

MODES: normal modes tool

Manual

Ben Hourahine

Contents

1	Introduction	2
2	Input for MODES	3
2.1	Main input	3
2.2	Geometry	4
2.3	SlaterKosterFiles	5
	Index	7

Chapter 1

Introduction

The MODES program calculates the vibrational modes using data created by a calculation performed with DFTB⁺.

Chapter 2

Input for MODES

The input file for MODES must be named `modes_in.hsd` and should be a Human-friendly Structured Data (HSD) formatted file.¹ The program can read the input in XML instead of HSD format if the input file is `modes_in.xml`. The input file must be present in the working directory. As with DFTB⁺ to prevent ambiguity, the parser refuses to read in any input if two types of input file are present.

The input file for MODES must contain the properties listed in the tables of the following sections. The first column of each table specifies the name of the property. The second column indicates the type of the expected value for that property. The letters “i”, “r”, “s”, “p”, “m” stand for integer, real, string, property list and method type, respectively. An optional prepended number specifies how often (if more than once) this type must occur. An appended “+” indicates arbitrary occurrence greater than zero, while “*” allows also for zero occurrence. Alternative types are separated by “|”. Parentheses serve only the delimitation of groups.

Sometimes a property is only interpreted if some conditions are met. If this is the case, the appropriate conditions are indicated in the third column. The fourth column contains the default value for the property. If no default value is specified (“-”), the user is required to assign a value to that property. The description of the properties immediately follows the table. If there is also a more detailed description available for a given keyword somewhere else, the appropriate page number appears in the last column.

2.1 Main input

The table below contains the list of the properties, which must occur in the input file `modes_in.hsd`:

Name	Type	Condition	Default	Page
Geometry	plm		-	4
Hessian	p		{}	6
SlaterKosterFiles	plm		-	5

Additionally optional definitions may be present:

Name	Type	Condition	Default	Page
DisplayModes	p		-	6
Atoms	i+lm		1:-1	

¹For the details of the HSD format, please consult the manual of the DFTB⁺ code at <http://www.dftb-plus.info>.

Geometry Specifies the geometry for the system to be calculated. See p. 4.

Hessian Contains the second derivatives matrix of the system energy with respect to atomic positions. See p. 6.

SlaterKosterFiles Name of the Slater-Koster files for every atom type pair combination. See 5.

DisplayModes Optional settings to plot the eigenmodes of the vibrations. See 6.

Atoms Optional list of atoms, ranges of atoms and/or the species of atoms for which the Hessian has been supplied.

2.2 Geometry

The geometry can be specified either directly by passing the appropriate list of properties or by using the `GenFormat{}` method.

2.2.1 Explicit geometry specification

If the geometry is being specified explicitly, the following properties can be set:

Periodic	1		No
LatticeVectors	9r	Periodic = Yes	-
TypeNames	s+		-
TypesAndCoordinates	(1i3r)+		-

Periodic Specifies if the system is periodic in all 3 dimensions or is to be treated as a cluster. If set to Yes, property `LatticeVectors{}` must be also specified.

LatticeVectors [*length*] The *x*, *y* and *z* components of the three lattice vectors if the system is periodic.

TypeNames List of strings with the names of the elements, which appear in your geometry.

TypesAndCoordinates [*relative|length*] For every atom the index of its type in the `TypeNames` list and its coordinates. If for a periodic system (`Periodic = Yes`) the modifier *relative* is specified, the coordinates are interpreted in the coordinate system of the lattice vectors.

Example: Geometry of GaAs:

```
Geometry = {
  TypeNames = { "Ga" "As" }
  TypesAndCoordinates [Angstrom] = {
    1 0.000000  0.000000  0.000000
    2 1.356773  1.356773  1.356773
  }
  Periodic = Yes
  LatticeVectors [Angstrom] = {
    2.713546  2.713546  0.
    0.        2.713546  2.713546
    2.713546  0.        2.713546
  }
```

```
}  
}
```

2.2.2 GenFormat{}

You can use the generic format to specify the geometry (see the DFTB⁺ manual). The geometry specification for GaAs would be the following:

```
Geometry = GenFormat {  
  2 S  
  Ga As  
  1 1  0.000000  0.000000  0.000000  
  2 2  1.356773  1.356773  1.356773  
  0.000000  0.000000  0.000000  
  2.713546  2.713546  0.  
  0.        2.713546  2.713546  
  2.713546  0.        2.713546  
}
```

It is also possible to include the gen-formatted geometry from a file:

```
Geometry = GenFormat {  
  <<< "geometry.gen"  
}
```

2.3 SlaterKosterFiles

The Slater-Koster files are required to determine the masses of the atoms for the vibrational modes. There are two different ways to specify the Slater-Koster files for the atom type pairs, explicit specification and using the `Type2FileNames{}` method.

