

目 次

op_tools ver0.2.0 ユーザーマニュアル	1
これは何？	2
インストール	2
必要なデータ	2
すべてのオーダーパラメーターで共通の設定	2
隣接粒子の設定	2
隣接粒子で平均化を行う回数の設定	3
各オーダーパラメーターの説明	4
<i>A</i> : neighborhood parameters	4
<i>B</i> : bond angle analysis (BAA)	4
<i>C</i> : centrosymmetry parameter analysis (CPA)	5
<i>D</i> : neighbor distance analysis (NDA)	5
<i>F</i> : angular Fourier series (AFS) like parameter	6
<i>H</i> : angle histogram analysis (AHA)	6
<i>I</i> : tetrahedron order parameter (TOP)	7
<i>Q</i> : Bond order parameter	7
<i>W</i> : Bond order parameter	8
<i>Q2, W2</i> : function weighting Bond order parameter	9
<i>LQ, LW</i> : local Bond order parameter	10
<i>S</i> : Onsager' s parameter	11
<i>T</i> : McMillan' s Sigma	12
<i>Z</i> : user define order parameter	12
大量の解析を行う設定	13
出力のフォーマット	14
Reference	14

op_tools ver0.2.0 ユーザーマニュアル

date: “2022 04/01” author: “土居 英男” (doi.hideo.chemistry@gmail.com)

これは何？

分子シミュレーションや粒子のシミュレーションにおける結晶構造を解析するための python モジュールです。

個々の粒子や原子の周囲の環境を評価し、数値に変換します。この数値はオーダーパラメーター（秩序変数, `order parameters`）とも呼ばれます。似通った並び方の粒子は同じ数値になり、違った並び方をしている粒子は違う数値になります。体心立方格子 (body-centered cubic, BCC) や面心立方格子 (face-centered cubic, FCC) ではオーダーパラメーターが特定の値になることが知られており、固体と液体の境界の位置などを調べる事ができます。このような解析は、固体金属を融解させるようなシミュレーションや、結晶の粒界を調べる解析などで使われます。

インストール

所定のサイトからパッケージをダウンロードし、次のコマンドを実行してください。

```
git clone https://github.com/hdoi/op_tools.git
cd op_tools
pip3 install -e .
```

必要なデータ

1. `coord` : 各粒子の座標のリスト。 [[0,0,0], [0,0,1], ...]
[`x1`, `y1`, `z1`], [`x2`, `y2`, `z2`], ...] というような順番になっている。
2. `direct` : 各粒子の方向ベクトル。3次元の単位ベクトルでも、四元数でも良い。次のようになっているれば良い。 [[`x1`, `y1`, `z1`], [`x2`, `y2`, `z2`], ...] か [[`w1`, `x1`, `y1`, `z1`], [`w2`, `x2`, `y2`, `z2`], ...] 質点や球状粒子の場合、方向ベクトルが存在しない。その場合は、`direct` = [] で良い。
3. `box_length` : シミュレーションボックスの長さ。 [3, 3, 3] xyz の順番になっている。
4. `op_settings` : 各 `order parameter` を計算するための設定

すべてのオーダーパラメーターで共通の設定

隣接粒子の設定

各粒子のオーダーパラメーターの数値は近くに存在する粒子の座標を使用して計算されます。このある粒子 i の近くにある粒子のことを、ある粒子の隣接粒子と呼びます。隣接粒子の決める方法として、このスクリプトでは3つの方法が実装されています。

1つめの方法は、粒子 i からの距離がある距離（隣接半径）以内にある粒子を、粒子 i の隣接粒子とする方法です。この方法の利点は、実装が比較的簡単でユーザーにとってもわかりやすいことです。欠点としては、隣接粒子の個数が各粒子で違う場合がある事と、スケール依存性があることです。例えば、隣接半径を 1 nm と 1 angstrom を指定するのでは結果が変わります。これをスケール依存性があると言います。

2つ目の方法は、粒子 i から距離の近い n 個の粒子を隣接粒子とする方法です。利点としては、スケール依存性がなく、長さの単位が変わっても距離の大小関係が変わらないため、隣接粒子が変化しません。欠点としては実装が簡単ではない、計算コストが数割程度大きい、論文での使用実績が存在するが少ないという点です。

3つ目の方法は Delaunay 分割 (Voronoi 分割, `Voronoi diagram`) を使う方法です。Voronoi 分割を行った後、各 Voronoi 面を共有する2つの粒子を隣接粒子とします。

