

# User manual for op\_tools ver 0.1

Hideo Doi

Feb 2019

## 目 次

What is this?	2
How to install	2
必要なデータ	2
Common settings for all order parameters	2
Neighborhood particle setting . . . . .	2
Setting for neighborhood radius . . . . .	2
Setting for the number of neighborhood particles . . . . .	3
Setting for the number of times for averaging . . . . .	3
Coordinates . . . . .	4
各オーダーパラメータの説明	5
<i>S</i> : Onsager' s parameter . . . . .	5
<i>T</i> : McMillan' s Sigma . . . . .	5
<i>A</i> : neighborhood parameters . . . . .	6
<i>B</i> : bond angle analysis (BAA) . . . . .	7
<i>C</i> : centrosymmetry parameter analysis (CPA) . . . . .	7
<i>D</i> : neighbor distance analysis (NDA) . . . . .	8
<i>F</i> : angular Fourier series (AFS) like parameter . . . . .	8
<i>H</i> : angle histogram analysis (AHA) . . . . .	9
<i>I</i> : tetrahedron order parameter (TOP) . . . . .	9
<i>Q</i> : Bond order parameter . . . . .	10
<i>W</i> : Bond order parameter . . . . .	11
<i>Z</i> : user define order parameter . . . . .	13
大量の解析を行う設定	13
出力のフォーマット	14
Reference	14

## What is this?

python module for order parameter analysis for molecular and particle simulation. this module evaluate the particle environment as the value. In molecular simulation, there are often cases where you want to mechanically determine the structure of a particle. For example, when melting simulating of solid metal, you may want to visualize how it melts. You can use such a module at such a time.

## How to install

```
$ cd order_tools
$ pip install -e .
```

## 必要なデータ

1. coord : List of coordinates of each particles. Such as [ [0,0,0], [0,0,1], ... ] like [[x1, y1, z1], [x2, y2, z2], ... ]
2. direct : Direction vector of each particle. It may be a three-dimensional unit vector or a quaternion. Such as [[1,0,0], ...] or [[1,0,0,0], ...]  
In the case of a three-dimensional unit vector, the order is [x, y, z]. In the case of quaternions, the order is [w, x, y, z]. In the case of a mass point or spherical particle, It has no direction vector. There is no problem with direct = [].
3. box\_length : The length of the simulation box. Such as [3, 3, 3]
4. op\_settings : Settings for calculating each order parameter

## Common settings for all order parameters

### Neighborhood particle setting

The first method is to set a particle whose distance from particle  $i$  is within a certain distance (neighborhood radius) as a neighborhood particle of particle  $i$ . The advantage of this method is that it is relatively easy to implement and easy for users to understand. The disadvantage is that the number of nearby particles can be different for each particle, and the number of nearby particles can vary depending on the unit of distance. For example, specifying the neighborhood radius as 1 nm, 1 angstrom will change the result. This is called scale dependency. Another method is to set  $n$  particles close to the particle  $i$  as neighborhood particles. The advantage is that neighborhood particles do not change because there is no scale dependency, and even if the unit of length changes, the magnitude relationship of distances does not change. The disadvantages are that it is relatively difficult to implement, the computational cost is relatively high (> 10%), and there are few usage records in articles.

### Setting for neighborhood radius

Specify calculation conditions for order parameters.

```
op_settings = {
    'radius' : [1.5, 1.75] ,
    ... ( other parameter settings ) }
```

User can specify in the list format, and it is also possible to specify multiple conditions at the same time. In the above example, the radius setting is 1.5 and 1.75. ..

Setting for the number of neighborhood particles

Specify calculation conditions for order parameters.

```
op_settings = {
    'neighbor' : [8, 12] ,
    ... ( other parameter settings ) }
```

User can specify in the list format, and it is also possible to specify multiple conditions at the same time. In the above example, the number of neighborhood particles setting is 8 and 12. ..

Setting for the number of times for averaging

When the classification performance of the order parameter of each particle does not enough performance. To increase the performance, user can perform the averaging calculation. The average is calculated with the order parameter of particle  $i$  and the order parameter of neighborhood particles of particle  $i$ .  $N_b(i)$  is a list of  $N$  neighborhood particles of particle  $i$ .  $\tilde{N}_b(i)$  is a list of  $N + 1$  particles that  $i$  itself is added to  $N_b(i)$ . The order parameter of the particles  $i$  before averaging is  $X(i)$  and the order parameter after averaging is  $Y(i)$ . The following calculation is performed.

