pyriodic

Release 0.2.1

Feb 15, 2020

Contents:

pyriodic is an in-development library to handle a database of three-dimensional structures. It also supports several simple manipulations of structures.

CHAPTER 1

Installation

Install *pyriodic* from source on github:

```
pip install git+https://github.com/klarh/pyriodic.git#egg=pyriodic-structures
```

By default, pyriodic only ships with a few very simple structures; other libraries can be added by installing other packages, such as pyriodic-aflow, which contains structures from the AFLOW project.

CHAPTER 2

API Reference

class pyriodic.**Database**

Manage an in-memory database of structures

Database objects wrap a sqlite database containing structure information. Structures can be added to and read from the database.

Databases should only be written to by a single thread at once.

Currently the only table populated in the database is *unit_cells*, with the fields:

- name (str): Short name of the structure type
- space_group (int): Integer representation of the space group of the structure
- size (int): Number of particles in the unit cell
- structure (`Structure`): Structure object

insert_unit_cell(*name*, *space_group*, *structure*, *cursor=None*)

Insert a unit cell into this database object

Parameters

- **name** – Short name of the structure
- **space_group** – Integer representation of the space group for the structure
- **structure** – `Structure` object to store
- **cursor** – Database connection cursor (optional)

classmethod make_standard()

Generate the standard database from all installed packages

query(*query*, **args*)

Execute a (sqlite) query on the database

Parameters are the same as for an *sqlite3* database.

class pyriodic.**Structure**(*positions*, *types*, *box*)

Container for a single set of coordinates

Structure objects hold all of the important quantities for a structural example, like coordinates and the system box.

add_gaussian_noise(*magnitude*)

Add gaussian noise to each particle

Parameters **magnitude** – Scale of the zero-mean gaussian nose

Returns A new *Structure* with the gaussian noise applied.

replicate(*nx=1*, *ny=1*, *nz=1*)

Replicate the system a given number of times in each dimension

Parameters

- **nx** – Number of times to replicate in the x direction
- **ny** – Number of times to replicate in the y direction
- **nz** – Number of times to replicate in the z direction

Returns A new *Structure* that has been replicated appropriately

replicate_upto(*N_target*)

Replicate the system to have at least a given number of particles

Replicas are iteratively added in the shortest dimension of the box until at least *N_target* particles are present.

Parameters **N_target** – Minimum number of particles to have in the resulting structure

Returns A new *Structure* that has been replicated appropriately

rescale_linear(*factor*)

Rescale all distances in the system by the given factor

The coordinates and box are scaled by the given factor.

Parameters **factor** – Number to scale all lengths in the system by

Returns a new *Structure* that has been scaled accordingly

rescale_number_density(*phi*)

Rescale the system to the given number density

The box and all coordinates are scaled by an appropriate factor to produce a box with the given number density (number of particles/volume).

Parameters **phi** – Number density of the resulting system

Returns a new *Structure* with the given density

rescale_shortest_distance(*l*)

Rescale the system to have the given shortest distance between points

The box and all coordinates are scaled by an appropriate factor to produce a system with the given shortest distance between any two points. This method is currently N^2 in the number of points, but may be improved in the future.

Parameters **l** – Shortest distance of the resulting system

Returns a new *Structure* with the given shortest distance

rescale_volume(*V*)

Rescale the system to the given volume

The box and all coordinates are scaled by an appropriate factor to produce a box with the given volume.

Parameters **V** – Volume of the resulting system

Returns a new *Structure* with the given volume

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index