



Scientific Computing & Modelling

DFTB Manual

**ADF Program System
Release 2012**

Scientific Computing & Modelling NV
Vrije Universiteit, Theoretical Chemistry
De Boelelaan 1083; 1081 HV Amsterdam; The Netherlands
WWW: www.scm.com
E-mail: support@scm.com

Copyright © 1993-2012: SCM / Vrije Universiteit, Theoretical Chemistry, Amsterdam, The Netherlands
All rights reserved

Table of Contents

DFTB Manual.....	1
Table of Contents	2
Introduction.....	3
Release 2014	3
DFTB-GUI	3
Input.....	4
Specification of the System.....	4
Specification of the computational Task.....	5
Changing the default Units.....	6
Setting DFTB Calculation details	6
Geometry optimization.....	10
Constrained optimization	12
TDDFTB excited states	13
Restart	16
Molecular Dynamics.....	16
Available Thermostats	18
Additional Periodicity Data.....	19
Timing details	20
Examples.....	21
Parameter files.....	22
Installing additional DFTB.org parameter files	22
Third Order parameter files	23
metainfo.yaml	23
References	25
DFTB: general description.....	25
DFTB: parameter sets	25
Dresden	25
QUASINANO2013.1	26
DFTB.org.....	26
Keywords	27

Introduction

Our implementation of the DFTB method can perform single point calculations, geometry optimizations, transition state searches, frequency calculations, and molecular dynamics. Molecules as well as periodic systems can be handled ensuring a smooth link with our full DFT codes ADF and BAND. It can be used as a stand-alone command line program, or from the graphical interface.

The DFTB program is orders of magnitude faster than DFT, but requires parameter files to be installed for all pair-wise combinations of atoms in a molecule. Many elements can be handled with the Dresden parameter set included in the distribution, while many other parameter sets (from DFTB.org) can be [enabled](#), free of charge for non-profit users. Alternatively, sets of parameters in the SKF format can be downloaded and used from third party sources.

Three models within the DFTB framework are available: standard DFTB, SCC-DFTB (DFTB with self-consistent-charge correction), and DFTB3 (SCC-DFTB with third-order correction). As they have been respectively parameterized, it is important to specify a proper parameter set when applying one of these models.

Release 2014

The 2014 release of our DFTB program offers the following new functionality:

Functionality

- TDDFTB excitation energies
 - singlet-singlet and singlet-triplet excitations
 - intensity selection of excitation energies
- constrained optimizations
- QUASINANO2013.1 DFTB parameters for almost the whole periodic table (only electronic part)
- Density matrix purification method during SCC using sparse matrix algebra (most useful for very large calculations of insulators or semi-conductors).

DFTB-GUI

Note that the graphical user interface DFTB-GUI enables all users to set up complicated calculations with a few mouse clicks, and provides graphical representations of calculated data fields, see the [DFTB-GUI tutorials](#).

Input

The input for DFTB slightly differs from the one found in ADF. General nomenclature and structure are however similar to those found in ADF. In the following sections, a list of the relevant keys and the contained sub-keys will be presented.

After the run, results of the computation are written to standard output. Binary information about the evaluation are also written to a keyed-file `dftb.rkf`.

Specification of the System

The input of the initial structure can be given with the key `System`. This key is generally mandatory, but for molecular dynamics restarts, it can be omitted. In that case, the molecular configuration of the last recorded iteration will be used.

```
System
  {Atoms
    Atom Coords
  End}
  {Charge NetQ}
  {Lattice
    Vectors
  End}

  {LatticeStrain
    eps1 value
    eps2 value
    eps3 value
    eps4 value
    eps5 value
    eps6 value
  }
  {FractionalCoords}
End
```

The `System` key accepts a set of sub-keys to specify various details of the chemical system under evaluation

Atoms

Specifies the geometry of the molecular system as a list of rows, one row per atom.

Atom

The name of an *atom type*. It must be the standard one- or two-characters symbol for the chemical element: H, He, Li, and so on.

Coords

This specifies the coordinates of the atom. The x, y, z values of the Cartesian coordinates are by default interpreted in Ångstrom.

Charge

The net charge of the molecule can be controlled with the optional sub-key `CHARGE`. If this sub-key is omitted the net total charge of the molecule is by default zero.

NetQ

The net total charge of the molecule.

Lattice

Information about the periodicity of the system is given through this sub-key. Its presence is optional, and it implies a periodic system. The subsequent computation will therefore evaluate the system accordingly. A list of up to three vectors (one per row) for the cell must be specified.

Vector

Three floating point values defining the periodicity vector along a given direction. One, two, or three vectors can be specified (each on a different row) to express linear, planar or bulk periodicity, respectively. If one vector is specified, periodicity must develop along the x axis. If two vectors are specified, periodicity must develop along the xy plane. The unit is the same of the Atoms section.

LatticeStrain

Allows the application of a strain tensor to the lattice. The values of eps1 to eps6 represent the unique elements of the strain tensor, as follows

```
eps1 eps6 eps5
eps6 eps2 eps4
eps5 eps4 eps3
```

FractionalCoords

This optional keyword modifies how the ATOMS coordinates are interpreted. When the keyword is present, coordinates will be interpreted as fractions of the periodicity lattice vectors, instead of absolute geometric positions in 3D space. Necessarily, the presence of this sub-key requires LATTICE to be specified.

Specification of the computational Task

The Task section is mandatory and allows to specify the computational task to perform. It accepts only one mandatory sub-key, runType.

```
Task
  runType type
End
```

type

The type of evaluation to perform.

- It can be:
- SinglePoint or SP
 - GeometryOptimization or GO
 - TransitionState or TS
 - Frequencies or F
 - MolecularDynamics or MD
 - Phonons

The Properties section is optional and allows to specify the property to calculate. At the moment the only property is Excitations, in which case the runType should be SinglePoint

```

Task
  RunType SP
END
Properties
  Excitations
End

```

Changing the default Units

The Units key is optional. It allows to specify different units for Length, Angle, and Time, in place of the default ones.

```

Units
  {length    angstrom|bohr}
  {angle     degree|radian}
  {time      femtosec|au}
End

```

The default values are angstrom, degree, and femtosec.