$$Y(i) = \frac{\sum_{j \in \tilde{N}_b(i)} X(j)}{N + 1}$$

This averaging calculation has the following advantages and disadvantages. The advantage is that the calculation cost is low, and it is possible to improve the classification performance of the same order parameter, Any numerical value can be used, and it can be averaged many times. On the other hand, the disadvantage is that more information is used to calculate the order parameters, and the resolution of the order parameters is decreased. The order parameters are calculated using the information of neighborhood particles, and the averaging calculation is calculated using the order parameters of the neighborhood particles. Each time the averaging operation is performed, the information on neighborhood particles of neighborhood particles is used. In other words, more particles are used to calculate order parameters, and information on far particles is used. This is not a problem in homogeneous systems, but it is a problem in inhomogeneous systems (such as interfaces). In inhomogeneous systems, it is likely to be used in many cases for the purpose of wanting to know the position where the abnormality is occurring or want to know the position of the interface, etc. Using distant particle information leads the indistinctness of position.

The number of times of this averaging is specified by op\_settings specifying the calculation condition of the order parameter as follows. Note that this value starts from 0, and in the case of 0, averaging is not performed.

```
op_settings = {
    'ave_times' : 1,
    ... ( other parameter settings )
}
```

## Coordinates

This software was developed to analyze particles (eg, ellipsoids, rods, arrows, disks, etc.) that have direction information, not mass points or spherical particles. Of course, analysis of mass points and spherical particles is also possible. If it is not a mass point or spherical particle, the coordinates you want to use for analysis may change. For example, in the case of rod-like particles, it may be considered that analysis is desired to be performed at the coordinates of the center of the particle, or analysis is desired to be performed at the coordinates of the tip. Also, the coordinates of one particle  $i$  may be considered to be the center, and the coordinates of the neighborhood particle  $j$  may be the tip. Therefore, `op_tools` uses the following formula for particle coordinates.

$$r^{(o_i)}(i) = r(i) + o_f o_i u(i)$$

The vector  $r^{(o_i)}(i)$  is the coordinate vector of the particle  $i$  for analysis, The vector  $r(i)$  is a coordinate vector of the particle  $i$  given as an argument to the python module. The variable  $o_f$  is a coefficient for the direction vector, the variable  $o_i$  is a variable indicating the position at the particle  $i$ , and the vector  $u(i)$  is unit direct vector of the particle  $i$ . The variable  $o_f$  represents the strength of the direction vector used throughout the analysis. The variable  $o_i$  is used to represent the position of the coordinate vector of the particle  $i$ . Usually, it looks like  $o_i \in \{1, 0, -1\}$ , and when  $o_f = 1, o_i = 1$ , the vector  $r^{(o_i)}(i)$  is the coordinates of the tip of the direction vector. Also, when  $o_f = 1, o_i = -1$ , the vector  $r^{(o_i)}(i)$  is the coordinate of the end of the direction vector. In this way, it is possible to switch which coordinate of the direction vector to use. In the case of a mass point or a spherical particle, it may be set as  $o_f = 0$  and  $o_i = 0$ .

In the case of an ellipsoid, specify the calculation condition of order parameter in `op_settings` as follows.

ベクトル  $r^{(o_i)}(i)$  は、解析に使用する粒子  $i$  の座標ベクトル、ベクトル  $r(i)$  は、python モジュールに引数として与えられた粒子  $i$  の座標ベクトルである。変数  $o_f$  は方向ベクトルの係数で、変数  $o_i$  は粒子  $i$  での位置を示す変数、ベクトル  $u(i)$  は粒子  $i$  の持つ方向ベクトルである。変数  $o_f$  は解析全体で用いる方向ベクトルの強さを表現している。変数  $o_i$  は、粒子  $i$  の座標ベクトルの位置を表現するために用いている。通常、 $o_i \in \{1, 0, -1\}$  のようになっており、 $o_f = 1, o_i = 1$  の時は、ベクトル  $r^{(o_i)}(i)$  は方向ベクトルの先端の座標になる。また、 $o_f = 1, o_i = -1$  の時は、ベクトル  $r^{(o_i)}(i)$  は方向ベクトルの後端の座標になる。このように、方向ベクトルのどの座標を使うか切り替える事ができる。尚、質点や球状粒子の場合は、 $o_f = 0, o_i = 0$  とすれば良い。

楕円体の場合は、オーダーパラメータの計算条件を指定する `op_settings` で次のように指定を行う。

```
op_settings = {
    ... ( 各オーダーパラメータの設定 ) ...
    'o_factor'      : [1.0 ],
    'oi_oj'         : [1, 0, -1],
}
```

この例では、座標として粒子の中心座標を入力とした場合、楕円体の先端、中心、後端の3通りを座標として使用したい場合の設定である。

質点または球状粒子の場合は、オーダーパラメータの計算条件を指定する `op_settings` で次のように指定を行う。

```
op_settings = {
    ... ( 各オーダーパラメータの設定 ) ...
```

```

'o_factor'      : [0],
'oi_oj'         : [0],
}

