions.

`Basis%nbset`

Type int

Description The total number of basis ‘sets’, where a ‘set’ here means a Cartesian function set (3 for a p-type function, 6 for a d-type function, and so on),

`Basis%norde`

Type int_array

Description Maximum of the main quantum number for all STO basis and fit functions corresponding to that atom type.

Shape [Geometry%nr of atomtypes]

`Basis%nprta`

Type int_array

Description nprta(i) contains number of Cartesian basis function i in an ordering in which all Cartesian core orthogonalization functions precede all Cartesian valence functions.

Shape [naos]

`Basis%nqbas`

Type int_array

Description Main quantum number of each basis set. A 4p function has main quantum number 4.

Shape [nbset]

BeckeGridConfig **Section content:** Configuration used to create the Becke grid.

`BeckeGridConfig%angLOrder`

Type int_array

Description ?.

Shape [:]

`BeckeGridConfig%beckeMapParams`

Type float_array

Description Mapping parameter per atom.

Shape [nAtoms]

`BeckeGridConfig%includeRadialWeights`

Type bool

Description Whether or not to include the radial weights. Normally you want this.

`BeckeGridConfig%isSymmetryUnique`

Type bool_array

Description Is an atom symmetry unique?

Shape [nAtoms]

`BeckeGridConfig%minimumRadius`

Type float

Description To solve the exact singularity a small hard sphere around the nuclei can be used. The partition function starts beyond this radius.

`BeckeGridConfig%mpvPartitionCheckSpheres`

Type bool

Description Whether or not to check the spheres for the MPV partitioning.

`BeckeGridConfig%nAtoms`

Type int

Description Number of atoms.

`BeckeGridConfig%nRadPoints`

Type int_array

Description Number of radial points per atom.

Shape [nAtoms]

`BeckeGridConfig%oper`

Type float_array

Description Point group part of the symmetry operators.

Shape [3, 3, :]

`BeckeGridConfig%partitionFunThresh`

Type float

Description Threshold for the partition function.

`BeckeGridConfig%qAtoms`

Type float_array

Description Atomic number per atom.

Shape [nAtoms]

`BeckeGridConfig%quality`

Type string_fixed_length

Description Quality used.

`BeckeGridConfig%transl`

Type float_array

Description Translational part of the symmetry operators.

Shape [3, :]

`BeckeGridConfig%vectors`

Type float_array

Description Lattice vectors

Unit bohr

`BeckeGridConfig%xyzAtoms`

Type float_array

Description Atom coordinates.

Unit bohr

Shape [3, nAtoms]

ConstrainedDFT **Section content:** Contrained DFT data.

ConstructPotential **Section content:** Data related to FDE potential reconstruction.

ConstructPotential_FunctionSet **Section content:** Data related to FDE potential reconstruction.

Core **Section content:** Description of the auxiliary core functions and core orthogonalization functions.

`Core%alfcor`

Type float_array

Description Alpha values (the exponent in the Slater type-function) for the auxiliary core functions set.

Shape [ncset]

`Core%ccor`

Type float_array

Description All core coefficients, which expresses the core orbitals in the auxiliary core functions.

`Core%cmat`

Type float_array

Description Overlap matrix between Cartesian core orbitals and Cartesian basis functions. In the list of Cartesian basis functions all Cartesian core orthogonalization functions precede all Cartesian valence basis functions (see array Basis%nprta).

Shape [ncos, Basis%naos]

`Core%cornrm`

Type float_array

Description Normalization factors for the auxiliary core functions.

Shape [ncset]

`Core%idfcor`

Type int

Description Integer that indicates whether there are d- and/or f-orbitals in the core. 1=yes, 0=no.

`Core%kcos`

Type int

Description Total number of spherical harmonics core orthogonalization functions, counting 1 d-orbital as 5 functions, and 1 f-orbital as 7 functions.

`Core%kinetic_energy`

Type float

Description Kinetic energy of the core orbitals.

`Core%lqcor`

Type int_array

Description Angular momentum quantum numbers (l-value) for the auxiliary core functions set.

Shape [ncset]

`Core%nccpt`

Type int_array

Description Index array. 1 + cumulative number of Cartesian core orthogonalization functions.

Shape [Geometry%ntyp+1]

`Core%ncorpt`

Type int_array

Description Index array. 1 + cumulative number of auxiliary core functions set.

Shape [Geometry%ntyp+1]

`Core%ncos`

Type int

Description Total number of Cartesian core orthogonalization functions.

`Core%ncptr`

Type int_array

Description Index array. 1 + cumulative number of Cartesian auxiliary core functions.

Shape [Geometry%ntyp+1]

`Core%ncset`

Type int

Description The total number of auxiliary core functions set.

`Core%nd`

Type int

Description Total number of d-type auxiliary core functions set.

`Core%ndd`

Type int

Description Equals nd

`Core%ndfun`

Type int_array

Description Array of indices of d-orbital Cartesian core orthogonalization functions (the last of the Cartesian subset of size 6) in an ordering in which all Cartesian core orthogonalization functions precede all Cartesian valence functions (see array Basis%nprta). Equals Size ndd+1. Last element is 0.

`Core%ndorb`

Type int_array

Description Array of indices of d-orbital Cartesian core orthogonalization functions (the first of the Cartesian subset of size 6) in an ordering in which all Cartesian core orthogonalization functions precede all Cartesian valence functions (see array Basis%nprta). Size nd+1. Last element is 0.

`Core%nf`

Type int

Description Total number of f-type auxiliary core functions set.

`Core%nff`

Type int

Description Equals nf

`Core%nffun`

Type int_array

Description Array of indices of f-orbital Cartesian core orthogonalization functions (the last of the Cartesian subset of size 10) in an ordering in which all Cartesian core orthogonalization functions precede all Cartesian valence functions (see array Basis%nprta). Size nff+1. Last element is 0.

`Core%nforb`

Type int_array

Description Array of indices of f-orbital Cartesian core orthogonalization functions (the first of the Cartesian subset of size 10) in an ordering in which all Cartesian core orthogonalization functions precede all Cartesian valence functions (see array Basis%nprta). Size nf+1. Last element is 0.

`Core%npos`

Type int_array

Description Index for each atom where core data can be found on TAPE12 file created with Dirac.

Shape [Geometry%nnuc]

`Core%nqcor`

Type int_array

Description Main quantum numbers (n-value) for the auxiliary core functions set.

Shape [ncset]

`Core%nrcorb`

Type int_array

Description The number of frozen core orbitals per l-value and atom type.

Shape [4, Geometry%ntyp]

`Core%nrcset`

Type int_array

Description The number of auxiliary core functions set per l-value and atom type.

Shape [4, Geometry%ntyp]

`Core%s`

Type float_array

Description Overlap matrix between spherical harmonics core orbitals and spherical harmonics core orthogonalization functions.

Shape [kcos, kcos]

COSMO **Section content:** COSMO solvation model related data.

`COSMO%Area`

Type float

Description COSMO cavity surface area.

Unit bohr^2

`COSMO%Atom Coordinates`

Type float_array

Description Atom coordinates.

Unit angstrom

Shape [3, Number of Atoms]

`COSMO%Atom COSMO Radii`

Type float_array

Description Atom COSMO radii.

Unit bohr

Shape [Number of Atoms]

`COSMO%Atom Type`

Type lchar_string_array

Description Atom type names.

Shape [Number of Atoms]

`COSMO%Bond Energy`

Type float

Description Bond energy (including COSMO).

Unit hartree

`COSMO%Gas Phase Bond Energy`

Type float

Description Gas phase bond energy

Unit hartree

`COSMO%Number of Atoms`

Type int

Description Number of atoms.

`COSMO%Number of Segments`

Type int

Description Number of segments (number of COSMO surface points).

`COSMO%Segment Area`

Type float_array

Description COSMO surface point area.

Unit angstrom^2

Shape [Number of Segments]

`COSMO%Segment Atom`

Type int_array

Description Atom associated with COSMO surface point.

Shape [Number of Segments]

`COSMO%Segment Charge`

Type float_array

Description COSMO surface point charge.

Unit e

Shape [Number of Segments]

`COSMO%Segment Charge Density`

Type float_array

Description COSMO surface point charge density.

Unit e/angstrom^2

Shape [Number of Segments]

`COSMO%Segment Coordinates`

Type float_array

Description COSMO surface point coordinates.

Unit bohr

Shape [Number of Segments, 3]

`COSMO%Segment Potential`

Type float_array

Description COSMO surface point solute potential (angstrom length scale).

Shape [Number of Segments]

`COSMO%Volume`

Type float

Description COSMO cavity volume.

Unit bohr^3

CSMRSP **Section content:** COSMO response data.

CurrentMatrix **Section content:** Current response data.

DIMQM **Section content:** Data related to the DIM/QM procedure.

Dipole velocity matrix **Section content:** Dipole velocity matrix.

EgoData **Section content:** Data for the EGO (excited states gradients) procedure.

Elec multipole ints OCCOCC **Section content:** Data for XES.

Elec multipole ints OCCVIR **Section content:** Data for XAS.

ElstatEmbed **Section content:** The electrostatic embedding (i.e. point charges, electric field, …) used in the ADF calculation.

`ElstatEmbed%eeAttachTo`

Type archived_int_array

Description A multipole may be attached to an atom. This influences the energy gradient.

`ElstatEmbed%eeChargeWidth`

Type float

Description If charge broadening was used for external charges, this represents the width of the charge distribution

`ElstatEmbed%eeEField`

Type float_array

Description The external homogeneous electric field

Unit hartree/(e*bohr)

Shape [3]

`ElstatEmbed%eeLatticeVectors`

Type archived_float_array

Description The lattice vectors used for the external point- or multipole- charges

Unit bohr

`ElstatEmbed%eeMultipoles`

Type archived_float_array

Description The multiple charges.

Unit bohr

`ElstatEmbed%eenMulti`

Type int

Description The number of multipoles.

`ElstatEmbed%eeUseChargeBroadening`

Type bool

Description Whether or not the external charges are point-like or broadened

`ElstatEmbed%eeXYZ`

Type archived_float_array

Description The position of the external point- or multipole- charges

Unit bohr

Energy **Section content:** Energy terms related to energy decomposition analysis.

`Energy%Bond Energy`

Type float

Description Total bonding energy, same as the 'SCF Bond Energy' variable.

Unit hartree

`Energy%Corr. due to Orthogonalization`

Type float

Description For analysis purposes, the concept of 'orthogonalized fragments' has been introduced and the bonding energy is split in a part that describes the difference between the sum-of-fragments situation and the orthogonalized-fragments density at the one hand, and the SCF relaxation (from the orthogonalized fragments density) at the other. Both terms contain a first order fit correction term. The result of adding the two parts is not identical to computing the total bonding energy directly and applying the first order correction to that approach. The difference is given by this term, which therefore corrects for the additional second order fit errors caused by using the orthogonalized fragments split-up.

Unit hartree

`Energy%Dispersion Energy`

Type float

Description Dispersion energy.

Unit hartree

`Energy%Ebond due to Efield`

Type float

Description Bond energy term due to any homogeneous electric field.

Unit hartree

`Energy%Electrostatic Energy`

Type float

Description Electrostatic Energy.

Unit hartree

`Energy%Electrostatic Interaction`

Type float

Description The electrostatic interaction energy including any first order fit correction (if computed from the fit density).

Unit hartree

`Energy%Elstat FitCorrection`

Type float

Description The first-order correction to the electrostatic interaction term (putting the fragments together, without any relaxation of Pauli orthogonalization), for the error in the Coulomb energy due to the fit incompleteness.

Unit hartree

`Energy%Elstat Interaction`

Type float

Description Elstat interaction.

Unit hartree

`Energy%Excited State Bond Energy`

Type float

Description Excited State Bond Energy.

Unit hartree

`Energy%HF Exen Ort`

Type float

Description HF exchange energy using orthonormalized orbitals.

Unit hartree

`Energy%HF Exen SCF`

Type float

Description HF exchange energy using SCF orbitals.

Unit hartree

`Energy%HF SO Exen SCF`

Type float

Description HF exchange energy using SCF orbitals related to spin-orbit coupling.

Unit hartree

`Energy%HF SOU Exen SCF`

Type float

Description HF exchange energy using SCF orbitals related to open shell spin-orbit coupling.

Unit hartree

`Energy%Hybrid Exen Ort`

Type float

Description HF exchange part hybrid exchange energy using orthonormalized orbitals.

Unit hartree

`Energy%Hybrid Exen SCF`

Type float

Description HF exchange part hybrid exchange energy using SCF orbitals.

Unit hartree

`Energy%Hybrid SO Exen SCF`

Type float

Description HF exchange part hybrid exchange energy using SCF orbitals related to spin-orbit coupling.

Unit hartree

`Energy%Hybrid SOU Exen SCF`

Type float

Description HF exchange part hybrid exchange energy using SCF orbitals related to open shell spin-orbit coupling.

Unit hartree

`Energy%Kinetic Energy`

Type float

Description Kinetic energy part of bonding energy.

Unit hartree

`Energy%MP2 energy`

Type float

Description MP2 energy.

Unit hartree

`Energy%Orb.Int. *`

Type float

Description

- stands for one of the irreps of the point group symmetry. The value gives the orbital interaction (SCF relaxation) term for that symmetry representation.

Unit hartree

`Energy%Orb.Int. Efield`

Type float

Description The contribution to the SCF relaxation energy (orbital interactions) due to any electric field.

Unit hartree

`Energy%Orb.Int. FitCorrection`

Type float

Description The first-order correction to the electrostatic interaction term in the SCF relaxation energy (Orbital Interactions), for the error in the Coulomb energy due to the fit incompleteness. This term is not printed (anymore) separately, but incorporated in the symmetry-specific interaction terms.

Unit hartree

`Energy%Orb.Int. Total`

Type float

Description The total orbital interaction energy.

Unit hartree

`Energy%Orb.Int. TSCorrection (LDA)`

Type float

Description The difference between the representation-specific orbital interaction terms added, and a straightforward computation of the SCF relaxation energy is the result of the neglect of higher order terms in the Taylor expansion that underlies the 'Transition State' method. This difference, therefore, corrects exactly this neglect. It is not printed separately anymore in the output, but incorporated in (distributed over) the representation-specific orbital interaction terms. LDA part.

Unit hartree

`Energy%Orb.Int. TSCorrection (NL)`

Type float

Description Similar as Orb.Int. TSCorrection (LDA), but then for GGA part.

Unit hartree

`Energy%Pauli Coulomb`

Type float

Description Coulomb energy term in the Pauli exchange interaction energy.

Unit hartree

`Energy%Pauli Efield`

Type float

Description The contribution to the Pauli interaction energy due to any electric field.

Unit hartree

`Energy%Pauli FitCorrection`

Type float

Description The first-order correction to the Pauli exchange interaction term, for the error in the Coulomb energy due to the fit incompleteness. This correction term is not printed in the output file but included in the Pauli interaction term.

Unit hartree

`Energy%Pauli Kinetic`

Type float

Description Kinetic energy term in the Pauli exchange interaction energy.

Unit hartree

`Energy%Pauli Kinetic+Coulomb`

Type float

Description Sum of the kinetic and Coulomb terms in the Pauli exchange interaction energy.

Unit hartree

`Energy%Pauli Total`

Type float

Description The Pauli exchange (orbital orthogonalization) interaction energy.

Unit hartree

`Energy%Pauli TS Correction (LDA)`

Type float

Description Correction to the 'Transition State' method to compute terms in the bonding energy, in this case the Pauli exchange energy term. The Pauli TS Correction is not separately printed in the standard output file, but included in the Pauli interaction term.

Unit hartree

`Energy%post-SCF Orb.Int. Correlation`

Type float

Description post-SCF orbital interaction Correlation energy.

Unit hartree

`Energy%post-SCF Orb.Int. Exchange`

Type float

Description post-SCF orbital interaction Exchange energy.

Unit hartree

`Energy%post-SCF Pauli Correlation`

Type float

Description post-SCF Pauli Correlation energy.

Unit hartree

`Energy%post-SCF Pauli Exchange`

Type float

Description post-SCF Pauli Exchange energy.

Unit hartree

`Energy%RPA energy`

Type float

Description RPA energy.

Unit hartree

`Energy%SCF Bond Energy`

Type float

Description Total bonding energy.

Unit hartree

`Energy%Solvation Energy (cd)`

Type float

Description COSMO Solvation Energy (cd).

Unit hartree

`Energy%Solvation Energy (el)`

Type float

Description COSMO Solvation Energy (el).

Unit hartree

`Energy%Solvation Energy (Gcds)`

Type float

Description SM12 Solvation Energy (Gcds).

Unit hartree

`Energy%Solvation Energy (Gp)`

Type float

Description SM12 Solvation Energy (Gp).

Unit hartree

`Energy%Steric Total`

Type float

Description The total steric interaction energy, consisting of the electrostatic and the Pauli interactions.

Unit hartree

`Energy%SumFragmentsSCF FitCorrection`

Type float

Description The 'true' first order fit correction for the complete bonding energy, resulting from a direct calculation that takes the sum-of-fragments as starting point and the SCF as final situation, without the intermediate step of orthogonalized fragments.

Unit hartree

`Energy%Total BE without XC`

Type float

Description Total bonding energy without XC energy.

Unit hartree

`Energy%Total BE without XC and Disp`

Type float

Description Total bonding energy without XC energy and without Dispersion energy.

Unit hartree

`Energy%XC energies`

Type float_array

Description exchange-correlation energies of various charge densities: first index: 1=exchange term, 2=correlation term second index: 1=lda tern, 2=gga term third index: 1=energy of fragments (summed over fragments), 2=energy of sum-of-fragments density, 3=energy of orthogonalized fragments, 4=SCF.

Unit hartree

Shape [2, 2, 4]

`Energy%XC Energy`

Type float

Description XC energy

Unit hartree

`Energy%XC SOU spin-polarization energy`

Type float_array

Description part of XC energy contributions in case of open shell spin-orbit coupling.

Unit hartree

Shape [2, 2]

ETS **Section content:** Data used for the energy decomposistion scheme (EDA), mostly related to core orbitals. ETS is the Extended Transition State method.

Excitations SO * **Section content:** Spin-orbit excitations for a given irrep

`Excitations SO *%coef_Trans #{ExcitationNumber}`

Type float_array

Description Transition density fit coefficients in case of STOFIT.

`Excitations SO *%contr #{ExcitationNumber}`

Type float_array

Description Large contributions excited state vector expressed in single-orbital transitions.

Shape [nr of contributions #{ExcitationNumber}]

`Excitations SO *%contr index #{ExcitationNumber}`

Type int_array

Description Indices of the occupied and virtual orbital (spinor) of the single-orbital transition for which the excited state vector has large contributions.

Shape [nr of contributions #{ExcitationNumber}, :]

`Excitations SO *%contr irep index #{ExcitationNumber}`

Type int_array

Description Symmetry indices of the occupied and virtual orbital (spinor) of the single-orbital transition for which the excited state vector has large contributions.

Shape [nr of contributions #{ExcitationNumber}, :]

`Excitations SO *%contr slab #`

Type lchar_string_array

Description Spin symmetry label for each contribution. Size [nr of contributions #].