Setting DFTB Calculation details

```

DFTB
  ResourcesDir relativepath
  {RadialExtrapolation none|linear|improved|original|bezier}
  {SCC
    {iterations NIter}
    {thirdorder}
    {converge charge=QDiff}
    {mixing Mix}
  End}
  {Purify tol=tol}
  {SparsityThreshold sparsitythreshold}
  {CPBO
    {iterations NIter}
    {converge charge=QDiff energy=EDiff}
    {weight min=minWeight max=maxWeight initial=initialWeight}
  End}
  {CPMD
    {iterations NIter}
    {timestep timestep}
    {orbitalsMass orbMass}
    {converge charge=QDiff}
  End}
  {UseSymmetry yes|no}
  {Repulsion
    forcePolynomial
  End}
  {Dispersion
    method UFF|D2|D3-BJ|D3-zero
  End}
  {Occupation aufbau|fermi {temperature=FermiTemp}}
End

```

This mandatory key allows to specify and control different aspects of the DFTB evaluation engine.

ResourcesDir

Allows to specify the path (relative to \$ADFRESOURCES/DFTB) of the directory containing DFTB parameter files. Different parameters may be suitable for different DFTB evaluations. It is important to choose the appropriate parameter set for the type of calculation and molecular system under study. Alternatively, an absolute path (starting with a slash character) can be specified to have complete freedom over the location of the directory. Examples:

```
ResourcesDir Dresden
```

Uses the Resource directory \$ADFRESOURCES/DFTB/Dresden

```
ResourcesDir /home/myusername/myskfdir
```

Uses the specified path /home/myusername/myskfdir as the resource directory

NOTE: each resource directory should contain a file called *metainfo.yaml*, which defines extra information (Hubbard derivatives, dispersion parameters, etc.) required by DFTB calculation. For details see [metainfo.yaml](#).

RadialExtrapolation

Advanced control option. Overrides the extrapolation method for Slater-Koster grid values between the end of the tabulated grid and the cutoff distance (value for which atoms are considered too far to interact). Depending on the structure of your Slater-Koster tables, a different radial extrapolation method may be needed in order to guarantee correct behavior, in particular for large and periodic systems. Five different extrapolation strategies are available:

```
none
```

Performs no extrapolation, the value being forced to zero at distances greater than the grid last position.

```
linear
```

performs a linear interpolation between the last point of the grid and the value of zero, at cutoff distance.

```
improved (default)
```

Perform a 9th grade polynomial interpolation between 6 points of the grid and three zeros. This interpolation may prove unstable for particular Slater-Koster data

```
original
```

same as improved, but reproducing behavior of previous reference programs. Should not be used in general.

```
bezier
```

Uses a Bézier curve passing through the last grid point and the cutoff point, guaranteeing continuity and smoothness. This is the suggested method in case of unexpected behavior.

SCC

The SCC key is optional. By its presence the SCC-DFTB model is used, where the self-consistent-charge (SCC) iterative procedure is performed. If this key is not present, DFTB will perform a non-SCC

evaluation (standard DFTB). Subkeys are optional for setting up the SCC details and choosing the SCC level:

`thirdorder`

This option turns on the DFTB3 model, which applies a third-order correction in addition to SCC-DFTB.

`iterations NIter`

Allows to specify the maximum number of SCC iterations. Default is 100 iterations, which is a very high number. Most computations will converge in less iterations. Lack of convergence within this limit may be due to use of aufbau Occupation. Fermi occupation may improve convergence. See the Occupation key below.

`converge charge=QDiff`

Specifies the tolerance for convergence on the variation of the atomic charges. The default is 1.0e-8.

`mixing Mix`

Mix is the Broyden mixing parameter, which should be real number between 0 and 1. The default value of 0.2 should in general be fine, but decreasing/increasing the mixing can sometimes help with SCC convergence problems.

`Purify tol=tol`

By default (when this key is not present), the next step density matrix is calculated from molecular orbitals obtained as eigenvectors of the charge-dependent Hamiltonian. An alternative way to obtain the density matrix is using an iterative purification procedure enabled by this keyword. The *tol* parameter defines the purification convergence threshold. Purification is considered converged when the trace of the density matrix becomes equal to the total number of electrons within *tol*. The default value is 1.e-8.

`SparsityThreshold sparsitythreshold`

The *sparsitythreshold* parameter defines the threshold below which small density matrix components are discarded after each matrix-matrix multiplication. The default value is 1.e-8.

`CPBO`

This key is mutually exclusive with the SCC and CPMD keys. Its presence enables the Car-Parrinello Born Oppenheimer method to perform optimization at the Self-Consistent Charge level (second order only). Details of the methodology can be found in the references section. The procedure is initiated via a single non-SCC evaluation. It is important to note that this methodology requires a aufbau occupation scheme. Additional optional keys can be used to control the procedure

`iterations NIter`

Allows to specify the maximum number of Car-Parrinello iterations to converge the atomic charges and energy. Default is 500 iterations. A Car-Parrinello evaluation may require a high number of iterations than SCC, but the individual iteration will be faster.

`converge charge=QDiff energy=EDiff`

Specifies the tolerance for convergence on the variation of the atomic charges. The default is 1.0e-7 for both charges and energy, in their respective units.

`weight min=minWeight max=maxWeight initial=initialWeight`

Specifies the weight coefficient for the internal Verlet algorithm used to propagate the orbitals according to the Car-Parrinello method. Default values are min=0.01, max=0.5, initial=0.5. The weight determines how strongly the orbitals change after each iteration. A small value implies a small variation, while a larger one implies a large one. The algorithm adjusts the weight for optimal convergence, increasing the current value at each iteration if the process is too slow, or decreasing it if the minimum is closer to the current value and the next step would increment the energy. The current weight is always clamped between the min and max values. The initial value specifies the value for the first Car-Parrinello iteration.