各方法の使用法は、以下の通りです。

隣接半径を指定する場合

オーダーパラメーターの計算条件を指定する `op_settings` で次のように指定を行います。

```
op_settings = {  
    'radius': [1.5, 1.75] ,  
    ... ( 各オーダーパラメーターの設定 ) }
```

list 形式で指定することが可能で、複数の条件を一度に指定する事も可能です。上の例では、隣接半径=1.5, 1.75 での指定を行っています。

隣接粒子の数を指定する場合

オーダーパラメーターの計算条件を指定する `op_settings` で次のように指定を行います。

```
op_settings = {  
    'neighbor': [8, 12] ,  
    ... ( 各オーダーパラメーターの設定 ) }
```

list 形式で指定することが可能で、複数の条件を一度に指定する事も可能です。上の例では、隣接粒子数 8 と 12 の指定を行っている。

Delaunay 分割を行う場合

オーダーパラメーターの計算条件を指定する `op_settings` で次のように指定を行います。

```
op_settings = {  
    'Delaunay': ['standard']  
    ... ( 各オーダーパラメーターの設定 ) }
```

隣接粒子で平均化を行う回数の設定

粒子 i のオーダーパラメーターとその隣接粒子のオーダーパラメーターで平均を計算する場合があります。粒子 i の N 個の隣接粒子の list を $N_b(i)$ 、 $N_b(i)$ に粒子 i 自身を追加した $N + 1$ 個の粒子の list を $\tilde{N}_b(i)$ とすると、オーダーパラメーターを $X(i)$ 、平均化した後のオーダーパラメーターを $Y(i)$ とすると、次の様な計算を行います。

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

この平均化の操作には以下の利点と欠点があります。利点としては、平均を計算するだけであるため計算負荷が軽い事、同じオーダーパラメーターの分類性能を上げる事が可能な事、何らかの数値であれば使用できる事、何回も平均化をする事が可能である事などです。一方、欠点としては、オーダーパラメーターの計算に使用する情報が多くなり、オーダーパラメーターの解像度が下がる事です。オーダーパラメーターは隣接粒子の情報を使って計算され、さらに平均化の計算では隣接粒子のオーダーパラメーターを使って計算されます。平均化操作を 1 回行うごとに、隣接粒子の隣接粒子の情報を使用することになる。つまり、オーダーパラメーターの計算に使用する粒子が多くなり、遠くの粒子の情報を使用することになる。これは、均一な系ではあまり問題にはならないが、不均一な系（界面など）では問題になります。不均一な系では、異常が起きている位置が知りたいとか、界面の位置を知りたいなどといった目的で使用される場合が多いですが、遠くの粒子の情報を使用すると、位置がはっきりとはわからなくなる。

この平均化を行います回数は、オーダーパラメーターの計算条件を指定する `op_settings` で次のように指定を行います。なお、この値は、0 から始まり、0 の場合は、平均化を行わない。

```

op_settings = {
    'ave_times' : 1,
    ... ( 各オーダーパラメーターの設定 )
}

```

各オーダーパラメーターの説明

A : neighborhood parameters

common neighborhood parameter A [1]、predominant common neighborhood parameter P [2]、another predominant common neighborhood parameter N [2] は非常に似ている式で計算される。

$$\begin{aligned}
 A^{(a=0, type=A, m)}(i) &= A^{(a=0, m)}(i) = \frac{1}{N} \sum_{j \in N_b(i)} \left| \sum_{k \in N_b(i, j)} (r_{ik} + r_{jk}) \right|^2 \\
 A^{(a=0, type=P, m)}(i) &= P^{(a=0, m)}(i) = \frac{1}{N} \sum_{j \in N_b(i)} \left| \sum_{k \in N_b(i, j)} (r_{ij} + r_{kj}) \right|^2 \\
 A^{(a=0, type=N, m)}(i) &= N^{(a=0, m)}(i) = \frac{1}{N} \left| \sum_{j \in N_b(i)} \sum_{k \in N_b(i, j)} (r_{ij} + r_{kj}) \right|^2
 \end{aligned}$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。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す。変数 $type$ で計算を行うオーダーパラメーターの種類を指定します。