Explicit specification

Every possible atom type pair connected by a dash must occur as property with the name of the corresponding file as assigned value.

Example (GaAs):

```
SlaterKosterFiles = {  
  Ga-Ga = "./Ga-Ga.skf"  
  Ga-As = "./Ga-As.skf"  
  As-Ga = "./As-Ga.skf"  
  As-As = "./As-As.skf"  
}
```

If you treat shells from different species as shells of one atom by using the `SelectedShells{}` keyword in the `MaxAngularMomentum{}` block, you have to specify more than one file name for certain specie pairs. (For details see the description about the `MaxAngularMomentum{}` keyword.)

Type2FileNames{}

You can use this method to generate the name of the Slater-Koster files automatically using the element names from the geometry input. You have to specify the following properties

Prefix	s	""
Separator	s	""
Suffix	s	""
LowerCaseTypeName	1	No

Prefix Prefix before the first type name, usually the path.

Separator Separator between the type names.

Suffix Suffix after the name of the second type, usually extension.

LowerCaseTypeName If the name of the types should be converted to lower case. Otherwise they are used in the same way, as they were specified in the geometry input.

Example (for producing the same file names as in the previous section):

```
SlaterKosterFiles = Type2FileNames {  
  Prefix = "./"  
  Separator = "-"  
  Suffix = ".skf"  
  LowerCaseTypeName = No  
}
```

The Type2FileNames method can not be used, if an extended basis was defined with the SelectedShells method.

2.3.1 Hessian{}

Contains the second derivatives of the energy supplied by DFTB⁺. The derivatives matrix must be stored as the following order: For the i, j and k directions of atoms $1 \dots n$ as

$$\frac{\partial^2 E}{\partial x_{i1} \partial x_{i1}} \quad \frac{\partial^2 E}{\partial x_{j1} \partial x_{i1}} \quad \frac{\partial^2 E}{\partial x_{k1} \partial x_{i1}} \quad \frac{\partial^2 E}{\partial x_{i2} \partial x_{i1}} \quad \frac{\partial^2 E}{\partial x_{j2} \partial x_{i1}} \quad \frac{\partial^2 E}{\partial x_{k2} \partial x_{i1}} \quad \dots \quad \frac{\partial^2 E}{\partial x_{kn} \partial x_{kn}}$$

Note: for supercell calculations, the modes are currently obtained at the $\mathbf{q} = 0$ point, irrespective of the k-point sampling used.

2.3.2 DisplayModes{}

Allows the eigenvectors of the system to be plotted out if present

PlotModes	i+lm	1:-1
Animate	1	Yes
XMakeMol	1	Yes

PlotModes Specifies list of which eigenmodes should be plotted as xyz files. Remember that there are $3N$ modes for the system (including translation and rotation).

Animate Produce separate animation files for each mode or a single file multiple modes where the mode vectors are marked for each atom.

XMakeMol Adapt xyz format output for XMakeMol dialect xyz files.

Index

Animate, [6](#)

Atoms, [3](#)

DisplayModes, [3](#)

DisplayModes{}, [6](#)

GenFormat{}, [5](#)

Geometry, [3](#), [4](#)

Hessian, [6](#)

Hessian, [3](#)

Hessian{}, [6](#)

LatticeVectors, [4](#)

LowerCaseTypeName, [6](#)

Periodic, [4](#)

PlotModes, [6](#)

Prefix, [6](#)

Separator, [6](#)

SlaterKosterFiles, [3](#), [5](#)

Suffix, [6](#)

Type2FileNames{}, [6](#)

TypeNames, [4](#)

TypesAndCoordinates, [4](#)

XMakeMol, [6](#)