CPMD

This key is mutually exclusive with the SCC and CPBO keys. Its presence enables the Car-Parrinello Molecular Dynamics method to propagate optimized orbitals during a molecular dynamics evaluation. As a consequence, CPMD cannot be used with the runType is anything else than MD. The procedure is initiated via two CPBO evaluations at successive geometries. It is important to note that this methodology requires a aufbau occupation scheme. Additional optional keys can be used to control the procedure

iterations NIter

Specifies the maximum number of iterations allowed during the initial two Car Parrinello Born Oppenheimer steps to initialize the CPMD procedure. The defaults are the same as for CPBO.

converge charge=QDiff energy=EDiff

Specifies the tolerance for convergence of the initial two Car Parrinello Born Oppenheimer steps to initialize the CPMD procedure. The defaults are the same as for CPBO.

weight min=minWeight max=maxWeight initial=initialWeight

Specifies the weight coefficients for the initial two Car Parrinello Born Oppenheimer steps to initialize the CPMD procedure. The defaults are the same as for CPBO.

orbitalsMass mass

Specifies the fictitious orbitals mass for the Car Parrinello MD propagation method. The default is $25 \times (\text{timestep})^2$. A small mass allows the orbitals to respond quickly to changes in geometry, but may confer excessive kinetic energy to the orbitals. A larger mass reduces this energy transfer but the orbitals have lower response to geometry change, requiring a smaller MD timestep.

timestep tstep

Specifies the timestep used for propagation of the orbitals. The default is the timestep as specified in the molecular dynamics (MD) section. It is generally required for the stability of the method for these two values to be the same.

UseSymmetry yes|no

Enables or disables the use of symmetry in systems with periodic boundary conditions. Not used in molecular calculations. The key is optional, and in its absence the default is yes. For MD calculations, NO symmetry will be used.

Repulsion

This key allows to specify some details about the repulsion contribution evaluation. It accepts only one sub-key "forcePolynomial", which forces the use of the polynomial representation (as defined in the header of the Slater-Koster parameter files), in place of the Spline description.

Dispersion

This key allows to specify options for the London dispersion correction. If it does not exist, no dispersion correction will be included.

```
method UFF|ULG|D2|D3-BJ
```

This subkey is used to specify a dispersion model. Please refer to the literature for details on the different methods:

- UFF: L. Zhechkov et al., *J. Chem. Theory Comput.*, **2005**, 1 (5), pp 841-847
- ULG: H. Kim et al., *J. Phys. Chem. Lett.*, **2012**, 3 (3), pp 360-363
- D2: S. Grimme, *J. Comput. Chem.*, **2006**, 27: 1787-1799
- D3-BJ: S. Grimme et al., *J. Comput. Chem.*, **2011**, 32: 1456-1465

If no method specified in the `Dispersion` block, a default dispersion correction defined in the `metainfo.yaml` file (depends on "ResourcesDir", see `metainfo.yaml`) will be used. Note that not all dispersion models are supported by all parameter sets. A list of supported models can be found in the parameter's `metainfo.yaml` file.

```
d3parameters s6=S6Param s8=S8Param a1=A1Param a2=A2Param
```

User-customized D3-BJ parameters can be specified with this subkey. Otherwise, the default parameters defined in `metainfo.yaml` are used. Note that all four parameters have to be specified if the `d3parameters` keyword is used.

Occupation

This optional key allows to specify the fill strategy to use for the orbitals. It can be either "aufbau", to fill the orbitals according to the Hund's rule, or "fermi", to perform electronic charge distribution over the orbitals. If "fermi" is specified, a further "temperature" option must be present, specifying the Fermi temperature in Kelvin (K). If this key is absent, the default is Fermi occupation with a temperature of 5 K. This option cannot be used with the Car Parrinello methods, requiring aufbau occupation.

Geometry optimization

```
Geometry
{Method      quasinewton|conjgrads}
{CGType
fletcherreeves|polakribiere|hestenesstiefel|perry|birginmartinez|
scaled|adaptedscaled1|adaptedscaled2|swartdyn1|swartdyn2}
{SwartIter   Niter}
{Optim       Cartesian|Delocal|Primitive|Internal}
{Iterations  Niter}
{Converge    {E=TolE} {Grad=TolG} {Rad=TolR}}
{Step        {TrustRadius=MaxRadius}}
{OptimizeLattice}
End
```

Geometry allows to specify information about the geometry optimization strategy. The keyword must be specified only for those Task runTypes requiring a geometry optimization (GeometryOptimization and TransitionState).

Method

```
quasinewton|conjgrads
```

Selects the optimization algorithm to use, between Quasi-Newton (Hessian-based) and Conjugate Gradients (gradient-based). The default is "quasinewton".

CGType

Defines the methodology to compute the β_n value to update the conjugate direction. This keyword is relevant only if the Method is conjgrads. By default, the Fletcher Reeves methodology is used.

fletcherreeves

Uses the Fletcher-Reeves formula $\beta_n = \frac{\Delta x_n^{\text{top}} \Delta x_n}{\Delta x_{n-1}^{\text{top}} \Delta x_{n-1}}$

polakribiere

Uses the Polak-Ribiere formula $\beta_n = \frac{\Delta x_n^{\text{top}} (\Delta x_n - \Delta x_{n-1})}{\Delta x_{n-1}^{\text{top}} \Delta x_{n-1}}$

hestenesstiefel

Uses the Hestenes-Stiefel formula $\beta_n = -\frac{\Delta x_n^{\text{top}} (\Delta x_n - \Delta x_{n-1})}{s_{n-1}^{\text{top}} (\Delta x_n - \Delta x_{n-1})}$

perry

Uses the Perry formula $\beta_n = -\frac{\Delta x_n^{\text{top}} (\Delta x_n - \Delta x_{n-1})}{s_{n-1}^{\text{top}} (\Delta x_n - \Delta x_{n-1})}$

birginmartinez

Similar to Perry, with a scaled $(\Delta x_n - \Delta x_{n-1})$ at the numerator by $\theta = \frac{s_{n-1}^{\text{top}} s_{n-1}}{s_{n-1}^{\text{top}} (\Delta x_n - \Delta x_{n-1})}$

scaled

Uses $\beta_n^{\text{Scaled}} = -\frac{\Delta x_n^{\text{top}} \Delta x_n - \Delta x_{n-1}^{\text{top}} \Delta x_n}{s_{n-1}^{\text{top}} \Delta x_{n-1}}$

adaptedscaled1

Same as scaled, with $\beta = \max\left(\{1 - \beta_n^{\text{Scaled}}, \beta_n^{\text{Scaled}}\}\right)$

adaptedscaled2

Same as scaled, with $\beta = 1 - \beta_n^{\text{Scaled}}$

swartdyn1

Uses adaptedscaled1 up to a given number of steps (specified as SwartIter), then uses Polak-Ribiere

swartdyn2

Uses adaptedscaled2 up to a given number of steps (specified as SwartIter), then uses Polak-Ribiere