```

また、粒子  $i$  の座標ベクトル  $r^{(o_i)}(i)$  から粒子  $j$  の座標ベクトル  $r^{(o_j)}(j)$  は以下のように計算する。

$$r^{(o_i, o_j)}(i, j) = r^{(o_j)}(j) - r^{(o_i)}(i)$$

## 各オーダーパラメータの説明

$S$  : Onsager' s parameter

Onsager のオーダーパラメータ  $S$  は次の式で計算される。[1][2]

$$S^{(a=1, n)}(i) = \frac{\sum_{j \in N_b(i)} P_n(\cos(\theta))}{N}, \text{even.}$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。関数  $P_n$  は Legendre polynomial である。変数  $n$  は Degree of the polynomial である。変数  $\theta$  は、粒子  $i$  の持つ方向ベクトル  $u(i)$  と粒子  $j$  の持つ方向ベクトル  $u(j)$  との角度である。変数  $\cos(\theta)$  は通常、 $u(i) \cdot u(j)$  で計算される。

$n = 2, 4$  の時、オーダーパラメータ  $S$  はそれぞれ以下の式で計算される。

$$S^{(a=1, n=2)}(i) = \frac{\sum_{j \in N_b(i)} [3 \cos^2(\theta) - 1]/2}{N}$$

$$S^{(a=1, n=4)}(i) = \frac{\sum_{j \in N_b(i)} [35 \cos^4(\theta) - 30 \cos^2(\theta) + 3]/8}{N}$$

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの設定として以下の例を示す。尚、このオーダーパラメータは計算のために方向ベクトルが必須であるため、質点や球状粒子の解析に使う事はできない。

```

op_settings = {
  'neighbor': [12],
  'radius': [2.0],
  'ave_times': 1,
  'oi_oj': [0],
  'o_factor': [0.00],
  'n_in_S' : [2],
  'analysis_type' : ['S']
}

```

$T$  : McMillan' s Sigma

McMillan のパラメータ  $T$  は次の式で計算される。[3]

$$T^{(a=1, n, o_i, o_j)} = \frac{\sum_{j \in N_b(i)} \cos(2\pi z^{(o_i, o_j)}(i, j)/d) P_n(\cos(\theta))}{N}$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。関数  $P_n$  は Legendre polynomial である。変数  $n$  は Degree of the polynomial である。変数  $\theta$  は、粒子  $i$  の持つ方向ベクトル  $u(i)$  と粒子  $j$  の持つ方向ベクトル  $u(j)$  との角度である。変数  $\cos(\theta)$  は通常、 $u(i) \cdot u(j)$  で計算される。変数  $z^{(o_i, o_j)}(i, j)$  は、ベクトル  $r^{(o_i)}(i)$  を通過し、ベクトル  $u(i)$  と垂直な平面  $P$  から、ベクトル  $r^{(o_j)}(j)$  までの距離である。変数  $d$  は、層から層までの距離である。

$n = 2$  の時、オーダーパラメータ  $T$  は以下の式で計算される。

$$T^{(a=1, n=2, o_i, o_j)} = \frac{\sum_{j \in N_b(i)} \cos(2\pi z^{(o_i, o_j)}(i, j)/d) [3 \cos^2(\theta) - 1]/2}{N}$$

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。尚、このオーダーパラメータは計算のために方向ベクトルが必須であるため、質点や球状粒子の解析に使う事はできない。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'oi_oj': [0],
    'o_factor': [0.00],
    'n_in_T': [2],
    'd_in_T': [1.0],
    'analysis_type' : ['T']
}
```

$A$  : neighborhood parameters

common neighborhood parameter  $A$  [4]、predominant common neighborhood parameter  $P$  [5]、another predominant common neighborhood parameter  $N$  [5] は非常に似ている式で計算される。 $P$  と  $N$  に関しては、 $A$  の一種として実装されている。

$$A_i^{(a=1, type=A, m, o_i, o_j, o_k)} = A_i^{(a=1, m, o_i, o_j, o_k)} = \frac{1}{N} \sum_{j \in N_b(i)} \left| \sum_{k \in N_b(i, j)} \{r^{(o_i, o_k)}(i, k) + r^{(o_j, o_k)}(j, k)\} \right|^2$$

$$A_i^{(a=1, type=P, m, o_i, o_j, o_k)} = P_i^{(a=1, m, o_i, o_j, o_k)} = \frac{1}{N} \sum_{j \in N_b(i)} \left| \sum_{k \in N_b(i, j)} \{r^{(o_i, o_k)}(i, j) + r^{(o_j, o_k)}(k, j)\} \right|^2$$

$$A_i^{(a=1, type=N, m, o_i, o_j, o_k)} = N_i^{(a=1, m, o_i, o_j, o_k)} = \frac{1}{N} \left| \sum_{j \in N_b(i)} \sum_{k \in N_b(i, j)} \{r^{(o_i, o_k)}(i, j) + r^{(o_j, o_k)}(k, j)\} \right|^2$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $N_b(i, j)$  は粒子  $i$ , 粒子  $j$  から近い  $m$  個の粒子のリストである。変数  $type$  で計算を行うオーダーパラメータの種類を指定する変数である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```
op_param = {
    'neighbor': [12],
    'radius': [1.5],
    'ave_times': 1,
    'oi_oj': [0],
}
```

```

'o_factor': [0.00],
'm_in_A': [2],
'types_in_A': ['A', 'P', 'N'],
'analysis_type': ['A']}