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

```

op_param = {
    'neighbor': [12],
    'radius': [1.5],
    'ave_times': 1,
    '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ます。 [3]

$$\begin{aligned}
 B^{(a=0, m, n, \phi)}(i) &= \frac{1}{N(N-1)/2} \sum_{j > k \in N_b(i)} f^{(m, n, \phi)}(\theta_{jik}) \\
 f^{(m, n, \phi)}(\theta_{jik}) &= \cos^n(m\theta_{jik} + \phi)
 \end{aligned}$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。変数 θ_{jik} ベクトル r_{ij} とベクトル r_{ik} との角度です。変数 m は、角度の係数です。変数 n は、 \cos 関数に使用する指数です。変数 ϕ は、関数の補正のために角度です。

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

```
op_param = {
  'neighbor': [12],
  'radius': [1.5],
  'ave_times': 1,
  '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ます。 [4]

$$C^{(a=0, type='half', mode)}(i) = \sum_{j \in M_b(i)} |r_{ij} + r_{ik}|^2$$

$$C^{(a=0, type='all', mode)}(i) = \sum_{j \in N_b(i)} |r_{ij} + r_{ik}|^2$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。変数 $M_b(i)$ は粒子 i の最近接の $N/2$ 個の隣接粒子のリストです。変数 $type$ は、ベクトルを足す回数を指定する変数です。変数 $mode$ は粒子 k の決め方を示す変数です。変数 $mode$ が 'dist' であるとき、粒子 k は、座標 r_j を座標 r_i に関して反対側に移動した座標 r'_j から距離の一番近い位置にあるリスト $N_b(i)$ にある粒子です。変数 $mode$ が 'angle' であるとき、粒子 k は、角度 θ_{jik} が一番 π に近いリスト $N_b(i)$ にある粒子です。これらは論文中で曖昧な部分があり実装することができなかったため、近接粒子の決める方法と反対側の粒子位置を決める方法として2つの方法を実装しています。

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

```
op_param = {
  'neighbor': [12],
  'radius': [2.0],
  'ave_times': 1,
  'types_in_C' : ['half'],
  'modes_in_C' : ['dist'],
  'analysis_type': ['C']}
```

D : neighbor distance analysis (NDA)

neighbor distance analysis D のような秩序変数として以下の秩序変数を実装しました。 [5] ovito を使用して NDA 解析を行うことを推奨します。

$$D_i^{(a=0, f_{ij}, f_{ik}, f_{jk})} = \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} f_{ij}(r_{ij}) f_{ik}(r_{ik}) f_{jk}(r_{jk})$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。関数 $f_{ij}(r_{ij}), f_{ik}(r_{ik}), f_{jk}(r_{jk})$ は、距離を引数とし、返り値として持つ何かの関数です。変数 r_{ij}, r_{ik}, r_{jk} は、それぞれベクトル r_{ij}, r_{ik}, r_{jk} の距離です。また、距離は、 $r_{ij} \leq r_{ik}$ を満たします。

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

```
def f1(r):
    judge = 0
    if 0.8 < r and r < 1.21:
        judge = 1
    return judge
```

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'function': [f1],
    '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。[6][7]

$$F_i^{(a=0, f_1, f_2, m)} = \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} f_1(r_{ij}) f_2(r_{ik}) \cos(l\theta_{jik})$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。関数 $f^1(r), f^2(r)$ は、距離を引数として値を返り値として持つ何かの関数です。変数 l は、角度の係数です。変数 θ_{jik} はベクトル r_{ij} とベクトル r_{ik} との角度です。距離は、 $r_{ij} \leq r_{ik}$ を満たします。

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

```
def f1(r):
    return r
```

```
op_settings = {
    'neighbor': [12],
    'radius': [2.0],
    'ave_times': 1,
    'l_in_F': [1],
    'function': [f1],
    'analysis_type': ['F']}
```

H : angle histogram analysis (AHA)

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

$$h^{(b=0, bin)}(i) = \frac{1}{N(N-1)/2} \sum_{j>k \in N_b(i)} histogram^{(bin)}(\theta_{jik})$$

$$h^{(b,bin)}(i) = \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} h^{(b-1,bin)}(j)$$

$$H^{(a=0,b,bin,\nu)}(i) = FFT_{amplitude}(h^{(b,bin)}(i))\delta(X - \nu)$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。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 です。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す。変数 θ_{jik} はベクトル r_{ij} とベクトル r_{ik} との角度です。関数 *histogram* はビンの数 bin のヒストグラムを作成する関数です。関数 $FFT_{amplitude}$ はヒストグラムをフーリエ変換し *amplitude* を計算する関数です。関数 δ はディラックのデルタ関数です。変数 ν 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す。周波数成分 X が ν と同じであった場合のみ、デルタ関数 δ は 1 になるためです。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメーター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,
    '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。[8][9]

$$I^{(a=0)}(i) = 1 - \frac{3}{8} \sum_{j>k \in N_b(i)} \{\cos(\theta_{jik}) + 1/3\}^2$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。変数 θ_{jik} はベクトル r_{ij} とベクトル r_{ik} との角度です。