SwartIter

Defines the number of steps required for the swartdyn CGTypes to switch methodology. Only relevant when the proper CGType is used. Default is 300

Optim

cartesian|delocal|primitive|internal

Optimization in delocalized coordinates (Delocal) can only be used in geometry optimizations or transition state searches with the quasi-newton method.

Iterations

Niter

The maximum number of geometry iterations allowed to locate the desired structure. The default is 50.

This is a fairly large number. If the geometry has not converged (at least to a reasonable extent) within that many iterations, there may be an underlying cause to consider, instead of simply increasing the allowed number of cycles.

Converge

Convergence is monitored for two items: the energy and the Cartesian gradients. Convergence criteria can be specified separately for each of these items:

TolE

The criterion for changes in the energy, in Hartrees. Default: 1e-5.

TolG

Applies to gradients, in Hartree/Ångstrom. Default: 1e-3.

TolR

The maximum Cartesian step allowed for a converged geometry, in Ångstrom. Default: 0.001 Ångstrom.

Step

Controls that changes in geometry from one cycle to another are not too large:

MaxRadius

By default, the trust radius is set to 0.2. Using the key, the user can override this, setting a constant value. A conservative value is 0.2. A large system (e.g., 100 atoms) typically needs a larger trust radius (e.g., 0.8).

OptimizeLattice

Enables optimization of the Lattice parameters, in addition to the molecular geometry. This can only be applied to periodic systems.

Constrained optimization

The CONSTRAINTS keyword allows geometry optimizations with constraints for the distance between two atoms, an angle defined by three atoms, or a dihedral angle defined by four atoms:

```
CONSTRAINTS
  DIST Ia1 Ia2 Ra
  ANGLE Ib1 Ib2 Ib3 Rb
```

```
DIHED Ic1 Ic2 Ic3 Ic4 Rc
end
```

The *DIST*, *ANGLE*, and *DIHED* constraints do not have to be satisfied at the start of the geometry optimization.

DIST

When *DIST* is specified, the distance between atoms Ia1 and Ia2 is constrained to the value Ra in Angstrom.

ANGLE

When *ANGLE* is specified, the angle between atoms Ib1, Ib2 and Ib3 (Ib1-Ib2-Ib3) is constrained to the value Rb in degrees.

DIHED

When *DIHED* is specified, the dihedral angle between atoms Ic1, Ic2, Ic3 and Ic4 (Ic1-Ic2-Ic3-Ic4) is restrained to the value Rc in degrees. The dihedral angle is projected onto the $[0, 2\pi]$ interval, so there should be no difference between specifying -30° or 330° .

TDDFTB excited states

DFTB now allows excited state calculations for molecular systems using single orbital transitions as well as time-dependent DFTB as published by Niehaus et al. in Phys. Rev. B 63, 085108 (2001). Singlet-singlet as well as singlet-triplet excitations can be calculated. A filter can be used to reduce computational costs, for example by using only single orbital transitions that have a minimal oscillator strength.

The TDDFTB implementation uses the PRIMME library (PREconditioned Iterative MultiMethod Eigensolver) by Andreas Stathopoulos and James R. McCombs, [PRIMME: PREconditioned Iterative MultiMethod Eigensolver: Methods and software description ACM Transaction on Mathematical Software Vol. 37, No. 2, \(2010\), 21:1–21:30.](#)

DFTB excited state calculations are controlled by the following keywords:

```
Properties
  Excitations
    {SingleOrbTrans
      {maxnum n}
      {Filter
        {dEMin r}
        {dEMax r}
        {OSMin r}
      }
    }
    {printlowest n}
  End}
{TDDFTB
  {calc [ singlet triplet ]}
  {lowest n}
  {upto r}
  {diagonalization exact|davidson|auto}
  {DavidsonConfig
    {ATCharges onthefly|precalc}
    {preconditioner exactdiag|approxdiag}
    {tolerance r}
    {maxBlockSize n}
  }
```

```

        {minRestartSize n}
        {maxBasisSize n}
    End}
    {print [ eigenvectors evcontributes omegamatrix omegadiagonal ]}
End}
End
End

```

SingleOrbTrans

The simplest approximation to the true excitations are the single orbital transitions (sometimes called Kohn-Sham transitions), that is transitions where a single electron is excited from an occupied Kohn-Sham orbital into a virtual orbital. The calculation of these transitions is configured in the `SingleOrbTrans` section. Note that the `SingleOrbTrans` section is optional even though the single orbital transitions are also needed for TDDFTB calculations. If the section is not present all single orbital transitions will still be calculated and used in a subsequent TDDFTB calculation, but no output will be produced.

`maxnum n`

The maximum number of calculated single orbital transitions. If the total number of single orbital transitions (that is the number of occupied orbitals times the number of virtual orbitals) is larger than this, the high energy end of the spectrum will be truncated. Accepts a single integer. The default is infinity, so that all possible transitions will be considered.

`Filter`

The `Filter` section allows to remove single orbital transitions based on certain criteria. Note that the filter is applied after the single orbital transition spectrum has been truncated via the `maxnum` keyword. The final number of single orbital transitions can therefore be less than `maxnum` if a filter is applied. All filters are disabled by default.

