

フル分子系アトムベクトルの数式解説 (atom_vec_full.cpp)

Open DEM Japan

2025年6月30日

本ファイルは LIGGGHTS/LAMMPS において分子系を扱う “AtomVecFull” クラスの実装であり、各粒子 i ($i = 0, 1, \dots, n_{\text{local}} - 1$) について

$\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$, $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3})$, $\mathbf{f}_i = (f_{i1}, f_{i2}, f_{i3})$, q_i, m_i , bond/angle 連結情報などを格納・通信・保存する。以下では各メンバ関数を対応する数学操作として記述する。

■メモリ拡張 grow(n) 必要最大数 n_{max} は

$$n_{\text{max}} \leftarrow \begin{cases} n_{\text{max}} + \text{DELTA} & (n = 0), \\ n & (n > 0), \end{cases} \quad (1)$$

と更新され、各配列 $\{\mathbf{x}, \mathbf{v}, \mathbf{f}, q, \dots\}$ を長さ n_{max} に再確保する。

■コピー copy($i \rightarrow j$)

$$\begin{aligned} \mathbf{x}_j &\leftarrow \mathbf{x}_i, & \mathbf{v}_j &\leftarrow \mathbf{v}_i, & q_j &\leftarrow q_i, \\ \text{bond}_j &\leftarrow \text{bond}_i, & \text{angle}_j &\leftarrow \text{angle}_i, & \text{他属性}_j &\leftarrow \text{他属性}_i. \end{aligned} \quad (2)$$

■座標パック pack_comm 通信リスト $\{j_l\}_{l=0}^{n-1}$ に対し、

$$\text{buf}_{m+3l+k} = x_{j_l k} + d_k, \quad k = 1, 2, 3, \quad (3)$$

ただし $\mathbf{d} = (d_1, d_2, d_3)$ は周期境界条件 (PBC) オフセット

$$\mathbf{d} = \begin{cases} \mathbf{0} & (\text{PBC 無}), \\ (p_x L_x, p_y L_y, p_z L_z) & (\text{直方体系}), \\ (\dots) & (\text{斜方格子の場合}), \end{cases}$$

である。

■速度付きパック pack_comm_vel 座標に加え速度を並べる。格子変形速度 $\dot{\mathbf{h}}$ が有効かつ粒子が変形グループに属するとき

$$\mathbf{v}_j \rightarrow \mathbf{v}_j + \dot{\mathbf{h}} \cdot \mathbf{p}, \quad (4)$$

を加算してバッファへ書き込む。

■アンパック unpack_comm 受信したバッファを

$$\mathbf{x}_{i+l} \leftarrow (\text{buf}_{m+3l}, \text{buf}_{m+3l+1}, \text{buf}_{m+3l+2}), \quad l = 0, \dots, n-1, \quad (5)$$

のように復元する (速度版も同様)。

■逆通信 pack/unpack_reverse スレッド間ないし MPI 間で力を集約する操作は

$$\mathbf{f}_i += \sum_{\text{送信元}} \mathbf{f}_i^{(\text{recv})}. \quad (6)$$

■境界パック pack_border タグ T_i , 型 s_i , マスク M_i , 電荷 q_i , 分子 ID m_i を含む

$$(\mathbf{x}_i + \mathbf{d}, T_i, s_i, M_i, q_i, m_i) \quad (7)$$

を順に配置する. 速度付き・ハイブリッド版も同型でベロシティ \mathbf{v}_i やサブスタイル特有量を追加する.

■領域交換 プロセッサ境界を越えた粒子について

$$\text{buf}_0 = m, \quad (\text{バッファ長}) \quad (1)$$

$$\text{buf}_{1:3} = \mathbf{x}_i, \quad \text{buf}_{4:6} = \mathbf{v}_i, \quad (2)$$

$$\text{buf}_{7:10} = (T_i, s_i, M_i, I_i), \quad (3)$$

$$\text{buf}_{11} = q_i, \text{buf}_{12} = m_i, \quad (4)$$

$$\text{buf}_{13:*} = \text{結合} \cdot \text{角度} \cdot \text{特殊近傍情報}, \quad (5)$$

を並べ, 受信側は逆操作で $n_{\text{local}} \rightarrow n_{\text{local}} + 1$ とする.

■リスタートサイズ 各原子が出力するダブル値の総数

$$N_{\text{rst},i} = 17 + 2N_i^{\text{bond}} + 4N_i^{\text{ang}} + 5N_i^{\text{dih}} + 5N_i^{\text{imp}} + \sum_{\xi} N_{\xi,i}^{\text{fix}}, \quad (8)$$

であり, プロセッサ内合計が `size_restart()` の返値となる.

■新規原子 与えられた座標 \mathbf{x}_* と型 s で

$$n_{\text{local}} \xleftarrow{+1}, \quad \mathbf{x}_{n_{\text{local}}} = \mathbf{x}_*, \quad \mathbf{v}_{n_{\text{local}}} = \mathbf{0}, \quad q_{n_{\text{local}}} = 0,$$

その他属性は零/既定値.

■Pack/Write Data データファイル用配列は

$$(T_i, m_i, s_i, q_i, \mathbf{x}_i, \mathbf{I}_i), \quad (9)$$

ここで \mathbf{I}_i は 3 方向のイメージフラグ (I_{ix}, I_{iy}, I_{iz}) である.

■メモリ計算 確保済みバイト数

$$B = \sum_{\alpha \in \mathcal{A}} s_{\alpha}(n_{\text{max}}) + 3s_{\mathbf{f}}(n_{\text{max}} N_{\text{th}}), \quad (10)$$

ただし \mathcal{A} は属性集合 $\{\mathbf{x}, \mathbf{v}, \mathbf{f}, q, m_i, \text{bond}, \dots\}$, $s_{\alpha}(N)$ は配列 α を長さ N で確保したバイト数である.

以上の数式で “AtomVecFull” の各関数が実装する操作の本質を網羅した. メモリ動的確保, MPI 通信パッキング, リスタート入出力など高レベル並列 DEM シミュレーションに必須の処理が, 配列-バッファ写像と整数/実数列操作に帰着していることが理解できる.