OpenDFT Documentation

Jannick Weisshaupt

Nov 26, 2018

Contents:

OpenDFT is a free and open source software that brings cutting edge solid state research to the people. It is a graphical program that interacts with various scientific terminal based solid state software packages. It visualizes inputs, such as the crystal structure and outputs, such as band structures and optical spectra and can start and control calculations, which are done by the respective scientific solid state package. OpenDFT is designed to be DFT engine agnostic so that you can easily switch between different scientific codes and compare results seamlessly.

CHAPTER 1

Introduction into OpenDFT

OpenDFT is a free and open source software that brings cutting edge solid state research to the people. It is a graphical program that interacts with various scientific terminal based solid state software packages. It visualizes inputs, such as the crystal structure and outputs, such as band structures and optical spectra and can start and control calculations, which are done by the respective scientific solid state package. OpenDFT is designed to be DFT engine agnostic so that you can easily switch between different scientific codes and compare results seamlessly.

CHAPTER 2

How to install OpenDFT

OpenDFT is very easy to install on its own. Just follow the steps below and it should work. OpenDFT relies on DFT codes, e.g. exciting, abinit etc. that are installed on your system. They must be available in the terminal that start OpenDFT with their standard name, i.e. `excitingser` for exciting, abinit for abinit etc., otherwise they cannot be used.

- Download the latest release from the OpenDFT release page and save it on your work station.
- Extract the archive (for linux use the .tar.gz one) somewhere on your machine
- Add the OpenDFT executable in the OpenDFT directory to your PATH variable or make an alias, e.g for bash open the ~/.bashrc file and add: `alias OpenDFT=/home/user/path/to/OpenDFT/OpenDFT`
- Reload the environmental variable of the terminal with `source ~/.bashrc` or simply restart the terminal.
- Type `OpenDFT` in your terminal to start OpenDFT

If OpenDFT does not start and some errors appear in the terminal try to install qt4 on your system, for e.g. ubuntu use **sudo apt install qt4-default**.

CHAPTER 3

DFT codes installation

OpenDFT is a graphical user interface to command line quantum chemistry and/or DFT codes. As such it requires those to be installed and accessible through their standard names in the terminal that started OpenDFT, i.e. through the environmental variables supplied to OpenDFT. Below is a "how to install" list of the supported codes.

3.1 Nwchem

`sudo apt-get install nwchem` or download the nwchem-release and follow these instructions Afterwards the command `nwchem` should work in the terminal.

3.2 Quantum espresso

`sudo apt-get install quantum-espresso` or download

CHAPTER 4

Solid state tools

These tools are used within OpenDFT to represent objects from solid state physics. They can be used for scripting within OpenDFT

class src.solid_state_tools.**CrystalStructure**(*lattice_vectors*, *atoms*, *relative_coords=True*, *scale=1.0*)

This class represents a crystal structure

Parameters

- **lattice_vectors** – a 3x3 numpy array with the lattice vectors as rows
- **atoms** – a Nx4 numpy array the atomic positions in [:,:3] as either relative (crystal) or cartesion coordinates
- **relative_coords** –
- **scale** – A scale of the whole unit cell. This parameter is only used to nicely format the lattice vectors. The lattice_vectors argument must still contain the correct unit vectors.

calc_absolute_coordinates(*repeat=(1, 1, 1)*, *offset=(0, 0, 0)*, *edges=False*)

Returns the cartesion coordinates

Parameters

- **repeat** – 3 element tuple which determines the number of repeated unit cells
- **offset** – optional offset from (0,0,0)
- **edges** – Determines whether atoms on the faces or edges should be plotted only ones (->False) or multiple times (->True)

Returns cartesian coordinates of the atoms

Return type Nx4 numpy array

find_bonds(*abs_coords*)

Searches for bonds between all atoms. This function uses literature values for the covalent bond radii.

Parameters **abs_coords** – cartesion coordinates of the atoms (Nx4) as returned by calc_absolute_coordinates

Returns Tuple of all bonds between the atoms by their index.

Return type A tuple of two-tuples, e.g. ((3,5),(1,2))

class src.solid_state_tools.**MolecularStructure**(*atoms*, *scale=1.0*)

Represents a molecular structure :param atoms – a (Nx4) numpy array with [x,y,z,type] as rows. Type is an integer with the atomic number (H=1,He=2 etc.).

calc_absolute_coordinates(*repeat=[1, 1, 1], edges=False*)

Returns the cartesion coordinates

find_bonds(*abs_coords*)

Returns a list of bonds between atom i and j, i.e. ((i,j),...). The bonds are calculated according to covalent radii from the literature.

return bonds a tuple of of two element tuples which represent the connections or bonds

symmetry_information()

Returns symmetry information of the molecule as for example its point group

Return info a dictionary with symmetry informations

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index

C

F

M

S