GFNFF Manual

Amsterdam Modeling Suite 2024.1

www.scm.com

Apr 05, 2024

CONTENTS

CHAPTER

ONE

OVERVIEW

The GFN-FF AMS engine is a generic force-field which can be used for elements up to radon (Z=86). It does not require any external force-field file because all parameters are built into the engine.

The main publication for GFN-FF can be found here: Angew. Chemie Int. Edit. 59, 15665-15673 (2020) (https://doi.org/10.1002/anie.202004239).

The version GFN-FF included in AMS is derived from the Jun 29 2021 version of the master branch of https://github.com/grimme-lab/xtb.git.

1.1 Atom typing

Connectivity, atom typing and atomic charges are automatically determined by the GFN-FF engine based on the geometry of the system (if you provide input bonds, they will **not** be used by GFN-FF).

Note: Since connectivity, atom typing and atomic charges are automatically deduced from the geometry, it is important for the input geometry to be "reasonable". If your initial geometry is a heavily distorted molecule, the automatic atom typing procedure might produce unreasonable atom types, which will lead to poor results.

1.2 Parallelization

The current implementation runs on a single processor core only and it should be able to treat systems up to several thousand atoms. For tasks that can take advantage of the AMS driver-level parallelism, you may want to run AMS in parallel.

Tip: The stress tensor is currently calculated numerically with the GFN-FF engine. Therefore you should definitely run in parallel (ideally with `NSCM=12`) for tasks that require its repeated calculated, such as lattice optimizations or MD calculations with a barostat.

1.3 Minimal input

You can set up GFN-FF calculations from AMSInput (see the GUI tutorials for more info) or you can create a run script:

```
#!/bin/sh

# Since GFN-FF runs on a single CPU only, here we set
# the number of cores used by AMS to 1.
# For tasks that can take advantage of driver-level
# parallelism, you may want to run AMS in parallel.

export NSCM=1


"$AMSBIN/ams" << eor

Task GeometryOptimization

System
    Atoms
        O 0.0  0.0      0.0
        H 0.0 -0.783836 0.554256
        H 0.0  0.783836 0.554256
    End
End

Engine GFNFF
    # Options for the GFN-FF engine can be specified here.
    # No options are required if you want to use the standard settings.
EndEngine
eor
```

A list of all available input options for GFN-FF can be found *here* (page 3).

CHAPTER

TWO

KEYWORDS

2.1 Summary of all keywords

2.1.1 Engine GFNFF

`Accuracy`

Type Float

Default value 1.0

Description Expert option: GFNFF accuracy parameter. Several thresholds within GFNFF depend on this accuracy parameter. Must be a positive number. Smaller values of Accuracy will result in larger distance thresholds.

`ForceFieldFile`

Type String

Default value

GUI name GFNFF parameter file

Description Path to a GFNFF parameter file

`Type`

Type Multiple Choice

Default value AngewChem2020_2

Options [AngewChem2020, AngewChem2020_1, AngewChem2020_2, FromFile]

Description Type of GFNFF to be used

CHAPTER

THREE

REQUIRED CITATIONS

When you publish results in the scientific literature that were obtained with programs of the AMS package, you are required to include references to the program package with the appropriate release number, and a few key publications.

For calculations using GFN-FF: S. Spicher, S. Grimme. *Robust Atomistic Modeling of Materials, Organometallic, and Biochemical Systems*, Angew. Chemie Int. Edit. 59, 15665-15673 (2020) (https://doi.org/10.1002/anie.202004239)

CHAPTER

FOUR

KF OUTPUT FILES

4.1 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'
}
>>> 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.

4.2 Sections and variables on gfnff.rkf

AMSResults **Section content:** Generic results of the ForceField Engine evaluation.

`AMSResults%BondInfo`

Type subsection

Description FIXME: this section should include the file shared/ArchivedBondInfo.json, but there is a problem: the variable 'BondInfo.LatticeDisplacements@dim ('BondInfo.LatticeDisplacements@dim)' is longer than 32 characters (the KF limit) and this messes up thigs. For now I'll just ignore all the variables in here…

`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%Config

Type subsection

Description Configuration of the GFNFF engine.