`dEMin r`

Removes single orbital transitions with an orbital energy difference smaller than `dEMin`. Accepts the minimum energy difference in Hartree as a single number.

`dEMax r`

Removes single orbital transitions with an orbital energy difference larger than `dEMax`. Accepts the maximum energy difference in Hartree as a single number.

`OSMin r`

Removes single orbital transitions with an oscillator strength smaller than `OSMin`.

`printlowest n`

The number of single orbital transitions that are printed to the screen and written to disk. Accepts a single integer. If the `TDDFTB` section does not exist, the default is to print the 10 lowest single orbital transitions. If it does exist it is assumed that the single orbital transitions are only used as an input for TDDFTB and nothing will be printed unless `printlowest` is specified explicitly.

TDDFTB

Calculations with time-dependent DFTB can be configured in the `TDDFTB` section and should in general give better results than the raw single orbital transitions. TDDFTB calculates the excitations in the basis of the single orbital transitions, whose calculation is configured in the `SingleOrbTrans` section. Using

truncation or a filter in `SingleOrbTrans` can therefore be used to reduce the size of the basis for TDDFTB. One possible application of this is to accelerate the calculation of electronic absorption spectra by removing single orbital transitions with small oscillator strengths from the basis. Note that the entire TDDFTB section is optional. If no TDDFTB section is found, the behaviour depends on the existence of the `SingleOrbTrans` section: If no `SingleOrbTrans` section is found (the `Excitations` section is completely empty then) a TDDFTB calculation with default parameters will be performed. If only the `SingleOrbTrans` section is present no TDDFTB calculation will be done.

`calc [singlet triplet]`

Specifies the multiplicity of the excitations to be calculated. Accepts the keys `singlet` and/or `triplet` separated by a space. The default is to calculate both singlet-singlet and singlet-triplet excitations.

`lowest n`

Specifies the number of excitations that are calculated. Accepts a single integer. The default is to calculate the 10 lowest excitations. Note that in case of the exact diagonalization all excitations are calculated, but only the lowest ones are printed to screen and written to the output file.

`upto r`

Attempts to calculate all excitations up to a given energy by calculating a number of excitations equal to the number of single orbital transitions in this window. This is only approximately correct, so one should always add some safety margin. Accepts a single real number which is the maximum excitation energy in Hartree. Note that if both `lowest` and `upto` are specified, DFTB will always use whatever results in the smaller number of calculated excitations.

`diagonalization exact|davidson|auto`

Specifies the method used to solve the TDDFTB eigenvalue equation. The most straightforward procedure is a direct diagonalization of the matrix from which the excitation energies and oscillator strengths are obtained. Since the matrix grows quickly with system size (number of used single orbital transitions squared), this option is possible only for small molecules. The alternative is the iterative Davidson method, which finds a few of the lowest excitations within an error tolerance without ever storing the full matrix. The default is to make this decision automatically based on the system size and the requested number of excitations.

`DavidsonConfig`

The `DavidsonConfig` section contains a number of keywords that can be used to override various internals of the Davidson eigensolver. The default values should generally be fine.

`ATCharges onthefly|precalc`

Controls whether the atomic transition charges are precalculated in advance or reevaluated during the iterations of the Davidson solver. Precalculating the charges will improve the performance, but requires additional storage. The default is to precalculate the atomic transition charges, but the precalculation can be disabled if not enough memory is available.

`preconditioner approxdiag|exactdiag`

Sets the elements of the diagonal matrix used as a preconditioner for the trial vectors. Using the exact diagonal of the Omega matrix is generally not recommended as it is slightly slower and does not speed up convergence. The default is to use the single orbital transition energies as an approximation to the diagonal of Omega.

tolerance r

Convergence criterion for the norm of the residual. Accepts a single real number. The default is 1e-5.

maxBlockSize n

Maximum number of trial vector multiplications done at once. Both too small and too large values might result in suboptimal performance. Accepts a single integer. The default is to use the number of requested excitations, but no more than 64.

minRestartSize n

Minimum number of retained Ritz vectors upon basis collapse. Accepts a single integer. The default is to retain 4 times maxBlockSize.

maxBasisSize n

Maximum number of basis vectors before the basis is collapsed. Larger values will decrease the number of required matrix-vector multiplications but require more memory and make the orthonormalization more costly. Accepts a single integer. The default is to use 10 times maxBlockSize.

print [eigenvectors evcontribs omegamatrix omegadiagonal]

Specifies whether to print the full eigenvectors, the contributions of the single orbital transitions to the excitations, the entire Omega matrix and/or its diagonal elements. The default is to print nothing. Note that the full Omega matrix cannot be printed when using Davidson diagonalization. Note also that for Davidson diagonalization the printed diagonal is the one used as the preconditioner, which is only the true diagonal of Omega if the exact preconditioner was selected in DavidsonConfig.

Restart

```
Restart
  {RestartFile}
  {RestartMulliken}
  {RestartOrbitals}
End
```

Molecular Dynamics

```
MD
Steps      NSteps
TimeStep   TStep
{Restart   file=path}
{Checkpoint frequency=ChkFreq}
{Trajectory samplingFreq=SFreq}
{Preserve  [TotalMomentum AngularMomentum CenterOfMass All None]}
{InitialVelocities zero|inline|random {temperature=InitTemp}}
{InlineVelocities
  velocityVector
End}
{Thermostat type=ThermoType {thermostat options}}
End
```


The DFTB program supports molecular dynamics (with Velocity Verlet) with and without thermostats. This key, used with Task runType is set to MD, allows to specify the information needed by the molecular dynamics evaluation. This implementation of MD supports periodic systems.

Steps NSteps

Specifies the number of steps to be taken in the MD simulations. It accepts a simple integer number NSteps.

TimeStep TStep

Specifies the time for each step. By default, the unit is femtoseconds. Through the Units key, it can be changed to atomic units of time.

Restart file=path

Triggers a restart procedure, recovering the latest known information from the specified file (either a final .rkf file, or a checkpoint .chk file). When this keyword is present, System, Velocity, previous average values and energy transfers will be recovered from the file, ignoring any redundant specification made in the input file. This is the only situation where the System keyword can be omitted.