`Excitations SO *%ct_at_pl #{ExcitationNumber}`

Type float

Description Charge transfer descriptor CT_AT: an atomic distance criterion is used, based on Plasser, Lischka, et al. A larger value means more charge transfer character.

`Excitations SO *%ct_pl #{ExcitationNumber}`

Type float

Description Charge transfer descriptor CT: fragment based, based on Plasser, Lischka, et al. A larger value means more charge transfer character.

`Excitations SO *%CV2DFT excenergies`

Type float_array

Description CV(2)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations SO *%Degeneracy`

Type int_array

Description MCD related data related to degeneracy of excited states.

Shape [nr of excenergies]

`Excitations SO *%electronic couplings`

Type float_array

Description Electronic couplings calculated localized excitation energies.

Shape [nr of localized excenergies, nr of localized excenergies]

`Excitations SO *%excenergies`

Type float_array

Description Excitation energies.

Unit hartree

Shape [nr of excenergies]

`Excitations SO *%gradient #{ExcitationNumber}`

Type float_array

Description Excited state gradients (input order).

Shape [3, Molecule%nAtoms]

`Excitations SO *%Gradients_CART #{ExcitationNumber}`

Type float_array

Description Excited state gradients (internal order).

Shape [3, Molecule%nAtoms]

`Excitations SO *%index in all localized`

Type int_array

Description Index array for one type of localized excitations in list of all localized excitations with the same irrep.

Shape [nr of excenergies]

`Excitations SO *%lambda_pt #{ExcitationNumber}`

Type float

Description Charge-transfer diagnostic overlap quantity LAMBDA, based on Peach, Tozer, et al. A smaller value means more charge transfer character.

`Excitations SO *%localized excenergies`

Type float_array

Description Localized excitation energies.

Shape [nr of localized excenergies]

`Excitations SO *%magnetic trans dip`

Type float_array

Description Magnetic transition dipole moment.

Shape [3, nr of excenergies]

`Excitations SO *%nr of contributions #{ExcitationNumber}`

Type int

Description Number of large contributions excited state vector expressed in single-orbital transitions.

`Excitations SO *%nr of densities`

Type int

Description Number of single-orbital transitions.

`Excitations SO *%nr of excenergies`

Type int

Description Number of calculated excitation energies.

`Excitations SO *%nr of localized excenergies`

Type int

Description Number of calculated localized excitation energies.

`Excitations SO *%nr of NTOs #{ExcitationNumber}`

Type int

Description Number of large contributions excited state vector expressed in NTO transitions.

`Excitations SO *%NTO #{NTOindex} occ #{ExcitationNumber}`

Type float_array

Description Occupied natural transition orbital (NTO) expressed in Cartesian basis functions (BAS). In case of spinor [Re alpha,Re beta,Im alpha,Im beta].

Shape [Basis%naos, :]

`Excitations SO *%NTO #{NTOindex} virt #{ExcitationNumber}`

Type float_array

Description Virtual natural transition orbital (NTO) expressed in Cartesian basis functions (BAS). In case of spinor [Re alpha,Re beta,Im alpha,Im beta].

Shape [Basis%naos, :]

`Excitations SO *%NTO contr #{ExcitationNumber}`

Type float_array

Description Large contributions excited state vector expressed in NTO transitions.

Shape [nr of NTOs #{ExcitationNumber}]

`Excitations SO *%oscillator strengths`

Type float_array

Description Oscillator strengths for absorption on ground state to excited state.

Unit

Shape [nr of excenergies]

`Excitations SO *%R-CVnDFT excenergies`

Type float_array

Description R-CV(n)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations SO *%rhe_pl #{ExcitationNumber}`

Type float

Description Charge-transfer descriptor R_HE: hole-electron distance based on Plasser, Lischka, et al.

`Excitations SO *%rhe_pt #{ExcitationNumber}`

Type float

Description Charge-transfer descriptor R_HE: hole-electron distance based on Guido, Adamo, et al.

`Excitations SO *%rotatory strengths`

Type float_array

Description Rotatory strengths.

Shape [nr of excenergies]

`Excitations SO *%SCF-CVnDFT excenergies`

Type float_array

Description SCF-CV(n)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations SO *%SOmat-I`

Type float_array

Description Imaginary part spin-orbit matrix in case of perturbative inclusion of spin-orbit coupling. Packed matrix.

Unit hartree

`Excitations SO *%SOmat-R`

Type float_array

Description Real part spin-orbit matrix in case of perturbative inclusion of spin-orbit coupling in the basis of solutions without spin-orbit coupling. On the diagonal the excitation energy without spin-irbit coupling is added. Packed matrix.

Unit hartree

`Excitations SO *%symmetry label`

Type string

Description Double group symmetry label.

`Excitations SO *%transition dipole moments`

Type float_array

Description Transition dipole moment between ground state and excited state.

Unit bohr

Shape [3, nr of excenergies]

`Excitations SO *%transition dipoles dip-vel`

Type float_array

Description Velocity formula used for calculated transition dipole moment between ground state and excited state.

Shape [3, nr of excenergies]

Excitations SS * **Section content:** Singlet-Singlet excitations for a given irrep.

`Excitations SS *%coef_Trans #{ExcitationNumber}`

Type float_array

Description Transition density fit coefficients in case of STOFIT.

`Excitations SS *%contr #{ExcitationNumber}`

Type float_array

Description Large contributions excited state vector expressed in single-orbital transitions.

Shape [nr of contributions #{ExcitationNumber}]

`Excitations SS *%contr index #{ExcitationNumber}`

Type int_array

Description Indices of the occupied and virtual orbital (spinor) of the single-orbital transition for which the excited state vector has large contributions.

Shape [nr of contributions #{ExcitationNumber}, :]

`Excitations SS *%contr irep index #{ExcitationNumber}`

Type int_array

Description Symmetry indices of the occupied and virtual orbital (spinor) of the single-orbital transition for which the excited state vector has large contributions.

Shape [nr of contributions #{ExcitationNumber}, :]

`Excitations SS *%contr spin #{ExcitationNumber}`

Type int_array

Description Integer related to spin symmetry of contribution.

Shape [nr of contributions #{ExcitationNumber}, 2]

`Excitations SS *%contr transdip #{ExcitationNumber}`

Type float_array

Description Contribution to transition dipole moment.

Shape [3, nr of contributions #{ExcitationNumber}]

`Excitations SS *%ct_at_pl #{ExcitationNumber}`

Type float

Description Charge transfer descriptor CT_AT: an atomic distance criterion is used, based on Plasser, Lischka, et al. A larger value means more charge transfer character.

`Excitations SS *%ct_pl #{ExcitationNumber}`

Type float

Description Charge transfer descriptor CT: fragment based, based on Plasser, Lischka, et al. A larger value means more charge transfer character.

`Excitations SS *%CV2DFT excenergies`

Type float_array

Description CV(2)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations SS *%Degeneracy`

Type int_array

Description MCD related data related to degeneracy of excited states.

Shape [nr of excenergies]

`Excitations SS *%eigenvector #{ExcitationNumber}`

Type float_array

Description Excited state eigenvector expressed in single-orbital transitions.

`Excitations SS *%electronic couplings`

Type float_array

Description Electronic couplings calculated localized excitation energies.

Shape [nr of localized excenergies, nr of localized excenergies]

`Excitations SS *%excenergies`

Type float_array

Description Excitation energies.

Unit hartree

Shape [nr of excenergies]

`Excitations SS *%Exctyp *`

Type int_array

Description Excited state type vector expressed in symmetry adapted single-orbital transitions.

Shape [nr of densities]

`Excitations SS *%F Vectors *`

Type float_array

Description Excited state eigenvector expressed in symmetry adapted single-orbital transitions.

Shape [nr of densities]

`Excitations SS *%gradient #{ExcitationNumber}`

Type float_array

Description Excited state gradients (input order).

Shape [3, Molecule%nAtoms]

`Excitations SS *%Gradients_CART #{ExcitationNumber}`

Type float_array

Description Excited state gradients (internal order).

Shape [3, Molecule%nAtoms]

`Excitations SS *%index in all localized`

Type int_array

Description Index array for one type of localized excitations in list of all localized excitations with the same irrep.

Shape [nr of excenergies]

`Excitations SS *%lambda_pt #{ExcitationNumber}`

Type float

Description Charge-transfer diagnostic overlap quantity LAMBDA, based on Peach, Tozer, et al. A smaller value means more charge transfer character.

`Excitations SS *%left eigenvector #{ExcitationNumber}`

Type float_array

Description Excited state eigenvector (X-Y) expressed in single-orbital transitions.

`Excitations SS *%localized excenergies`

Type float_array

Description Localized excitation energies.

Shape [nr of localized excenergies]

`Excitations SS *%magnetic trans dip`

Type float_array

Description Magnetic transition dipole moment.

Shape [3, nr of excenergies]

`Excitations SS *%nr of contributions #{ExcitationNumber}`

Type int

Description Number of large contributions excited state vector expressed in single-orbital transitions.

`Excitations SS *%nr of densities`

Type int

Description Number of single-orbital transitions.

`Excitations SS *%nr of excenergies`

Type int

Description Number of calculated excitation energies.

`Excitations SS *%nr of localized excenergies`

Type int

Description Number of calculated localized excitation energies.

`Excitations SS *%nr of NTOs #{ExcitationNumber}`

Type int

Description Number of large contributions excited state vector expressed in NTO transitions.

`Excitations SS *%NTO #{NTOindex} occ #{ExcitationNumber}`

Type float_array

Description Occupied natural transition orbital (NTO) expressed in Cartesian basis functions (BAS). In case of spinor [Re alpha,Re beta,Im alpha,Im beta].

Shape [Basis%naos, :]

`Excitations SS *%NTO #{NTOindex} virt #{ExcitationNumber}`

Type float_array

Description Virtual natural transition orbital (NTO) expressed in Cartesian basis functions (BAS). In case of spinor [Re alpha,Re beta,Im alpha,Im beta].

Shape [Basis%naos, :]

`Excitations SS *%NTO contr #{ExcitationNumber}`

Type float_array

Description Large contributions excited state vector expressed in NTO transitions.

Shape [nr of NTOs #{ExcitationNumber}]

`Excitations SS *%oscillator strengths`

Type float_array

Description Oscillator strengths for absorption on ground state to excited state.

Unit

Shape [nr of excenergies]

`Excitations SS *%R-CVnDFT excenergies`

Type float_array

Description R-CV(n)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations SS *%rhe_pl #{ExcitationNumber}`

Type float

Description Charge-transfer descriptor R_HE: hole-electron distance based on Plasser, Lischka, et al.

`Excitations SS *%rhe_pt #{ExcitationNumber}`

Type float

Description Charge-transfer descriptor R_HE: hole-electron distance based on Guido, Adamo, et al.

`Excitations SS *%rotatory strengths`

Type float_array

Description Rotatory strengths.

Shape [nr of excenergies]

`Excitations SS *%SCF-CVnDFT excenergies`

Type float_array

Description SCF-CV(n)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations SS *%transition dipole moments`

Type float_array

Description Transition dipole moment between ground state and excited state.

Unit bohr

Shape [3, nr of excenergies]

`Excitations SS *%transition dipoles dip-vel`

Type float_array

Description Velocity formula used for calculated transition dipole moment between ground state and excited state.

Shape [3, nr of excenergies]

`Excitations SS *%unrelaxed dipole moments`

Type float_array

Description Unrelaxed dipole moments of the excited state.

Shape [3, nr of excenergies]

Excitations ST * **Section content:** Singlet-Triplet excitations for a given irrep.

`Excitations ST *%coef_Trans #{ExcitationNumber}`

Type float_array

Description Transition density fit coefficients in case of STOFIT.

`Excitations ST *%contr #{ExcitationNumber}`

Type float_array

Description Large contributions excited state vector expressed in single-orbital transitions.

Shape [nr of contributions #{ExcitationNumber}]

`Excitations ST *%contr index #{ExcitationNumber}`

Type int_array

Description Indices of the occupied and virtual orbital (spinor) of the single-orbital transition for which the excited state vector has large contributions.

Shape [nr of contributions #{ExcitationNumber}, :]

`Excitations ST *%contr irep index #{ExcitationNumber}`

Type int_array

Description Symmetry indices of the occupied and virtual orbital (spinor) of the single-orbital transition for which the excited state vector has large contributions.

Shape [nr of contributions #{ExcitationNumber}, :]

`Excitations ST *%contr spin #{ExcitationNumber}`

Type int_array

Description Integer related to spin symmetry of contribution.

Shape [nr of contributions #{ExcitationNumber}, 2]

`Excitations ST *%contr transdip #{ExcitationNumber}`

Type float_array

Description Contribution to transition dipole moment.

Shape [3, nr of contributions #{ExcitationNumber}]

`Excitations ST *%ct_at_pl #{ExcitationNumber}`

Type float

Description Charge transfer descriptor CT_AT: an atomic distance criterion is used, based on Plasser, Lischka, et al. A larger value means more charge transfer character.

`Excitations ST *%ct_pl #{ExcitationNumber}`

Type float

Description Charge transfer descriptor CT: fragment based, based on Plasser, Lischka, et al. A larger value means more charge transfer character.

`Excitations ST *%CV2DFT excenergies`

Type float_array

Description CV(2)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations ST *%Degeneracy`

Type int_array

Description MCD related data related to degeneracy of excited states.

Shape [nr of excenergies]

`Excitations ST *%eigenvector #{ExcitationNumber}`

Type float_array

Description Excited state eigenvector expressed in single-orbital transitions.

`Excitations ST *%electronic couplings`

Type float_array

Description Electronic couplings calculated localized excitation energies.

Shape [nr of localized excenergies, nr of localized excenergies]

`Excitations ST *%excenergies`

Type float_array

Description Excitation energies.

Unit hartree

Shape [nr of excenergies]

`Excitations ST *%Exctyp *`

Type int_array

Description Excited state type vector expressed in symmetry adapted single-orbital transitions.

Shape [nr of densities]

`Excitations ST *%F Vectors *`

Type float_array

Description Excited state eigenvector expressed in symmetry adapted single-orbital transitions.

Shape [nr of densities]

`Excitations ST *%gradient #{ExcitationNumber}`

Type float_array

Description Excited state gradients (input order).

Shape [3, Molecule%nAtoms]

`Excitations ST *%Gradients_CART #{ExcitationNumber}`

Type float_array

Description Excited state gradients (internal order).

Shape [3, Molecule%nAtoms]

`Excitations ST *%index in all localized`

Type int_array

Description Index array for one type of localized excitations in list of all localized excitations with the same irrep.

Shape [nr of excenergies]

`Excitations ST *%lambda_pt #{ExcitationNumber}`

Type float

Description Charge-transfer diagnostic overlap quantity LAMBDA, based on Peach, Tozer, et al. A smaller value means more charge transfer character.

`Excitations ST *%left eigenvector #{ExcitationNumber}`

Type float_array

Description Excited state eigenvector (X-Y) expressed in single-orbital transitions.

`Excitations ST *%localized excenergies`

Type float_array

Description Localized excitation energies.

Shape [nr of localized excenergies]

`Excitations ST *%magnetic trans dip`

Type float_array

Description Magnetic transition dipole moment.

Shape [3, nr of excenergies]

`Excitations ST *%nr of contributions #{ExcitationNumber}`

Type int

Description Number of large contributions excited state vector expressed in single-orbital transitions.

`Excitations ST *%nr of densities`

Type int

Description Number of single-orbital transitions.

`Excitations ST *%nr of excenergies`

Type int

Description Number of calculated excitation energies.

`Excitations ST *%nr of localized excenergies`

Type int

Description Number of calculated localized excitation energies.

`Excitations ST *%nr of NTOs #{ExcitationNumber}`

Type int

Description Number of large contributions excited state vector expressed in NTO transitions.

`Excitations ST *%NTO #{NTOindex} occ #{ExcitationNumber}`

Type float_array

Description Occupied natural transition orbital (NTO) expressed in Cartesian basis functions (BAS). In case of spinor [Re alpha,Re beta,Im alpha,Im beta].

Shape [Basis%naos, :]

`Excitations ST *%NTO #{NTOindex} virt #{ExcitationNumber}`

Type float_array

Description Virtual natural transition orbital (NTO) expressed in Cartesian basis functions (BAS). In case of spinor [Re alpha,Re beta,Im alpha,Im beta].

Shape [Basis%naos, :]

`Excitations ST *%NTO contr #{ExcitationNumber}`

Type float_array

Description Large contributions excited state vector expressed in NTO transitions.

Shape [nr of NTOs #{ExcitationNumber}]

`Excitations ST *%oscillator strengths`

Type float_array

Description Oscillator strengths for absorption on ground state to excited state.

Unit

Shape [nr of excenergies]

`Excitations ST *%R-CVnDFT excenergies`

Type float_array

Description R-CV(n)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations ST *%rhe_pl #{ExcitationNumber}`

Type float

Description Charge-transfer descriptor R_HE: hole-electron distance based on Plasser, Lischka, et al.

`Excitations ST *%rhe_pt #{ExcitationNumber}`

Type float

Description Charge-transfer descriptor R_HE: hole-electron distance based on Guido, Adamo, et al.

`Excitations ST *%rotatory strengths`

Type float_array

Description Rotatory strengths.

Shape [nr of excenergies]

`Excitations ST *%SCF-CVnDFT excenergies`

Type float_array

Description SCF-CV(n)-DFT excitation energies.

Shape [nr of excenergies]

`Excitations ST *%transition dipole moments`

Type float_array

Description Transition dipole moment between ground state and excited state.

Unit bohr

Shape [3, nr of excenergies]

`Excitations ST *%transition dipoles dip-vel`

Type float_array

Description Velocity formula used for calculated transition dipole moment between ground state and excited state.

Shape [3, nr of excenergies]

`Excitations ST *%unrelaxed dipole moments`

Type float_array

Description Unrelaxed dipole moments of the excited state.

Shape [3, nr of excenergies]

FDE Energy **Section content:** FDE Related data.

FDEIntegrals **Section content:** FDE Related data.

Fit **Section content:** Fit functions data for the STO fit procedure.

FitCoefficients **Section content:** Fit coefficients for STO fit procedure

FitFit **Section content:** Data for polTDDFT.

FQQM **Section content:** Data related to QM/FQ

`FQQM%chemical hardnesses`

Type float_array

Description ?

`FQQM%electronegativities`

Type float_array

Description ?

`FQQM%external xyz`

Type float_array

Description ?

`FQQM%fqind`

Type float_array

Description ?

`FQQM%fqind_r`

Type float_array

Description ?

`FQQM%group names`

Type string

Description ?

`FQQM%group num`

Type int_array

Description ?

`FQQM%n atoms per mol`

Type int_array

Description ?

`FQQM%n mol index`

Type int_array

Description ?

`FQQM%name`

Type string

Description ?

`FQQM%total charges`

Type float_array

Description ?

`FQQM%type index`

Type int_array

Description ?

`FQQM%type name`

Type string

Description ?

Fragdepend **Section content:** Information regarding almost linear dependent orbitals which are removed from fragments.