AMSResults%Config%accuracy

Type float

Description Accuracy parameter of GFNFF.

AMSResults%Config%includeBend

Type bool

Description Whether the following forcefield term was included: bending.

AMSResults%Config%includeBondedATM

Type bool

Description Whether the following forcefield term was included: bonded ATM

AMSResults%Config%includeBonding

Type bool

Description Whether the following forcefield term was included: bond potentials.

AMSResults%Config%includeDispersion

Type bool

Description Whether the following forcefield term was included: dispersion.

AMSResults%Config%includeEField

Type bool

Description Whether the following forcefield term was included: E field.

AMSResults%Config%includeElstat

Type bool

Description Whether the following forcefield term was included: electrostatic.

AMSResults%Config%includeHBonding

Type bool

Description Whether the following forcefield term was included: HB.

AMSResults%Config%includeRepulsion

Type bool

Description Whether the following forcefield term was included: bonded repulsion.

AMSResults%Config%includeSolvation

Type bool

Description Whether the following forcefield term was included: solvation.

`AMSResults%Config%includeTors`

Type bool

Description Whether the following forcefield term was included: torsions.

`AMSResults%Config%includeXBonding`

Type bool

Description Whether the following forcefield term was included: XB.

`AMSResults%Config%version`

Type int

Description Number indicating the version of the gfnff parameters used.

`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%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%Gradients`

Type float_array

Description The nuclear gradients.

Unit hartree/bohr

Shape [3, Molecule%nAtoms]

`AMSResults%Hessian`

Type float_array

Description The Hessian matrix

Unit hartree/bohr^2

Shape [3*Molecule%nAtoms, 3*Molecule%nAtoms]

`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%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%ShearModulus`

Type float

Description The Shear modulus (conversion factor from hartree/bohr^3 to GPa: 29421.026)

Unit hartree/bohr^3

`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

BZcell(primitive cell) **Section content:** The Brillouin zone of the primitive cell.

`BZcell(primitive cell)%boundaries`

Type float_array

Description Normal vectors for the boundaries.

Shape [ndim, nboundaries]

`BZcell(primitive cell)%distances`

Type float_array

Description Distance to the boundaries.

Shape [nboundaries]

`BZcell(primitive cell)%idVerticesPerBound`

Type int_array

Description The indices of the vertices per bound.

Shape [nvertices, nboundaries]

`BZcell(primitive cell)%latticeVectors`

Type float_array

Description The lattice vectors.

Shape [3, :]

`BZcell(primitive cell)%nboundaries`

Type int

Description The nr. of boundaries for the cell.

`BZcell(primitive cell)%ndim`

Type int

Description The nr. of lattice vectors spanning the Wigner-Seitz cell.

`BZcell(primitive cell)%numVerticesPerBound`

Type int_array

Description The nr. of vertices per bound.

Shape [nboundaries]

BZcell(primitive cell)%nvertices

Type int

Description The nr. of vertices of the cell.

BZcell(primitive cell)%vertices

Type float_array

Description The vertices of the bounds.

Unit a.u.

Shape [ndim, nvertices]

DOS_Phonons **Section content:** Phonon Density of States

DOS_Phonons%DeltaE

Type float

Description The energy difference bewteen sampled DOS energies. When there is no DOS at all a certain energy range can be skipped.

Unit hartree

DOS_Phonons%Energies

Type float_array

Description The energies at which the DOS is sampled.

Unit hartree

Shape [nEnergies]

DOS_Phonons%Fermi Energy

Type float

Description The fermi energy.

Unit hartree

DOS_Phonons%IntegrateDeltaE

Type bool

Description If enabled it means that the DOS is integrated over intervals of DeltaE. Sharp delta function like peaks cannot be missed this way.