Checkpoint frequency=ChkFreq

Sets the frequency (in steps) for checkpoint the current status to a file. This allows to restart from an intermediate configuration in case of a crash of the program or the system. The keyword is optional; if not specified, by default is equal to the number of steps divided by 4. Only the most recent checkpoint is preserved. In case of crash, the checkpoint may be found in the execution temporary directory, instead of the working path. Checkpoint files can be inspected with the GUI for the latest configuration.

Trajectory samplingFreq=SFreq

Sets the frequency for printing to stdout and storing the molecular configuration on the .rkf file. This keyword is optional, and the default is the number of steps divided by 1000 (minimum one).

Preserve [TotalMomentum AngularMomentum CenterOfMass All None]

Constrains the molecular dynamics simulation to preserve different whole-system parameters. Note that this option has poor meaning for periodic systems. The keys can be given as a sequence out of the allowed list, with words separated by spaces

TotalMomentum

removes the overall velocity of the system from the atomic velocities.

AngularMomentum

removes the overall angular velocity of the system from the atomic velocities.

CenterOfMass

keeps the molecular system centered on the current center of mass.

All

Specifying "All" is equivalent of specifying all of the above keywords

None

None of the above options will be enabled. This is the default setup if the Preserve keyword is not specified.

`InitialVelocities zero|inline|random {temperature=InitTemp}`

Specifies the initial velocities to assign to the atoms. Three methods to assign velocities are available

`zero`

All atom's velocities are set to zero

`inline`

Atom's velocities are set to the values specified in the key `InlineVelocities` (see below)

`random temperature=InitTemp`

Atom's velocities are set to random values according to the specified temperature `InitTemp`, in kelvin. The temperature keyword is mandatory for this choice.

`InlineVelocities`

This optional key is read when `InitialVelocities inline` option is used. It allows to specify the velocities for each atom. Each row must contain three floating point values (corresponding to the x,y,z component of the velocity vector) and a number of rows equal to the number of atoms must be present, given in the same order as the System Atoms specification.

Available Thermostats

The key `Thermostat` allows to specify the use of a thermostat during the simulation. Depending on the selected thermostat type, different additional options may be needed to characterize the specific thermostat behavior. At the moment, the following choices for the type parameter are available

`None`

No thermostat applied. This is the default if no `Thermostat` key is present.

`Scale`

Applies a scaling of the velocities in agreement to the specified temperature. The following options are required for this thermostat

`frequency=NSteps`

This parameter is optional. If specified, the thermostat will be applied every `NSteps`, using that step's ensemble temperature and the specified thermostat temperature to compute the scaling factor. If not specified, the thermostat will be applied at every step, using the mean temperature of the ensemble and the specified thermostat temperature to compute the scaling factor.

`temperature=Temp`

Specifies the temperature of the thermostat, in kelvin. This parameter is mandatory.

`Berendsen`

Applies the Berendsen thermostat. The following options are required for this thermostat

`tau`

Specifies the initial tau parameter for the Berendsen thermostat, in femtoseconds (can be changed via Units key).

```
apply=local|global
```

Defines the scope of application of the scaling correction, either per-atom-velocity (option local) or on the molecular system as a whole (option global)

```
temperature=Temp
```

Specifies the temperature of the thermostat, in kelvin. This parameter is mandatory.

Additional Periodicity Data

```
Periodic
  {KSpace NK}
  {BZStruct {enabled=yes|no} {automatic=yes|no} {interpol=intVal}}
  {Phonon reorderatoms=yes|no interpol=N rcelx=N stepsize=N}
  {BZPath
    KMesh NMesh
    {Path
      End}
  End}
  {SuperCell
  End}
  {LatticeStepSize}
  {stressTensor}
  {Screening}
End
```

KSpace

This parameter controls the number of k-points used in the calculation. For very small unit cells (one atom wide) a value of 5 is advised. For medium sized unit cells 3 is adequate. For very large ones (10 atoms wide) kspace=1 suffices. (Default is 5)

BZStruct

This controls the path taken through the Brillouin zone (for plotting purposes). It has no effect on the calculated energy. Specifying *intVal* the band structure is interpolated along the path (so extra k-points are generated). You can also specify a path by hand, setting *automatic* to *no*. The points in the path should be entered in the BZPath key.

enabled

By default this feature is enabled.

automatic

Whether or not to use the automatic path generation.

interpol

Level of interpolation to use along the path

Phonon

This enables a phonon run. One should start from a completely optimized system. Next one should choose a super cell. The phonon spectrum converges with super cell size. How big it should be depends on the system.

`reorderatoms`

Technical option: put atoms of the same type after each other.

`stepsize`

Step size to be taken to obtain the force constants (second derivative) from the analytical gradients.

`BZPath`

Allows the user to specify manually a path through the BZ. The points are in terms of reciprocal lattice vectors.

`KMesh`

The amount of points on each line segment

`Path`

Each path consists of a number of points, which are assumed to be connected.

`SuperCell`

Used for the phonon run. The super lattice is expressed in the lattice vectors. Most people will find a diagonal matrix easiest to understand.

Timing details

For developers: one can get timing details to see which part of the code takes most time. Default value is `None`.

```
| Print  
|   timers {None | Normal | Detail | TooMuchDetail}  
| End
```

Examples

The \$ADFBHOME/examples/dftb directory contains many different example files, covering various DFTB options.