隣接粒子数 4、隣接半径 2.0 でのオーダーパラメーター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,
    'analysis_type': ['I'] }
```

Q : Bond order parameter

Bond order parameter Q は以下の式で計算される。[10][11]

$$Q_i^{(l,a=0,b)} = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^l |q_{lm}^{(l,a=0,b)}(i)|^2}$$

$$q_{lm}^{(l,a=0,b)}(i) = \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} q_{lm}^{(l,a=0,b-1)}(j)$$

$$q_{lm}^{(l,a=0,b=1)}(i) = \frac{1}{N} \sum_{j \in N_b(i)} Y_{lm}(\theta_{ij}, \phi_{ij})$$

変数 a は、オーダーパラメーターの値を隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。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す。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す。関数 Y は球面調和関数です。変数 θ_{ij}, ϕ_{ij} はベクトル r_{ij} の球面座標系での表現における角度で、 θ は z 軸からの角度、 ϕ は x 軸からの角度です。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメーター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,
  'l_in_Q': [4],
  'b_in_Q': 1,
  'analysis_type': ['Q'] }
```

Steinhardt のオーダーパラメーター [10] 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,
  'l_in_Q' : [4,6],
  'b_in_Q' : 0,
  'analysis_type' : ['Q'] }
```

Lechner のオーダーパラメーター [11] 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,
  'l_in_Q' : [4,6],
  'b_in_Q' : 1,
  'analysis_type' : ['Q'] }
```

W : Bond order parameter

Bond order parameter W は以下の式で計算される。 [10][11]

$$W_i^{(l,a=0,b)} = \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} q_{lm_1}^{(l,a=0,b)}(i) q_{lm_2}^{(l,a=0,b)}(i) q_{lm_3}^{(l,a=0,b)}(i)}{\left(\sum_{m=-l}^l |q_{lm}^{(l,a=0,b)}(i)|^2 \right)^{3/2}}$$

$$q_{lm}^{(l,a=0,b)}(i) = \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} q_{lm}^{(l,a=0,b-1)}(j)$$

$$q_{lm}^{(l,a=0,b=0)}(i) = \frac{1}{N} \sum_{j \in N_b(i)} Y_{lm}(\theta_{ij}, \phi_{ij})$$

変数 a は、オーダーパラメーターの値を隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。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す。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、球面調和関数です。変数 θ_{ij}, ϕ_{ij} はベクトル r_{ij} の球面座標系での表現における角度で、 θ は z 軸からの角度、 ϕ は x 軸からの角度です。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 です。

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメーター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,
    'l_in_Q': [4],
    'b_in_Q': 1,
    'analysis_type': ['W'] }
```

Steinhardt のオーダーパラメーター [10] 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,
    'l_in_Q': [4,6],
    'b_in_Q': 0,
    'analysis_type': ['W'] }
```

Lechner のオーダーパラメーター [11] 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,
    'l_in_Q': [4,6],
    'b_in_Q': 1,
    'analysis_type': ['W'] }
```

$Q2, W2$: function weighting Bond order parameter

$$Q2_i^{(l,a=0,b)} = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^l |q_{lm}^{(l,a=0,b)}(i)|^2}$$

$$q_{lm}^{(l,a=0,b)}(i) = \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} q_{lm}^{(l,a=0,b-1)}(j)$$

$$q_{lm}^{(l,a=0,b=0)}(i) = \sum_{j \in N_b(i)} F(i,j) Y_{lm}(\theta_{ij}, \phi_{ij})$$

粒子 i と粒子 j の間の重み付けを設定する関数 $F(i, j)$ を使用することができる。

変数 a は、オーダーパラメーターの値を隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。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す。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す。関数 Y は、球面調和関数です。変数 θ_{ij}, ϕ_{ij} はベクトル r_{ij} の球面座標系での表現における角度で、 θ は z 軸からの角度、 ϕ は x 軸からの角度です。

また、オーダーパラメーター $W2$ に関しては以下のように、計算する。

$$W2_i^{(l,a=0,b)} = \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} q_{lm_1}^{(l,a=0,b)}(i) q_{lm_2}^{(l,a=0,b)}(i) q_{lm_3}^{(l,a=0,b)}(i)}{\left(\sum_{m=-l}^l |q_{lm}^{(l,a=0,b)}(i)|^2\right)^{3/2}}$$

隣接条件 Delaunay 分割を利用した場合、Mickel のオーダーパラメーターを利用することができる。以下にオプションを示す。[12]

