# DFTB Manual

*Release 2017*

www.scm.com

Apr 12, 2018

# CONTENTS

CHAPTER

ONE

# GENERAL

## 1.1 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 pairwise combinations of atoms in a molecule. Many elements can be handled with the parameter sets included in the distribution. 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 parametrized, it is important to specify a proper parameter set when applying one of these models.

## 1.2 Release 2017

- New features:
  - *spin-polarized calculations* (page 6)
  - *DFTB3 for periodic systems* (page 6)
  - *l-dependent SCC-DFTB and TD-DFTB* (page 6)
  - *user defined masses in the input file, allowing the calculation of systems with different isotopes of the same element* (page 4)
  - *calculation of fat-bands* (page 9)
  - *massively improved performance of the D3-BJ dispersion correction for periodic systems* (page 6)
  - *improved performance of lattice optimizations* (page 12)
  - *Mayer bond order analysis* (page 25)
  - *interface to the NBO6 program* (page 25)
  - *support for parameter sets with only partially present repulsive potentials (e.g. halorg-0-1)* (page 31)
  - *support for restarting geometry optimizations* (page 26)
  - *reading of initial Mulliken charges from the input file* (page 6)
- All *DFTB.org parameter sets* (page 29) are now Creative Commons licensed and no longer require a special license agreement. The following new parameter sets are available:

  - 3ob-ophyd: modified O-P for 3ob (improves description of pentavalent phosphorus species)
  - auorg: for gold-thiolate compounds
  - borg: boron systems (solids and molecules)
  - halorg: for halogens
  - magsil: for chrisotyle nanotubes
- New defaults:
  - the default model Hamiltonian is now SCC-DFTB (used to be DFTB0)
  - k-space sampling is now disabled by default for periodic systems (only gamma-point will be sampled)
- Bugfixes:
  - fixed an issue with the band structure calculation for SCC-DFTB
  - fixed lattice optimization for Gamma-only periodic DFTB calculations

CHAPTER

TWO

# SPECIFICATION OF THE SYSTEM

## 2.1 System

The input of the initial structure can be given with the following key System. Apart from cases in which the system is retrieved by restarting a previous calculations this specification is always required.

```
System
    {Atoms
        Atom Coords
    End}

    {Charge NetQ}

    {Lattice
        Vectors
    End}

    {LatticeStrain
         eps1 value
         eps2 value
         eps3 value
         eps4 value
         eps5 value
         eps6 value
    }

    {FractionalCoords}

    {AtomMasses
      Atom Mass
    End}
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 this optional sub-key. 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 as in 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.

**AtomMasses** The default atomic masses used in DFTB are taken from Nuclear Physics A 565, 1-65 (1993) (http://dx.doi.org/10.1016/0375-9474(93)90024-R). The atomic masses of selected atom types can be redefined via the `AtomMasses` block key. The units of the atomic mass is *unified atomic mass unit*.

**Example**: Three different hydrogen isotopes:

```
Atoms
   N         0.000000    0.000000    0.010272
   H        -0.471582   -0.816803    0.407861
   H.D       0.943163    0.000000    0.407861
   H.T      -0.471582    0.816803    0.407861
End
AtomMasses
   H.D 2.01410
   H.T 3.01605
End
```

## 2.2 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**.

## CHAPTER THREE

# DFTB MODEL HAMILTONIAN

The mandatory `DFTB` key allows to specify and control different aspects of how the DFTB Hamiltonian is created and how the stationary Schroedinger equation is solved.

```
DFTB
  ResourcesDir  relativepath
  {Model DFTB0|SCC-DFTB|DFTB3
  {SCC
     {Unrestricted yes|no}
     {OrbitalDependent yes|no}
     {HXdamping}
     {iterations NIter}
     {converge charge=QDiff}
     {mixing Mix}
     {ndiis}
     {InitialCharges
        ...
     End}
  End}
  {Occupation aufbau|fermi {temperature=FermiTemp}}
  {UnpairedElectrons Number}
  {Dispersion
     method UFF|ULG|D2|D3-BJ
  End}
  {Purify
     tolerance r
     iterations n
  End}
  {SparsityThreshold sparsitythreshold}
  {Repulsion
     forcePolynomial
  End}
  {RadialExtrapolation  none|linear|improved|bezier|original}
  {UseSymmetry  yes|no}
  {StoreMatrices yes|no}
End
```

**`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, see *parameter sets* (page 29). 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 must contain a file called *metainfo.yaml*, which specifies the capabilities of the parameter set. For details see *metainfo.yaml* (page 31).

**Model** Selects the model Hamiltonian used in the DFTB calculation. Can be either `DFTB0` for the *classic* DFTB method without self-consistency cycle, `SCC-DFTB` for the self-consistent charge method, or `DFTB3` for the self-consistent charge method with third-order constributions. The default is `SCC-DFTB`.

**SCC** This optional section configures various details of the self-consistent charge cycle. If `Model` is set to `DFTB0`, none of this information is used and the entire section will be ignored.

**Unrestricted yes|no** Enables spin unrestricted calculations. The default is a restricted calculation. Only collinear spin polarization is supported, see Theor Chem Acc (2016) 135: 232, (http://dx.doi.org/10.1007/s00214-016-1991-9) for details. Must be supported by the chosen *parameter set* (page 29). Not yet compatible with DFTB3 or k-space sampling periodic calculations.

**OrbitalDependent yes|no** Activates or disables orbital resolved calculations. Standard is set by the chosen parameter set, only set this option if you want to change the default (see *metainfo.yaml* (page 31)).

**hxdamping** This option activates the H-X damping. Note that this treatment is also automatically enabled if the `model` key is set to `DFTB3`.

**iterations NIter** Allows to specify the maximum number of SCC iterations. Default is 100 iterations, which suffices for most standard calculations. If convergence issues are encountered, choosing a smaller value mixing parameter (see below) often provides a more stable but slower way to converge the SCC cycle. Convergence issues may also arise due to the use of the Aufbau occupations for systems with small HOMO-LUMO gaps. In this case the use of a Fermi broadening strategy may improve convergence (see below).

**converge charge=QDiff** SCC convergence threshold in terms of maximum change in the atomic charges (in atomic units) between two succeeding SCC cycles. Default converge charge 1.0E-8.

**ndiis Ndiis** Specifies the maximum number of samples considered during the direct inversion of iteration of subspace (DIIS) extrapolation of the atomic charges in during the SCC iterations (default ndiis 20). A smaller number of samples potentially leads to a more aggressive convergence acceleration, while a larger number often guarantees a more stable iteration. Due to often occurring linear dependencies within the set of sample vectors, the maximum number of samples is reached only in very rare cases.

**mixing Mix** Mix is the parameter used to mix the DIIS linear combination of previously sampled atomic charge vectors with an analogous linear combination of charge vectors resulting from population analysis combination. The default for the mixing parameter is 0.2 and it can assume real values between 0 and 1.

**InitialCharges** This section allows to set initial Mulliken charges for the SCC cycle. Specify one number per line and atom (or shell if running an orbital-resolved calculation). The order has to be the same as in the atoms block.

**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 5K.

**UnpairedElectrons** This optional key allows to specify the number of unpaired electrons (not the multiplicity!). This number will then be used in the orbital-filling strategy, even if it is unreasonable. Has to be compatible with the total number of electrons, meaning it must be an even number (or zero) if the total number of electrons is even and odd if the total number is odd. Must be an integer value. Note that this does not activate spin polarization, it only affects the filling of the orbitals.

**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* (page 31)) 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.

**Purify** By default (when the `Purify` section is not present), the next step's 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. Note also that molecular orbitals are not calculated when using the density matrix purification method, meaning that any subsequent calculations using molecular orbitals (e.g. TD-DFTB or Mayer bond order analysis) are incompatible with this keyword. Density matrix purification can be considerably faster than diagonalization for molecular of gamma-only periodic systems with large HOMO-LUMO gaps. Note that density matrix purification is faster for systems where the density matrix is sparse. The fill-in of the density matrix is printed to the standard output during the calculation. If it is too large (e.g. >50%), it is probably faster to use normal diagonalization.

**tolerance r** Accepts a single real number `r` that 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 `r`. The default value is 1.0e-8.

**iterations** Accepts a single integer number `n` that determines the maximum number of steps in the purification cycle. If the desired tolerance is not reached during `n` steps, the calculation is aborted. The default value is 50.

**Repulsion** This key allows to specify some details about the repulsion contribution evaluation. It currently 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.

**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. All matrix elements for distances greater than the range of the tables in the Slater-Koster files are zero.

**linear** Performs a linear interpolation between the last point of the grid and the value 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.

**bezier** Uses a Bezier 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.

**original** Same as improved, but reproducing behavior of previous reference programs. This method is only available for backwards compatibility and should not be used anymore.

**UseSymmetry yes|no** Enables or disables the use of symmetry. Currently only used in periodic calculations with >1 k-point. The key is optional, and in its absence the default is yes. For MD calculations, no symmetry will be used.

**`StoreMatrices yes|no` Default: no**. If *yes*, save the Hamiltonian and Overlap matrices to the results file *dftb.rkf*.

CHAPTER

FOUR

# ADDITIONAL PERIODICITY DATA