DOS_Phonons%nEnergies

Type int

Description The nr. of energies to use to sample the DOS.

DOS_Phonons%nSpin

Type int

Description The number of spin components for the DOS.

Possible values [1, 2]

DOS_Phonons%Total DOS

Type float_array

Description The total DOS.

Shape [nEnergies, nSpin]

General **Section content:** General information about the GFNFF calculation.

`General%account`

Type string

Description Name of the account from the license

`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%jobid`

Type int

Description Unique identifier for the job.

`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%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%version`

Type int

Description Version number?

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.

kspace(primitive cell) **Section content:** should not be here!!!

`kspace(primitive cell)%avec`

Type float_array

Description The lattice stored as a 3xnLatticeVectors matrix. Only the ndimk,ndimk part has meaning.

Unit bohr

Shape [3, :]

`kspace(primitive cell)%bvec`

Type float_array

Description The inverse lattice stored as a 3x3 matrix. Only the ndimk,ndimk part has meaning.

Unit 1/bohr

Shape [ndim, ndim]

`kspace(primitive cell)%kt`

Type int

Description The total number of k-points used by the k-space to sample the unique wedge of the Brillouin zone.

`kspace(primitive cell)%kuniqu`

Type int

Description The number of symmetry unique k-points where an explicit diagonalization is needed. Smaller or equal to kt.

`kspace(primitive cell)%ndim`

Type int

Description The nr. of lattice vectors.

`kspace(primitive cell)%ndimk`

Type int

Description The nr. of dimensions used in the k-space integration.

`kspace(primitive cell)%xyzpt`

Type float_array

Description The coordinates of the k-points.

Unit 1/bohr

Shape [ndimk, kt]

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.

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

MoleculeSuperCell **Section content:** The system used for the numerical phonon super cell calculation.

`MoleculeSuperCell%AtomicNumbers`

Type int_array

Description Atomic number ‘Z’ of the atoms in the system

Shape [nAtoms]

`MoleculeSuperCell%AtomMasses`

Type float_array

Description Masses of the atoms

Unit a.u.

Values range [0, '\infinity']

Shape [nAtoms]

`MoleculeSuperCell%AtomSymbols`

Type string

Description The atom's symbols (e.g. 'C' for carbon)

Shape [nAtoms]

`MoleculeSuperCell%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

`MoleculeSuperCell%Charge`

Type float

Description Net charge of the system

Unit e

`MoleculeSuperCell%Coords`

Type float_array

Description Coordinates of the nuclei (x,y,z)

Unit bohr

Shape [3, nAtoms]

`MoleculeSuperCell%eeAttachTo`

Type int_array

Description A multipole may be attached to an atom. This influences the energy gradient.

`MoleculeSuperCell%eeChargeWidth`

Type float

Description If charge broadening was used for external charges, this represents the width of the charge distribution.

`MoleculeSuperCell%eeEField`

Type float_array

Description The external homogeneous electric field.

Unit hartree/(e*bohr)

Shape [3]

`MoleculeSuperCell%eeLatticeVectors`

Type float_array

Description The lattice vectors used for the external point- or multipole- charges.

Unit bohr

Shape [3, eeNLatticeVectors]

`MoleculeSuperCell%eeMulti`

Type float_array

Description The values of the external point- or multipole- charges.

Unit a.u.

Shape [eeNZlm, eeNMulti]

`MoleculeSuperCell%eeNLatticeVectors`

Type int

Description The number of lattice vectors for the external point- or multipole- charges.

`MoleculeSuperCell%eeNMulti`

Type int

Description The number of external point- or multipole- charges.

`MoleculeSuperCell%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).

`MoleculeSuperCell%eeUseChargeBroadening`

Type bool

Description Whether or not the external charges are point-like or broadened.

`MoleculeSuperCell%eeXYZ`

Type float_array

Description The position of the external point- or multipole- charges.