Fragments **Section content:** May contain variable Pmat_SumFrag written as packed reals.

Freq **Section content:** Internal ADF data concerning frequencies. Use AMSResults data.

`Freq%DipoleDerivatives_RAW`

Type float_array

Description Internal ADF data concerning dipole derivatives. Use AMSResults data.

Freq Symmetry **Section content:** Symmetry information for frequencies…?

Ftyp* **Section content:** Fragment type data.

General **Section content:** General information about the ADF calculation.

`General%account`

Type string

Description Name of the account from the license

`General%doublehybrid`

Type string

Description Double hybrid XC functional used in the calculation

`General%electrons`

Type float

Description Number of (valence) electrons in the calcualtion. Note that this is not necessarily the same as what may consider, chemically, as the valence space. Rather, it equals the total number of electrons in the calculation minus the electrons in the frozen core orbitals

`General%engine input`

Type string

Description The text input of the engine.

`General%engine messages`

Type string

Description Message from the engine. In case the engine fails to solves, this may contains extra information on why.

`General%file-ident`

Type string

Description The file type identifier, e.g. RKF, RUNKF, TAPE21…

`General%ggaen`

Type string

Description GGA density functional part for ENERGY

`General%ggapot`

Type string

Description GGA potential used.

`General%ggapotfd`

Type string

Description FDE GGA potential used in kinetic energy approximant

`General%hybrid`

Type string

Description Hybrid XC functional used.

`General%iopcor`

Type int

Description Deprecated.

`General%iopnuc`

Type int

Description Nuclear model used. 1: point charge nucleus, 2: Uniform Sphere nucleus, 3: Gaussian nucleus model.

`General%ioprel`

Type int

Description Integer code for relativistic option used. 0: non-relativistic, 1: scalar Pauli + sum of frozen core pot., 3: scalar ZORA + MAPA, 4: scalar ZORA + full pot. (not supported anymore), 5: scalar ZORA + APA (Band), 6: scalar X2C + MAPA, 7: scalar X2C ZORA + MAPA, 11: spin-orbit Pauli + sum of frozen core pot., 13: spin-orbit ZORA + MAPA, 14: spin-orbit ZORA + full pot. (not supported anymore), 15: spin-orbit ZORA + APA (Band), 16: spin-orbit X2C + MAPA, 17: spin-orbit X2C ZORA + MAPA

`General%isounr`

Type int

Description Integer code for how to treat the spin in the xc functional with spin-orbit coupling.

`General%itnad`

Type int

Description FDE integer used in kinetic energy approximant

`General%jobid`

Type int

Description Unique identifier for the job.

`General%lcjcorr`

Type bool

Description FDE use long-distance correction in kinetic energy approximant

`General%ldaen`

Type int

Description Integer code for type of local XC energy.

`General%ldapot`

Type int

Description Integer code for type of local XC potential.

`General%ldapot_fde`

Type int

Description FDE Integer code for type of local XC functional used in kinetic energy approximant

`General%ldoublehybrid`

Type bool

Description Whether a double hybrid XC functional was used in the calculation.

`General%lfrozend`

Type bool

Description Whether Frozen Density Embedding (FDE) is used.

`General%lhybrid`

Type bool

Description Whether a hybrid functional was used in the calculation.

`General%lunrfrag`

Type bool

Description Whether real unrestricted fragments were used.

`General%Molecular_Weight`

Type float

Description The molecular weight of the molecule.

Unit dalton

`General%nspin`

Type int

Description Number of 'spins' in the calculation. 1: spin-restricted calculation, 2: spin-unrestricted calculation.

Possible values [1, 2]

`General%nspinf`

Type int

Description Same as nspin but for a fragment. In case of FRAGOCCUPATIONS nspinf = 2.

`General%program`

Type string

Description The name of the program/engine that generated this kf file.

`General%release`

Type string

Description The version of the program that generated this kf file (including svn revision number and date).

`General%runtype`

Type string

Description Run type for ADF calculation. 'CREATE' in the create run of an atom, otherwise 'SINGLE POINT'.

`General%scfmod`

Type string

Description String indicating whether SCF converged (moderately) or not.

`General%termination status`

Type string

Description The termination status. Possible values: 'NORMAL TERMINATION', 'NORMAL TERMINATION with warnings', 'NORMAL TERMINATION with errors', 'ERROR', 'IN PROGRESS'.

`General%title`

Type string

Description Title of the calculation.

`General%uid`

Type string

Description SCM User ID

`General%unit of angle`

Type float

Description Deprecated: conversion factor for unit of angle

`General%unit of length`

Type float

Description Deprecated: conversion factor for unit of length

`General%use_libxc`

Type bool

Description Whether the LIBXC library was used.

`General%use_xcfun`

Type bool

Description Whether XCFUN automatic differentiation to generate functional and derivatives is used.

`General%use_xcfun_fde_nadxc`

Type bool

Description FDE XCfun is used in kinetic energy approximant

`General%user input`

Type string

Description User input ADF part.

`General%version`

Type int

Description Version number?

`General%xcpare`

Type float

Description Parameter for X-alpha (only relevant for X-alpha) energy evaluation

`General%xcparv`

Type float

Description Parameter for X-alpha (only relevant for X-alpha) potential

`General%xcparv_fde`

Type float

Description FDE parameter for X-alpha used in kinetic energy approximant

GenptData Section content: Data related to grid generation for the auxiliary programs. Technical.

Geometry Section content: The geometry of the system. Note: ADF internally sorts the atoms in what is referred to as 'internal order' (as opposed to 'input order', which is the order in which atoms were specified in the input). Atomic arrays in this section are (unless otherwise specified) stored in 'internal order'. The variable 'atom order index' contains the mapping from 'input order' to 'internal order' (and viceversa).

`Geometry%atom order index`

Type int_array

Description Mapping from input order to internal order. atom order index(:,1): input atom order –> internal atom order; atom order index(:,2) is the inverse.

Shape [nr of atoms, 2]

`Geometry%Atomic Distances`

Type float_array

Description Inter-atomic distances between the atoms, shape: [0:nr of atoms, 0:nr of atoms]. The elements (iAtom,jAtom) are the interatomic distances between the two atoms (the index of the first atom being 1). The elements (0,k), (k,0): nearest neighbour for atom k. The element (0,0) is the overall smallest interatomic distance.

Shape [nr of atoms+1, nr of atoms+1]

`Geometry%atomtype`

Type lchar_string_array

Description Atom type names.

Shape [nr of atomtypes]

`Geometry%atomtype effective charge`

Type float_array

Description Effective charge of the atom types, i.e. nuclear charge minus the number of electrons in the frozen core.

Unit e

Shape [nr of atomtypes]

`Geometry%atomtype total charge`

Type float_array

Description Nuclear charge of the atom types.

Unit e

Shape [nr of atomtypes]

`Geometry%charge`

Type float_array

Description Atomic charges for the atom types. The first row is the nuclear charge of the atom type. The second row is the effective nuclear charge (i.e. nuclear charge minus the number of electrons in the frozen core). The third: ?.

Unit e

Shape [nr of atomtypes, 3]

`Geometry%cum nr of atoms`

Type int_array

Description Cumulative number of atoms, up to a certain atomtype. The first element of this array is always zero. The last element is always the total number ot atoms.

Shape [nr of atomtypes+1]

`Geometry%cum nr of fragments`

Type int_array

Description Cumulative number of fragments of a certain type.

Shape [nr of fragmenttypes+1]

`Geometry%fframe`

Type lchar_string_array

Description Signals whether or not special local coordinate frames are used for the atoms. Usually this is not so, in which case the variable has the value DEFAULT. See the 'z=' option (Orientation of Local Atomic Coordinates)

Shape [nr of atoms]

`Geometry%fragment and atomtype index`

Type int_array

Description Array containing, for each atom, the index of it's fragment type and atom type ((iAtom,1) -> fragment index, (iAtom,2) -> atom type index).

Shape [nr of atoms, 2]

`Geometry%fragmenttype`

Type lchar_string_array

Description The fragment types (i.e. the fragment names).

Shape [nr of fragmenttypes]

`Geometry%Geometric Symmetry`

Type string

Description Auto-determined ('true') symmetry (considering the nuclear frame and any external fields, but not taking into account any user-defined MO occupation numbers and hence the electronic charge distribution.)

`Geometry%geometry id`

Type string

Description String id that identifies the structure

`Geometry%grouplabel`

Type string

Description Symmetry used in the electronic structure calculation (this is a subgroup of 'Geometric Symmetry').

`Geometry%lrotat`

Type bool

Description Whether or not a rotation has been applied between the input frame and the internally used frame.

`Geometry%mass`

Type float_array

Description Atomic masses for the atom types.

Unit dalton

Shape [nr of atomtypes]

`Geometry%nfragm`

Type int

Description Number of fragments (same as 'nr of fragments'?). Sometimes it is not, e.g. in Test/FDE_EO_FHF… why?

`Geometry%nnuc`

Type int

Description Number of nuclei.

`Geometry%nofrag_1`

Type int_array

Description Array specifying for each non-dummy atom the fragment it belongs to.

Shape [nr of atoms]

`Geometry%nofrag_2`

Type int_array

Description Array specifying for each non-dummy atom the fragment type it belongs to.

Shape [nr of atoms]

`Geometry%nqptr`

Type int_array

Description Index of the first atom of particular atom type.

Shape [nr of atomtypes+1]

`Geometry%nr of atoms`

Type int

Description Number of atoms.

`Geometry%nr of atomtypes`

Type int

Description Number of atom types.

`Geometry%nr of dummy fragments`

Type int

Description Number of dummy fragments.

`Geometry%nr of dummy fragmenttypes`

Type int

Description Number of dummy fragment types.

`Geometry%nr of fragments`

Type int

Description Number of fragments.

`Geometry%nr of fragmenttypes`

Type int

Description Number of fragment types.

`Geometry%ntyp`

Type int

Description Number of atom types (same as 'nr of atomtypes').

`Geometry%nuclab`

Type lchar_string_array

Description Nuclear labels (how is this different from atomtype? is it just the atomic symbol?).

Shape [nr of atomtypes]

`Geometry%NumberofPointCharges`

Type int

Description Number of point charges.

`Geometry%oinver`

Type float_array

Description Inverse of 'orient'.

Shape [3, 4]

`Geometry%orient`

Type float_array

Description Transform matrix from input to standard orientation.

Shape [3, 4]

`Geometry%PointCharges`

Type float_array

Description Positions and values of the point charges. For each point charge, the first 3 values are the coordinates (units: Bhor) and the 4th value is the value of the point charge (units: e).

Shape [4, NumberofPointCharges]

`Geometry%qeff`

Type float_array

Description Effective nuclear charge for the variaous atom types (i.e. nuclear charge minus the number of electrons in the frozen core).

Shape [nr of atomtypes]

`Geometry%qtch`

Type float_array

Description Nuclear charge for the variaous atom types.

Shape [nr of atomtypes]

`Geometry%symmetry tolerance`

Type float

Description ADF threshold for allowed deviation of input atomic coordinates from symmetry to be detected or verified.

`Geometry%xaxis`

Type float_array

Description Point in plane of local x-axis for each atom.

Shape [3, nr of atoms]

`Geometry%xyz`

Type float_array

Description Atom coordinates in internal order.

Unit bohr

Shape [3, nr of atoms]

`Geometry%xyz InputOrder`

Type float_array

Description Atom coordinates in input order.

Unit bohr

Shape [3, nr of atoms]

`Geometry%zaxis`

Type float_array

Description Orientation of local z-axis for each atom.

Shape [3, nr of atoms]

GeoOpt **Section content:** This seems to contain some data for excited state GOs?

Gradient **Section content:** The various contributions to the nuclear gradients. The atom-order of the arrays in this section is ‘internal order’. This is stored only if the input option ‘Debug Gradients’ is specified.

`Gradient%3D-RISM`

Type float_array

Description Nuclear gradients contribution from the 3D-RISM model. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%B-matrix`

Type float_array

Description B-matrix contribution to the nuclear gradients?. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%COSMO`

Type float_array

Description Nuclear gradients contribution from the COSMO solvation model. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Darwin`

Type float_array

Description Nuclear gradients contribution from Darwin (pauli?). Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%DIMQM`

Type float_array

Description Nuclear gradients contribution from the DIMQM model. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Dispersion`

Type float_array

Description xxx contribution to the nuclear gradients. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Electric Field`

Type float_array

Description Contribution to the nuclear gradients from the interaction from external electrostatic fields (homogeneous, point charges, …). Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Electrostatic Energy`

Type float_array

Description Electrostatic (pair?) contribution to the nuclear gradients. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Elstat Interaction`

Type float_array

Description Electrostatic interaction (non pair?) contribution to the nuclear gradients. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%HF Energy`

Type float_array

Description Nuclear gradients contribution from the Hartree-Fock exchange energy. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Kinetic Energy`

Type float_array

Description Kintetic energy contribution to the nuclear gradients. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Mass-Velocity`

Type float_array

Description Nuclear gradients contribution from the Mass-Velocity (pauli?). Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%P-matrix [W-energy]`

Type float_array

Description P-matrix [W-energy] contribution to the nuclear gradients? . Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%SM12`

Type float_array

Description Nuclear gradients contribution from the SM12 solvation method. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%Total Gradient`

Type float_array

Description Total nuclear gradients. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`Gradient%XC Energy`

Type float_array

Description Nuclear gradients contribution from the exchange correlation energy. Order: internal atom order.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

gtens Section content: Data for spin-orbit unrestricted calculation of ESR g-tensor

GW Section content: ?

`GW%freqGrid`

Type float_array

Description ?

`GW%G0W0_QP_hole_ener`

Type float_array

Description ?

`GW%G0W0_QP_hole_ener_dif`

Type float_array

Description ?

`GW%G0W0_QP_hole_ener_sp_A`

Type float_array

Description ?

`GW%G0W0_QP_hole_ener_sp_A_dif`

Type float_array

Description ?

`GW%G0W0_QP_hole_ener_sp_B`

Type float_array

Description ?

`GW%G0W0_QP_hole_ener_sp_B_dif`

Type float_array

Description ?

`GW%G0W0_QP_hole_energies`

Type float

Description ?

`GW%G0W0_QP_hole_energies_diff`

Type float

Description ?

`GW%G0W0_QP_part_ener`

Type float_array

Description ?

`GW%G0W0_QP_part_ener_dif`

Type float_array

Description ?

`GW%G0W0_QP_part_ener_sp_A`

Type float_array

Description ?

`GW%G0W0_QP_part_ener_sp_A_dif`

Type float_array

Description ?

`GW%G0W0_QP_part_ener_sp_B`

Type float_array

Description ?

`GW%G0W0_QP_part_ener_sp_B_dif`

Type float_array

Description ?

`GW%G0W0_QP_particle_energies`

Type float

Description ?

`GW%G0W0_QP_particle_energies_diff`

Type float

Description ?

`GW%GLRatio`

Type float_array

Description ?

`GW%nBas`

Type int_array

Description ?

`GW%nFit`

Type int

Description ?

`GW%nFreq`

Type int

Description ?

`GW%nFreqTotal`

Type int

Description ?

`GW%nInnerLoopIterations`

Type int

Description ?

`GW%nInnerLoopIterationsTotal`

Type int

Description ?

`GW%nIterations`

Type int

Description ?

`GW%normV`

Type float

Description ?

`GW%normW0`

Type float

Description ?

`GW%nremov`

Type int

Description ?

`GW%nStates`

Type int

Description ?

`GW%nTime`

Type int

Description ?

`GW%nTimeTotal`

Type int

Description ?

`GW%QPocc`

Type float_array

Description ?

`GW%QPocc_A`

Type float_array

Description ?

`GW%QPocc_B`

Type float_array

Description ?

`GW%QPun`

Type float_array

Description ?

`GW%QPun_A`

Type float_array

Description ?

`GW%QPun_B`

Type float_array

Description ?

`GW%SCGW_QP_hole_ener`

Type float_array

Description ?

`GW%SCGW_QP_hole_ener_dif`

Type float_array

Description ?

`GW%SCGW_QP_hole_energies`

Type float_array

Description ?

`GW%SCGW_QP_hole_energies_diff`

Type float_array

Description ?

`GW%SCGW_QP_part_ener`

Type float_array

Description ?

`GW%SCGW_QP_part_ener_dif`

Type float_array

Description ?

`GW%SCGW_QP_particle_energies`

Type float_array

Description ?

`GW%SCGW_QP_particle_energies_diff`

Type float_array

Description ?

`GW%spectral_*`

Type float_array

Description ?

`GW%SSOXcorrection`

Type float_array

Description ?

`GW%SSOXgreater`

Type float_array

Description ?

`GW%SSOXlesser`

Type float_array

Description ?

Hessian **Section content:** Data related to the analytical calculation of the hessian / frequencies.

`Hessian%Analytical Dipole Derivative`

Type float_array

Description Dipole derivatives computed anaytically by ADF. Atom ordering: internal order.

Shape [3, 3, Molecule%nAtoms]

`Hessian%Analytical Hessian`

Type float_array

Description Hessian computed anaytically by ADF. Atom ordering: internal order.

Unit hartree/bohr^2

Shape [3*Molecule%nAtoms, 3*Molecule%nAtoms]

`Hessian%Any User Selected Atoms?`

Type bool

Description Whether second derivatives are only computed for a user-defined sub-set of atoms.

`Hessian%Number of User Selected Atoms`

Type int

Description Number of user selected atoms for second derivatives.

`Hessian%User Selected Atoms`

Type int_array

Description Indices (in internal order) of the atoms for which second derivatives are computed.

Shape [Number of User Selected Atoms]

HFConfigADF **Section content:** Configuration for the RI procedure used for computing the Hartree-Fock exchange matrix.

HFdependency **Section content:** Data related to the linear-dependency within the 'old' RI Hartree-Fock scheme.

Irred dip vel matrix elem **Section content:** Dipole velocity matrix, irreducible matrix elements.

Irred magnetic matrix elem **Section content:** Magnetic moment matrix, irreducible matrix elements.

Irred matrix elements **Section content:** Section with irreducible matrix elements.

Irreducible CurrentMatrix **Section content:** Current response data.

Irreducible CurrentMatrix Factor **Section content:** Current response data.

KFDefinitions **Section content:** The definitions of the data on this file

`KFDefinitions%json`

Type string

Description The definitions of the data on this file in json.

LF_diag **Section content:** All LFDFT energies (including degeneracies) and eigenvectors.

`LF_diag%eigenvec_imag`

Type float_array

Description Imaginary part LF eigenvector.

Shape [nr_of_energies, nr_of_energies]

`LF_diag%eigenvec_real`

Type float_array

Description Real part LF eigenvector.

Shape [nr_of_energies, nr_of_energies]

`LF_diag%energies`

Type float_array

Description All LF energies (including degeneracies) wrt to AOC energy.

Unit hartree

Shape [nr_of_energies]

`LF_diag%nr_of_energies`

Type int

Description Total number of energies (including degeneracies).

`LF_diag%reference_energy`

Type float

Description Average of Configuration (AOC) energy, should be equal to Energy%Bond Energy.

Unit hartree

LF_energies **Section content:** LFDFT energies and properties.