```
Periodic
     {KSpace NK}
     {BZStruct {enabled=yes|no} {automatic=yes|no} {fatbands=yes|no}
↪{UseSymmetry=yes|no} {DeltaK=realVal}}
     {BZPath
         {Path
         End}
     End}
     {DOS {nSteps=N} {EMin=r} {EMax=r}}
     {Phonon interpol=N stepsize=N nSides=1|2 UseSymmetry=yes|no}
     {SuperCell
     End}
     {LatticeStepSize}
     {Screening
         {rmadel}
         {dmadel}
     End}
End
```

**`KSpace`** This parameter controls the number of k-points used in the calculation. By default DFTB does not do any k-space sampling and uses only the Gamma-point as the only k-point. This should be sufficient for systems with large unit cells. For smaller systems, k-space sampling can be enabled explicitly using this keyword. For very small unit cells (one atom wide) a value of 5 is advised. For medium sized unit cells 3 is adequate.

The k-space sampling is relatively new in DFTB and as of the ADF2017 release still has some incompatibilities with other features: At the moment it is not possible to use k-space sampling in combination with DFTB3, spin-polarization, l-dependent SCC cycles or density matrix purification. Furthermore, k-space sampling calculations are not parallelized and should be run in serial, e.g. by setting the `NSCM` environment variable to `1`.

**`BZStruct`** Options for band structure plotting. This has no effect on the calculated energy.

**Warning:** The band structure is only computed in case of k-space sampling, i.e. it is not computed for Gamma-only calculations (see: `KSpace`).

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`** (**Default: yes**) Whether or not to calculate the band structure.

**`automatic`** (**Default: yes**) If `yes` DFTB will automatically generate the standard path through the Brillouin zone. If `no` DFTB will use the user-defined path in *BZPath*.

**`FatBands`** (**Default: yes**) If `yes` DFTB will compute the fat bands (note: only if `BZStruct%Enabled=yes`). The Fat Bands are the periodic equivalent of the Mulliken population analysis (https://en.wikipedia.org/wiki/Mulliken_population_analysis). The definition of the fat bands can be found in the Band Documentation.

**`UseSymmetry`** (**Default: yes**) If `yes` only the irreducible wedge of the Wigner-Seitz cell is sampled. If `no`, the whole (inversion-unique) Wigner-Seitz cell is sampled.

**`DeltaK`** (**Default: 0.1**) Step (in reciprocal space, unit: 1/Bohr) for band structure interpolation. Using a smaller number (*e.g.* 0.03) will yield to *smoother* band curves, but will increase the computation time.

**`BZPath`** If `BZStruct%Automatic` is `no`, DFTB will compute the band structure for the user-defined path in the `BZPath` block. You should define the vertices of your path in fractional coordinates (wrt the reciprocal lattice vectors) in the `Path` sub-block. If you want to make a *jump* in your path, you need to specify a new `Path` sub-block.

In the following example we define the path `Gamma-X-W-K|U-X` for a FCC lattice:

```
BZPath
   Path
      0.000   0.000   0.000
      0.500   0.000   0.500
      0.500   0.250   0.750
      0.375   0.375   0.750
   SubEnd
   Path
      0.625   0.250   0.625
      0.500   0.000   0.500
   SubEnd
End
```

**`DOS`** The subkeys of `DOS` allow to customize the calculation of the density of states. The density of states is calculated at `nSteps` energies in the interval `EMin` to `EMax`. Energies are given in Hartree and relative to the Fermi energy. By default the density of states is evaluated at 300 points in the interval from -0.75Ha to +0.75Ha.

**`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. For the study of isotopic shift of phonons, see the *AtomMasses key* (page 4).

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

**`nSides`** (**Default: 2**) By default a two-sided (or quadratic) numerical differentiation of the nuclear gradients is used. Using a single-sided (or linear) numerical differentiation is computationally faster but much less accurate. **Note**: In older versions of the program only the single-sided option was available.

**`useSymmetry`** (**Default: Yes**) Whether or not to exploit the symmetry of the system in the phonon calculation.

**`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.

**`LatticeStepSize`** (**Default: 1e-4**) Step size for the numerical calulation of lattice gradients.

**`Screening`** For SCC-DFTB in periodic systems the Coulomb interaction is screened with a Fermi-Dirac like function defined as $S(r) = \left(\exp\left(\frac{r-r_{\text{madel}}}{d_{\text{madel}}}\right)+1\right)^{-1}$. Screening is always enable, even if this section is absent. This section allows to change some details of the screening procedure.

**`rmadel`** Sets the range $r_{\text{madel}}$ of the screening function. The default is 2x the norm of the longest lattice vector.

**`dmadel`** Sets the smoothness $d_{\text{madel}}$ of the screening function. The default is 1/10 of $r_{\text{madel}}$.

CHAPTER

FIVE

# STRUCTURE AND REACTIVITY

## 5.1 Computational Task

The section Task is used to specify the computational task to perform with the DFTB program. This section and its only sub-key, runType, are mandatory therefore.