Unit bohr

Shape [3, eeNMulti]

`MoleculeSuperCell%EngineAtomicInfo`

Type string_fixed_length

Description Atom-wise info possibly used by the engine.

`MoleculeSuperCell%fromAtoms`

Type int_array

Description Index of the first atom in a bond. See the bondOrders array

`MoleculeSuperCell%latticeDisplacements`

Type int_array

Description The integer lattice translations for the bonds defined in the variables bondOrders, fromAtoms and toAtoms.

MoleculeSuperCell%LatticeVectors

Type float_array

Description Lattice vectors

Unit bohr

Shape [3, nLatticeVectors]

MoleculeSuperCell%nAtoms

Type int

Description The number of atoms in the system

MoleculeSuperCell%nAtomsTypes

Type int

Description The number different of atoms types

MoleculeSuperCell%nLatticeVectors

Type int

Description Number of lattice vectors (i.e. number of periodic boundary conditions)

Possible values [0, 1, 2, 3]

MoleculeSuperCell%toAtoms

Type int_array

Description Index of the second atom in a bond. See the bondOrders array

phonon_curves **Section content:** Phonon dispersion curves.

phonon_curves%brav_type

Type string

Description Type of the lattice.

phonon_curves%Edge_#_bands

Type float_array

Description The band energies

Shape [nBands, nSpin, :]

phonon_curves%Edge_#_direction

Type float_array

Description Direction vector.

Shape [nDimK]

phonon_curves%Edge_#_kPoints

Type float_array

Description Coordinates for points along the edge.

Shape [nDimK, :]

phonon_curves%Edge_#_labels

Type lchar_string_array

Description Labels for begin and end point of the edge.

Shape [2]

phonon_curves%Edge_#_lGamma

Type bool

Description Is gamma point?

phonon_curves%Edge_#_nKPoints

Type int

Description The nr. of k points along the edge.

phonon_curves%Edge_#_vertices

Type float_array

Description Begin and end point of the edge.

Shape [nDimK, 2]

phonon_curves%Edge_#_xFor1DPlotting

Type float_array

Description x Coordinate for points along the edge.

Shape [:]

phonon_curves%indexLowestBand

Type int

Description ?

phonon_curves%nBands

Type int

Description Number of bands.

phonon_curves%nBas

Type int

Description Number of basis functions.

phonon_curves%nDimK

Type int

Description Dimenion of the reciprocal space.

phonon_curves%nEdges

Type int

Description The number of edges. An edge is a line-segment through k-space. It has a begin and end point and possilby points in between.

phonon_curves%nEdgesInPath

Type int

Description A path is built up from a number of edges.

`phonon_curves%nSpin`

Type int

Description Number of spin components.

Possible values [1, 2]

`phonon_curves%path`

Type int_array

Description If the (edge) index is negative it means that the vertices of the edge abs(index) are swapped e.g. path = (1,2,3,0,-3,-2,-1) goes though edges 1,2,3, then there's a jump, and then it goes back.

Shape [nEdgesInPath]

`phonon_curves%path_type`

Type string

Description ?

Phonons **Section content:** Information on the numerical phonons (super cell) setup. NB: the reciprocal cell of the super cell is smaller than the reciprocal primitive cell.

`Phonons%Modes`

Type float_array

Description The normal modes with the translational symmetry of the super cell.

Shape [3, nAtoms, 3, NumAtomsPrim, nK]

`Phonons%nAtoms`

Type int

Description Number of atoms in the super cell.

`Phonons%nK`

Type int

Description Number of gamma-points (of the super cell) that fit into the primitive reciprocal cell.

`Phonons%NumAtomsPrim`

Type int

Description Number of atoms in the primitive cell.

`Phonons%xyzKSuper`

Type float_array

Description The coordinates of the gamma points that fit into the primitive reciprocal cell.

Shape [3, nK]

Properties **Section content:** Generic container for properties. The program band uses different rules for Types and Subtypes.

`Properties%nEntries`

Type int

Description Number of properties.

`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.

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]

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