```

$B$  : bond angle analysis (BAA)

bond angle analysis  $B$  は次の式で計算される。 [6]

$$B_i^{(a=1,m,n,\phi,o_i,o_j,o_k)} = \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} f^{(m,n,\phi,o_i,o_j,o_k)}(\theta^{(o_i,o_j,o_k)}(j,i,k))$$

$$f^{(m,n,\phi,o_i,o_j,o_k)}(\theta(j,i,k)) = \cos^n(m\theta^{(o_i,o_j,o_k)}(j,i,k) + \phi)$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $\theta^{(o_j,o_i,o_k)}(j,i,k)$  はベクトル  $r^{(o_i,o_j)}(i,j)$  とベクトル  $r^{(o_i,o_k)}(i,k)$  との角度である。変数  $m$  は、角度の係数である。変数  $n$  は、 $\cos$  関数に使用する指数である。変数  $\phi$  は、関数の補正のために角度である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```

op_param = {
  'neighbor': [12],
  'radius': [1.5],
  'ave_times': 1,
  'oi_oj': [0],
  'o_factor': [0.00],
  'm_in_B': [2],
  'phi_in_B': [0],
  'n_in_B': [1, 2],
  'analysis_type': ['B']}

```

$C$  : centrosymmetry parameter analysis (CPA)

centrosymmetry parameter  $C$  は次の式で計算される。 [7]

$$C_i^{(a=1,o_i,o_j,o_k)} = \sum_{j \in M_b(i)} |r^{(o_i,o_j)}(i,j) + r^{(o_i,o_k)}(i,k)|^2$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $M_b(i)$  は粒子  $i$  の  $N/2$  個の隣接粒子のリストである。

粒子  $k$  の計算の方法は 2 通り実装している。座標  $r^{(o_j)}(j)$  の、ベクトル  $r^{(o_i)}(i)$  に関して反対側に移動した  $r^{(o_j)}(j)$  から一番近い位置にある  $N_b(i)$  リストにある粒子である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```

op_param = {
  'neighbor': [12],
  'radius': [2.0],
  'ave_times': 1,

```

```
'oi_oj': [0],
'o_factor': [0.00],
'analysis_type': ['C'] }
```

$D$  : neighbor distance analysis (NDA)

neighbor distance analysis  $D$  は次の式で計算される。[8]

$$D_i^{(a=1, o_i, o_j, o_k, f_{ij}, f_{ik}, f_{jk})} = \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} f_{ij}(r^{(o_i, o_j)}(i, j)) f_{ik}(r^{(o_i, o_k)}(i, k)) f_{jk}(r^{(o_j, o_k)}(j, k))$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。関数  $f_{ij}(r(i, j)), f_{ik}(r(i, k)), f_{jk}(r(j, k))$  は、距離を引数とし、返り値として持つ何かの関数である。変数  $r(i, j), r(i, k), r(j, k)$  は、ベクトル  $r^{(o_i, o_j)}(i, j), r^{(o_i, o_k)}(i, k), r^{(o_j, o_k)}(j, k)$  の距離である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。尚、下記の例は、関数  $\{f_{ij}(r), f_{ik}(r), f_{jk}(r)\} = r$  の設定での例である。

```
def f_1(r):
    return r

op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'oi_oj': [0],
    'o_factor': [0.00],
    'function': [f_1],
    'analysis_type': ['D'] }
```