The run script below is for the geometry optimization of aspirin at the SCC-DFTB level:

```
$ADFBIN/dftb << eor

Task
  RunType GO
End

System
  Atoms
    C      0.000000  0.000000  0.000000
    C      1.402231  0.000000  0.000000
    C      2.091015  1.220378  0.000000
    C      1.373539  2.425321  0.004387
    C     -0.034554  2.451759  0.016301
    C     -0.711248  1.213529  0.005497
    O     -0.709522  3.637718  0.019949
    C     -2.141910  1.166077 -0.004384
    O     -2.727881  2.161939 -0.690916
    C     -0.730162  4.530447  1.037168
    C     -0.066705  4.031914  2.307663
    H     -0.531323 -0.967191 -0.007490
    H      1.959047 -0.952181 -0.004252
    H      3.194073  1.231720 -0.005862
    H      1.933090  3.376356 -0.002746
    O     -2.795018  0.309504  0.548870
    H     -2.174822  2.832497 -1.125018
    O     -1.263773  5.613383  0.944221
    H     -0.337334  4.693941  3.161150
    H      1.041646  4.053111  2.214199
    H     -0.405932  3.005321  2.572927
  End
End

DFTB
  ResourcesDir Dresden
  SCC
    iterations 200
    converge charge=1.0e-8
  End
End

Geometry
  iterations 100
End

eor
```

Parameter files

The so called Dresden set of DFTB parameter files available in the ADF package were designed by J. Frenzel, A.F. Oliveira, N. Jardillier, T. Heine, and G. Seifert, mainly at the Technische Universität in Dresden, Germany, see also some [additional information about the generation of these parameter files](#). These parameter files are kept in the directory \$ADFHOME/atomicdata/DFTB/Dresden.

You can also use a different set of parameter files. Note that different sets of parameter files are often not compatible. Note also that often parameter files were designed for a specific purpose, which may be different than your application, and therefore may give not the desired accuracy.

The QUASINANO2013.1 set of DFTB parameter files available in the ADF package are designed by Mohammad Wahiduzzaman et al. contains parameters for a large part of the periodic table (no f-elements). Note that the QUASINANO2013.1 set only contains the electronic part of the interaction, so that only the spectrum for a given geometry can be calculated, but no total energy, and thus also no forces. These parameters can be used in TDDFTB calculations, for example. Additional licensing requirements may however be needed to access the content of the files. Please contact our licensing department to evaluate the available options.

The DFTB implementation shipped by SCM provides the most up-to-date parameter sets available on the DFTB.org website. Additional licensing requirements may however be needed to access the content of the files. Please contact our licensing department to evaluate the available options.

The following sets are currently shipped

- mio-0-1 and mio-1-1 (H, C, N, O, S, P): for organic molecules
- pbc-0-3 (Si, F, O, N, C, H, Fe): for solid and surfaces
- matsci-0-3 (Al, Si, Cu, Na, Ti, Ba): for various compounds in material science

In addition, the following extension sets are provided to the mio set (either version 0-1 or 1-1):

- hyb-0-1 (Ag, Ga, As, Si) + mio-0-1: for organic and inorganic hybrid systems
- chalc-0-1 (As, S) + mio-0-1: for Chalcogenide glasses
- miomod-hh-0-1 + mio-1-1: contains a modified parameter set for H2
- miomod-nh-0-1 + mio-1-1: contains a modified parameter set for N-H to improve N-H binding energies
- tiorg-0-1 (Ti-(C, H, N, O, S, Ti)) + mio-0-1 : for Ti bulk, TiO2 bulk, TiO2 surfaces, and TiO2 with organic molecules
- trans3d-0-1 (Sc, Ti, Fe, Co, Ni) + mio-0-1: Transition metal elements for biological systems
- znorg-0-1 (Zn-(C, H, N, O, S, Zn)) + mio-0-1: for Zn bulk, ZnO bulk, ZnO surfaces, and ZnO with organic molecules

We recommend to visit the [DFTB.org](#) web site for more detailed information about each set. Please note that our implementation of DFTB does not support parameter sets files containing f-functions, such as the "rare" set.

Installing additional DFTB.org parameter files

To install new parameter sets released by the DFTB.org website in the future, we recommend the following procedure

1. Unpack the tar.gz file containing the parameters (for example, newset-0-1.tar.gz) with the command `tar -C $ADFPRESOURCES/DFTB/DFTB.org -xzvf newset-0-1.tar.gz`.
2. Make sure the files have the name in the format X-Y.skf, with X and Y element symbols (for example, C-C.skf, C-H.skf, Al-H.skf). If this is not the case, rename the files to follow this naming.

3. Take note of the new directory name created in \$ADFRESOURCES/DFTB/DFTB.org while unpacking (for example, newset-0-1)

The new parameter set can be now specified with the key ResourcesDir

```
| ResourcesDir DFTB.org/newset-0-1
```

Third Order parameter files

The parameter files for third-order evaluation are available under a separate license agreement. Contact our licensing department for more information. Third-order parametrization uses the values classified as "DFTB3 fit" in the reference paper (Gaus, Cui, and Elstner). The special parameters for DFTB3 (Hubbard derivatives and Zeta) are defined in file *metainfo.yaml* (see below).

metainfo.yaml

There should be a file named *metainfo.yaml* in each resources directory (see [ResourcesDir](#)), for example *DFTB.org/3ob-1-1/metainfo.yaml*. The file is in accordance with the [YAML](#) syntax convention. It contains necessary information required for DFTB calculation. An example with explanation comments (beginning with #) is shown here:

```
# version info
metainfo_version: 1

# SCC models the current resources directory supports ("thirdorder" is aliased to "dftb3")
# possible models: noscc, scc, thirdorder
supports: [ thirdorder ]

# Hubbard derivatives for DFTB3, see DOI: 10.1021/ct300849w
# note: the list must be {} rather than []
Hubbard_derivatives:
  zeta: 4.000
  parameters: { H: -0.1857, C: -0.1492, N: -0.1535, O: -0.1575, P:
-0.14, S: -0.11 }

# magnetic Hubbard parameters for TD-DFTB, see DOI: 10.1103/PhysRevB.63.085108
# note: the list must be {} rather than []
magnetic_Hubbard:
  parameters: { H: -1.9700, C: -0.6180, N: -0.6940, O: -0.7590 }

# London dispersion correction models
dispersion:
# default method to use if none is specified in the input script
  default: D3-BJ
# list of supported dispersion corrections
  UFF:
  ULG:
  D2:
  D3-BJ:
# D3-BJ dispersion parameters fitted by S. Grimme
# see: http://www.thch.uni-bonn.de/tc/index.php?section=downloads&subsection=DFT-D3
  parameters: { s6: 1.0000, s8: 0.5883, a1: 0.5719, a2: 3.6017 }
```

references

reference: |

M. Gaus, A. Goez, M. Elstner

"Parameterization and Benchmark of DFTB3 for Organic Molecules"

J. Chem. Theory Comput. 2013, 9 (1), pp 338-354.

M. Gaus, X. Lu, M. Elstner, Q. Cui

"Parameterization of DFTB3/3OB for Sulfur and Phosphorus for Chemical and Biological Applications"

J. Chem. Theory Comput. 2014, 10 (4), pp 1518-1537

references in short format

short_reference: |

[O-N-C-H] J. Chem. Theory Comput. 2013, 9, 338-354.