```
Task
  runType type
End
```

Currently the DFTB program supports the following values for `type`

- `SinglePoint` or `SP` for *single point* (page 5) energy calculations
- `GeometryOptimization` or `GO` for *geometry optimizations* (page 11)
- `TransitionState` or `TS` for transition state searches
- `Frequencies` or `F` for frequency and `Phonons` for phonon calculations
- `MolecularDynamics` or `MD` for *molecular dynamics simulations* (page 13)

## 5.2 Geometry Optimization

Geometry optimizations are enabled by setting the runType key to “GO”.

```
Task
  runType GO
End
```

The actual behavior of the geometry optimization can be specified in more detail by the following input block.

```
Geometry
  {Method      quasinewton|conjgrads|gdiis}
  {CGType      fletcherreeves|polakribiere|hestenesstiefel|perry|birginmartinez|
                scaled|adaptedscaled1|adaptedscaled2|swartdyn1|swartdyn2}
  {SwartIter   Niter}
  {Optim       cartesian|delocal|internal}
  {initialHessian       swart|unitmatrix}
  {Iterations  Niter}
  {Converge    {E=TolE} {Grad=TolG} {Rad=TolR}}
  {Step        {TrustRadius=MaxRadius}}
  {OptimizeLattice [Yes|No]}
End
```

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

**`Method quasinewton|conjgrads|gdiis`** Selects the optimisation algorithm employed for the geometry relaxation. Currently supported are the Hessian-based Quasi-Newton-type BFGS algorithm and its variant based on the direct inversion of iterative subspace (GDIIS), as well as the purely gradient-based conjugate gradients method. The default is “quasinewton”.

**`CGType`** Defines the methodology to compute the $\beta_n$ 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 $\beta_n = \frac{\Delta x_n^\top \Delta x_n}{\Delta x_{n-1}^\top \Delta x_{n-1}}$.
>
> **`polakribiere`** Uses the Polak-Ribiere formula $\beta_n = \frac{\Delta x_n^\top (\Delta x_n - \Delta x_{n-1})}{\Delta x_{n-1}^\top \Delta x_{n-1}}$.
>
> **`hestenesstiefel`** Uses the Hestenes-Stiefel formula $\beta_n = -\frac{\Delta x_n^\top (\Delta x_n - \Delta x_{n-1})}{s_{n-1}^\top (\Delta x_n - \Delta x_{n-1})}$.
>
> **`perry`** Uses the Perry formula $\beta_n = -\frac{\Delta x_n^\top (\Delta x_n - \Delta x_{n-1})}{s_{n-1}^\top (\Delta x_n - \Delta x_{n-1})}$.
>
> **`birginmartinez`** Similar to Perry, with a scaled $(\Delta x_n - \Delta x_{n-1})$ at the numerator by $\theta = \frac{s_{n-1}^\top s_{n-1}}{s_{n-1}^\top (\Delta x_n - \Delta x_{n-1})}$.
>
> **`scaled`** Uses $\beta_n^{Scaled} = -\frac{\Delta x_n^\top \Delta x_n - \Delta x_{n-1}^\top \Delta x_n}{s_{n-1}^\top \Delta x_{n-1}}$.
>
> **`adaptedscaled1`** Same as scaled, with $\beta = \max\left(1 - \beta_n^{Scaled}, \beta_n^{Scaled}\right)$.
>
> **`adaptedscaled2`** Same as scaled, with $\beta = 1 - \beta_n^{Scaled}$.
>
> **`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|internal`** Optimization in delocalized coordinates (`delocal`) can only be used for geometry optimizations or transition state searches of molecular systems with the Quasi-Newton method.

**`InitialHessian swart|unitmatrix`** Selects the type of the initial model Hessian when optimizing periodic systems with either the Quasi-Newton or the GDIIS method.

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

**`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 Hartree. 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 MaxRadius`** Controls that changes in geometry from one cycle to another are not too large. 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** Default: No. Enables optimization of the lattice parameters, in addition to the molecular geometry. This can only be applied to periodic systems.

### 5.2.1 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 A B Rab
  Angle A B C alphaABC
  Dihed A B C D dihABDC
end
```

Atoms are specified using their index (starting at 1) based on the order given in the `Atoms` block. The `Dist`, `Angle`, and `Dihed` constraints do not have to be satisfied at the start of the geometry optimization.

**Dist** When specified, the distance between atoms A and B is constrained to the value Rab in Ångstrom.

**Angle** When specified, the angle between atoms A, B and C (A-B-C) is constrained to the value alphaABC in degrees.

**Dihed** When specified, the dihedral angle between atoms A, B, C and D (A-B-C-D) is restrained to the value dihABCD in degrees. The dihedral angle is projected onto the [0, 2 $\pi$] interval, so there is no difference between specifying -30 degrees or 330 degrees.

## 5.3 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 the velocity-Verlet algorithm) with and without thermostats. This key (used with `runType` set to `MD`) allows to specify the details of the molecular dynamics calculation.

**Steps NSteps** Specifies the number of steps to be taken in the MD simulation. 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 file 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 this 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 `Atoms` block.

**Thermostat** 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:

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

**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.

**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 (`local`) or on the molecular system as a whole (`global`).

**`temperature=Temp`** Specifies the temperature of the thermostat, in kelvin. This parameter is mandatory.

CHAPTER

SIX

# SPECTROSCOPIC PROPERTIES

## 6.1 Excited States with Time Dependent DFTB

DFTB allows for excited state calculations on molecular systems by means of 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. DFTB also supports the calculation of excited state gradients, which allows geometry optimizations and vibrational frequency calculations for excited states.

The TD-DFTB implementation uses the PRIMME library (PReconditioned Iterative MultiMethod Eigensolver) by Andreas Stathopoulos and James R. McCombs, PRIMME: PReconditioned Iterative MultiMethod Eigensolver (http://www.cs.wm.edu/~andreas/publications/primmeTOMS.pdf): Methods and software description ACM Transaction on Mathematical Software Vol. 37, No. 2, (2010), 21:1–21:30 (http://dx.doi.org/10.1145/1731022.1731031).