$F$  : angular Fourier series (AFS) like parameter

angular Fourier series parameter  $F$  は次の式で計算される。[9][10]

$$F_i^{(a=1, o_i, o_j, o_k, f_{ij}, f_{ik}, m)} = \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} f_{ij}(r^{(o_i, o_j)}(i, j)) f_{ik}(r^{(o_i, o_k)}(i, k)) \cos(l\theta^{(o_j, o_i, o_k)}(j, i, k))$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。関数  $f_{ij}(r), f_{ik}(r), f_{jk}(r)$  は、ベクトルを引数として値を返り値として持つ何かの関数である。変数  $l$  は、角度の係数である。変数  $\theta^{(o_j, o_i, o_k)}(j, i, k)$  はベクトル  $r^{(o_i, o_j)}(i, j)$  とベクトル  $r^{(o_i, o_k)}(i, k)$  との角度である。関数  $f_{ij}(r(i, j)), f_{ik}(r(i, k))$  は、距離を引数とし、返り値として持つ何かの関数である。変数  $r(i, j), r(i, k)$  は、ベクトル  $r^{(o_i, o_j)}(i, j), r^{(o_i, o_k)}(i, k)$  の距離である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。尚、下記の例は、関数  $\{f_{ij}(r), f_{ik}(r)\} = r$  の設定での例である。

```
def f_1(r):
    return r
```



```

op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'oi_oj': [0],
    'o_factor': [0.00],
    'l_in_F': [1],
    'function': [f_1],
    'analysis_type': ['F']}

```

$H$  : angle histogram analysis (AHA)

angle histogram analysis  $H$  は次の式で計算される。

$$\begin{aligned}
h^{(b=1, bin, o_i, o_j, o_k)}(i) &= \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} histogram^{(bin)}(\theta^{(o_i, o_j, o_k)}(j, i, k)) \\
h^{(b, bin, o_i, o_j, o_k)}(i) &= \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} h^{(b-1, bin, o_i, o_j, o_k)}(j) \\
H_i^{(a=1, b, bin, o_i, o_j, o_k, \nu)} &= FFT_{amplitude}(h^{(b, bin, o_i, o_j, o_k)}(i))\delta(X - \nu)
\end{aligned}$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $\tilde{N}_b(i)$  は変数  $N_b(i)$  に粒子  $i$  自身を追加した  $N+1$  個の粒子の list である。変数  $b$  は、各粒子のヒストグラムを隣接粒子のヒストグラムで平均化を行う回数である。変数  $\theta^{(o_j, o_i, o_k)}(j, i, k)$  はベクトル  $r^{(o_i, o_j)}(i, j)$  とベクトル  $r^{(o_i, o_k)}(i, k)$  との角度である。関数  $histogram$  はビンの数  $bin$  のヒストグラムを作成する関数である。関数  $FFT_{amplitude}$  はヒストグラムをフーリエ変換し  $amplitude$  を計算する関数である。関数  $\delta$  はディラックのデルタ関数である。変数  $\nu$  はフーリエ変換された後のヒストグラムの周波数成分である。周波数成分  $X$  が  $\nu$  と同じであった場合のみ、デルタ関数  $\delta$  は 1 になるためである。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```

op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'oi_oj': [0],
    'o_factor': [0.00],
    'b_in_H': 1,
    'bin_in_H': [24],
    'nu_in_H': [3],
    'analysis_type': ['H'] }

```

$I$  : tetrahedron order parameter (TOP)

Tetrahedron order parameter  $I$  は以下の式で計算される。[11][12]

$$I_i^{(a=1, o_i, o_j, o_k)} = 1 - \frac{3}{8} \sum_{j>k \in N_b(i)} \{ \cos(\theta^{(o_i, o_j, o_k)}(j, i, k)) + 1/3 \}^2$$

変数  $a$  は、隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $\theta^{(o_j, o_i, o_k)}(j, i, k)$  はベクトル  $r^{(o_i, o_j)}(i, j)$  とベクトル  $r^{(o_i, o_k)}(i, k)$  との角度である。

隣接粒子数 4、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```
op_settings = {
  'neighbor': [4],
  'radius': [2.0],
  'ave_times': 1,
  'oi_oj': [0],
  'o_factor': [0.00],
  'analysis_type': ['I'] }
```

$Q$  : Bond order parameter

Bond order parameter  $Q$  は以下の式で計算される。[13][14]

$$Q_i^{(l, a=1, b, o_i, o_j, p)} = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^l |q_{lm}^{(l, a=1, b, o_i, o_j, p)}(i)|^2}$$

$$q_{lm}^{(l, a=1, b, o_i, o_j, p)}(i) = \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} q_{lm}^{(l, a=1, b-1, o_i, o_j, p)}(j)$$

$$q_{lm}^{(l, a=1, b=1, o_i, o_j, p)}(i) = \frac{1}{N+c} \left[ \sum_{j \in N_b(i)} Y_{lm}(\theta^{(o_i, o_j)}(i, j), \phi^{(o_i, o_j)}(i, j)) + p \sum_{v \in O} Y_{lm}(\theta^{(o_i, v)}(i, i), \phi^{(o_i, v)}(i, i)) \right]$$

