
Order Documentation

Release 0.0.1

Pu Du

Jul 13, 2017

Contents

1	order package	1
1.1	Submodules	1
1.2	order.XYZ module	1
1.3	order.avc module	1
1.4	order.lsi module	2
1.5	order.order module	2
1.6	order.oto module	2
1.7	order.plot module	2
1.8	order.tto module	2
1.9	order.util module	3
1.10	Module contents	3
2	Order	5
2.1	Getting started	5
	Python Module Index	7

Submodules

order.XYZ module

XYZ trajectory reader

```
class order.XYZ.XYZLoader(filename)  
    Bases: object  
  
    close()  
        close XYZ file if it was open  
  
    read_all_frames()  
        read all frames of XYZ trajectory  
  
    read_n_frames()  
        get the starting position of each frame  
  
    read_next_frame(frame)  
        read a frame of XYZ trajectory
```

order.avc module

```
class order.avc.VoronoiCell(filename, center, bins=100)  
    Bases: order.oto.Orientational  
  
    asphericity of the Voronoi cell  
  
    asphericity(freq=1)  
        compute asphericity of the Voronoi cell
```

compute_vc (*points*)
compute the Voronoi cell

polyhedron (*coords, j, L*)
find the polyhedron for center molecule

wrap_box (*c_coord, coords, L*)
wrap the simulation box

order.lsi module

order.order module

`order.order.command_line_runner()`
`order.order.get_parser()`

order.oto module

Oriental Tetrahedral Order q

class `order.oto.Oriental` (*trajectory, center, bins=100*)
Bases: `object`

orientational tetrahedral order parameter

four_neighbors (*coords, L*)
compute four nearest water oxygen neighbors

orientational_param (*freq=1*)
compute orientational order parameter

out_put (*taskname='OTO', param_name='Q'*)
output raw data and distribution

order.plot module

order.tto module

Translational Tetrahedral Order Sk

class `order.tto.Translational` (*filename, center, bins=100*)
Bases: `order.oto.Oriental`

translational tetrahedral order parameter

translational_param (*freq=1*)
compute translational order parameter

order.util module

`order.util.cos_angle(v1, v2)`
compute the cos angle of two giving vectors

`order.util.output_end(t_start, t_end)`
print total running time

`order.util.output_system_info(filename, n_atoms, n_frames)`
print system information

`order.util.output_task(name, freq, bins, center)`
print task information

`order.util.output_welcome()`
print welcome information

`order.util.pbc(dx, dy, dz, L)`
periodic boundary conditions

Module contents

Order : A tool to characterize the local structure of liquid water by geometric order parameters.

Getting started

Installation:

```
$ pip install iorder
```

Resources

- [GitHub](#)
- [Issues](#)

O

- `order`, 3
- `order.avc`, 1
- `order.lsi`, 2
- `order.order`, 2
- `order.oto`, 2
- `order.plot`, 2
- `order.tto`, 2
- `order.util`, 3
- `order.XYZ`, 1

A

asphericity() (order.avc.VoronoiCell method), 1

C

close() (order.XYZ.XYZLoader method), 1
command_line_runner() (in module order.order), 2
compute_vc() (order.avc.VoronoiCell method), 1
cos_angle() (in module order.util), 3

F

four_neighbors() (order.oto.Orientational method), 2

G

get_parser() (in module order.order), 2

O

order (module), 3
order.avc (module), 1
order.lsi (module), 2
order.order (module), 2
order.oto (module), 2
order.plot (module), 2
order.tto (module), 2
order.util (module), 3
order.XYZ (module), 1
Orientational (class in order.oto), 2
orientational_param() (order.oto.Orientational method), 2
out_put() (order.oto.Orientational method), 2
output_end() (in module order.util), 3
output_system_info() (in module order.util), 3
output_task() (in module order.util), 3
output_welcome() (in module order.util), 3

P

pbv() (in module order.util), 3
polyhedron() (order.avc.VoronoiCell method), 2

R

read_all_frames() (order.XYZ.XYZLoader method), 1

read_n_frames() (order.XYZ.XYZLoader method), 1
read_next_frame() (order.XYZ.XYZLoader method), 1

T

Translational (class in order.tto), 2
translational_param() (order.tto.Translational method), 2

V

VoronoiCell (class in order.avc), 1

W

wrap_box() (order.avc.VoronoiCell method), 2

X

XYZLoader (class in order.XYZ), 1