DFTB excited state calculations are controlled by the following keywords:

```
Properties
    Excitations
       {SingleOrbTrans
          {Filter
             {dEMin r}
             {dEMax r}
             {OSMin r}
          End}
          {printlowest n}
       End}
       {TDDFTB
          calc singlet|triplet
          {lowest n}
          {upto r}
          {diagonalization exact|davidson|auto}
          {DavidsonConfig
             {ATCharges onthefly|precalc}
             {tolerance r}
             {safetymargin n}
          End}
          {print evcontribs}
       End}
       {TDDFTBGradients
          {excitation n}
          {eigenfollow true|false}
       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 TD-DFTB calculations. If the section is not present all single orbital transitions will still be calculated and used in a subsequent TD-DFTB calculation, but no output will be produced.

**Filter** The Filter section allows to remove single orbital transitions based on certain criteria. 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 TD-DFTB 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. TD-DFTB calculates the excitations in the basis of the single orbital transitions, whose calculation is configured in the `SingleOrbTrans` section. Using a filter in `SingleOrbTrans` can therefore be used to reduce the size of the basis for TD-DFTB. 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 behavior depends on the existence of the `SingleOrbTrans` section: If no `SingleOrbTrans` section is found (the `Excitations` section is completely empty then) a TD-DFTB calculation with default parameters will be performed. If only the `SingleOrbTrans` section is present no TD-DFTB calculation will be done.

**calc singlet|triplet** Specifies the multiplicity of the excitations to be calculated. Accepts the values `singlet` or `triplet`. Note that this key is not optional: If the TD-DFTB section is present, this keyword also has to be there.

**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 TD-DFTB 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 not enough memory is available.

**tolerance r** Convergence criterion for the norm of the residual. Accepts a single real number. The default is 1e-9.

**safetymargin n** The number of eigenvectors the Davidson method will calculate in addition to the ones requested by the user. With the Davidson eigensolver it is generally a good idea to calculate a few more eigenvectors than requested with the `lowest` keyword, as depending on the initial guess for the eigenvectors it can happen that the found ones are not exactly the lowest ones. This problem is especially prominent if one wants to calculate only a small number of excitations for a symmetric molecule, where the initial guesses for the eigenvectors might have the wrong symmetry. The default is to calculate 4 more eigenvectors than requested. Note that the additionally calculated excitations will neither be written to the result file nor be visible in the output.

**print evcontribs** Specifies whether to print details on the contribution of the individual single orbital transitions to the calculated excitations. The default is to print nothing.

**TDDFTBGradients** Use this input to calculate analytical gradients for the TD-DFTB excitation energies, which allows the optimization of excited state geometries and the calculation of vibrational frequencies in excited states (see J. Comput. Chem., 28: 2589–2601 (http://dx.doi.org/10.1002/jcc.20697)). If the gradients are calculated, they will automatically be used for geometry optimizations or vibrational frequency calculations, if the corresponding `runType` is selected. Vibrationally resolved UV/Vis spectroscopy (Franck-Condon Factors) can be calculated in combination with the FCF program. See the ADF documentation on Vibrationally resolved electronic spectra.

**excitation n** Determine which excited state to calculate the gradients for. Gradients can only be calculated for an excited states that has been calculated in the `TDDFTB` block, so make sure that enough excitations are calculated. The default is to calculate the gradients for the first excited state.

**eigenfollow true|false** If this is set to true, DFTB uses the transition density in atomic orbital basis to follow the initially selected excited state during a geometry optimization. This is useful if excited state potential energy surfaces cross each other and you want to follow the surface you started on. Eigenvector following is disabled by default.

CHAPTER

SEVEN

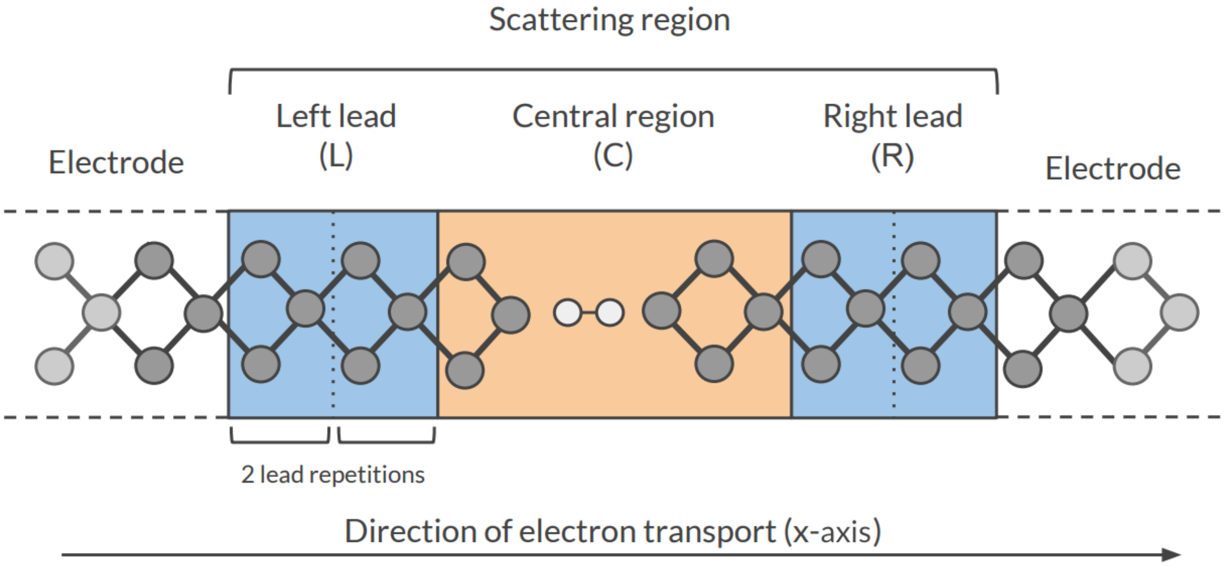
# ELECTRONIC TRANSPORT (NEGF)

**See also:**

DFTB-NEGF GUI tutorial

## 7.1 Transport with NEGF in a nutshell

The **Non-Equilibrium Green's Functions** formalism (**NEGF**) is a theoretical framework for modeling electron transport through nano-scale devices. Electron transport is treated as a one-dimensional coherent scattering process in the "scattering region" for electrons coming in from the electrodes:



Our goal is to compute the **transmission function** $T(E)$, which describes the rate at which electrons of energy $E$ are transferred from the left electrode to the right electrode by propagating through the scattering region. From the transmission function we can calculate the electric current for given **Bias Voltage** $V$ applied between the electrodes:

$$I(V) = \frac{2e}{h} \int_{-\infty}^{\infty} T(E) \left( f(E-\mu_L) - f(E-\mu_R) \right) dE$$

where $f(E)$ is the Fermi-Dirac distribution function for a given temperature, and $\mu_L$ ($\mu_R$) is $\epsilon_F + eV/2$ ($\epsilon_F - eV/2$), $\epsilon_F$ being the Fermi energy of the electrodes.

The transmission function $T(E)$ can be computed from the **Green's function** of our system.

The Green's function $G(E)$ of the scattering region is obtained solving the following equation:

$$(ES - H)G(E) = I$$

where $S$ is the overlap matrix, $H$ is the Hamiltonian and $I$ is the identity matrix. The Hamiltonian is composed as follows (**L**, **C** and **R** denote the **left lead**, the **central region** and the **right lead** respectively):

$$H = \begin{pmatrix} H_L + \Sigma_L & H_{LC} & 0 \\ H_{LC} & H_C & H_{RC} \\ 0 & H_{RC} & H_R + \Sigma_R \end{pmatrix}$$

The two *self-energies* $\Sigma_L$ and $\Sigma_R$ model the two semi-infinite electrodes.

The transmission function $T(E)$ can be calculated from the Green's function $G(E)$ and the so-called *coupling matrices* $\Gamma_L(E)$ and $\Gamma_R(E)$ (which are related to $\Sigma_L$ and $\Sigma_R$):

$$T(E) = Tr[G(E)\Gamma_R(E)G(E)\Gamma_L(E)]$$

**See also:**

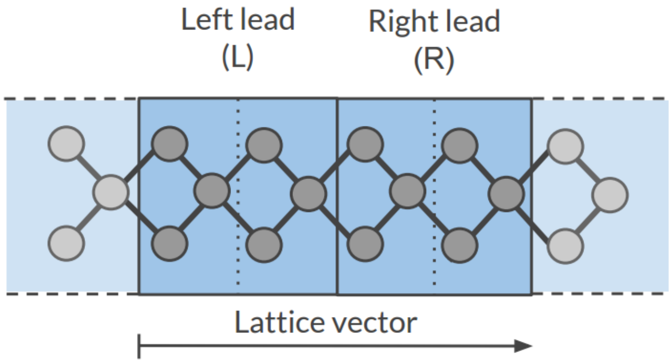
PhD Thesis (https://opus.jacobs-university.de/frontdoor/index/index/docId/478) of Mahdi Ghorbani-Asl (DFTB-NEGF developer)

## 7.2 Simulations work flow

The computation of the transmission function $T(E)$ within the DFTB-NEGF formalisms requires three individual simulations.

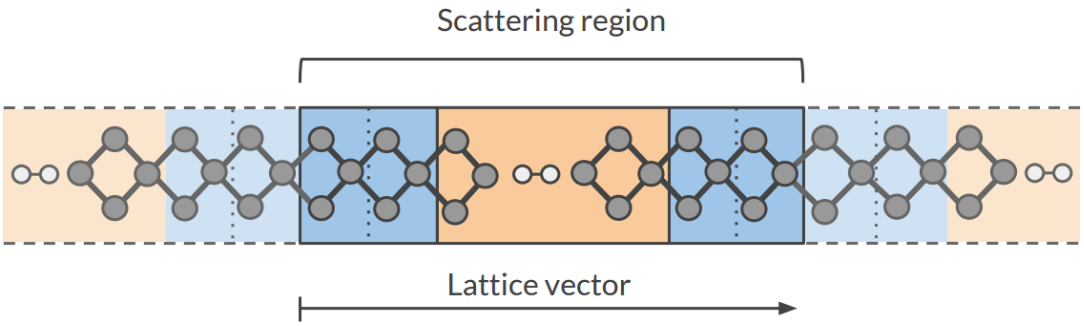
**Tip:** Use ADFInput (GUI) to set up your DFTB-NEGF calculation (see the DFTB-NEGF GUI tutorial)

**1): DFTB leads calculation** A 1D-periodic DFTB calculation of the leads (*StoreMatrices* (page 7): yes, *KSpace* (page 9) sampling 13):



The Hamiltonian matrices $H_L$ and $H_R$ and the Fermi energy of the electrode $\epsilon_F$ are computed in this calculation ($H_L$, $H_R$ and $H_{LR}$ are also used to compute the surface Green's functions $g_L$ and $g_R$ of the semi-infinite electrodes).

**2): DFTB scattering-region calculation** A a 1D-periodic DFTB calculation of the scattering region (*StoreMatrices* (page 7): yes, gamma-only, *i.e.*, no *KSpace* (page 9) sampling):



The Hamiltonian matrices $H_{LC}$ and $H_{RC}$ and $H_C$ are computed in this calculation.

**3): Conductance calculation** The **Conductance program** computes the NEGF transmission function $T(E)$ using the Hamiltonians and Overlap matrices from the previous two DFTB calculations.

## 7.3 Conductance Input options

The **Conductance** program computes the transmission function using the NEGF approach. This is the input structure of the **conductance** program:

```
$ADFBIN/conductance <<EOF > conductance.out

  EnergyGrid Min=value Max=value Num=value

  Files
     Leads       /path/DFTB_lead_filename.rkf
     Scattering  /path/DFTB_scattering_filename.rkf
  End

  Technical
     Eta                     value
     OverwriteLeads          True|False
     SetOffDiagonalToZero    True|False
  End

  end input
EOF
```

**EnergyGrid** Defines the energies for which the **Transmission function** $T(E)$ is computed.

**Min** **Default: -5 eV**. Minimum value for the energy grid.

**Max** **Default: 5 eV**. Maximum value for the energy grid.

**Num** **Default: 200**. Number of energy values in which the interval Min-Max is subdivided.

**Files** DFTB results files (.rkf) containing the Hamiltonians and overlap matrices.

**Leads** Path (either absolute or relative) of the results file (.rkf) of the **DFTB leads calculation** (first calculation).

**Scattering** Path (either absolute or relative) of the results file (.rkf) of the **DFTB scattering-region calculation** (second calculation).

**Technical**

`Eta` **Default: 1.0E-5**. To avoid poles of the Green's function, a small imaginary number is added to the energy, $E := E + i\eta$. If you have matrix inversion problems in Conductance, try changing this parameter (e.g., 1.0E-4 or 1.0E-6)

`OverwriteLeads` **Default: True**. If True, the Hamiltonians $H_L$ and $H_R$ are taken from the *DFTB-leads* calculation. If False, they are taken from the *DFTB scattering-region* calculation.

`SetOffDiagonalToZero` **Default: True**. If true, $H_{LR}$ and $S_{LR}$ are explicitly set to zero. If False, they are taken from the *DFTB scattering-region* calculation.

## 7.4 Miscellaneous remarks on DFTB-NEGF

- You should make sure that your results are converged with respect to the number of lead repetitions; the results should not change significantly if you increase the number of lead repetitions.
- It's good practice to include at least one lead repetition in the central region.
- The transmission function is computed at zero bias voltage. The zero-bias transmission function is then used for computing the electric current for non-zero bias voltage.

CHAPTER

EIGHT

# ADDITIONAL INPUT OPTIONS

## 8.1 System Properties

DFTB can calculate various properties of the simulated system.