`LF_energies%degeneracies`

Type int_array

Description Degenaracy of LF levels.

LF_energies%energies

Type float_array

Description Unique LF energies wrt to GS energy.

Unit hartree

Shape [nr_of_energies]

LF_energies%J2

Type float_array

Description Expectation value <J2>. If a level is degenerate <J2> is calculated as the maximum value for one of these levels.

Shape [nr_of_energies]

LF_energies%L2

Type float_array

Description Expectation value <L2>. If a level is degenerate <L2> is calculated as the maximum value for one of these levels.

Shape [nr_of_energies]

LF_energies%nr_of_energies

Type int

Description Number of unique LF energies. Degenerate levels count as 1.

LF_energies%oscillator strengths

Type float_array

Description Oscillator strengths for absorption of light on GS to an excited state with the same (atomic) electron configuration. This is zero in the electric dipole approximation.

Shape [nr_of_energies]

LF_energies%reference_energy

Type float

Description Ground state (GS) energy. Should be equal to LF_diag%reference_energy + LF_diag%energies(1).

Unit hartree

LF_energies%S2

Type float_array

Description Expectation value <S2>. If a level is degenerate <S2> is calculated as the maximum value for one of these levels.

Shape [nr_of_energies]

LF_energies%transition dipole moments

Type float_array

Description Transition dipole moment between GS and an excited state with the same (atomic) electron configuration. This is zero in the electric dipole approximation.

Shape [3, nr_of_energies]

LF_excitations **Section content:** LFDFT excitation energies and oscillator strengths between two atomic multiplet states which come from different electron configurations of the same molecule.

`LF_excitations%circular left`

Type float_array

Description Oscillator strengths (arbitrary units) circular left for absorption on GS 1 to excited state 2.

`LF_excitations%circular right`

Type float_array

Description Oscillator strengths (arbitrary units) circular right for absorption on GS 1 to excited state 2.

`LF_excitations%degeneracies`

Type int_array

Description Degenaracy of levels in the excited state 2.

`LF_excitations%degeneracy gs`

Type int

Description Degeneracy GS.

`LF_excitations%energies`

Type float_array

Description LF energies of the excited state wrt ground state 1.

Unit hartree

Shape [nr_of_energies]

`LF_excitations%excited state`

Type string

Description Excited state 2 electron configuration.

`LF_excitations%ground state`

Type string

Description Ground state (GS) 1 electron configuration.

`LF_excitations%J2 gs`

Type float

Description Expectation value <J2> GS 1. If the GS 1 is degenerate <J2> is calculated as the maximum value for one of these levels.

`LF_excitations%L2 gs`

Type float

Description Expectation value <L2> GS 1. If the GS 1 is degenerate <L2> is calculated as the maximum value for one of these levels.

`LF_excitations%nr_of_energies`

Type int

Description Number of unique energies in the excited state 2. Degenerate levels count as 1.

`LF_excitations%oscillator strengths`

Type float_array

Description Oscillator strengths (arbitrary units) for absorption on GS 1 to excited state 2.

Shape [nr_of_energies]

`LF_excitations%S2 gs`

Type float

Description Expectation value <S2> GS 1. If the GS 1 is degenerate <S2> is calculated as the maximum value for one of these levels.

`LF_excitations%transition dipole moments`

Type float_array

Description Transition dipole moment [1:3,:] (arbitrary units) between GS 1 and excited state 2. [4:5,:] used for circular right (x+iy)/sqrt(2) and circular left (x-iy)/sqrt(2) polarized light.

Shape [5, nr_of_energies]

LF_input **Section content:** ?

`LF_input%active_electrons`

Type int

Description Total number of electrons.

`LF_input%electron_configuration`

Type string

Description Electron configuration.

`LF_input%electrons_per_shell`

Type int_array

Description Number of electrons for each shell.

Shape [number_of_shells]

`LF_input%l_values`

Type int_array

Description Angular momentum quantum numbers (l-value) for each shell.

Shape [number_of_shells]

`LF_input%n_values`

Type int_array

Description The main quantum number n for each shell. For frozen core calculations the number n is reduced with the number of core levels with angular momentum l that are in the frozen core.

Shape [number_of_shells]

`LF_input%number_of_shells`

Type int

Description Number of shells

LibXCConfig **Section content:** ?

`LibXCConfig%densityThreshold`

Type float

Description ?

`LibXCConfig%flag`

Type bool_array

Description ?

Shape [42]

`LibXCConfig%fractionHF`

Type float

Description ?

`LibXCConfig%func`

Type int_array

Description ?

Shape [2]

`LibXCConfig%lRangeSep`

Type bool

Description ?

`LibXCConfig%nfunc`

Type int

Description ?

`LibXCConfig%RangeSepline`

Type string_fixed_length

Description ?

`LibXCConfig%xc_family`

Type int_array

Description ?

LinearScaling **Section content:** Technical parameters concerning the linear scaling.

LMO_P1 **Section content:** Data for localized orbitals.

LocOrb **Section content:** Data for localized orbitals.

locpert_data **Section content:** Data for localized orbitals.

Low Frequency Correction **Section content:** Configuration for the Head-Gordon Dampener-powered Free Rotor Interpolation.

`Low Frequency Correction%Alpha`

Type float

Description Exponent term for the Head-Gordon dampener.

`Low Frequency Correction%Frequency`

Type float

Description Frequency around which interpolation happens, in 1/cm.

`Low Frequency Correction%Moment of Inertia`

Type float

Description Used to make sure frequencies of less than ca. 1 1/cm don't overestimate entropy, in kg m^2.

LqbasxLqfitx_xyznuc **Section content:** Contains data for the grid generation for the auxiliary programs. Technical.

Magn multipole ints OCCOCC **Section content:** Data for XES.

Magn multipole ints OCCVIR **Section content:** Data for XAS.

Magnetic moment matrix **Section content:** Magnetic moment matrix.

MCD Elements **Section content:** Data related to MCD calculation.

`MCD Elements%B Fields`

Type float_array

Description ?

`MCD Elements%B Term`

Type float_array

Description ?

`MCD Elements%B Term SOS`

Type float_array

Description ?

`MCD Elements%C Term`

Type float_array

Description ?

`MCD Elements%C Term SOS`

Type float_array

Description ?

`MCD Elements%C Term SOS AnisoMCD*`

Type float_array

Description ?

`MCD Elements%C Term temperature coefficients`

Type float_array

Description ?

`MCD Elements%dipAJ`

Type float_array

Description ?

`MCD Elements%dipAJgr`

Type float_array

Description ?

`MCD Elements%dipAJgrSO`

Type float_array

Description ?

`MCD Elements%Magnetic Field*`

Type float

Description ?

`MCD Elements%nMCDTerm`

Type int

Description ?

`MCD Elements%Temperature*`

Type float

Description ?

`MCD Elements%Temperatures`

Type float_array

Description ?

METAGGA bonding energies **Section content:** Post-SCF bonding energies computed for a set of META-GGA XC functionals.

MetaGGAConfig **Section content:** Configuration for MetaGGA XC functionals.

Mobile Block Hessian **Section content:** Mobile Block Hessian.

`Mobile Block Hessian%Coordinates Internal`

Type float_array

Description ?

`Mobile Block Hessian%Free Atom Indexes Input`

Type int_array

Description ?

`Mobile Block Hessian%Frequencies in atomic units`

Type float_array

Description ?

`Mobile Block Hessian%Frequencies in wavenumbers`

Type float_array

Description ?

`Mobile Block Hessian%Input Cartesian Normal Modes`

Type float_array

Description ?

`Mobile Block Hessian%Input Indexes of Block #`

Type int_array

Description ?

`Mobile Block Hessian%Intensities in km/mol`

Type float_array

Description ?

`Mobile Block Hessian%MBH Curvatures`

Type float_array

Description ?

`Mobile Block Hessian%Number of Blocks`

Type int

Description Number of blocks.

`Mobile Block Hessian%Sizes of Blocks`

Type int_array

Description Sizes of the blocks.

Shape [Number of Blocks]

Molecule **Section content:** The input molecule of the calculation.

`Molecule%AtomicNumbers`

Type int_array

Description Atomic number 'Z' of the atoms in the system

Shape [nAtoms]

`Molecule%AtomMasses`

Type float_array

Description Masses of the atoms

Unit a.u.

Values range [0, '\infinity']

Shape [nAtoms]

`Molecule%AtomSymbols`

Type string

Description The atom's symbols (e.g. 'C' for carbon)

Shape [nAtoms]

`Molecule%bondOrders`

Type float_array

Description The bond orders for the bonds in the system. The indices of the two atoms participating in the bond are defined in the arrays 'fromAtoms' and 'toAtoms'. e.g. bondOrders[1]=2, fromAtoms[1]=4 and toAtoms[1]=7 means that there is a double bond between atom number 4 and atom number 7

`Molecule%Charge`

Type float

Description Net charge of the system

Unit e

`Molecule%Coords`

Type float_array

Description Coordinates of the nuclei (x,y,z)

Unit bohr

Shape [3, nAtoms]

`Molecule%eeAttachTo`

Type int_array

Description A multipole may be attached to an atom. This influences the energy gradient.

`Molecule%eeChargeWidth`

Type float

Description If charge broadening was used for external charges, this represents the width of the charge distribution.

`Molecule%eeEField`

Type float_array

Description The external homogeneous electric field.

Unit hartree/(e*bohr)

Shape [3]

`Molecule%eeLatticeVectors`

Type float_array

Description The lattice vectors used for the external point- or multipole- charges.

Unit bohr

Shape [3, eeNLatticeVectors]

`Molecule%eeMulti`

Type float_array

Description The values of the external point- or multipole- charges.

Unit a.u.

Shape [eeNZlm, eeNMulti]

`Molecule%eeNLatticeVectors`

Type int

Description The number of lattice vectors for the external point- or multipole- charges.

`Molecule%eeNMulti`

Type int

Description The number of external point- or multipole- charges.

`Molecule%eeNZlm`

Type int

Description When external point- or multipole- charges are used, this represents the number of spherical harmonic components. E.g. if only point charges were used, eeNZlm=1 (s-component only). If point charges and dipole moments were used, eeNZlm=4 (s, px, py and pz).

`Molecule%eeUseChargeBroadening`

Type bool

Description Whether or not the external charges are point-like or broadened.

`Molecule%eeXYZ`

Type float_array

Description The position of the external point- or multipole- charges.

Unit bohr

Shape [3, eeNMulti]

`Molecule%EngineAtomicInfo`

Type string_fixed_length

Description Atom-wise info possibly used by the engine.

`Molecule%fromAtoms`

Type int_array

Description Index of the first atom in a bond. See the bondOrders array

`Molecule%latticeDisplacements`

Type int_array

Description The integer lattice translations for the bonds defined in the variables bondOrders, fromAtoms and toAtoms.

`Molecule%LatticeVectors`

Type float_array

Description Lattice vectors

Unit bohr

Shape [3, nLatticeVectors]

`Molecule%nAtoms`

Type int

Description The number of atoms in the system

`Molecule%nAtomsTypes`

Type int

Description The number different of atoms types

`Molecule%nLatticeVectors`

Type int

Description Number of lattice vectors (i.e. number of periodic boundary conditions)

Possible values [0, 1, 2, 3]

`Molecule%toAtoms`

Type int_array

Description Index of the second atom in a bond. See the bondOrders array

MP2 energies Section content: ?

`MP2 energies%Contribution to DH energy`

Type float

Description ?

`MP2 energies%LT-MP2 energy`

Type float

Description ?

`MP2 energies%os LT-MP2 energy`

Type float

Description ?

`MP2 energies%os RI-MP2 energy`

Type float

Description ?

`MP2 energies%RI-MP2 energy`

Type float

Description ?

`MP2 energies%ss LT-MP2 energy`

Type float

Description ?

`MP2 energies%ss RI-MP2 energy`

Type float

Description ?

MP2configADF Section content: ?

`MP2configADF%Dependency`

Type bool

Description ?

`MP2configADF%EmpiricalScaling`

Type string

Description ?

`MP2configADF%LT`

Type bool

Description ?

`MP2configADF%Memory`

Type bool

Description ?

`MP2configADF%nTime`

Type int

Description ?

`MP2configADF%RI`

Type bool

Description ?

`MP2configADF%UseScaledZORA`

Type bool

Description ?

Multipole matrix elements **Section content:** Section with multipole matrix elements.

NBOs **Section content:** ?

`NBOs%Label_#{LocalizedOrbitalNumber}`

Type string

Description Label localized orbital.

`NBOs%Label_A_#{LocalizedOrbitalNumber}`

Type string

Description Label localized orbital.

`NBOs%Label_B_#{LocalizedOrbitalNumber}`

Type string

Description Label localized orbital, for spin B.

`NBOs%nocc_A`

Type int

Description Number of occupied orbitals.

`NBOs%nocc_B`

Type int

Description Number of occupied orbitals, for spin B.

`NBOs%Orbitals_A_#{LocalizedOrbitalNumber}`

Type float_array

Description Localized orbital expressed in Cartesian basis functions (BAS).

Shape [Basis%naos]

`NBOs%Orbitals_B_#{LocalizedOrbitalNumber}`

Type float_array

Description Localized orbital expressed in Cartesian basis functions (BAS), for spin B.

Shape [Basis%naos]

NLMOs **Section content:** ?

`NLMOs%Label_#{LocalizedOrbitalNumber}`

Type string

Description Label localized orbital.

`NLMOs%Label_A_#{LocalizedOrbitalNumber}`

Type string

Description Label localized orbital.

`NLMOs%Label_B_#{LocalizedOrbitalNumber}`

Type string

Description Label localized orbital, for spin B.

`NLMOs%nocc_A`

Type int

Description Number of occupied orbitals.

`NLMOs%nocc_B`

Type int

Description Number of occupied orbitals, for spin B.

`NLMOs%Orbitals_A_#{LocalizedOrbitalNumber}`

Type float_array

Description Localized orbital expressed in Cartesian basis functions (BAS).

Shape [Basis%naos]

`NLMOs%Orbitals_B_#{LocalizedOrbitalNumber}`

Type float_array

Description Localized orbital expressed in Cartesian basis functions (BAS), for spin B.

Shape [Basis%naos]

NOCV **Section content:** Results from the NOCV (Natural Orbitals for Chemical Valence) procedure.

`NOCV%Label_A_#`

Type string_fixed_length

Description Label of the NOCV (spin-alpha in case of spin-restricted calcualtion, or the spin-restricted in case of spin-restricted calcualtion).

`NOCV%Label_B_#`

Type string_fixed_length

Description Label of the spin-beta NOCV.

`NOCV%NOCV_eigenvalues_alpha`

Type float_array

Description NOCV eigenvalues for alpha spin (for spin-unrestricted calculations).

Unit hartree

`NOCV%NOCV_eigenvalues_beta`

Type float_array

Description NOCV eigenvalues for beta spin (for spin-unrestricted calculations).

Unit hartree

`NOCV%NOCV_eigenvalues_restricted`

Type float_array

Description NOCV eigenvalues (in case of spin-restricted calculation).

Unit hartree

`NOCV%NOCV_oi_alpha`

Type float_array

Description Orbital interaction contributions from each NOCV for alpha spin (for spin-unrestricted calculations).

Unit hartree

`NOCV%NOCV_oi_beta`

Type float_array

Description Orbital interaction contributions from each NOCV for beta spin (for spin-unrestricted calculations).

Unit hartree

`NOCV%NOCV_oi_restricted`

Type float_array

Description Orbital interaction contributions from each NOCV (in case of spin-restricted calculation).

Unit hartree

`NOCV%NOCV_T_ALPHA`

Type float_array

Description ?

`NOCV%NOCV_T_BETA`

Type float_array

Description ?

`NOCV%NOCV_V_ALPHA`

Type float_array

Description ?

NOCV%NOCV_V_BETA

Type float_array

Description ?

NOCV%Orbitals_A_#

Type float_array

Description Orbital coefficients the NOCV (spin-alpha in case of spin-restricted calcualtion, or the spin-restricted in case of spin-restricted calcualtion).

Shape [Basis%naos]

NOCV%Orbitals_B_#

Type float_array

Description Orbital coefficients the spin-beta NOCV.

Shape [Basis%naos]

Num Int Params **Section content:** Technical parameters concerning the numerical integration procedure of ADF.

OrbitalInfo **Section content:** Section containing the orbital information related to a specific symmetry group

OrbitalInfo%AOBas-I_A

Type float_array

Description Imaginary alpha and beta part spinor expressed in Cartesian basis functions (BAS).

Shape [Basis%naos, 2, nmo_A, nrdim]

OrbitalInfo%AOBas-R_A

Type float_array

Description Real alpha and beta part spinor expressed in Cartesian basis functions (BAS).

Shape [Basis%naos, 2, nmo_A, nrdim]

OrbitalInfo%Bas-I_A *

Type float_array

Description Spin-orbit data.

OrbitalInfo%Bas-R_A *

Type float_array

Description Spin-orbit data.

OrbitalInfo%BasPhi-*

Type float_array

Description Spin-orbit data related to large component spinor.

OrbitalInfo%BasXPhi-*

Type float_array

Description Spin-orbit data related to pseudo-large component spinor.

OrbitalInfo%cmatab_A

Type float_array

Description Data relevant for unrestricted fragments.

Shape [nmo_A, nmo_A]

`OrbitalInfo%cmatab_B`

Type float_array

Description Data relevant for unrestricted fragments.

Shape [nmo_A, nmo_A]

`OrbitalInfo%Eig-CoreSFO_A`

Type float_array

Description MOs expressed in core orthogonalization functions and SFOs, for spin-A MOs.

Shape [nt, nmo_A]

`OrbitalInfo%Eig-CoreSFO_B`

Type float_array

Description Same as ‘Eig-CoreSFO_A’, but for spin B.

Shape [nt, nmo_A]

`OrbitalInfo%Eigen-Bas_A`

Type float_array

Description MO expansion coefficients in the BAS representation for all nmo_A orbitals (in unrestricted case for spin A). The coefficients run over all BAS functions indicated by npart.

Shape [nbas, nmo_A]

`OrbitalInfo%Eigen-Bas_B`

Type float_array

Description Unrestricted case: same as ‘Eigen-Bas_A’, but for spin B.

Shape [nbas, nmo_A]

`OrbitalInfo%Eigen-BasPhi_A`

Type float_array

Description Large component expressed in Cartesian basis functions (BAS).

Shape [nbas, nmo_A]

`OrbitalInfo%Eigen-BasPhi_B`

Type float_array

Description Large component expressed in Cartesian basis functions (BAS) for spin B.

Shape [nbas, nmo_A]

`OrbitalInfo%Eigen-BasXPhi_A`

Type float_array

Description Pseudo-large component expressed in Cartesian basis functions (BAS).

Shape [nbas, nmo_A]

`OrbitalInfo%Eigen-BasXPhi_B`

Type float_array

Description Pseudo-large component expressed in Cartesian basis functions (BAS) for spin B.

Shape [nbas, nmo_A]

`OrbitalInfo%eps_A`

Type float_array

Description The orbital energies for the nmo_A orbitals (in unrestricted case for spin A). When they result from a ZORA calculations, the non-scaled values are stored on file, see qscal how to scale.