```
def f1(j, voronoi_area_list, distance_list):
    weight = voronoi_area_list[j] / np.sum(voronoi_area_list)
    return weight

op_settings = {
    'Delaunay': ['standard'],
    'ave_times': 1,
    'l_in_Q': [4],
    'function_in_Q2': [f1],
    'analysis_type': ['Q2', 'W2'] }
```

他にも、距離に応じた重み付けなどができる。以下の例は、粒子 i と粒子 j 間の距離が近い場合はより重く、遠い場合はより軽くするような設定です。

```
def f1(j, voronoi_area_list, distance_list):
    sum_weight = 0
    for dist in distance_list:
        sum_weight += 1/dist
    weight = (1/distance_list[j]) / sum_weight
    return weight

op_settings = {
    'Delaunay': ['standard'],
    'radius': [2.0],
    'ave_times': 1,
    'l_in_Q': [4],
    'function_in_Q2': [f1],
    'analysis_type': ['Q2'] }
```

LQ, LW : local Bond order parameter

このオーダーパラメーターは、[13][14][15] を模して実装したものです。

$$\begin{aligned}
LQ_i^{(l,a=0,b)} &= \frac{1}{N} \sum_{j \in N_b(i)} \frac{q_{lm}^{(l,a,b)}(i,j)}{|q_{lm}^{(l,a,b)}(i)| |q_{lm}^{(l,a,b)}(j)|} \\
q_{lm}^{(l,a,b)}(i,j) &= \sum_{m=-l}^l q_{lm}^{(l,a,b)}(i) q_{lm}^{*(l,a,b)}(j) \\
q_{lm}^{(l,a=0,b)}(i) &= \frac{1}{N+1} \sum_{j \in \tilde{N}_b(i)} q_{lm}^{(l,a=0,b-1)}(j) \\
q_{lm}^{(l,a=0,b=0)}(i) &= \frac{1}{N} \sum_{j \in N_b(i)} Y_{lm}(\theta_{ij}, \phi_{ij})
\end{aligned}$$

変数 a は、オーダーパラメーターの値を隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。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す。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す。関数 Y は、球面調和関数です。変数 θ_{ij}, ϕ_{ij} はベクトル r_{ij} の球面座標系での表現における角度で、 θ は z 軸からの角度、 ϕ は x 軸からの角度です。

$$\begin{aligned}
LW_i^{(l,a=0,b)} &= \frac{\sum_{m_1+m_2+m_3=0} \begin{pmatrix} l & l & l \\ m_1 & m_2 & m_3 \end{pmatrix} lq_{lm_1}^{(l,a=0,b)}(i) lq_{lm_2}^{(l,a=0,b)}(i) lq_{lm_3}^{(l,a=0,b)}(i)}{\left(\sum_{m=-l}^l |lq_{lm}^{(l,a=0,b)}(i)|^2 \right)^{3/2}} \\
lq_{lm}^{(l,a,b)}(i) &= \frac{1}{N} \sum_{j \in N_b(i)} \frac{q_{lm}^{(l,a,b)}(i) q_{lm}^{*(l,a,b)}(j)}{|q_{lm}^{(l,a,b)}(i)| |q_{lm}^{(l,a,b)}(j)|}
\end{aligned}$$

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメーターの計算条件として以下の例を示す。[13] と同じようなオーダーパラメーター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,
    'l_in_Q': [4,6],
    'b_in_Q': 0,
    'analysis_type': ['LQ', 'LW'] }

```

S : Onsager' s parameter

Onsager のオーダーパラメーター S は次の式で計算される。[16][17]

$$S^{(a=0,n)}(i) = \frac{\sum_{j \in N_b(i)} P_n(\cos(\theta))}{N}$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。関数 P_n は Legendre polynomial です。変数 n は Degree of the polynomial で偶数です。変数 θ 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 との角度です。

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

$$S^{(a=0,n=2)}(i) = \frac{\sum_{j \in N_b(i)} [3 \cos^2(\theta) - 1]/2}{N}$$

$$S^{(a=0,n=4)}(i) = \frac{\sum_{j \in N_b(i)} [35 \cos^4(\theta) - 30 \cos^2(\theta) + 3]/8}{N}$$