```
Properties
  {BondOrders {true|false}}
  {NBOInput {true|false}}
  {DipoleMoment {true|false}}
End
```

Specifying neither `true` nor `false` is the same as specifying `true`.

**`BondOrders`** (**Default: false**) Whether or not Mayer bond orders are calculated based on the final molecular orbitals. The calculated bond orders will be picked up by the GUI and shown in ADFMovie for geometry optimzations and molecular dynamics calculations.

**`NBOInput`** (**Default: false**) Whether or not an input file for the NBO program is written to disk as `nboInput.FILE74`. The input file follows the `FILE47 format` as described in the NBO6 manual (http://nbo6.chem.wisc.edu/nbo6ab_man.pdf) available on nbo6.chem.wisc.edu (http://nbo6.chem.wisc.edu/). By default, only the calculation of the natural bond orbitals and the natural localized molecular orbitals is enabled, but the `nboInput.FILE47` file can be edited by hand to enable other analysis models. Please refer to the NBO6 manual for details.

**`DipoleMoment`** (**Default: true**) Whether of not the electric dipole moment is calculated from the calculated Mulliken charges. While it is technically possible to calculate the dipole moment with the DFTB0 model, it is not recommended and the SCC-DFTB or DFTB3 model should be used instead. For periodic systems the dipole moment is ill-defined and should not be interpreted.

## 8.2 Restart

Allows to restart DFTB-SCC (and DFTB3) calculations from atomic or shell charges (or spin populations) generated in earlier runs.

```
Restart
      {RestartFile RestartFileName}
      {RestartMulliken true|false}
      {RestartSpinPops true|false}
      {RestartGeometry true|false}
End
```

For spin-restricted calculations restarting requires requires a valid .rkf file to be present at the (absolute or relative) path specified by `RestartFileName` and the key `RestartMulliken` to be enabled. In the context of spin-resolved DFTB runs a restart can either performed via `RestartMulliken` and/or `RestartSpinPops`. In the latter case the restart .rkf file is required to contain spin populations computed in a previous spin-resolved calculation. The `RestartGeometry` key can be used to read the last nuclear coordinates (and potentially the lattice) from the restart file. This can be used to restart a failed or non-converged geometry optimization from the last known coordinates. Note that the *entire* `System` block must still be present, even when restarting with the geometry from another file. The coordinates given in the `Atoms` block (and potentially the lattice) will be ignored, but the number the type and the order of the atoms is used and must match with the system in the restart file.

## 8.3 Developer options

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

Printing timing details to see how much time is spend in which part of the code. Default value is `None`.

CHAPTER

# NINE

# EXAMPLES

The $ADFHOME/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
    Model SCC-DFTB
End

Geometry
    Iterations 100
End
```

```
eor
```

CHAPTER

TEN

# DFTB PARAMETERS

Most popular DFTB parameter sets are *distributed with the ADF package* (page 29). Other parameter sets can easily be *added* (page 30).

Often parameter files are designed for a specific purpose, which may be different from your application, and therefore may give not the desired accuracy. Note that parameter files coming from different parameter sets are in general not compatible with each other and should not be mixed.

Additional licenses may be required to use some of the distributed parameter sets. Please contact us at license@scm.com for details.

## 10.1 Available parameter sets

### 10.1.1 DFTB.org

The DFTB implementation shipped by SCM provides the most up-to-date parameter sets available on the DFTB.org website. The following sets are currently shipped:

- 3ob-3-1 (Br, C, Ca, Cl, F, H, I, K, Mg, N, Na, O, P, S, Zn): general purpose set for the DFTB3 method
- mio-1-1 (H, C, N, O, S, P): for bio and organic molecules with SCC-DFTB
- 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, we ship the following specific purpose parameter sets:

- 3ob-freq: modified 3ob parameters for a better description of vibrational frequencies
- 3ob-hhmod: modified H-H for 3ob (for a better description of H2)
- 3ob-nhmod: modified N-H for 3ob (improves sp3-N proton affinities)
- 3ob-ophyd: modified O-P for 3ob (improves description of pentavalent phosphorus species)
- auorg (Au + mio): for gold-thiolate compounds
- borg (B, H): boron systems (solids and molecules)
- chalc-0-1 (As + mio): for chalcogenide glasses
- halorg (F, Cl, Br, I + mio): for halogens
- hyb-0-1 (Ag, Ga, As, Si + mio): for organic and inorganic hybrid systems
- magsil (Mg, Si, O, H, Mg): for chrisotyle nanotubes
- miomod-hh: contains a modified parameter set for H2

- miomod-nh: contains a modified parameter set fo N-H to improve N-H binding energies
- siband (Si, O, H): electronic parameters for accurate silicon and silicon dioxide band structures
- tiorg-0-1 (Ti + mio): for Ti bulk, TiO2 bulk, TiO2 surfaces, and TiO2 with organic molecules
- trans3d-0-1 (Sc, Ti, Fe, Co, Ni + mio): transition metal elements for biological systems
- znorg-0-1 (Zn + mio): for Zn bulk, ZnO bulk, ZnO surfaces, and ZnO with organic molecules

We recommend to visit the DFTB.org (http://www.dftb.org) web site for more detailed information about each set. We are committed to shipping all DFTB.org parameter sets in their latest version. If you miss one of the DFTB.org parameter sets in our distribution, please contact us at support@scm.com. Please note that our implementation of DFTB currently does not support parameter sets containing f-functions, such as the "rare" set.

### 10.1.2 QUASINANO

The *QUASINANO2013.1* (page 37) 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 TD-DFTB calculations, for example.

The *QUASINANO2015* (page 38) parameter set extends the QUASINANO2013.1 parameter set, and includes terms that are needed to compute the total energy and its gradient.

### 10.1.3 Dresden

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* (page 38). These parameter files are kept in the directory $ADFHOME/atomicdata/DFTB/Dresden.

## 10.2 Files in the resources directory

This section contains a technical description of all the files and their formats which together constitute a DFTB parameter set. The parameter sets *distributed with the ADF package* (page 29) are ready to be used out-of-the box, and no knowledge about their format should be necessary to run DFTB calculations. However, users who want to use their own DFTB parameters with our implementation, will need to package them in a way that is understood by it.

DFTB parameter sets in the ADF package have up to four components: The *Slater-Koster files* (page 30), the *metainfo.yaml file* (page 31) and the optionally some *additional .yaml files* (page 32) as well as *binary .rkf files containing the basis functions* (page 33).

### 10.2.1 Slater-Koster files

Most of the data consituting a DFTB parameter set is contained in the so called Slater-Koster files. These are typically text files with the file extension `.skf`. For legal reasons, some parameter sets that are shipped with ADF the Slater-Koster files are encrypted though, in which case their file extension is `.ske`.

There is generally a Slater-Koster file per **pair of elements** supported by the parameter set, e.g. for a set supporting the four elements C,H,O,N there will be 16 Slater-Koster files in total. The Slater-Koster file names contain the symbols of the elements, e.g. `C-H.skf`, `H-O.skf` and `C-C.skf`. Note that files for both element orders, e.g. `C-H.skf` and `H-C.skf`, are needed and differ in general. The Slater-Koster files contain the matrix elements of the Hamiltonian operator and the overlap between between basis functions centered on two atoms, tabulated for different distances. They also contain a description of a repulsive potential between the two atoms. Furthermore the one element Slater-Koster files (like `H-H.skf` and `C-C.skf`) contain some information about the individual atom, e.g. orbital energies of the atomic orbitals. A detailed description (http://www.dftb.org/fileadmin/DFTB/public/misc/slakoformat.pdf) of the Slater-Koster file format can be found at DFTB.org (http://www.dftb.org).