変数  $a$  は、オーダーパラメータの値を隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $\tilde{N}_b(i)$  は  $N_b(i)$  に粒子  $i$  自身を追加した  $N+1$  個の粒子のリストである。変数  $b$  は、球面調和関数を隣接粒子で平均化を行う回数である。関数  $Y$  は、球面調和関数である。変数  $\theta^{(o_i, o_j)}(i, j), \phi^{(o_i, o_j)}(i, j)$  はベクトル  $r^{(o_i, o_j)}(i, j)$  の球面座標系での表現における角度で、 $\theta$  は  $z$  軸からの角度、 $\phi$  は  $x$  軸からの角度である。変数  $o$  は、粒子の位置を表現する数値の集合  $O$  かつ、 $o_i \neq O$  であるような  $b$  個の数値の集合である。例えば、 $O = \{1, 0, -1\}$  かつ  $o_i = 0$  であった、 $o = \{1, -1\}$  となる。変数  $p$  は、粒子  $i$  自身の他の位置ベクトル重さである。変数  $c$  は、規格化のための係数で、 $c = bp$  となるような数値である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```
op_settings = {
  'neighbor': [12],
  'radius': [2.0],
  'ave_times': 1,
  'oi_oj': [0],
  'o_factor': [0.00],
  'l_in_Q': [4],
  'b_in_Q': 1,
  'p_in_Q': [0],
  'analysis_type': ['Q'] }
```

尚、変数 `p_in_Q` では、変数として隣接粒子数  $N$  を使用することが可能であり、次のように指定する事も可能である。

```
op_settings = {
    'p_in_Q' : [0, 'N', 'N/2', '2*N'],
    ... ( 各オーダーパラメータのパラメータ ) }
```

Steinhardt のオーダーパラメータ [13] は以下のオプションで計算できる。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 0,
    'oi_oj': [0],
    'o_factor': [0.00],
    'l_in_Q' : [4,6],
    'b_in_Q' : 0,
    'p_in_Q' : [0],
    'analysis_type' : ['Q'] }
```

Lechner のオーダーパラメータ [14] は以下のオプションで計算できる。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 0,
    'oi_oj': [0],
    'o_factor': [0.00],
    'l_in_Q' : [4,6],
    'b_in_Q' : 1,
    'p_in_Q' : [0],
    'analysis_type' : ['Q'] }
```

$W$  : Bond order parameter

Bond order parameter  $W$  は以下の式で計算される。 [13][14]

$$W_i^{(l,a=1,b,o_i,o_j,p)} = \frac{\sum_{m_1+m_2+m_3=0} \binom{l}{m_1} \binom{l}{m_2} \binom{l}{m_3} q_{lm_1}^{(l,a=1,b,o_i,o_j,p)}(i) q_{lm_2}^{(l,a=1,b,o_i,o_j,p)}(i) q_{lm_3}^{(l,a=1,b,o_i,o_j,p)}(i)}{\left( \sum_{m=-l}^l |q_{lm}^{(l,a=1,b,o_i,o_j,p)}(i)|^2 \right)^{3/2}}$$

$$q_{lm}^{(l,a=1,b,o_i,o_j,p)}(i) = \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} q_{lm}^{(l,a=1,b-1,o_i,o_j,p)}(j)$$

$$q_{lm}^{(l,a=1,b=1,o_i,o_j,p)}(i) = \frac{1}{N+c} \left[ \sum_{j \in N_b(i)} Y_{lm}(\theta^{(o_i,o_j)}(i,j), \phi^{(o_i,o_j)}(i,j)) + p \sum_{v \in o} Y_{lm}(\theta^{(o_i,v)}(i,i), \phi^{(o_i,v)}(i,i)) \right]$$

変数  $a$  は、オーダーパラメータの値を隣接粒子で平均化を行う回数。変数  $N_b(i)$  は粒子  $i$  の  $N$  個の隣接粒子のリストである。変数  $\tilde{N}_b(i)$  は  $N_b(i)$  に粒子  $i$  自身を追加した  $N+1$  個の粒子のリストである。変数  $b$  は、球

面調和関数を隣接粒子で平均化を行う回数。関数  $Y$  は、球面調和関数である。変数  $\theta^{(o_i, o_j)}(i, j)$ ,  $\phi^{(o_i, o_j)}(i, j)$  はベクトル  $r^{(o_i, o_j)}(i, j)$  の球面座標系での表現における角度で、 $\theta$  は  $z$  軸からの角度、 $\phi$  は  $x$  軸からの角度である。変数  $o$  は、粒子の位置を表現する数値の集合  $O$  かつ、 $o_i \neq O$  であるような  $b$  個の数値の集合である。例えば、 $O = \{1, 0, -1\}$  かつ  $o_i = 0$  であった、 $o = \{1, -1\}$  となる。変数  $p$  は、粒子  $i$  自身の他の位置ベクトル重さである。変数  $c$  は、規格化のための係数で、 $c = bp$  となるような数値である。変数  $m_1, m_2, m_3$  は  $-l$  から  $l$  までの値をとるが、 $m_1 + m_2 + m_3 = 0$  の時だけ計算される。行列  $\begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix}$  は Wigner 3- $j$  symbol である。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'oi_oj': [0],
    'o_factor': [0.00],
    'l_in_Q': [4],
    'b_in_Q': 1,
    'p_in_Q': [0],
    'analysis_type': ['W'] }
```

