

“ComputePropertyLocal” の数式解説 (compute_property_local.cpp)

Open DEM Japan

2025年6月30日

本稿では、離散要素法 (DEM) コード LIGGGHTS に実装されている `ComputePropertyLocal` クラスの計算法を数式により体系的に記述する。本クラスは局所的な相互作用 (粒子対, 結合, 角度, ジヘドラル, インプロパー) の **属性ベクトルまたは行列** を生成し, 可視化や後解析に供する。以下では相互作用の集合と属性写像を定義し, アルゴリズムを厳密な集合演算として示す。

■準備：粒子集合とグループ選択 計算時点のローカル粒子集合を

$$\mathcal{A} = \{i \mid 0 \leq i < N_{\text{local}} \wedge i \in \text{proc}\}, \quad (1)$$

とする。ここで N_{local} は当該プロセスが保持する粒子数である。粒子グループビットを用いて演算対象となる部分集合

$$\mathcal{G} = \{i \in \mathcal{A} \mid i \in \text{group}\} \quad (2)$$

を定義する。以下, すべての相互作用は \mathcal{G} に属する粒子のみを考慮する。

■ペア (近接粒子) 集合 半近接リストから得られる隣接対を

$$\mathcal{L}_{\text{pair}} = \{(i, j) \mid i \in \mathcal{G}, j \in \mathcal{G}, j \in \mathcal{N}(i), \|\mathbf{r}_i - \mathbf{r}_j\|^2 < r_{\text{cut}}^2(t_i, t_j), \sigma(i, j)\}, \quad (3)$$

と定義する。ここで $\mathcal{N}(i)$ は近接リスト, r_{cut} は型依存カットオフ距離, $\sigma(i, j)$ は `newton_pair` の有無に応じた重複除去条件である。集合の要素数

$$N_{\text{pair}} = |\mathcal{L}_{\text{pair}}| \quad (4)$$

がローカルペア数となる。

■結合 (Bond) 集合 粒子 i が保持する結合インデックス j を巡り,

$$\mathcal{L}_{\text{bond}} = \{(i, j) \mid i \in \mathcal{G}, j < n_{\text{bond}}(i), k = \text{map}(\text{bond_atom}_{i,j}), k \in \mathcal{G}, \tau(i, k)\}, \quad (5)$$

と置く。ここで $\tau(i, k)$ は `newton_bond` に基づく重複除去条件である。同様に角度 $\mathcal{L}_{\text{angle}}$, ジヘドラル $\mathcal{L}_{\text{dihedral}}$, インプロパー $\mathcal{L}_{\text{improper}}$ を, それぞれ第2粒子 (中心粒子) がローカルに存在し全粒子が \mathcal{G} に属する条件で定義する。

■統一のインデックス集合と個数 フラグ

$$\text{kindflag} \in \{\text{NEIGH}, \text{PAIR}, \text{BOND}, \text{ANGLE}, \text{DIHEDRAL}, \text{IMPROPER}\}$$

に応じて

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{pair}} & (\text{kindflag} = \text{NEIGH or PAIR}), \\ \mathcal{L}_{\text{bond}} & (\text{kindflag} = \text{BOND}), \\ \mathcal{L}_{\text{angle}} & (\text{kindflag} = \text{ANGLE}), \\ \mathcal{L}_{\text{dihedral}} & (\text{kindflag} = \text{DIHEDRAL}), \\ \mathcal{L}_{\text{improper}} & (\text{kindflag} = \text{IMPROPER}), \end{cases} \quad (6)$$

とし、局所インタラクション数

$$N = |\mathcal{L}| \quad (7)$$

を得る.

■属性写像 ユーザは n_{val} ($1 \leq n_{\text{val}} \leq 2$) 個のキーワード

$$\kappa = (\kappa_1, \dots, \kappa_{n_{\text{val}}})$$

を与える. 各キーワード κ_s に対し属性写像

$$P_{\kappa_s} : \mathcal{L} \rightarrow \mathbb{R} \quad (8)$$

を定義する. 例えば

$$P_{\text{patom1}}(i, j) = \text{tag}(i), \quad (9)$$

$$P_{\text{ptype2}}(i, j) = \text{type}(j), \quad (10)$$

$$P_{\text{btype}}(i, j) = \text{bond_type}_{i,j}, \quad (11)$$

などが該当する.

■出力ベクトル／行列の構築 (1) 属性が1種類するとき (vector 出力)

$$\mathbf{v}[m] = P_{\kappa_1}(\ell_m), \quad m = 1, \dots, N, \quad (12)$$

ただし ℓ_m は \mathcal{L} の m 番目の要素である.

(2) 属性が複数するとき (array 出力)

$$\mathbf{A}[m, s] = P_{\kappa_s}(\ell_m), \quad m = 1, \dots, N, \quad s = 1, \dots, n_{\text{val}}. \quad (13)$$

■メモリ使用量 前節の構造体はメモリを

$$\text{bytes} = N n_{\text{val}} \times \text{sizeof}(\text{double}) + N \times 2 \times \text{sizeof}(\text{int}) \quad (14)$$

だけ消費する. ここで第1項は属性値, 第2項はインデックス配列である.

■再割当て 必要メモリ N が現在確保量 N_{max} を超える場合,

$$N_{\text{max}} \leftarrow N_{\text{max}} + \Delta N, \quad \Delta N = 10\,000, \quad (15)$$

となる最小の N_{max} を再確保し, **vector** または **array** を拡張する.

■アルゴリズムの要旨

1. 近接リストあるいはトポロジ情報を参照して \mathcal{L} を構築し, N を決定する.
2. 要求属性に応じて P_{κ_s} を適用し, \mathbf{v} または \mathbf{A} を充填する.
3. 必要に応じて動的メモリを再割当てし, 省メモリかつ再利用可能なデータ構造を維持する.

以上により `ComputePropertyLocal` は局所的な相互作用に関する属性ベクトル／行列を厳密に定義し, 離散要素法計算における後処理の基盤を提供する.