### 10.2.2 The metainfo.yaml file

There should be a file named `metainfo.yaml` in each resources directory (see *ResourcesDir* (page 5)), for example *DFTB.org/3ob-3-1/metainfo.yaml*. The file is in accordance with the YAML (http://www.yaml.org) syntax convention. In older versions of ADF this file was optional and its format was slightly different. Starting with the 2017 release of ADF, the `metainfo.yaml` file is required to use a parameter set.

The following entries `metainfo.yaml` specify the capabilities of the parameter set:

**`format: txt|txtn|txtq`** Specifies which format is used for the Slater-Koster files of the parameter set. Use `txt` for normal text files with extension `.skf`. `txtq` is used for the encrypted Slater-Koster files from the QUASINANO parameter sets. Encrypted Slater-Koster files have the extension `.ske`. If the `format` entry is not there, normal (unencrypted) text files are assumed.

**`supports: [dftb0, scc-dftb, dftb3] (or old aliases: [ noscc, scc, thirdorder ])`** A comma-separated list of model Hamiltonians for which the parameter set can be used. If the parameter set only supports a single model Hamiltonian, the enclosing `[ ]` can be dropped. This entry is mandatory. Without it, DFTB will refuse to use the parameter set.

**`repulsion: no|partial|yes`** Specifies whether the parameter set has repulsive potentials for all pairs of elements. Parameter sets that do not have repulsive potentials (e.g. QUASINANO2013.1) can not be used in geometry optimizations, molecular dynamics or frequency calculations. Howver, they can still be used in single point calculations, e.g. for UV/Vis absorption spectra of molecules or band structures of solids. Some parameter sets (e.g. DFTB.org/halorg-0-1) have most, *but not quite all* repulsive potentials. If `repulsion` is set to `partial`, DFTB will allow calculations with run-types normally requiring repulsive potentials and will only print a warning about which pairs are missing. It is then the user's responsibility to ensure that element pairs for which the repulsion is missing do not get too close during the calculation. If the `repulsion` entry is not found, DFTB will assume that there are repulsive potentials for all element pairs.

**`spin_polarization: no|yes`** Whether the parameter set supports unrestricted calculations. The default value is `no`.

**`orbital_dependence: [ noldep, ldep ]`** Whether the parameter set supports an atomic and/or shell-resolved SCC cycle. If only one of the two is supported, the enclosing `[ ]` can be dropped and the `default_orbital_dependence` entry (see below) does not need to be specified. If the `orbital_dependence` key is not found, DFTB assumes that the parameter set only supports atomic SCC cycles.

**`default_orbital_dependence: noldep|ldep`** The default mode for the SCC cycle.

**`dispersion: [ uff, ulg, d2, d3-bj ]`** A comma-separated list of London dispersion corrections supported by the parameter set. If only one method is supported, the enclosing `[ ]` can be dropped and the `default_dispersion` entry (see below) does not need to be specified.

**`default_dispersion: uff|ulg|d2|d3-bj`** The default dispersion method to be used if the user enables dispersion but does not specify a method explicitly. If the `default_dispersion` entry is not found and more than one method is supported according to `dispersion` entry, **no** dispersion correction will be used by default. The user then has to select a dispersion method explicitly in the input file.

In addition to specifying the parameter set's capabilities, the `metainfo.yaml` file should also contain references to the scientific publication describing the parameter set:

```
url: http://www.scm.com

reference: |
  M. Wahiduzzaman, A. F. Oliveira, P. Philipsen, L. Zhechkov, E. van Lenthe, H. Witek,
↪ T. Heine
  "DFTB Parameters for the Periodic Table: Part 1, Electronic Structure",
  J. Chem. Theory Comput., 9, 2013, 4006-4017, DOI: 10.1021/ct4004959

short_reference: J. Chem. Theory Comput., 9, 2013, 4006-4017
```

All these entries are optional and at the moment only the `reference` entry is read by DFTB (and reproduced verbatim in its output). Note that the pipe symbol `|` is necessary to start a multiline entry (with preserved line breaks) and that the following lines have to be indented by at least one space.

### 10.2.3 Additional .yaml files

The Slater-Koster file format (http://www.dftb.org/fileadmin/DFTB/public/misc/slakoformat.pdf) is relatively old and very inflexible. Over the years extensions of the DFTB method (e.g. spin-polarization, DFTB3, dispersion corrections) have been developed that require parameters which do not have a place in the Slater-Koster files. In the ADF implementation of DFTB, these parameters are stored in additional `.yaml` files in the resources directory of the parameter set: The `additional_parameters.yaml` file as well as per element `.yaml` files, e.g. `H.yaml` and `C.yaml`.

The `additional_parameters.yaml` file contains anything that applies to the entire parameter set and does not depend on the individual elements. At the moment this is:

**`grimme_d3bj_params: s6 s8 a1 a2`** The fitting parameters for Grimme's D3-BJ dispersion correction. This entry is mandatory if the `metainfo.yaml` file lists D3-BJ as a supported dispersion correction methods.

**`zeta_Hcorr: zeta`** A single number `zeta` used in the HX-damping usually applied in DFTB3 calculations.

The per element `.yaml` files may contain the following entries:

**`hubbard: U_atom`** The atomic Hubbard parameter used in a normal, atomic SCC cycle is specified in the element's `.yaml` file as the `hubbard` entry. It is quite surprising that such a commonly used parameter does not have its place in the Slater-Koster files, which only hold the shell-dependent Hubbard parameters. For atomic SCC cycles it is common practice to use the Hubbard parameter of the s-shell as the atomic Hubbard parameter, even though the two values are not strictly related. For consistency with other DFTB implementations, ADF will do the same if the atomic Hubbard parameter is not found in the element's `.yaml` file. However, it will also notify the user about this potentially questionable behavior.

**`hubbard_derivative: dUdq`** The derivative of the the atomic hubbard parameter with respect to the atomic charge. This is information is required to perform DFTB3 calculations.

**`magnetic_hubbard` and `magnetic_hubbard_ldep`** The magnetic Hubbard parameters (often abbreviated W in the literature). These are required for unrestricted calculations and TD-DFTB singlet-triplet excitations. Depending on whether the parameter set allows atomic and/or shell resolved SCC cycles, the magnetic Hubbard parameter is given as a single number and/or a small matrix:

```
magnetic_hubbard: W_atom

magnetic_hubbard_ldep: >
    W_ss  W_sp
    W_ps  W_pp
```

The size of the matrix is determined by the number of basis functions on the element. Note that the > is essential to start a multiline entry (in which line breaks are ignored).