Shape [nmo_A]

`OrbitalInfo%eps_B`

Type float_array

Description Unrestricted case: same as ‘eps_A’, but for spin B.

`OrbitalInfo%escale_A`

Type float_array

Description ZORA only. Scaled orbital eigenvalues.

Shape [nmo_A]

`OrbitalInfo%escale_A *`

Type float_array

Description Spin-orbit data.

`OrbitalInfo%escale_B`

Type float_array

Description ZORA only. Scaled orbital eigenvalues for spin B

Shape [nmo_A]

`OrbitalInfo%FragBas-I_A`

Type float_array

Description Imaginary part spinor expressed in SFOs. Note, only used in case fragment is the full molecule.

Shape [SFOs%number, 2, nmo_A, nrdim]

`OrbitalInfo%FragBas-R_A`

Type float_array

Description Real part of spinor expressed in SFOs. Note, only used in case fragment is the full molecule.

Shape [SFOs%number, 2, nmo_A, nrdim]

`OrbitalInfo%froc`

Type float_array

Description The occupation numbers of the MOs in the irrep.

Shape [nmo_A]

`OrbitalInfo%froc_A`

Type float_array

Description The occupation numbers of the MOs in the irrep (in unrestricted case for spin A).

Shape [nmo_A]

`OrbitalInfo%froc_B`

Type float_array

Description Unrestricted case: the occupation numbers of the MOs in the irrep, for spin B

Shape [nmo_A]

`OrbitalInfo%frocf`

Type float_array

Description The occupation numbers of the Lowdins and SFOs in this irrep.

Shape [nmo_A]

`OrbitalInfo%frocf_A`

Type float_array

Description The occupation numbers of the Lowdins and SFOs in this irrep for spin A.

Shape [nmo_A]

`OrbitalInfo%frocf_B`

Type float_array

Description The occupation numbers of the Lowdins and SFOs in this irrep for spin B.

Shape [nmo_A]

`OrbitalInfo%frocor`

Type float_array

Description Non-empty Lowdin and SFO occupation numbers.

`OrbitalInfo%frocor_A`

Type float_array

Description Non-empty Lowdin and SFO occupation numbers for spin A.

`OrbitalInfo%frocor_B`

Type float_array

Description Non-empty Lowdin and SFO occupation numbers for spin B.

`OrbitalInfo%frocr_A *`

Type float_array

Description Spin-orbit data.

`OrbitalInfo%large QP-Eigen-MO_A`

Type float_array

Description Large coefficients quasi particles expressed in MOs.

`OrbitalInfo%Low-Bas`

Type float_array

Description The Lowdin orbitals expressed in the BAS representation: the matrix to transform the MOs from Lowdin representation (Lowdin = orthonormalized SFOs) to the BAS representation.

Shape [nmo_A, nbas]

`OrbitalInfo%Low-Bas_B`

Type float_array

Description Same as Low-Bas, but for spn B.

Shape [nmo_A, nbas]

`OrbitalInfo%mo_index large QP-Eigen-MO_A`

Type int_array

Description Large coefficients index array quasi particles expressed in which MO.

`OrbitalInfo%nbas`

Type int

Description Number of primitive STOs in this symmetry group.

`OrbitalInfo%ncbas`

Type int

Description Number of core orthogonalization functions in this symmetry group.

`OrbitalInfo%nmo_A`

Type int

Description Number of alpha molecular orbitals in this symmetry group (in unrestricted case for spin A).

`OrbitalInfo%nmo_B`

Type int

Description Unrestricted case: number of beta molecular orbitals in this symmetry group. Should be equal to nmo_A.

`OrbitalInfo%NOCV-Eigen-MO_A`

Type float_array

Description NOCV orbitals expressed in MOs.

Shape [nmo_A, nmo_A]

`OrbitalInfo%NOCV-Eigen-MO_B`

Type float_array

Description Same as NOCV-Eigen-MO_A, but for spin B.

Shape [nmo_A, nmo_A]

`OrbitalInfo%npart`

Type int_array

Description A list of indices of the BAS functions that are used in this irrep.

Shape [nbas]

OrbitalInfo%nrdim

Type int

Description Dimension (number of subspecies) of this double-group irrep.

OrbitalInfo%nt

Type int

Description Number of core orthogonalization functions plus number of SFOs in this symmetry group.

OrbitalInfo%number large QP-Eigen-MO_A

Type int_array

Description Number of large coefficients quasi particles expressed in MOs.

OrbitalInfo%Orth-Bas

Type float_array

Description The (non-empty) orthogonalized fragment orbitals expressed in the BAS representation.

Shape [nbas, :]

OrbitalInfo%Orth-Bas_A

Type float_array

Description The (non-empty) orthogonalized fragment orbitals expressed in the BAS representation for spin A.

Shape [nbas, :]

OrbitalInfo%Orth-Bas_B

Type float_array

Description The (non-empty) orthogonalized fragment orbitals expressed in the BAS representation for spin B.

Shape [nbas, :]

OrbitalInfo%QP-Eigen-Bas_A

Type float_array

Description Quasi particles expressed in Cartesian basis functions (BAS).

Shape [nbas, nmo_A]

OrbitalInfo%QP-Eigen-MO_A

Type float_array

Description Quasi particles expressed in MOs.

Shape [nmo_A, nmo_A]

OrbitalInfo%QP-eps_A

Type float_array

Description Quasi particles orbital energies.

Shape [nmo_A]

OrbitalInfo%QP-eps_B

Type float_array

Description Same as QP-eps_A, but for spin B.

Shape [nmo_A]

OrbitalInfo%qscal_A

Type float_array

Description Used only for ZORA. Scaled eps is (eps/(1+qscal)).

Shape [nmo_A]

OrbitalInfo%qscal_B

Type float_array

Description Unrestricted case: same as 'qscal_A', but for spin B.

Shape [nmo_A]

OrbitalInfo%qscalr_A *

Type float_array

Description Spin-orbit data.

OrbitalInfo%S-CoreSFO

Type float_array

Description Overlap matrix of both core orthogonalization functions and SFOs. Size [nt*(nt+1)/2].

OrbitalInfo%S-CoreSFO_B

Type float_array

Description Same as 'S-CoreSFO_A', but for spin B.

OrbitalInfo%SFO

Type float_array

Description Coeffs for the symmetrized fragment orbital expressed in Cartesian basis functions (BAS) in this symmetry group. Note these are core orthogonalized SFOs (CSFOs).

Shape [nmo_A, nbas]

OrbitalInfo%SFO_B

Type float_array

Description Same as SFO, but for spin B.

Shape [nmo_A, nbas]

OrbitalInfo%smx

Type float_array

Description Overlap matrix of both core orthogonalization functions and core orthogonalized SFOs (CSFOs). Size [nt*(nt+1)/2].

OrbitalInfo%smx_B

Type float_array

Description Same as smx, but for spin B.

OrbitalsCoulombInteraction **Section content:** ?

OrbitalsCoulombInteraction%IntEnergy_#

Type float

Description ?

OrbitalsCoulombInteraction%nPairs

Type int

Description ?

OrbitalsCoulombInteraction%OrbitalIndices_#

Type int_array

Description ?

Ort C energies **Section content:** Correlation energies from orthonormalized density.

Ort X energies **Section content:** Exchange energies from orthonormalized density.

Ort XC energies **Section content:** XC energies from orthonormalized density.

Point_Charges **Section content:** Data related to point charges.

POLTDDFT **Section content:** Data related to the PolTD-DFT procedure.

Properties **Section content:** A collection of properties computed by ADF.

Properties%Aromaticity Ring #

Type float_array

Description The first value in the array is the ring index (Iring). The second value is the multi center index (MCI). (Aromaticity indices with QTAIM results)

Shape [2]

Properties%AtomCharge CM5

Type float_array

Description Net atomic charges atomic charges from the Charge Model 5 (CM5) method. (e.g., the charges for a water molecule might be [-0.6, 0.3, 0.3]). Atom order: internal order.

Unit e

Shape [Molecule%nAtoms]

Properties%AtomCharge Mulliken

Type float_array

Description ?

Shape [Molecule%nAtoms]

Properties%AtomCharge_initial Voronoi

Type float_array

Description Voronoi atomic charges for the Initial (sum-of-fragments) density.

Shape [Molecule%nAtoms]

Properties%AtomCharge_SCF Voronoi

Type float_array

Description Voronoi atomic charges for the SCF density.

Shape [Molecule%nAtoms]

`Properties%AtomIndex Ring #`

Type int_array

Description Indices of atoms in a ring for which the aromaticity indices are computed. The aromaticity indices are saved in the variable 'Aromaticity Ring #' (Aromaticity indices with QTAIM results)

`Properties%AtomSpinDen Mulliken`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Bader atomic charges`

Type float_array

Description Atomic charges computed using the QTAIM approach, atoms in the internal ADF order.

`Properties%Bader atomic dipole moment`

Type float_array

Description Atomic dipole moments computed using the QTAIM approach, atoms in the internal ADF order.

`Properties%Bader atomic quadrupole moment`

Type float_array

Description ?

`Properties%Bader atomic spin densities`

Type float_array

Description ?

`Properties%Bader ELF`

Type float_array

Description ?

`Properties%Bader Esc`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Bader Eslc`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Bader EVF`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Bader Laplacian`

Type float_array

Description ?

`Properties%Bader MO contrib`

Type float_array

Description ?

`Properties%Bader Tc`

Type float_array

Description ?

`Properties%Bader Ts`

Type float_array

Description ?

`Properties%Bader Tsl`

Type float_array

Description ?

`Properties%BP atoms`

Type int_array

Description ?

`Properties%BP number of`

Type int

Description ?

`Properties%BP shift`

Type float_array

Description ?

`Properties%BP step number`

Type int_array

Description ?

`Properties%BPs and their properties`

Type float_array

Description ?

`Properties%Cond Linear Response`

Type float_array

Description ?

`Properties%CP code number for (Rank,Signatu`

Type float_array

Description ?

`Properties%CP coordinates`

Type float_array

Description ?

`Properties%CP density at`

Type float_array

Description ?

`Properties%CP density gradient at`

Type float_array

Description ?

`Properties%CP density Hessian at`

Type float_array

Description ?

`Properties%CP number of`

Type int

Description ?

`Properties%Dipole`

Type float_array

Description Dipole moment.

Shape [3]

`Properties%EFG asym. par. eta InputOrder`

Type float_array

Description EFG asymmetry parameter eta, which is the difference between the lowest 2 principal values of the EFG divided by the largest principal value of the EFG, atoms in input order.

Shape [Molecule%nAtoms]

`Properties%EFG NQCC (MHz) InputOrder`

Type float_array

Description Nuclear Quadrupole Coupling Constant (NQCC) (in MHz) is the largest value of the principal values of the EFG (in a.u.) times 234.9647 times the nuclear quadrupole moment (in barn units), atoms in input order.

Shape [Molecule%nAtoms]

`Properties%EFG tensor (au) InputOrder`

Type float_array

Description Electric field gradient (EFG) at nuclei (a.u.), atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%EFG tensor (MHz) InputOrder`

Type float_array

Description Electric field gradient (EFG) at nuclei (MHz), includes factor Q/(2I(2I-1)), atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%EFG Vzz (au) InputOrder`

Type float_array

Description The largest value of the principal values of the EFG (in a.u.), atoms in input order.

Shape [Molecule%nAtoms]

`Properties%Electron Density at Nuclei`

Type float_array

Description Average electron density at a small sphere around the center of a nucleus.

Shape [Molecule%nAtoms]

`Properties%Electronegativity`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Electronegativity(omega)`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Electrostatic Pot.at Nuclei`

Type float_array

Description Electrostatic Potential at Nuclei. The contribution from the nucleus itself is excluded.

Shape [Molecule%nAtoms]

`Properties%ESR A-iso (au) InputOrder`

Type float_array

Description Isotropic value A-tensor (a.u.), not multiplied by g_n/S, atoms in input order.

Shape [Molecule%nAtoms]

`Properties%ESR A-iso (MHz) InputOrder`

Type float_array

Description Isotropic value A-tensor (MHz), includes factor g_n/S, atoms in input order.

Shape [Molecule%nAtoms]

`Properties%ESR A-tens (au) InputOrder`

Type float_array

Description ESR A-tensor (a.u.), not multiplied by g_n/S, atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%ESR A-tens (MHz) InputOrder`

Type float_array

Description ESR A-tensor (MHz), includes factor g_n/S, atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%ESR g-iso`

Type float

Description Isotropic value ESR g-tensor.

`Properties%ESR g-tensor`

Type float_array

Description ESR g-tensor.

Shape [3, 3]

`Properties%Excited State Dipole`

Type float_array

Description Excited state dipole moment.

Shape [3]

`Properties%FermiLevel`

Type float_array

Description Fermi energy per spin, typically exactly halfway between HOMO and LUMO energy per spin.

Shape [General%nspin]

`Properties%Fragment Voronoi chrg per irrep`

Type bool

Description Whether Voronoi (and other) atomic charges per irreducible representation are written to file.

`Properties%FragmentCharge Hirshfeld`

Type float_array

Description Hirshfeld fragment charges.

Shape [Geometry%nfragm]

`Properties%Fukui Fminus`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%Fukui Fplus`

Type float_array

Description ?

Shape [Molecule%nAtoms]

`Properties%GophinatanJug BO between frag`

Type float_array

Description Matrix containing the Gopinathan-Jug bond order between fragments. Diagonal elements of the BO matrix=?. Atoms order: interal order.

Shape [Geometry%nr of fragments, Geometry%nr of fragments]

`Properties%Hirshfeld Atomic Charge`

Type float_array

Description ?

`Properties%Hirshfeld Effective Volume`

Type float_array

Description ?

`Properties%Hirshfeld Free Volume`

Type float_array

Description ?

`Properties%Hirshfeld Fuzzy Bond Orders`

Type float_array

Description Matrix containing the Hirshfeld Fuzzy Bond Orders.

Shape [Geometry%nr of fragments, Geometry%nr of fragments]

`Properties%Hirshfeld Fuzzy Valences`

Type float_array

Description Hirshfeld Fuzzy Valences (internal order).

Shape [Geometry%nr of fragments]

`Properties%HOMO`

Type float

Description Highest occupied molecular orbital (HOMO) energy.

`Properties%IQA atom-atom total`

Type float_array

Description Total interaction energy of atoms I and J

Shape [:]

`Properties%IQA Coulomb total`

Type float_array

Description ?

`Properties%IQA e-e Coulomb`

Type float_array

Description Coulomb interaction energy of electron densities on atoms I and J

Shape [:]

`Properties%IQA e-e exchange`

Type float_array

Description Exchange interaction energy of electron densities on atoms I and J

Shape [:]

`Properties%IQA e-e total`

Type float_array

Description Total interaction energy of electron densities on atoms I and J

Shape [:]

`Properties%IQA electron-nucleus`

Type float_array

Description Interaction energy of the electron density of atom I with the nucleus of atom J

Shape [Molecule%nAtoms, Molecule%nAtoms]

`Properties%IQA kinetic energy`

Type float_array

Description Kinetic energy of all electrons of the atom

Shape [Molecule%nAtoms]

`Properties%IQA pairs disp`

Type float_array

Description ?

`Properties%Koopmans DD`

Type float_array

Description Fukui Fplus - Fukui Fminus?

Shape [Molecule%nAtoms]

`Properties%LI-DI indices`

Type float_array

Description The matrix of localization/delocalization indices (Aromaticity indices with QTAIM results). Note: atom order might be 'internal'?

Shape [Molecule%nAtoms, Molecule%nAtoms]

`Properties%LI-DI packed`

Type float_array

Description Localization and delocalization indices (matrix elements). Order?

`Properties%LUMO`

Type float

Description Lowest unoccupied molecular orbital (LUMO) energy.

`Properties%Mayer BO between frag`

Type float_array

Description Matrix containing the Mayer bond order between fragments. Diagonal elements of the BO matrix=?. Atoms order: interal order.

Shape [Geometry%nr of fragments, Geometry%nr of fragments]

`Properties%MayerBondOrders_A`

Type float_array

Description Mayer bond orders matrix. In case of spin-restricted calculation, this contains the full bond orders. In case of spin-unrestricted, this contains the bond orders for the alpha spinns. Possibly in internal order? check.

Shape [Molecule%nAtoms, Molecule%nAtoms]

`Properties%MayerBondOrders_B`

Type float_array

Description Spin-beta Mayer bond orders matrix (only for spin-unrestricted calculations). Possibly in internal order? check.

Shape [Molecule%nAtoms, Molecule%nAtoms]

`Properties%MDC-d charges`

Type float_array

Description MDC-d charges reconstruct monopoles and dipoles.

Shape [Molecule%nAtoms]

`Properties%MDC-d charges_A`

Type float_array

Description Spin alpha MDC-d charges.

Shape [Molecule%nAtoms]

`Properties%MDC-d charges_B`

Type float_array

Description Spin beta MDC-d charges.

Shape [Molecule%nAtoms]

`Properties%MDC-m charges`

Type float_array

Description Multipole derived atomic charges (MDC) reconstruct monopoles.

Shape [Molecule%nAtoms]

`Properties%MDC-m charges_A`

Type float_array

Description Spin alpha MDC-m charges.

Shape [Molecule%nAtoms]

`Properties%MDC-m charges_B`

Type float_array

Description Spin beta MDC-m charges.

Shape [Molecule%nAtoms]

`Properties%MDC-q charges`

Type float_array

Description MDC-d charges reconstruct monopoles, dipoles and quadrupoles.

Shape [Molecule%nAtoms]

`Properties%MDC-q charges_A`

Type float_array

Description Spin alpha MDC-q charges.

Shape [Molecule%nAtoms]

`Properties%MDC-q charges_B`

Type float_array

Description Spin beta MDC-q charges.

Shape [Molecule%nAtoms]

`Properties%Nalewajski1 BO between frag`

Type float_array

Description Matrix containing the Nalewajski-Mrozek-1 bond order between fragments. Diagonal elements of the BO matrix=?. Atoms order: interal order.

Shape [Geometry%nr of fragments, Geometry%nr of fragments]

`Properties%Nalewajski2 BO between frag`

Type float_array

Description Matrix containing the Nalewajski-Mrozek-1 bond order between fragments. Diagonal elements of the BO matrix=?. Atoms order: interal order.

Shape [Geometry%nr of fragments, Geometry%nr of fragments]

`Properties%Nalewajski3 BO between frag`

Type float_array

Description Matrix containing the Nalewajski-Mrozek-3 bond order between fragments. Diagonal elements of the BO matrix=?. Atoms order: interal order.

Shape [Geometry%nr of fragments, Geometry%nr of fragments]

`Properties%NBO natural charges`

Type float_array

Description NBO natural charges.

`Properties%nEntries`

Type int

Description Number of properties.

`Properties%NFOD`

Type float

Description NFOD decsriptor: Integrated fractional orbital density.

`Properties%NMR Coupling J const InputOrder`

Type float_array

Description NMR Nuclear Spin-spin Coupling Constants, atoms in input order.

