Cgenarris Documentation

Marom group

June 9, 2019

Contents

1 Features

- Support for special positions.
- Parallelized using OpenMP.
- Fast and efficient structure checking.
- Can generate in all possible Z (or NMPC) with Z' <= 1.

2 Installation

2.1 Cgenarris

Requirements

Any C compiler which supports ANSI C99 / GNU99 standard.

Using Makefile

1. Uncompress the tar file.
2. Execute ’make cgenarris’.
3. This will create cgenarris.x which is the desired executable.

NOTE:

1. You may change the C compiler using the environment variable CC. You may also uncomment the first line of the makefile and set the compiler.
2. Remove object files using ’make clean’

2.2 Pygenarris

Requirements

1. Any C compiler which supports ANSI C99 / GNU99 standard.
2. SWIG (Simplified Wrapper code and Interface Generator).
3. Numpy
4. Distutils for installation through setup.py

Method 1: Using Makefile

1. Uncompress the tar file.
2. Paste the location of Python.h headerfile in the Makefile. (for Anaconda v3.7 it should be ’ anaconda/include/python3.7m/ ’)
3. Execute ’make pygenarris’.
4. This will create pygenarris.so library from which you can import pygenarris.

Method 2: Using Distutils

1. Uncompress the tar file.
2. Execute ' python setup.py build_ext --inplace'.
3. This will create pygenarris.so library from which you can import pygenarris.

NOTE

1. You may change the C compiler using the environment variable CC.
2. pygenarris can be configured with both python2 and python3.

3 Pygenarris

Pygenarris is a python API for C structure generator and associated functions.

3.1 Generate a pool of random molecular crystals

generate_molecular_crystals(filename, num_structures, Z, volume_mean, volume_std, sr, tol, max_attempts)

Description

Generate random molecular crystals by space groups. First, the generator first identifies space groups that are compatible with molecular symmetry and given number of molecules in the unit cell. Structures are generated sequentially from lowest space group to the highest. Cell volumes are sampled from a normal distribution. The attempted structures are checked for closeness of molecules. If an atom of a molecule is too close to its own periodic image or another atom of a different molecule in a cell, the structure is discarded. The closeness checks are controlled by the specific radius proportion (sr). If the generation of a space group fails after max_attempts times, the generator moves to the next higher space group. The generated structures are printed to the file in FHI-aims geometry.in format.

Input

1. Geometry of the molecule is read from the file *geometry.in* from the working directory.
2. *filename* is the name of the file to which generated structures are printed. Type: string
3. *num_structures* is the number of structures from each space group. Type: integer
4. *Z* number of molecules in the conventional cell. Type: integer
5. *volume_mean* is the mean of the normal distribution from which volume is sampled. Type: float
6. *volume_std* is the standard deviation of the volume distribution. Type: float
7. *sr* is specific radius proportion. See Genarris paper for definition. Type: float
8. *tol* is the tolerance for special position generation and space group detection. Type: float
9. *max_attempts* is the maximum number of attempts before moving to the next space group. Type: integer
10. The number of threads can be set by using the environment variable *OMP_NUM_THREADS.*

Output

1. A file with all the generated structures in FHI-aims geometry format.

3.2 Identification of compatible space groups given molecular symmetry

find_allowed_positions_using_molecular_symmetry(point_group, Z, Z")

Description

This function finds the compatible space group positions using molecule's point group.

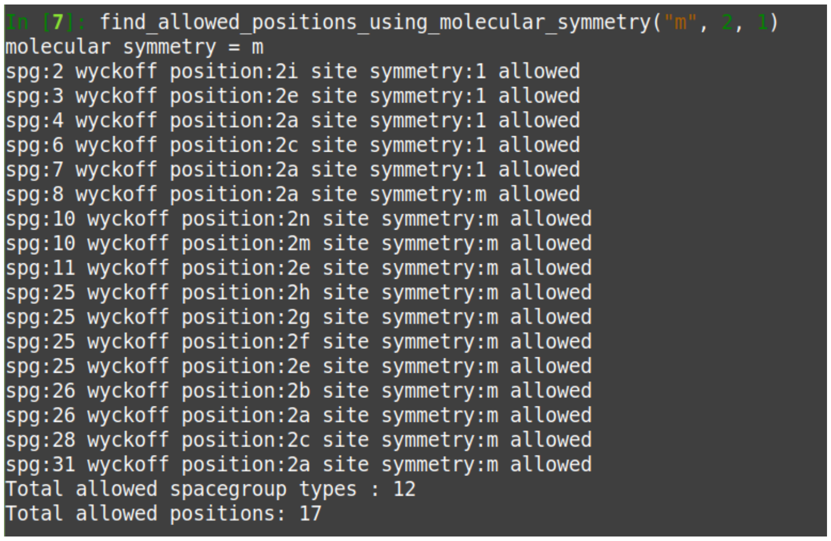
Input

1. *point_group* is the point_group of the molecule. Eg: "mmm" for tetracene. Type: String.
2. *Z* is the number of molecules in the conventional cell. Type: integer
3. *Z"* is the number of inequivalent molecules in the cell. *Not implemented! Set any integer.*

Output

1. Compatible Wyckoff positions and space groups are printed.

Example



```
In [7]: find_allowed_positions_using_molecular_symmetry("m", 2, 1)
molecular symmetry = m
spg:2 wyckoff position:2i site symmetry:1 allowed
spg:3 wyckoff position:2e site symmetry:1 allowed
spg:4 wyckoff position:2a site symmetry:1 allowed
spg:6 wyckoff position:2c site symmetry:1 allowed
spg:7 wyckoff position:2a site symmetry:1 allowed
spg:8 wyckoff position:2a site symmetry:m allowed
spg:10 wyckoff position:2n site symmetry:m allowed
spg:10 wyckoff position:2m site symmetry:m allowed
spg:11 wyckoff position:2e site symmetry:m allowed
spg:25 wyckoff position:2h site symmetry:m allowed
spg:25 wyckoff position:2g site symmetry:m allowed
spg:25 wyckoff position:2f site symmetry:m allowed
spg:25 wyckoff position:2e site symmetry:m allowed
spg:26 wyckoff position:2b site symmetry:m allowed
spg:26 wyckoff position:2a site symmetry:m allowed
spg:28 wyckoff position:2c site symmetry:m allowed
spg:31 wyckoff position:2a site symmetry:m allowed
Total allowed spacegroup types : 12
Total allowed positions: 17
```