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメーター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ー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,
  'n_in_S' : [2],
  'analysis_type' : ['S']
}
```

T : McMillan' s Sigma

McMillan のパラメーター T は次の式で計算される。 [18]

$$T^{(a=0,n)}(i) = \frac{\sum_{j \in N_b(i)} \cos(2\pi z(i,j)/d) P_n(\cos(\theta))}{N}$$

変数 a は、隣接粒子で平均化を行う回数です。変数 $N_b(i)$ は粒子 i の N 個の隣接粒子のリストです。関数 P_n は Legendre polynomial です。変数 n は Degree of the polynomial です。変数 θ 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 との角度です。変数 $z(i,j)$ は、粒子 i を通過し、ベクトル u_i と垂直な平面 P から、粒子 j までの距離です。変数 d は、液晶の smectic 相の層から層までの距離です。

$n = 2$ の時、オーダーパラメーター T は以下の式で計算される。

$$T^{(a=0,n=2)}(i) = \frac{\sum_{j \in N_b(i)} \cos(2\pi z(i,j)/d) [3 \cos^2(\theta) - 1]/2}{N}$$

隣接粒子数 12、隣接半径 2.0 でのオーダーパラメーター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ー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,
  'n_in_T': [2],
  'd_in_T': [1.0],
  'analysis_type' : ['T']
}
```

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ー 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ー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ーの計算部分としては、op_tools/op_z_user_define.py です。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ーを実装することが可能です。

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 f1(r):
    return r
```

```
def f2(j, voronoi_area_list, distance_list):
    weight = voronoi_area_list[j] / np.sum(voronoi_area_list)
    return weight
```

```
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ーを周囲の粒子で平均を計算する回数
    # A
    'op_types'      : ['A', 'P', 'N'], # オーダーパラメーター 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
    'types_in_C'    : ['half'],      # ゆらぎを表現したベクトルを何回足すかというパラメーター
    'modes_in_C'    : ['dist'],      # ある粒子の反対にある粒子の探し方
    # D
    'function'      : [f1]           # 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た
    # 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ー l
    # Q2 W2
    'function_in_Q2' : [f2],         # 重み関数の指定
```

```

# LQ LW
# S
'n_in_S'      : [2],          # Degree of Legendre_polynomials
# T
'n_in_T'      : [2],          # Degree of Legendre_polynomials
'd_in_T'      : [1.0],        # smectic 相の層と層との間の距離
'analysis_type': ['A', 'B', 'C', 'D', 'F', 'H', 'I',
'Q', 'W', 'Q2', 'W2', 'LQ', 'LW', 'S', 'T']] # 解析するオーダーパラメーターの種類

```

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_tools.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ーが計算される。各粒子のオーダーパラメーターには次のようにアクセスできる。

```
order_param_data['Q_N6_a=1_b=1']
```

- Q : オーダーパラメーターの種類
- N6 : 隣接粒子の条件
- 'a=1_b=1' : オーダーパラメーターの計算に使用したパラメーター一覧

Reference

- [1] 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).
- [2] 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).
- [3] 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).
- [4] 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).
- [5] 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).
- [6] 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).
- [7] 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).
- [8] 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).

- [9] 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).
- [10] 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).
- [11] 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).
- [12] W. Mickel, S.C. Kapfer, G.E. Schröder-Turk, K. Mecke, Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter, *Journal of Chemical Physics*. 138 (2013) 044501. doi:[10.1063/1.4774084](https://doi.org/10.1063/1.4774084).
- [13] E.B. Moore, E. De La Llave, K. Welke, D.A. Scherlis, V. Molinero, Freezing, melting and structure of ice in a hydrophilic nanopore, *Physical Chemistry Chemical Physics*. 12 (2010) 4124-4134. doi:[10.1039/b919724a](https://doi.org/10.1039/b919724a).
- [14] M. Fitzner, P. Pedevilla, A. Michaelides, Predicting heterogeneous ice nucleation with a data-driven approach, *Nature Communications*. 11 (2020) 1-9. doi:[10.1038/s41467-020-18605-3](https://doi.org/10.1038/s41467-020-18605-3).
- [15] G.A. Tribello, F. Giberti, G.C. Sossa, M. Salvalaglio, M. Parrinello, Analyzing and Driving Cluster Formation in Atomistic Simulations, *Journal of Chemical Theory and Computation*. 13 (2017) 1317-1327. doi:[10.1021/acs.jctc.6b01073](https://doi.org/10.1021/acs.jctc.6b01073).
- [16] 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).
- [17] 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.
- [18] 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).