Shape [Molecule%nAtoms, Molecule%nAtoms]

`Properties%NMR Coupling J tens InputOrder`

Type float_array

Description NMR Nuclear Spin-spin Coupling tensors, atoms in input order.

Shape [3, 3, Molecule%nAtoms, Molecule%nAtoms]

`Properties%NMR Coupling K const InputOrder`

Type float_array

Description NMR reduced Nuclear Spin-spin Coupling Constants, atoms in input order.

Shape [Molecule%nAtoms, Molecule%nAtoms]

`Properties%NMR Coupling K tens InputOrder`

Type float_array

Description NMR reduced Nuclear Spin-spin Coupling tensors, atoms in input order.

Shape [3, 3, Molecule%nAtoms, Molecule%nAtoms]

`Properties%NMR Shielding Tensor InputOrder`

Type float_array

Description NMR chemical shielding tensors, atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%NMR Shieldings InputOrder`

Type float_array

Description NMR chemical shieldings, atoms in input order.

Shape [Molecule%nAtoms]

`Properties%OverlapPop`

Type float_array

Description ?

Shape [Geometry%nr of fragments, Geometry%nr of fragments, 2]

`Properties%pNMR Shielding Tensor InputOrder`

Type float_array

Description pNMR chemical shielding tensors, paramagnetic molecules, atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%pNMR Shieldings InputOrder`

Type float_array

Description pNMR chemical shieldings, paramagnetic molecules, atoms in input order.

Shape [Molecule%nAtoms]

`Properties%pNMR sigma_p InputOrder`

Type float_array

Description Temperature dependent part pNMR chemical shieldings, paramagnetic molecules, atoms in input order.

Shape [Molecule%nAtoms]

`Properties%pNMR sigma_p Tensor InputOrder`

Type float_array

Description Temperature dependent part pNMR chemical shielding tensors, paramagnetic molecules, atoms in input order.

Shape [3, 3, Molecule%nAtoms]

`Properties%pNMR Temperature`

Type float

Description Temperature used in the calculation of the temperature dependent part pNMR chemical shieldings, paramagnetic molecules.

`Properties%Polarizability`

Type float_array

Description Polarizability at first frequency PolarizabilityAtFrequency

Unit a.u.

Shape [3, 3]

`Properties%Polarizability #`

Type float_array

Description Polarizability at frequency PolarizabilityAtFrequency #.

Unit a.u.

Shape [3, 3]

`Properties%PolarizabilityAtFrequency`

Type float

Description First frequency at which Polarizability is calculated.

Unit a.u.

`Properties%PolarizabilityAtFrequency #`

Type float

Description Frequency at which Polarizability # is calculated.

Unit a.u.

`Properties%PolarizabilityImag`

Type float_array

Description Imaginary part polarizability at first frequency PolarizabilityAtFrequency.

Unit a.u.

Shape [3, 3]

`Properties%Quadrupole`

Type float_array

Description Quadrupole moment (Buckingham convention).

Shape [6]

`Properties%RhoDipole`

Type float_array

Description Dipole moment from electron density only.

Shape [3]

`Properties%RISM box`

Type float_array

Description ?

`Properties%RISM buv`

Type float_array

Description ?

`Properties%RISM cuv`

Type float_array

Description ?

`Properties%RISM epsu`

Type float_array

Description ?

`Properties%RISM ExChemPotential`

Type float

Description ?

`Properties%RISM grid dims`

Type int_array

Description ?

`Properties%RISM guv`

Type float_array

Description ?

`Properties%RISM guv integral #`

Type float_array

Description ?

`Properties%RISM guv rdf #`

Type float_array

Description ?

`Properties%RISM huv`

Type float_array

Description ?

`Properties%RISM Ndata`

Type int_array

Description ?

`Properties%RISM nga`

Type int

Description ?

`Properties%RISM ngk`

Type int

Description ?

`Properties%RISM num plane waves`

Type int

Description ?

`Properties%RISM num solute sites`

Type int

Description ?

`Properties%RISM num solvent sites`

Type int

Description ?

`Properties%RISM PartMolVol[A^3]`

Type float

Description ?

`Properties%RISM qu`

Type float_array

Description ?

`Properties%RISM radial grid #`

Type float_array

Description ?

`Properties%RISM ratu`

Type float_array

Description ?

`Properties%RISM sigu`

Type float_array

Description ?

`Properties%RISM SiteDegeneracy`

Type int_array

Description ?

`Properties%RISM SiteExChemPotential`

Type float_array

Description ?

`Properties%RISM uuv`

Type float_array

Description ?

`Properties%S2calc`

Type float

Description Electron spin S**2 expectation value.

`Properties%S2pure`

Type float

Description Exact electron spin S**2 expectation value s(s+1), where s is a half-integer value.

`Properties%Source Function (cp,nuc)`

Type float_array

Description ?

`Properties%Subtype(#)`

Type string_fixed_length

Description Extra detail about the property. For a charge propety this could be Mulliken.

`Properties%Type(#)`

Type string

Description Type of the property, like energy, gradients, charges, etc.

`Properties%Value(#)`

Type float_array

Description The value(s) of the propety.

`Properties%VDDBondInd`

Type int_array

Description Indicies of atoms for VVD bond (internal atom order).

Shape [VDDBonds, 2]

`Properties%VDDBonds`

Type int

Description Number of VDD bonds?

`Properties%VDDBondVal`

Type float_array

Description Values of the VVD bonds.

Shape [VDDBonds]

`Properties%Voronoi chrg per irrep`

Type float_array

Description Voronoi (and other) atomic charges per irreducible representation.

Shape [Molecule%nAtoms, Symmetry%nsym, 8]

QMFQ **Section content:** Why is this in the ams.rkf file and not in the adf.rkf file?

`QMFQ%atoms to index`

Type int_array

Description ?

`QMFQ%atoms to mol label`

Type int_array

Description ?

`QMFQ%charge constraints`

Type float_array

Description ?

`QMFQ%external xyz`

Type float_array

Description ?

`QMFQ%fde atoms to index`

Type int_array

Description ?

`QMFQ%fde atoms to mol label`

Type int_array

Description ?

`QMFQ%fde charge constraints`

Type float_array

Description ?

`QMFQ%fde external xyz`

Type float_array

Description ?

`QMFQ%fde index to mol label`

Type int_array

Description ?

`QMFQ%fde type index`

Type int_array

Description ?

`QMFQ%index to mol label`

Type int_array

Description ?

`QMFQ%type alpha`

Type float_array

Description ?

`QMFQ%type chi`

Type float_array

Description ?

`QMFQ%type eta`

Type float_array

Description ?

`QMFQ%type index`

Type int_array

Description ?

`QMFQ%type name`

Type string

Description ?

`QMFQ%type rmu`

Type float_array

Description ?

`QMFQ%type rq`

Type float_array

Description ?

Reference Eigenvector **Section content:** Reference eigenvector to Excited State Geometry Oprimization.

`Reference Eigenvector%Eigenvector`

Type float_array

Description ?

`Reference Eigenvector%Energy`

Type float

Description ?

`Reference Eigenvector%nstate`

Type int

Description ?

Response Data **Section content:** ?

`Response Data%npnts`

Type int

Description ?

`Response Data%polarizabilities`

Type float_array

Description ?

RPA energies Section content: ?

`RPA energies%Direct RPA correlation`

Type float

Description ?

`RPA energies%GM Delta P correlation`

Type float_array

Description ?

`RPA energies%RPA correlation`

Type float

Description ?

`RPA energies%RPA exchange`

Type float

Description ?

`RPA energies%RPA xc`

Type float_array

Description ?

`RPA energies%SOS-MP2 correlation`

Type float

Description ?

`RPA energies%SOX`

Type float

Description ?

SCF Section content: SCF related data

sDFT Energy Section content: FDE related data

`sDFT Energy%Electron electron repulsion`

Type float_array

Description Electron electron repulsion.

`sDFT Energy%Non-additive kinetic energy`

Type float

Description Non-additive kinetic energy.

`sDFT Energy%Non-additive xc energy`

Type float

Description Non-additive xc energy.

`sDFT Energy%Nuclear electron attraction`

Type float_array

Description Nuclear-electron attractions.

`sDFT Energy%Nuclear repulsion`

Type float_array

Description Nuclear-nuclear repulsions.

`sDFT Energy%Subsystem total energies`

Type float_array

Description Subsystem total energies.

`sDFT Energy%Total elel repulsion`

Type float

Description Total electron-electron repulsion.

`sDFT Energy%Total elstat interaction`

Type float

Description Total elstat interaction.

`sDFT Energy%Total interaction energy`

Type float

Description Total interaction energy.

`sDFT Energy%Total nucel attraction`

Type float

Description Total nuclear-electron attraction.

`sDFT Energy%Total nuclear repulsion`

Type float

Description Total nuclear-nuclear repulsion.

`sDFT Energy%Total sDFT energy`

Type float

Description Total sDFT energy.

SFO popul **Section content:** SFO population analysis.

`SFO popul%number of contributions`

Type int_array

Description Number of large SFO contributions per MO. Typically size [SFOs%number], in spin-orbit coupled calculation size [2*SFOs%number].

`SFO popul%sfo_grosspop`

Type float_array

Description Gross populations of both core orthogonalization functions and SFOs per symmetry.

`SFO popul%sfo_index`

Type int_array

Description SFO index for each large SFO comtribution.

`SFO popul%sfo_pop`

Type float_array

Description Large SFO contributions in MOs.

SFO_Fock **Section content:** Fock matrix on SFO basis.

SFO_Overlap **Section content:** SFO overlap matrix.

SFOs **Section content:** SFO related data.

`SFOs%energy`

Type float_array

Description Fragment orbital energies.

Shape [number]

`SFOs%energy_B`

Type float_array

Description Fragment orbital energies, for spin B.

Shape [number]

`SFOs%escale`

Type float_array

Description ZORA only: scaled ZORA fragment orbital energies.

Shape [number]

`SFOs%escale_B`

Type float_array

Description ZORA only: scaled ZORA fragment orbital energies for spin B.

Shape [number]

`SFOs%fragment`

Type int_array

Description Fragment number. Only the number of the first fragment is given in case of symmetry-related fragments.

Shape [number]

`SFOs%fragorb`

Type int_array

Description Fragment orbital number in fragment. Only the number on the first fragment is given in case of symmetry-related fragments.

Shape [number]

`SFOs%fragtype`

Type lchar_string_array

Description Fragment type for each SFO.

Shape [number]

`SFOs%ifo`

Type int_array

Description SFO numbers in symmetry irrep of the total system.

Shape [number]

`SFOs%isfo`

Type int_array

Description Fragment orbital numbers in symmetry irrep of the fragment.

Shape [number]

`SFOs%number`

Type int

Description Total number of (symmetrized) fragment orbitals (SFOs).

`SFOs%occupation`

Type float_array

Description Fragment orbital occupation numbers.

Shape [number]

`SFOs%occupation_B`

Type float_array

Description Fragment orbital occupation numbers, for spin B.

Shape [number]

`SFOs%site_energy`

Type float_array

Description Site energy.

Shape [number]

`SFOs%site_energy_B`

Type float_array

Description Site energies, for spin B.

Shape [number]

`SFOs%site_lowdin_energy`

Type float_array

Description Site Lowdin energies

Shape [number]

`SFOs%site_lowdin_energy_B`

Type float_array

Description Site Lowdin energies, for spin B

Shape [number]

`SFOs%subspecies`

Type lchar_string_array

Description Fragment orbital symmetry label including subspecies for each SFO.

Shape [number]

Spin operator matrix **Section content:** Spin operator matrix.

Spin_orbit **Section content:** Section for spin-orbit coupling.

`Spin_orbit%Bcoef`

Type float_array

Description Arrays with B-coefficients as given in PhD thesis J.G. Snijders (1979) p102.

SUB1!* **Section content:** Data for SUBEXCI.

SUB2!* **Section content:** Data for SUBEXCI.

SUBEXCI **Section content:** Data for SUBEXCI.

SumFrag C energies **Section content:** Sum of fragment correlation energies.

SumFrag X energies **Section content:** Sum of fragment exchange energies.

SumFrag XC energies **Section content:** Sum of fragment XC energies.

SymFit **Section content:** Data for the STO fit procedure.

Symmetry **Section content:** Data related to the symmetry of the molecule.

`Symmetry%faith`

Type float_array

Description The symmetry operator matrices.

Shape [3, 3, nogr]

`Symmetry%grouplabel`

Type string

Description Schoenflies symbol of the symmetry group.

`Symmetry%idcg`

Type int_array

Description Array with for each combination of a rel. symmetry and a non-relativistic one a 1 if this non-relat. symmetry is in the rel. symmetry else a 0.

Shape [nrsym, nsym]

`Symmetry%igr`

Type int

Description Point group identification number. 1: atom, 10: c(lin), 20: d(lin), 30: t(d), 60: o(h), 100: c(n), 200: c(nh), 400: c(nv), 450: d(n), 500: d(nh), 600: d(nd), 700: c(i), 800: c(s), 999: nosym

`Symmetry%isingtrip excitations`

Type int

Description Integer indicating what kind of TD-DFT excitations are calculated.

`Symmetry%ja1ok`

Type int_array

Description An array (1:npeq), with values 0 or 1. 1=the pair density can be fitted using A1 fit functions only. 0=all fit functions (on the involved atoms) are to be used. The value 1 may arise because of symmetry properties, or because the distance between the atoms is so large that the inaccuracy from using only A1 fit functions can be neglected.

Shape [npeq]

`Symmetry%jasym`

Type int_array

Description An array that runs over the npeq sets of equivalent atom pairs. Its value gives for the indicated the set the number of pairs in that set.

Shape [npeq]

`Symmetry%jjsym`

Type int_array

Description An array that runs over the npeq sets of symmetry equivalent atom pairs. Its value gives for the indicated set the index of a (c.f. the first) atom pair in that set.

`Symmetry%jsyml`

Type int_array

Description For each of the nsym representations: if it belongs to a one-dimensional irrep, the value is 1, otherwise: for the first subspecies in the irrep the value is the dimension of the irrep, for the other subspecies in the same irrep the value is 0.

Shape [nsym]

`Symmetry%lnoci`

Type bool

Description If symmetry is NOSYM or C(I) lnoci = true, otherwise false.

`Symmetry%lrep2do excitations`

Type bool_array

Description Which irrep subspecies [irrep,subspecies] calculated in TD-DFT excitations.

Shape [20, 5]

`Symmetry%lrep2do excitations ST`

Type bool_array

Description Which irrep subspecies [irrep,subspecies] calculated in TD-DFT singlet-triplet excitations.

Shape [20, 5]

`Symmetry%napp`

Type int_array

Description An array that stores for each atom the number of the symmetry set it belongs to.

Shape [Geometry%nnuc]

`Symmetry%ncbs`

Type int_array

Description The number of core orthogonalization functions in the corresponding irrep.

Shape [nsym]

`Symmetry%ndim excitations`

Type int_array

Description Dimension (number of subspecies) of the irreps used in TD-DFT excitations.

Shape [nsym excitations]

`Symmetry%nfcn`

Type int_array

Description Number of primitive functions in the corresponding irrep.

Shape [nsym]

`Symmetry%ngr`

Type int

Description One of the integer-code components that fix the symmetry group. See routine adf/maisya

`Symmetry%noat`

Type int_array

Description Map between normal list of atoms and symmetry sets.

Shape [Geometry%nnuc]

`Symmetry%nogr`

Type int

Description The number of symmetry operators. NB, for the special cases of infinite symmetries, only the operators corresponding to finite elements are counted. Therefore, ATOM has nogr=1 (only the unit operator); C(LIN) has nogr=1, D(LIN) has nogr=2

`Symmetry%norb`

Type int_array

Description For each of the nsym representations the number of basis function combinations (SFOs) that belong to it.

Shape [nsym]

`Symmetry%norboc`

Type int_array

Description An array (-2:2,nsym). The column runs over the symmetry representations. The positive row indices (1,2) specify for spin-A and spin-B (the latter only if the calculation is spin-unrestricted), the highest non-empty orbital. The negative indices (-1,-2) specify for spin-A and spin-B (if the unrestricted fragment option is used) the total number of non-empty SFOs.

The zero row index specifies the number of non-empty SFOs, before applying any fragment occupation changes.

`Symmetry%notyps`

Type int_array

Description For each set of symmetry equivalent atoms, the atom type to which the set belongs.

Shape [nsetat]

`Symmetry%npeq`

Type int

Description The number of symmetry unique pairs of atoms.

`Symmetry%nratst`

Type int_array

Description Number of atoms in each set of symmetry equivalent atoms.

Shape [nsetat]

`Symmetry%nrorb`

Type int_array

Description For each of the nrsym double group representations the number of spinors that belong to it.

Shape [nrsym]

`Symmetry%nroroc`

Type int_array

Description The indices (1,irsym) specifies for each double group irrep irsym the highest non-empty spinor. Indices (2,irsym) unused.

Shape [2, nrsym]

`Symmetry%nrsym`

Type int

Description Number of double group irreps.

`Symmetry%nsetat`

Type int

Description Number of sets of symmetry equivalent atoms.

`Symmetry%nsym`

Type int

Description Number of irreps.

`Symmetry%nsym excitations`

Type int

Description Number of irreps used in TD-DFT excitations.

`Symmetry%nsymdav excitations`

Type int_array

Description Calculated nr of TD-DFT excitations for each irrep subspecies [irrep,subspecies].

Shape [20, 5]

`Symmetry%nsymdav excitations ST`

Type int_array

Description Calculated nr of TD-DFT singlet-triplet excitations for each irrep subspecies [irrep,subspecies].

Shape [20, 5]

`Symmetry%nsymden excitations`

Type int_array

Description Maximum nr of TD-DFT excitations for each irrep subspecies [irrep,subspecies].

Shape [20, 5]

`Symmetry%ntr`

Type int_array

Description For each atom A and each symmetry operator R, the atom onto with A is mapped by R. The row index runs over all symmetry operators, the column index over the atoms.

Shape [nogr, Geometry%nnuc]

`Symmetry%ntr_setat`

Type int_array

Description A condensed variety of array ntr: the columns are not the atoms, but the nsetat sets of symmetry equivalent atoms. The value is the index of the atom, onto which a representative (c.f. the first) atom of the indicated symmetry set is mapped by the given symmetry operator.

`Symmetry%rep-D`

Type int

Description max. length of array for repr. matrices.

`Symmetry%rep-I`

Type float_array

Description Imaginary part array with double group repr. matrices.

Shape [rep-D+1]

`Symmetry%rep-R`

Type float_array

Description Real part array with double group repr. matrices.

Shape [rep-D+1]

`Symmetry%rep-S`

Type float_array

Description Representation matrices for single group.

Shape [nogr, nogr]

`Symmetry%repr`

Type float_array

Description Representation matrices.

`Symmetry%reprdim`

Type int_array

Description Dimension (number of subspecies) of the double-group irreducible representation.

Shape [nrsym]

`Symmetry%symlab`

Type lchar_string_array

Description Labels of the irreps.

Shape [nsym]

`Symmetry%symlab excitations`

Type lchar_string_array

Description Labels of the irreps used in TD-DFT excitations.

