

# 「近接原子数」の数式解説 (compute\_coord\_atom.cpp)

Open DEM Japan

2025年6月30日

本稿では, LIGGGHTS に実装されている ComputeCoordAtom クラス(ソースファイル compute\_coord\_atom.cpp) が計算する近接原子数 (coordination number) を数式として定式化し, アルゴリズムを教科書的に説明する.

まず, 系に  $N$  個の粒子 (原子) が存在し, 粒子  $i$  の位置を  $\mathbf{x}_i \in \mathbb{R}^3$ , 型 (type) を  $T_i \in \{1, \dots, N_{\text{types}}\}$ , 計算対象粒子集合 (LAMMPS の group) を  $\mathcal{G}$  とする. ComputeCoordAtom は以下の入力パラメータを持つ.

- カットオフ距離  $r_c > 0$  (コード中 cutoff)
- 型区間  $\Lambda_k = [t_k^{\text{lo}}, t_k^{\text{hi}}]$  ( $k = 1, \dots, n_{\text{col}}$ )
- mix フラグ (真なら異種粒子のみを数える)

■距離判定 粒子  $i$  と  $j$  のユークリッド距離

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (1)$$

がカットオフ  $r_c$  未満であれば「近接」と判定する. 計算量削減のため, 実装ではその二乗  $r_{ij}^2 < r_c^2$  を cutsq と比較している.

■型フィルタ 型区間  $\Lambda_k$  に属するか否かを

$$\delta_k(j) = \begin{cases} 1, & T_j \in \Lambda_k, \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

で表す. mix が真のときは, 同種粒子を無視するため

$$\delta_k^{\text{mix}}(i, j) = \delta_k(j) (1 - \delta_{T_i T_j}), \quad (3)$$

ここで  $\delta_{pq}$  はクロネッカーのデルタである. mix が偽のときは  $\delta_k^{\text{mix}}(i, j) = \delta_k(j)$  と同一視する.

■配位数 (スカラー版  $n_{\text{col}} = 1$ ) 区間  $\Lambda_1$  に対する粒子  $i$  の配位数は

$$C_i = \sum_{j \neq i} H(r_c - r_{ij}) \delta_1^{\text{mix}}(i, j), \quad (4)$$

ただし  $H$  はヘヴィサイド関数  $H(x) = 1 (x \geq 0), 0 (x < 0)$  である. コードでは  $C_i$  を配列 cvec[i] に格納し, グループ外粒子 ( $i \notin \mathcal{G}$ ) には 0 を代入する.

■配位数 (多列版  $n_{\text{col}} > 1$ ) 複数区間を同時に評価する場合,

$$C_i^{(k)} = \sum_{j \neq i} H(r_c - r_{ij}) \delta_k^{\text{mix}}(i, j), \quad k = 1, \dots, n_{\text{col}}, \quad (5)$$

を計算し `carray[i][k]` に格納する. このとき演算回数は  $n_{\text{col}}$  倍になるが, 距離計算は一度だけ行いループ内で共有される.

■隣接リスト 実装は LAMMPS の 完全隣接リスト (full neighbor list) を利用し, 粒子  $i$  に対する隣接インデックス集合

$$\mathcal{N}_i = \{j \mid r_{ij} < r_c + \text{skin}\} \quad (6)$$

を事前に構築する. ここで `skin` はスキン距離であり, `init()` 内のチェック `sqrt(cutsq) > pair->cutforce + skin` により計算安定性が保証される.

■メモリ使用量 ローカルプロセスが保持する最大粒子数を  $n_{\text{max}}$  とすると, 必要メモリは

$$M = n_{\text{col}} n_{\text{max}} \text{sizeof}(\text{double}) \quad [\text{bytes}] \quad (7)$$

であり, メンバ関数 `memory_usage()` は式 (7) を返す.

■まとめ `ComputeCoordAtom` は式 (4), (5) に示した配位数  $C_i$  (または  $C_i^{(k)}$ ) を各タイムステップで評価し, 原子属性として出力する. `mix` フラグにより同種粒子を除外できる点が特徴であり, 多成分系の界面解析や結合確率計算に有用である.