### 10.2.4 Basis function information in .rkf files

The QUASINANO parameter sets additionally have per element `.rkf` files in the resources directory, e.g. `H.rkf` and `C.rkf`. These binary files, which can be opened in the GUI with KFBrowser, contain information about the basis functions used to calculate the matrix elements in the Slater-Koster files. While this information is not needed to perform the DFTB calculation itself, it is used by the GUI in order to visualize properties like molecular orbitals or densities.

CHAPTER

ELEVEN

# 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.

## 11.1 General References

**For calculations with the Density Functional Tight Binding (DFTB) program:** ADF DFTB 2017, SCM, Theoretical Chemistry, Vrije Universiteit, Amsterdam, The Netherlands, http://www.scm.com Optionally, you may add the following list of authors and contributors: R. Rüger, A. Yakovlev, P. Philipsen, S. Borini, P. Melix, A. F. Oliveira, M. Franchini, T. Soini, M. de Reus, M. Ghorbani Asl, D. McCormack, S. Patchkovskii, T. Heine.

**For TD-DFTB, cite:** R. Rüger, E. van Lenthe, Y. Lu, J. Frenzel, T. Heine, and L. Visscher, *Efficient Calculation of Electronic Absorption Spectra by Means of Intensity-Selected Time-Dependent Density Functional Tight Binding*, J. Chem. Theory Comp., 2015, 11 (1), pp 157-167 (http://dx.doi.org/10.1021/ct500838h).

**For DFTB-NEGF, cite:** Mahdi Ghorbani-Asl *Electronic transport through two-dimensional transition-metal chalcogenides*, PhD Thesis (2014) (https://opus.jacobs-university.de/frontdoor/index/index/docId/478)

## 11.2 Parameter References

If you use one of the included parameter sets you must also add the proper reference for it.

**QUASINANO2015** A.F. Oliveira, P. Philipsen, T. Heine. *DFTB Parameters for the Periodic Table, Part 2: Energies and Energy Gradients from Hydrogen to Calcium*, Journal of Chemical Theory and Computation 11 (11), pp 5209–5218 (2015) (http://dx.doi.org/10.1021/acs.jctc.5b00702)

**QUASINANO2013.1** 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) (http://dx.doi.org/10.1021/ct4004959)

**Dresden (same origin as matsci-0-3 parameters in DFTB.org)** J. Frenzel, A. F. Oliveira, N. Jardillier, T. Heine, 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**

J. Frenzel, A. F. Oliveira, H. A. Duarte, T. Heine, G. Seifert, *Structural and electronic properties of bulk gibbsite and gibbsite surfaces*, Z. Anorg. Allg. Chem. 631, 1267-1271 (2005) (http://dx.doi.org/10.1002/chin.200529002)

L. Guimaraes, A. N. Enyashin, J. Frenzel, T. Heine, H. A. Duarte, G. Seifert, *Imogolite Nanotubes: Stability, electronic and mechanical properties*, Nano 1, 362-368 (2007) (http://dx.doi.org/10.1021/nn700184k)

R. Luschtinetz, A. F. Oliveira, J. Frenzel, J. Joswig, G. Seifert, H. A. Duarte, *Adsorption of phosphonic and ethylphosphonic acid on aluminum oxide surfaces*, Surf. Sci. 602, 1347-1359 (2008) (http://dx.doi.org/10.1016/j.susc.2008.01.035)

R. Luschtinetz, J. Frenzel, T. Milek, G. Seifert *Adsorption of phosphonic acid at the TiO2 anatase (101) and rutile (110) surface*, J. Phys. Chem. C 113, 5730-5740 (2009) (http://dx.doi.org/10.1021/jp8110343)

**DFTB.org** Required citations for the various DFTB.org parameter sets can be found on the official DFTB webpage: DFTB.org (http://www.dftb.org).

# CHAPTER TWELVE

# REFERENCES

## 12.1 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) (http://link.aps.org/doi/10.1103/PhysRevB.58.7260)

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) (http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1521-3951(200001)217:1%3C41::AID-PSSB41%3E3.0.CO;2-V/abstract)

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) (http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1521-3951(200001)217:1%3C357::AID-PSSB357%3E3.0.CO;2-J/abstract)

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) (http://www.rsc.org/publishing/journals/CP/article.asp?doi=b105782k)

T. Frauenheim, G. Seifert, M. Elstner, T. Niehaus, C. Kohler, M. Armkreutz, 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) (http://www.iop.org/EJ/abstract/0953-8984/14/11/313)

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) (http://pubs.acs.org/doi/abs/10.1021/ct100684s)

T. A. Niehaus, S. Suhai, F. Della Sala, P. Lugli, M. Elstner, G. Seifert, and Th. Frauenheim, *Tight-binding approach to time-dependent density-functional response theory*, Phys. Rev. B 63, 085108 (2001) (https://doi.org/10.1103/PhysRevB.63.085108)

D. Heringer, T. A. Niehaus, M. Wanko, Th. Frauenheim *Analytical excited state forces for the time-dependent density-functional tight-binding method*, J. Comput. Chem., 28: 2589–2601 (http://dx.doi.org/10.1002/jcc.20697)

## 12.2 DFTB: parameter sets

### 12.2.1 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) (http://dx.doi.org/10.1021/ct4004959)

### 12.2.2 QUASINANO2015

The DFTB parameter files in $ADFHOME/atomicdata/DFTB/QUASINANO2015 are distributed with the ADF package. The QUASINANO2015 parameter set extends the QUASINANO2013.1 parameter set, and includes terms that are needed to compute the total energy and its gradient.

A. F. Oliveira, P. Philipsen, T. Heine. *DFTB Parameters for the Periodic Table, Part 2: Energies and Energy Gradients from Hydrogen to Calcium*, Journal of Chemical Theory and Computation 11 (11), pp 5209–5218 (2015) (http://dx.doi.org/10.1021/acs.jctc.5b00702)

### 12.2.3 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 fr Anorganische und Allgemeine Chemie 631, 1267 (2005) (http://dx.doi.org/10.1002/zaac.200500051)

For construction and application of integral tables for Al-Si-O-H: L. Guimares, A. N. Enyashin, J. Frenzel, T. Heine, H. A. Duarte, and G. Seifert, *Imogolite Nanotubes: Stability, electronic and mechanical properties*, Nano 1, 362 (2007) (http://dx.doi.org/10.1021/nn700184k)

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) (http://dx.doi.org/10.1016/j.susc.2008.01.035)

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) (http://dx.doi.org/10.1021/jp8110343)

### 12.2.4 DFTB.org

For detailed information please visit the official DFTB webpage: www.dftb.org (http://www.dftb.org/parameters/). Detailed references of each specific parameter set are available in the corresponding *metainfo.yaml* file.

## 12.3 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

THIRTEEN

# KEYWORDS

# INDEX