Shape [nsym excitations]

`Symmetry%symlabr`

Type lchar_string_array

Description Labels of the double group irreps.

Shape [nrsym]

`Symmetry%vecdimension excitations`

Type int

Description Maximum number of the dimension of an eigenvector in TD-DFT excitations.

Thermodynamics **Section content:** Thermodynamic properties computed from normal modes.

`Thermodynamics%Enthalpy`

Type float_array

Description Enthalpy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Entropy rotational`

Type float_array

Description Rotational contribution to the entropy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Entropy total`

Type float_array

Description Total entropy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Entropy translational`

Type float_array

Description Translational contribution to the entropy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Entropy vibrational`

Type float_array

Description Vibrational contribution to the entropy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Gibbs free Energy`

Type float_array

Description Gibbs free energy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Heat Capacity rotational`

Type float_array

Description Rotational contribution to the heat capacity.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Heat Capacity total`

Type float_array

Description Total heat capacity.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Heat Capacity translational`

Type float_array

Description Translational contribution to the heat capacity.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Heat Capacity vibrational`

Type float_array

Description Vibrational contribution to the heat capacity.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Inertia direction vectors`

Type float_array

Description Inertia direction vectors.

Shape [3, 3]

`Thermodynamics%Internal Energy rotational`

Type float_array

Description Rotational contribution to the internal energy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Internal Energy total`

Type float_array

Description Total internal energy.

Unit a.u.

`Thermodynamics%Internal Energy translational`

Type float_array

Description Translational contribution to the internal energy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Internal Energy vibrational`

Type float_array

Description Vibrational contribution to the internal energy.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%lowFreqEntropy`

Type float_array

Description Entropy contributions from low frequencies (see ‘lowFrequencies’).

Unit a.u.

Shape [nLowFrequencies]

`Thermodynamics%lowFreqHeatCapacity`

Type float_array

Description Heat capacity contributions from low frequencies (see ‘lowFrequencies’).

Unit a.u.

Shape [nLowFrequencies]

`Thermodynamics%lowFreqInternalEnergy`

Type float_array

Description Internal energy contributions from low frequencies (see ‘lowFrequencies’).

Unit a.u.

Shape [nLowFrequencies]

`Thermodynamics%lowFrequencies`

Type float_array

Description Frequencies below 20 cm^-1 (contributions from frequencies below 20 cm^-1 are not included in vibrational sums, and are saved separately to 'lowFreqEntropy', 'lowFreqInternalEnergy' and 'lowFreqInternalEnergy'). Note: this does not apply to RRHO-corrected quantities.

Unit cm^-1

Shape [nLowFrequencies]

`Thermodynamics%Moments of inertia`

Type float_array

Description Moments of inertia.

Unit a.u.

Shape [3]

`Thermodynamics%nLowFrequencies`

Type int

Description Number of elements in the array lowFrequencies.

`Thermodynamics%nTemperatures`

Type int

Description Number of temperatures.

`Thermodynamics%Pressure`

Type float

Description Pressure used.

Unit atm

`Thermodynamics%RRHOCorrectedHeatCapacity`

Type float_array

Description Heat capacity T*S corrected using the 'low vibrational frequency free rotor interpolation corrections'.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%RRHOCorrectedInternalEnergy`

Type float_array

Description Internal energy T*S corrected using the 'low vibrational frequency free rotor interpolation corrections'.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%RRHOCorrectedTS`

Type float_array

Description T*S corrected using the 'low vibrational frequency free rotor interpolation corrections'.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%Temperature`

Type float_array

Description List of temperatures at which properties are calculated.

Unit a.u.

Shape [nTemperatures]

`Thermodynamics%TS`

Type float_array

Description T*S, i.e. temperature times entropy.

Unit a.u.

Shape [nTemperatures]

Total C energies **Section content:** Total correlation energies.

Total Energy **Section content:** Data related to a total energy calculation.

Total X energies **Section content:** Total exchange energies.

Total XC energies **Section content:** Total XC energies.

TransferIntegrals **Section content:** Charge transfer integrals relevant for hole or electron mobility calculations. Electronic coupling V (also known as effective (generalized) transfer integrals J_eff) V = (J-S(e1+e2)/2)/(1-S^2). For electron mobility calculations the fragment LUMOs are considered. For hole mobility calculations the fragment HOMOs are considered.

`TransferIntegrals%e1(electron)`

Type float

Description Site energy LUMO fragment 1.

Unit hartree

`TransferIntegrals%e1(hole)`

Type float

Description Site energy HOMO fragment 1.

Unit hartree

`TransferIntegrals%e2(electron)`

Type float

Description Site energy LUMO fragment 2.

Unit hartree

`TransferIntegrals%e2(hole)`

Type float

Description Site energy HOMO fragment 2.

Unit hartree

`TransferIntegrals%J(charge recombination 12)`

Type float

Description Charge transfer integral HOMO fragment 1 - LUMO fragment 2 for charge recombination 1-2.

Unit hartree

`TransferIntegrals%J(charge recombination 21)`

Type float

Description Charge transfer integral LUMO fragment 1 - HOMO fragment 2 for charge recombination 2-1.

Unit hartree

`TransferIntegrals%J(electron)`

Type float

Description Charge transfer integral LUMO fragment 1 - LUMO fragment 2 for electron transfer.

Unit hartree

`TransferIntegrals%J(hole)`

Type float

Description Charge transfer integral HOMO fragment 1 - HOMO fragment 2 for hole transfer.

Unit hartree

`TransferIntegrals%S(charge recombination 12)`

Type float

Description Overlap integral HOMO fragment 1 - LUMO fragment 2 for charge recombination 1-2.

`TransferIntegrals%S(charge recombination 21)`

Type float

Description Overlap integral LUMO fragment 1 - HOMO fragment 2 for charge recombination 2-1.

`TransferIntegrals%S(electron)`

Type float

Description Overlap integral LUMO fragment 1 - LUMO fragment 2.

`TransferIntegrals%S(hole)`

Type float

Description Overlap integral HOMO fragment 1 - HOMO fragment 2.

`TransferIntegrals%V(charge recombination 12)`

Type float

Description Effective charge transfer integral HOMO fragment 1 - LUMO fragment 2 for charge recombination 1-2.

Unit hartree

`TransferIntegrals%V(charge recombination 21)`

Type float

Description Effective charge transfer integral LUMO fragment 1 - HOMO fragment 2 for charge recombination 2-1.

Unit hartree

`TransferIntegrals%V(electron)`

Type float

Description Effective transfer integral LUMO fragment 1 - LUMO fragment 2 for electron transfer.

Unit hartree

`TransferIntegrals%V(hole)`

Type float

Description Effective transfer integral HOMO fragment 1 - HOMO fragment 2 for hole transfer.

Unit hartree

`TransferIntegrals%Vtot(charge recombination 12)`

Type float

Description Total electronic coupling for charge recombination 1-2.

Unit hartree

`TransferIntegrals%Vtot(charge recombination 21)`

Type float

Description Total electronic coupling for charge recombination 2-1.

Unit hartree

`TransferIntegrals%Vtot(electron)`

Type float

Description Total electronic coupling for electron transfer.

Unit hartree

`TransferIntegrals%Vtot(hole)`

Type float

Description Total electronic coupling for hole transfer.

Unit hartree

VCDTools **Section content:** VCDTools data: VCD analysis.

Vibrations **Section content:** Information related to molecular vibrations.

`Vibrations%ExcitedStateLifetime`

Type float

Description Raman excited state lifetime.

Unit hartree

`Vibrations%ForceConstants`

Type float_array

Description The force constants of the vibrations.

Unit hartree/bohr^2

Shape [nNormalModes]

`Vibrations%Frequencies[cm-1]`

Type float_array

Description The vibrational frequencies of the normal modes.

Unit cm^-1

Shape [nNormalModes]

`Vibrations%Intensities[km/mol]`

Type float_array

Description The intensity of the normal modes.

Unit km/mol

Shape [nNormalModes]

`Vibrations%IrReps`

Type lchar_string_array

Description Symmetry symbol of the normal mode.

Shape [nNormalModes]

`Vibrations%ModesNorm2`

Type float_array

Description Norms of the rigid motions.

Shape [nNormalModes+nRigidModes]

`Vibrations%ModesNorm2*`

Type float_array

Description Norms of the rigid motions (for a given irrep…?).

Shape [nNormalModes+nRigidModes]

`Vibrations%nNormalModes`

Type int

Description Number of normal modes.

`Vibrations%NoWeightNormalMode(#)`

Type float_array

Description ?.

Shape [3, Molecule%nAtoms]

`Vibrations%NoWeightRigidMode(#)`

Type float_array

Description ?

Shape [3, Molecule%nAtoms]

`Vibrations%nRigidModes`

Type int

Description Number of rigid modes.

`Vibrations%nSemiRigidModes`

Type int

Description Number of semi-rigid modes.

`Vibrations%PVDOS`

Type float_array

Description Partial vibrational density of states.

Values range [0.0, 1.0]

Shape [nNormalModes, Molecule%nAtoms]

`Vibrations%RamanDepolRatioLin`

Type float_array

Description Raman depol ratio (lin).

Shape [nNormalModes]

`Vibrations%RamanDepolRatioNat`

Type float_array

Description Raman depol ratio (nat).

Shape [nNormalModes]

`Vibrations%RamanIncidentFreq`

Type float

Description Raman incident light frequency.

Unit hartree

`Vibrations%RamanIntens[A^4/amu]`

Type float_array

Description Raman intensities

Unit A^4/amu

Shape [nNormalModes]

`Vibrations%ReducedMasses`

Type float_array

Description The reduced masses of the normal modes.

Unit a.u.

Values range [0, '\infinity']

Shape [nNormalModes]

`Vibrations%RotationalStrength`

Type float_array

Description The rotational strength of the normal modes.

Shape [nNormalModes]

`Vibrations%TransformationMatrix`

Type float_array

Description ?

Shape [3, Molecule%nAtoms, nNormalModes]

`Vibrations%VROACIDBackward`

Type float_array

Description VROA Circular Intensity Differential: Backward scattering.

Unit 10¯3

Shape [nNormalModes]

`Vibrations%VROACIDDePolarized`

Type float_array

Description VROA Circular Intensity Differential: Depolarized scattering.

Unit 10¯3

Shape [nNormalModes]

`Vibrations%VROACIDForward`

Type float_array

Description VROA Circular Intensity Differential: Forward scattering.

Unit 10¯3

Shape [nNormalModes]

`Vibrations%VROACIDPolarized`

Type float_array

Description VROA Circular Intensity Differential: Polarized scattering.

Unit 10¯3

Shape [nNormalModes]

`Vibrations%VROADeltaBackward`

Type float_array

Description VROA Intensity: Backward scattering.

Unit 10¯3 A^4/amu

Shape [nNormalModes]

`Vibrations%VROADeltaDePolarized`

Type float_array

Description VROA Intensity: Depolarized scattering.

Unit 10¯3 A^4/amu

Shape [nNormalModes]

`Vibrations%VROADeltaForward`

Type float_array

Description VROA Intensity: Forward scattering.

Unit 10¯3 A^4/amu

Shape [nNormalModes]

`Vibrations%VROADeltaPolarized`

Type float_array

Description VROA Intensity: Polarized scattering.

Unit 10¯3 A^4/amu

Shape [nNormalModes]

`Vibrations%ZeroPointEnergy`

Type float

Description Vibrational zero-point energy.

Unit hartree

XAS DATA **Section content:** XAS data.

XCRangeSeparated **Section content:** ?

`XCRangeSeparated%CAMTypeCalculation`

Type bool

Description ?

`XCRangeSeparated%currentRegion`

Type int

Description ?

`XCRangeSeparated%DFTLongRange`

Type bool

Description ?

`XCRangeSeparated%DFTOnly`

Type bool

Description ?

`XCRangeSeparated%ExactLongRange`

Type bool

Description ?

`XCRangeSeparated%ExcludeHF`

Type bool

Description ?

`XCRangeSeparated%HFOnly`

Type bool

Description ?

`XCRangeSeparated%RangeSeparated`

Type bool

Description ?

`XCRangeSeparated%Region_coefficient`

Type float_array

Description ?

Shape [10]

`XCRangeSeparated%Region_gamma`

Type float_array

Description ?

Shape [10]

`XCRangeSeparated%Region_IsDFT`

Type bool_array

Description ?

Shape [10]

`XCRangeSeparated%Region_IsExactExchange`

Type bool_array

Description ?

Shape [10]

XES DATA **Section content:** XES data.

ZFS **Section content:** ?

`ZFS%Dtensor`

Type float_array

Description ?

`ZFS%DtensorDiag`

Type float_array

Description ?

`ZFS%ZFSPrincipleAxis`

Type float_array

Description ?

ZlmFit_* **Section content:** Data related to the density fitting procedure (ZlmFit).

ZlmFitConfig **Section content:** Configuration options for the Zlm density fit.

`ZlmFitConfig%densityThresh`

Type float_array

Description Threshold for the density.

Shape [nAtoms]

ZlmFitConfig%gridAngOrder

Type int_array

Description Angular order (Lebedev grid) per atom.

Shape [nAtoms]

ZlmFitConfig%lMaxExpansion

Type int_array

Description Maximum l-value for the fit functions per atom.

Shape [nAtoms]

ZlmFitConfig%nAtoms

Type int

Description Number of atoms.

ZlmFitConfig%nRadialPoints

Type int_array

Description Number of radial points per atom.

Shape [nAtoms]

ZlmFitConfig%partitionFunThresh

Type float_array

Description Threshold for the partition function.

Shape [nAtoms]

ZlmFitConfig%partitionSizeAdjustment

Type bool

Description Atom dependent parition size?

ZlmFitConfig%potentialThresh

Type float_array

Description Threshold for the potential.

Unit a.u.

Shape [nAtoms]

ZlmFitConfig%pruning

Type bool

Description Whether or not to prune.

ZlmFitConfig%pruningGridAngOrder

Type int

Description ?.

`ZlmFitConfig%pruningL`

Type int

Description ?.

`ZlmFitConfig%pruningThreshDist`

Type float

Description Distance threshold for pruning.

Unit bohr

CHAPTER

EIGHTEEN

FAQ

18.1 My ADF input does not work in AMS2020?

In our AMS2020 release, *ADF has been fully integrated to the AMS driver* (page 9), simplifying the input. If you run through the GUI this will not affect you. If you use your own command line input with scripts, your old inputs are not compatible with AMS2020 or beyond. Old ADF inputs can be *converted with a simple tool* (page 13). Let us know at support@scm.com with your old input if that does not work.

18.2 Can I use Gaussian type basis (6-31G*, TZVP, etc.) functions in ADF?

No, Gaussian-type orbitals can not be used in ADF, since ADF uses Slater Type Orbitals (STOs).

Slater orbitals are better from a theoretical point of view since they have the correct behavior near the nucleus (cusp) and the correct asymptotic long-range behavior. Consequently, typically fewer STOs than GTOs are needed to reach a certain level of accuracy, as discussed in this news item Slaters beat Gaussians (https://www.scm.com/highlights/slaters-beat-gaussians-fe-complexes/) and a paper benchmarking DFT methods for a catalytic reaction (http://dx.doi.org/10.1002/jcc.23212).

18.3 Which basis set should I use?

There is no silver bullet. Some general recommendations follow below, which are by no means definite. Please check the literature carefully and check the robustness of the results by using a large basis set eventually, if possible. See also *a more extended discussion on what basis set to use* (page 349).

The nomenclature of the STO basis sets in ADF: D, T, and Q stand for double, triple, and quadruple, and (n)P stands for n polarization functions. For example, the DZP basis has double zeta + 1 polarization function. Inner electrons of (atomic) fragments can be frozen during the molecular calculation, this is denoted with ‘small’, ‘medium’ or ‘large’, or by specifically specifying up to which shell the electrons are frozen. In general, we recommend the use of frozen core basis sets for LDA and GGA functionals if available. All electron basis sets are required in case of SAOP, meta-GGA and meta-hybrid functionals, functionals that use LibXC, post-KS calculations like GW, RPA, MP2 or double hybrids. For hybrid functionals, the use of frozen cores usually is fine for geometries and non-core spectroscopic properties.

There are also specialized basis sets, e.g. containing diffuse functions (“AUG”, for describing anions and diffuse excitations), even-tempered basis sets (“ET”, for describing core electron binding energies), and optimized for ESR A-tensor and NMR spin-spin coupling calculations (ZORA/TZ2P-J and ZORA/QZ4P-J). *See also the documentation* (page 37) for more technical details and the basis set database in ADF.

The DZP basis set could be a good starting point, especially for geometry optimization. See also this paper looking at catalytic reactions (http://dx.doi.org/10.1002/jcc.23212). In general, this basis it is expected to be slightly better than the often used 6-31G* basis set in codes which use Gaussian basis sets. The DZP basis defaults to TZP for transition metals. The frozen core approximation generally does not affect geometries much, but a too large frozen core can lead to inaccuracies.

For the most accurate predictions of spectroscopic properties, the largest QZ4P basis is recommended, although often the TZ2P basis is already close to the basis set limit. Also, usually an all-electron basis set is necessary to properly describe properties related to inner electrons (NMR, EPR, X-ray absorption).

18.4 How do I calculate UV/VIS spectra with ADF?

The tutorial on calculating UV/VIS spectra with ADF will get you started with the basics. For getting more accurate results, however, you need to consider the technical details used in TDDFT, such as basis set, exchange correlation functional, and other settings.

There are also faster approximate TDDFT methods available, such as TDDFTB (see tutorial), sTDDFT, TDDFT+TB (webinar (https://www.scm.com/news/ams2023-webinar-recordings/)), and POLTDDFT (tutorial).

We also have an advanced tutorial / case study on TDDFT methods.

One can also go beyond TDDFT using quasiparticle self-consistent GW with the Bethe-Salpeter Equation (qsGW-BSE: high-lights (https://www.scm.com/highlights/large-scale-self-consistent-gw-bethe-salpeter-equation-calculations-in-adf/), tutorial), which could be the most accurate approach.

You are advised to check the literature on which approaches are suitable for similar compounds. Here are some general recommendations on what could work well in ADF:

1) optimize your molecule. Scalar relativistic, TZP, small core (although a large frozen core could also work), and numerical quality 'Normal' would typically be quite accurate. As to the functional, a GGA perhaps with dispersion corrections could give very good results (e.g. PBE-D3(BJ)). GGAs are much faster in ADF and can yield better geometries than hybrids. Note that the dispersion interactions affect the geometry, and thereby indirectly the electronic structure, including excitations.
2) TDDFT calculation for UV/VIS. Scalar relativistic up to 4d-elements is usually sufficient. For heavier elements you may want to check the effect of spin-orbit coupling on the spectrum (see also highlight on conversion efficiency of solar cell dyes (https://www.scm.com/highlights/spin-orbit-coupling-increases-dye-sensitized-solar-cells-efficiency/)). You may want a larger basis set (e.g. TZ2P) and perhaps all-electron. Also a higher accuracy may be desirable (good, or excellent). As to the number of excitations, this depends a bit on the electronic ground state and which excitations you are interested in. You could start with 30 or so to see how much to the blue/UV will be covered. Finally the functional could play an important role. The asymptotically correct SAOP model potential is often used, since this is an accurate Kohn-Sham potential well suited to describe virtual orbitals (see publication by Baerends et al. (http://dx.doi.org/10.1021/ct500727c)). You could also try hybrids (e.g. B3LYP) or metahybrids (M06-2X). Range-separated hybrids (CAMY-B3LYP) are sometimes advised for excitations with long-range charge transfer character (you can also tune range-separated hybrids (https://www.scm.com/news/tuning-range-separation-lc-wpbe-organic-electronics/)). SAOP calculations will be much faster in ADF than hybrids.