尚、変数 `p_in_Q` では、変数として隣接粒子数  $N$  を使用することが可能であり、次のように指定する事も可能である。

```
op_settings = {
    'p_in_Q' : [0, 'N', 'N/2', '2*N'],
    ... ( 各オーダーパラメータのパラメータ ) }
```

Steinhardt のオーダーパラメータ [13] は以下のオプションで計算できる。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 0,
    'oi_oj': [0],
    'o_factor': [0.00],
    'l_in_Q': [4, 6],
    'b_in_Q': 0,
    'p_in_Q': [0],
    'analysis_type': ['W'] }
```

Lechner のオーダーパラメータ [14] は以下のオプションで計算できる。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 0,
    'oi_oj': [0],
    'o_factor': [0.00],
    'l_in_Q': [4, 6],
```

```
'b_in_Q': 1,
'p_in_Q': [0],
'analysis_type': ['W'] }
```

$Z$  : user define order parameter

ユーザーが定義するためのオーダーパラメータ  $Z$  を用意している。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメータの計算条件として以下の例を示す。

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'analysis_type': ['Z'] }
```

オーダーパラメータの計算部分としては、 `op_tools/op_z_user_define.py` である。このファイルを編集し、ユーザーの考えたオーダーパラメータを実装することが可能である。

## 大量の解析を行う設定

現実的に実行するためには、非常に長い計算が必要だろう。

```
def f_1(r):
    return r
op_settings = {
    # 隣接粒子半径の設定
    'neighbor'      : [8],          # 隣接粒子数
    'radius'        : [1.5],        # 隣接半径
    'ave_times'     : 1,            # あるオーダーパラメータを周囲の粒子で平均を計算
    # する回数
    'oi_oj'         : [1,0,-1],     # 粒子の座標をどの点にするかという設定 粒
    # 子 => [0], 直線 => [1,-1] (先端、後端), 楕円体 => [0, 1, -1] (中心、先端、後端)
    'o_factor'      : [0.5, 1.0, 1.5], # 方向ベクトルの長さの設定
    # A
    'op_types'      : ['A','P','N'],  # オーダーパラメータ A での解析の種類の指定
    'm_in_A'        : [2, 4],        # 粒子 i, 粒子 j からの距離の近い粒子 j の粒子数
    # B
    'm_in_B'        : [2],           # 角度の係数
    'n_in_B'        : [1, 2],        # cosine 関数の指数
    'phi_in_B'      : [0],           # 角度の offset
    # C
    # D
    'function'      : [f_1]          # 関数の種類
    # F
    'l_in_F'        : [1],           # 角度の係数
    # H
    'b_in_H'        : 1,             # 角度のヒストグラムを平均化する回数
    'bin_in_H'      : [24],          # ヒストグラムのビンの数
```

```

    'nu_in_H' : [3],                                # 抜き出す角度の周波数成分。この例では  $\pi/3$  の周
波数の成分の指定になっている。
    # I
    # Q
    'b_in_Q'      : 1,                                # 球面調和関数を平均化する回数
    'l_in_Q'      : [2, 4, 6],                        # 球面調和関数の次数を指定するパラメータ l
    'p_in_Q'      : [0],                                # その粒子の方向ベクトルに関する重み
    # S
    'n_in_S'      : [2],                                # Degree of Legendre_polynomials
    # T
    'n_in_T'      : [2],                                # Degree of Legendre_polynomials
    'd_in_T': [1.0],
    'analysis_type': ['A', 'B', 'C', 'D', 'F', 'H', 'I', 'Q', 'W', 'S', 'T']} # 解析する
オーダーパラメータの種類

```

## 出力のフォーマット

次のように使用する。