[P-S] J. Chem. Theory Comput. 2014, 10, 1518-1537

web link

url: http://www.dftb.org/parameters/download/3ob/3ob_2_1/

Slater-Koster files: "txtn" indicates encrypted, "txt" indicates unencrypted

format: txtn

References

DFTB: general description

M. Elstner, D. Porezag, G. Jungnickel, J. Elsner, M. Haugk, T. Frauenheim, S. Suhai, and G. Seifert, *Self-consistent charge density functional tight-binding method for simulation of complex material properties*, *Physical Review B* **58**, 7260 (1998)

T. Frauenheim, G. Seifert, M. Elstner, Z. Hajnal, G. Jungnickel, D. Porezag, S. Suhai, and R. Scholz, *A self-consistent charge density-functional based tight-binding method for predictive materials simulations in physics, chemistry and biology*, *Physica Status Solidi (b)* **217**, 41 (2000)

M. Elstner, T. Frauenheim, E. Kaxiras, G. Seifert, and S. Suhai, *A self-consistent charge density-functional based tight-binding scheme for large biomolecules*, *Physica Status Solidi (b)* **217**, 357 (2000)

C. Koehler, G. Seifert, U. Gerstmann, M. Elstner, H. Overhof, and T. Frauenheim, *Approximate density-functional calculations of spin densities in large molecular systems and complex solids*, *Physical Chemistry Chemical Physics* **3**, 5109 (2001)

T. Frauenheim, G. Seifert, M. Elstner, T. Niehaus, C. Kohler, M. Armkrecht, M. Sternberg, Z. Hajnal, A. di Carlo, and S. Suhai, *Atomistic Simulations of complex materials: ground and excited state properties*, *Journal of Physics: Condensed Matter* **14**, 3015 (2002)

M. Gaus, Q. Cui, and M. Elstner, *DFTB3: Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method (SCC-DFTB)*, *Journal of Chemical Theory and Computation* **7**, 931 (2011)

DFTB: parameter sets

Dresden

The DFTB parameter files in \$ADFHOME/atomicdata/DFTB/Dresden are distributed with the ADF package. For more detailed information, see also the README file in the directory \$ADFHOME/atomicdata/DFTB/Dresden.

General reference for the construction of all integral tables in \$ADFHOME/atomicdata/DFTB/Dresden: J. Frenzel, A. F. Oliveira, N. Jardillier, T. Heine, and G. Seifert, *Semi-relativistic, self-consistent charge Slater-Koster tables for density-functional based tight-binding (DFTB) for materials science simulations*, TU-Dresden 2004-2009.

For construction and application of integral tables for Al-O-H:

J. Frenzel, A. F. Oliveira, H. A. Duarte, T. Heine, and G. Seifert, *Structural and electronic properties of bulk gibbsite and gibbsite, surfaces*, *Zeitschrift für Anorganische und Allgemeine Chemie* **631**, 1267 (2005)

For construction and application of integral tables for Al-Si-O-H:

L. Guimarães, A. N. Enyashin, J. Frenzel, T. Heine, H. A. Duarte, and G. Seifert, *Imogolite Nanotubes: Stability, electronic and mechanical properties*, *Nano* **1**, 362 (2007)

For construction and application of integral tables for Al-O-P-C-H:

R. Luschtinetz, A. F. Oliveira, J. Frenzel, J. Joswig, G. Seifert, and H. A. Duarte, *Adsorption of phosphonic and ethylphosphonic acid on aluminum oxide surfaces*, *Surface Science* **602**, 1347 (2008)

For construction and application of integral tables for Ti-O-P-C-H:

R. Luschtinetz, J. Frenzel, T. Milek, and G. Seifert, *Adsorption of phosphonic acid at the TiO₂ anatase (101) and rutile (110) surface*, [Journal of Physical Chemistry C](#) **113**, 5730 (2009)

QUASINANO2013.1

The DFTB parameter files in \$ADFHOME/atomicdata/DFTB/QUASINANO2013.1 are distributed with the ADF package. These are parameters only for the electronic part of the DFTB method that covers almost the complete periodic table (no f-elements). No forces can be calculated. These parameters can be used in TDDFTB calculations, for example.

M. Wahiduzzaman, A.F. Oliveira, P.H.T. Philipsen, L. Zhechkov, E. van Lenthe, H.A. Witek, T. Heine, *DFTB Parameters for the Periodic Table: Part 1, Electronic Structure*, [Journal of Chemical Theory and Computation](#) **9**, 4006 (2013)

DFTB.org

For detailed information please visit the official DFTB webpage: www.dftb.org. Detailed references of each specific parameter set are available in the corresponding *metainfo.yaml* file (See Section 'ResourcesDir' and 'metainfo.yaml' in this manual).

The newest parameter set (H, C, N, and O) for DFTB3 (DFTB with third-order correction to SCC) *DFTB.org/3ob-1-1*:

M. Gaus, A. Goez, and M. Elstner *Parametrization and Benchmark of DFTB3 for Organic Molecules*, [Journal of Chemical Theory and Computation](#) **9**, 338 (2013)

Keywords

CONSTRAINTS 12
DFTB 6
GEOMETRY 10
MD 16

PERIODIC 19
PRINT 20
PROPERTIES 6
RESOURCE DIR 23

RESTART 16
SYSTEM 4
TASK 5
UNITS 6