Further comments:

- for symmetric systems, ALLOWEDONLY can reduce the computation time by excluding the excitations which are symmetry-forbidden
- for very large systems you may consider trying out TD-DFTB in combination with filtering out the single orbital transitions with very low oscillator strength
- vibronic fine structure may be included by calculating Franck-Condon Factors. Since this involves calculating excited state frequencies (numerically), the calculations can be very demanding for large systems

18.5 What settings are recommended for NMR calculations?

See also the tutorial on relativistic NMR calculations. Perhaps the unique capabilities of ADF to analyze orbital contributions to NMR are of interest, too. We also have a tutorial on how to visualize spin-spin coupling.

An all-electron basis set of TZP or higher. TZ2P is usually a good choice. For heavy metals you may consider the QZ4P basis set. For NMR spin-spin couplings you may consider the ZORA/TZ2P-J or ZORA/QZ4P-J basis.

PBE0 is often a good functional for NMR. If you can't afford hybrids you could try PBE or SAOP.

For the relativistic approach we recommend at least scalar ZORA. Spin-orbit (SO) ZORA should be used if you are looking at the shift of an heavy atom, or a light atom near a heavy atom. If SO effects are large, a QZ4P or QZ4P-J basis will improve the description of the induced spin density on the probe nucleus. You may also want to try out the *'fxc' kernel option* (page 207) – note that this has a small gauge dependency.

The numerical accuracy should be better than default. You could try NUMERICALQUALITY good or verygood to set the grid and density fitting accuracy.

Disclaimer: These general recommendations are based on the work of Jochen Autschbach's group (https://ja01.chem.buffalo.edu/). You should always carefully try out a few different options to test the sensitivity of the DFT methods on the NMR predictions for your own system. Some suggested references:

- http://dx.doi.org/10.1002/chem.201502346
- http://dx.doi.org/10.1021/ja3040762
- http://dx.doi.org/10.1002/anie.201005431

18.6 Do you have tips on how to find a Transition State with ADF?

Finding a Transition State (TS) can be difficult. Two things will help the TS search:

- Get a geometry close to the TS
- Get a reasonable Hessian (energy curvature)

1. to get a reasonable starting geometry you could e.g. start with a linear transit, nudged elastic band or from a previous TS geometry of a similar reaction.
2. defining the reaction coordinate (TSRC) may be the easiest option to get a reasonable Hessian. You could also calculate the full Hessian (frequency calculation), a partial Hessian or a Mobile Block Hessian, which should be better-conditioned.

For all initial Hessian calculations you could consider using a lower-accuracy method (e.g. lower integration accuracy, smaller basis sets, or even using DFTB). You should however always check the Hessian, e.g. by visualizing the frequencies, to check if it has one single negative eigenvalue and whether that eigenmode corresponds to your reaction. For the TS search itself you should switch to higher accuracy – especially think about the numerical integration and density fitting.

See also the Tutorial Tips and Tricks for Transition State Searches for Click Reactions. See also some older Tips, tricks, examples and exercises for finding Transition States for an older version of ADF (before the AMSification, which changed the input a lot) (`pdf` (https://www.scm.com/wp-content/uploads/TransitionStateSearch_ADF_tips_tricks_exercises_examples.pdf), `input files` (https://www.scm.com/wp-content/uploads/TransitionStateSearch_Inputs.zip))

18.7 What's the difference between COSMO and COSMO-RS? Do I need COSMO-RS to include solvent effects?

You can do COSMO calculations with ADF. The COntinuum SOlvation MOdel treats the solvent as a polarizable continuum. You can also use other solvation models (SM12, 3D-RISM) or subsystem DFT (Frozen Density Embedding) to include environment effects.

COSMO-RS (https://www.scm.com/product/cosmo-rs/) (Realistic Solvents) is for calculating thermodynamic properties of (mixed) fluids. It uses post-ADF results (sigma profiles) for determining interactions between species in the liquid phase. You will typically not need COSMO-RS if you want to include solvent effects in your DFT calculation, but some properties such as solvation free energies may be more accurately treated with COSMO-RS than with COSMO.

18.8 What does a small, medium, or large frozen core mean?

To speed up a molecular or periodic DFT calculation, inner electrons of atoms can be frozen. Note that this is not a pseudopotential: all electrons are still present, the cores are just frozen at their optimized atomic configuration. The effect on the equilibrium geometry and properties of valence electrons is usually small. For spectroscopic properties of core electrons, or electrons close to the nucleus, typically an all electron (AE) approach is necessary.

In the atomic database (you can browse $AMSHOME/atomicdata to see which basis sets are available) the AE basis sets are denoted as the element names, while the name is extended with the highest shell that is kept frozen for the frozen core basis sets.

small core: the smallest core available in the basis set database, for example:

- Rh.3d: all 28 electrons up to 3d are frozen: [Ar]3d10. The valence orbitals are: 4s, 4p, 5s, 4d, 5p.
- Au.4d: all 46 electrons up to 4d are frozen: [Kr]4d10. The valence orbitals are: 5s, 5p, 6s, 4f, 5d, 6p.

large core: the largest core available in the basis set database, for example:

- Rh.4p: all 36 electrons up to 4p are frozen: [Kr]. The valence orbitals are: 5s, 4d, 5p.
- Au.4f: all 60 electrons up to 4f are frozen, [Kr]4d104f14. The valence orbitals are: 5s, 5p, 5d, 6s, 6p.

Note that for some elements and basis sets there is just 1 core available (e.g. 1s for C), in which case all core options default to that specific frozen core.

When there are two options available (e.g. Rh.3d and Rh.4p), the **medium** core option typically defaults to the large core option. The use of **medium** core is therefore not recommend, we suggest to choose your specific basis set, either from the GUI in the basis set details panel, or by browsing the available basis sets and specifically defining the appropriate core.

18.9 How did the keywords in ADF change from 2017 to 2018?

In 2018 a few keywords changed their format and *input parsing was made more strict* (page 31). See the *input parsing in the ADF documentation* (page 25).

18.10 How to do an EDA or ETS-NOCV calculation with open-shell fragments?

Fragment files in ADF no longer need to be spin-restricted. So as of ADF2019, you can use unrestricted fragments.

18.11 Can ADF do restricted open shell Kohn-Sham (ROKS)?

In AMS2023 *ROKS for high spin open shell molecules* (page 57) has been implemented. Note, however, that this is work-in-progress.

When you do a spin-restricted calculation in ADF with an odd number of electrons, you do not get the energy expression that is used in ROKS. In ROKS, the spin-unrestricted energy is optimized with alpha orbitals restricted to have the same spatial form as the beta orbitals. In ADF using a restricted calculations with an odd number of electrons will just result in fractional occupations of the frontier orbital. The energy of this electronic configuration is not the same as in a ROKS calculation.

Unrestricted calculations (UKS) in ADF should lead to proper integer occupations of the alpha and beta electrons, which are treated independently and can have spatially different forms.

18.12 Can I use GPU acceleration in ADF?

You can for certain types of calculations (energies, forces and Hessians for GGAs only). You will need a Linux OS with an NVidia CUDA GPU that does double precision, see the Installation manual for more details.

18.13 Where can I find the total energy in the output?

ADF does not calculate total energies. It calculates difference energies with respect to fragment energies. By default, these are the spherical spin-restricted atoms. For this reason total energies from other programs cannot be compared to ADF directly. Only energy difference comparisons are meaningful. These are the only energies that play a role in chemistry of course.

If you do need the total energy, there is a workaround: calculate total energies of the atomic fragments and add them to the bonding energy.

Because the total energy of an atom is, by definition, the energy difference between the atom and the (nucleus+free electrons) system one can calculate it by calculating a single atom with the charge equal to the number of electrons. The 'bonding energy' of such an 'atom' will then be equal to negative of the total energy of the atomic fragment. Care should be taken to apply this trick to frozen-core fragments. In this case, it only makes sense to remove the valence electrons and leave the frozen core.

18.14 How do I calculate atomization energies with ADF?

You need to subtract the bonding energies of the individual atoms from the bonding energy of the molecule. Care must be taken to get the correct atomic reference state. Usually you will need to break the symmetry (otherwise spherical symmetry is used) and note that ADF can not perform restricted open-shell Kohn-Sham (ROKS) calculations. An example input for the triplet oxygen atom:

```
System
   Atoms
   O 0.0 0.0 0.0
   End
End
Task SinglePoint
Engine adf
   Relativity
      Level None
   End
   SpinPolarization 2
   Unrestricted Yes
   Basis
      Type QZ4P
      Core None
   End
   IrrepOccupations
      A 5 // 3
   End
   numericalquality good
   symmetry nosym
   xc
      hybrid B3LYP
   End
EndEngine
```

18.15 I have problems with SCF convergence ("more iterations needed. not converged"). What to do?

There are a few options to aid SCF convergence:

- modern SCF algorithms: A-DIIS is usually a good option, and is default since ADF2016. Alternative options are LISTi (with good scaling) and ARH (slower per SCF cycle). E-DIIS is similar to A-DIIS but slower
- use the OCCUPATIONS key to specify occupation numbers
- use electron SMEARING for systems with low HOMO-LUMO gap
- use KEEPORBITALS to keep the same orbitals occupied after a certain number of steps

See the *SCF troubleshooting section* (page 356) of the ADF manual and send us (support@scm.com) relevant input and output if you keep having problems.

18.16 The geometry of my molecule is converged but when I do a frequency calculation, ADF finds one or more imaginary frequencies. Why?

There may be a few reasons why imaginary frequencies are found after geometry optimization:

- Visually inspect the negative normal mode in AMS Spectra by selecting it. Pause at some frame to get the distorted structure, and use Play -> Update geometry in input to use that as a new starting point for geometry optimization. If it's a frustrated rotation, you can also in the GUI manually distort it by changing the dihedral.
- The optimization threshold is not strict enough for the molecule's potential energy surface (PES). For example, if there are almost no-barrier rotational modes (e.g. methyl groups in some situations) or cyclopentane-like fragments, then the PES in such cases may be almost flat. In this case, much more strict convergence criteria (and higher integration accuracy) may be necessary to get to the energy minimum.
- Even with tight convergence criteria, one may get a small imaginary frequency due to numerical noise, which is always present in every calculation. In this case, it may be useful to scan the normal mode corresponding to the imaginary frequency and recheck is the frequency is indeed imaginary by finite differences. The latter is done using the SCANFREQ keyword, which is enabled by default in the AMSinput GUI module.
- If the scan shows an imaginary frequency, then it means that the preceding geometry optimization converged to a saddle point. This may happen when optimization was started close to the saddle point without a good Hessian estimate. In such a case, ADF did not make enough steps to get a good Hessian approximation and figure out that it has negative eigenvalue(s).

18.17 Are TDDFT calculations compatible with COSMO?

Yes, it is be possible to combine COSMO and TDDFT. This will modify the zeroth-order KS orbitals and their energies, which is usually the most important solvation effect. COSMO could also modify the first-order change in the KS potential. This effect has been implemented and will certainly be important in case of a high value for the dielectric constant. The response of the COSMO surface charges due to the TDDFT-induced modified charges may be switched off (NOCSMRSP), which is probably a better representation of the solvent response for instantaneous (de)excitation.

In the response part of the calculations, one can also use a different dielectric constant . The reason for using two different dielectric constants is that the electronic transition can be so fast that only the electronic component of the solvent dielectric can respond, i.e., one should use the optical part of the dielectric constant. This is typically referred to as non-equilibrium solvation. If one optimizes an excited state, the solvent dielectric has time to fully respond and the same dielectric constant should be used..

18.18 Can I optimize excited states with TDDFT in ADF?

Yes. See the tutorial and *documentation* (page 185) on excited state geometry optimizations.

18.19 Can I determine Raman intensities for open-shell systems?

Yes, this can be done numerically by distorting the structure along the normal modes. First calculate the normal modes "q" with ADF. Then calculate the polarizability tensor by applying finite electric fields at the geometries q_o + dq and q_o – dq. From this one can obtain, by double finite difference, the derivative of the polarizability w.r.t. a particular normal mode of interest. This should be less time-consuming then a regular Raman calculation as well.

18.20 How do I calculate the frequency-dependent second hyperpolarizability, GAMMA?

The beta tensor can be obtained by using the input key RESPONSE, for example:

```
RESPONSE
   HYPERPOL 0.03
   DYNAHYP
   ALLCOMPONENTS
END
```

In order to obtain gamma, one should calculate beta in small electric fields. Such a field can be switched on by specifying, for example:

```
System
   ElectrostaticEmbedding
      ElectricField 0.0 0.0 0.001
   End
End
```

in the AMS part of the input for an electric field of in the z-direction. The beta calculation needs to be repeated for different field directions. gamma_zzzz can be obtained from (for example): gamma_zzzz = beta_zzz (E=0.001 z) – beta_zzz (E=-0.001z) / (2 * 0.001)

Similar relations hold for other tensor components.

The group of Lasse Jensen has implemented damped quadratic and cubic response, which gives access to many non-linear optical properties including two-photon absorption. These methods are available in ADF2017 and later.

18.21 How can I do a spin restricted open-shell calculation for the calculation of the A-tensor?

For a spin restricted open-shell calculation, save adf.rkf (TAPE21), and use the whole molecule as a fragment in a SR unrestricted calculation with the keywords:

```
scf
   Iterations 0
end
```

ADF will use the fragment density coming from the spin-restricted calculation as start-up density in the unrestricted calculation. iter 0 means that only 1 scf-iteration will be performed with the use of the fragment density (one should use the same XC-potential in both calculations of course), such that the alpha-orbital energies remain degenerate with the beta-orbitals.

18.22 How do I read data from the binary KF files (adf.rkf, TAPE21, .runkf, .rkxf, .rkf or .t21 result files)?

After a calculation with ADF, BAND, ReaxFF, DFTB etc you will have a result file (a .t21, .runkf, .rxkf, or .rkf file). These are binary files, and like all other binary files used by the ADF package they are written using the KF library. The data are organized in sections, and each section can contain variables (either real, integer, string or logical). They may be transported between different platforms, conversion is performed automatically when needed.

Not all data on the files are intended to be used directly by end users, the detailed meaning of many data items will not be clear to end users who do not have access to the source code. Preferably use the GUI utilities to visualize the results or scripting tools to get relevant results from a batch of calculations.

The KF Browser is a useful tool to browse through our binary files. The available data can be searched, copied as text and plotted as graphs. It can be started from the GUI (SCM → KF Browser) or from the command line ($ADFBIN/kfbrowser filename). To see all raw data, switch to the Expert mode (File → Expert Mode in KF Browser).

amsreport is a command line tool to quickly access results from different result files (.t21, .runkf, .rkf, .rxkf) in a uniform manner. You can get any variable from a result file by key value, or you can use predefined report commands (like geometry, charges, energy, etc). Run with the -h flag to get help and a detailed description of all predefined commands). Note: if you include the -h flag and a filename, you will get specific help for the contents of that file. To use amsreport via the command line:

```
$ADFBIN/amsreport [-h] filename
```

KF Utilities are a set of command line utilities to handle KF files.

- pkf: print the table of contents
- dmpkf: dump the contents of all or part of a KF file to an ASCII file
- udmpkf: make a binary KF file out of a ASCII file (typically a dumped KF File)
- cpkf: copy all or part of one KF file to another KF file, or remove data from a KF file.

`KFReader` (http://downloads.scm.com/Downloads/KFReader-20140106.zip): C routines to read binary KF files

`kf.py` (http://downloads.scm.com/Downloads/kf.py): python interface for reading binary KF files (requires AMS installation).

18.23 I get the error: NO TRANSFORMATION FOUND in a fragment-based calculation. What to do?

A possible origin of this problem is that ADF can symmetrize the input coordinates of fragments which are nearly symmetric. As a consequence, the fragment coordinates can no longer be transformed to their position in the full calculation of the complex.

The workaround is to use SymmetryTolerance 1.0e-10 (in older ADF versions SYMMETRY TOL=1.0e-10) in the calculation of the fragments, such that the coordinates in the fragments are not symmetrized.

18.24 What does the error message "BAD ETA INTEGRALS" mean?

This message shows that the numerical integration is not as accurate as required (currently tested only with spin-orbit calculations). You can force the calculation to continue using the ALLOW BADINTEGRALS input key. This will force the calculation to continue, but remember there are possible inaccuracies due to these integrals. You can probably increase the accuracy of the eta integrals by increasing the overall integration accuracy (using the integration key). One will then probably still get the BAD ETA INTEGRALS message, but this is relative to the required accuracy. One can check the accuracy (it is printed on output), and if the accuracy is good enough one can use the ALLOW BADINTEGRALS key to force the calculation to continue.

18.25 ADF will not start when the user's login name has non-ASCII characters in it. Sometimes a message "QUICKX terminates" is printed to the logfile.

One of the reasons can be that the SCM_TMPDIR environment variable also contains non-ASCII characters and ADF cannot use it. The solution is to set the SCM_TMPDIR value to a path that does not contain non-ASCII characters, for example, C:TMP:

- Right click on My Computer
- Select Properties item in the pop-up menu
- Click on the "Advanced" tab
- Click on the "Environment Variables" button
- In the upper half under "User variables for …", click the New button
- Enter SCM_TMPDIR in the variable name and C:/TMP in the variable value field. Please note that the path in the variable value must not contain spaces. It is also advised to use forward slashes as a path separator instead of the backslash
- Make sure the C:TMP directory exists and is writable by anyone
- Click on the Ok button in all windows to confirm your change
- If you use the GUI, restart it so the change takes effect

18.26 What does this error mean Int. Energy with ZlmFit+LargeFragment+VoronoiGrid not implemented.

Most probably you use the INTEGRATION key for a fragment analysis and thereby using an incompatible combination of ZlmFit and Voronoi grid for fragment calculations. You should either use Voronoi + STOfit or Becke + ZlmFit (which is the default since ADF2014).

Explanation of the terms in the error: "Int. Energy with ZlmFit+LargeFragment+VoronoiGrid not implemented"

1. ZlmFit means the way the Coulomb potential is fitted. ZlmFit is the default in ADF2014 and beyond. One can also use the STOFit, using the key STOFIT. If one would use STOFIT this error would not appear.
2. LargeFragment means a fragment larger than an atom.
3. VoronoiGrid is the integration method. The VoronoiGrid will be used if one is using the INTEGRATION key. In ADF2013 and beyond the Becke grid has become the default. If one would use the default integration method in ADF, which is the BeckeGrid, this error would not appear.

Recommended is to use the BeckeGrid and ZlmFit if one has larger fragments, possibly using quality good.

INDEX