```

import order_tools
order_param_data = \
    op_analyze(coord, direct, box_length, op_settings)

```

出力として各粒子のオーダーパラメータが計算される。各粒子のオーダーパラメータにアクセスするためには、

```
order_param_data['Q_N6_a=1_b=1_of=1.00_oi=0_oj=0']
```

- Q : オーダーパラメータの種類
- N6 : 隣接粒子の条件
- 'a=1\_b=1\_of=1.00\_oi=0\_oj=0' : オーダーパラメータの計算に使用したパラメータ一覧 でアクセスできる。

## Reference

- [1] L. Onsager, THE EFFECTS OF SHAPE ON THE INTERACTION OF COLLOIDAL PARTICLES, Annals of the New York Academy of Sciences. 51 (1949) 627-659. doi:[10.1111/j.1749-6632.1949.tb27296.x](https://doi.org/10.1111/j.1749-6632.1949.tb27296.x).
- [2] C. Zannoni, Distribution function and order parameters, in: G.R. Luchhurst, G.W. Gray (Eds.), The Molecular Physics of Liquid Crystals, Academic Press Inc. (London) Ltd., 1979: p. 51.
- [3] W. McMillan, Simple Molecular Model for the Smectic A Phase of Liquid Crystals, Physical Review A. 4 (1971) 1238-1246. doi:[10.1103/PhysRevA.4.1238](https://doi.org/10.1103/PhysRevA.4.1238).
- [4] J.D. Honeycutt, H.C. Andersen, Molecular dynamics study of melting and freezing of small Lennard-Jones clusters, The Journal of Physical Chemistry. 91 (1987) 4950-4963. doi:[10.1021/j100303a014](https://doi.org/10.1021/j100303a014).
- [5] A. Radhi, K. Behdian, Identification of crystal structures in atomistic simulation by predominant common neighborhood analysis, Computational Materials Science. 126 (2017) 182-190. doi:[10.1016/j.commatsci.2016.09.035](https://doi.org/10.1016/j.commatsci.2016.09.035).

- [6] G.J. Ackland, A.P. Jones, Applications of local crystal structure measures in experiment and simulation, *Physical Review B – Condensed Matter and Materials Physics*. 73 (2006) 1-7. doi:[10.1103/PhysRevB.73.054104](https://doi.org/10.1103/PhysRevB.73.054104).
- [7] C.L. Kelchner, S.J. Plimpton, J.C. Hamilton, Dislocation nucleation and defect structure during surface indentation, *Physical Review B*. 58 (1998) 11085-11088. doi:[10.1103/PhysRevB.58.11085](https://doi.org/10.1103/PhysRevB.58.11085).
- [8] A. Stukowski, Structure identification methods for atomistic simulations of crystalline materials, *Modelling and Simulation in Materials Science and Engineering*. 20 (2012) 045021. doi:[10.1088/0965-0393/20/4/045021](https://doi.org/10.1088/0965-0393/20/4/045021).
- [9] A.P. Bartók, R. Kondor, G. Csányi, On representing chemical environments, *Physical Review B – Condensed Matter and Materials Physics*. 87 (2013) 1-16. doi:[10.1103/PhysRevB.87.184115](https://doi.org/10.1103/PhysRevB.87.184115).
- [10] A. Seko, A. Togo, I. Tanaka, Descriptors for Machine Learning of Materials Data, in: *Nanoinformatics*, Springer Singapore, Singapore, 2018: pp. 3-23. doi:[10.1007/978-981-10-7617-6\\_1](https://doi.org/10.1007/978-981-10-7617-6_1).
- [11] P.-L. CHAU, A.J. HARDWICK, A new order parameter for tetrahedral configurations, *Molecular Physics*. 93 (1998) 511-518. doi:[10.1080/002689798169195](https://doi.org/10.1080/002689798169195).
- [12] E. Duboué-Dijon, D. Laage, Characterization of the Local Structure in Liquid Water by Various Order Parameters, *The Journal of Physical Chemistry B*. 119 (2015) 8406-8418. doi:[10.1021/acs.jpcb.5b02936](https://doi.org/10.1021/acs.jpcb.5b02936).
- [13] P.J. Steinhardt, D.R. Nelson, M. Ronchetti, Bond-orientational order in liquids and glasses, *Physical Review B*. 28 (1983) 784-805. doi:[10.1103/PhysRevB.28.784](https://doi.org/10.1103/PhysRevB.28.784).
- [14] W. Lechner, C. Dellago, Accurate determination of crystal structures based on averaged local bond order parameters, *Journal of Chemical Physics*. 129 (2008) 114707. doi:[10.1063/1.2977970](https://doi.org/10.1063/1.2977970).