

AtomVecCharge の数式解説 (atom_vec_charge.cpp)

Open DEM Japan

2025 年 6 月 30 日

LIGGGHTS/LAMMPS における AtomVecCharge クラスは、電荷を持つ剛体球系の状態ベクトルを MPI 並列環境で効率的に格納・通信・入出力するための数値アルゴリズムを実装している。以下ではコードの各機能を抽象化し、離散要素法 (DEM) の状態遷移写像

$$\mathbf{S}(t) = \{ \mathbf{x}_i(t), \mathbf{v}_i(t), \mathbf{f}_i(t), q_i(t) \}_{i=1}^N$$

を伝搬・交換・保存する数学的操作として記述する。

1. メモリ拡張写像

登録可能粒子数 n_{\max} は

$$n_{\max}^{\text{new}} = \begin{cases} n_{\max}^{\text{old}} + \delta, & n = 0, \\ n, & n > 0, \end{cases} \quad \delta = 10^4 \quad (1)$$

で更新される。このとき配列 $\{\mathbf{x}, \mathbf{v}, \mathbf{f}, q, \dots\}$ は

$$\mathbf{A}^{\text{new}} = \text{Grow}(\mathbf{A}^{\text{old}}, n_{\max}^{\text{new}}) \quad (2)$$

により再配置される。写像 $\text{Grow}(\cdot)$ は再割当て - コピーに等しい。

2. 周期境界条件の変位写像

プロセッサ境界通信では、周期境界条件 (PBC) に起因する並進

$$\begin{aligned} \Delta x &= p_x L_x + p_{yz} L_{xy} + p_{xz} L_{xz}, \\ \Delta y &= p_y L_y + p_{yz} L_{yz}, \\ \Delta z &= p_z L_z, \end{aligned} \quad (3)$$

を用いて、

$$\mathbf{x}'_i = \mathbf{x}_i + (\Delta x, \Delta y, \Delta z) \quad (4)$$

が送信座標となる。三斜格子系では $L_{xy}, L_{xz}, L_{yz} \neq 0$, 直交系ではそれらが 0 となる。

3. パッキング写像

粒子インデックス集合 $\mathcal{I} = \{j_1, \dots, j_n\}$ に対し、

$$\mathbf{B} = \text{Pack}(\{\mathbf{x}'_{j_k}, \mathbf{v}_{j_k}, q_{j_k}\}_{k=1}^n) \quad (5)$$

が生成される。写像 Pack の出力は次元バッファ $\mathbf{B} = [B_1, \dots, B_m]$ であり,

$$m = 3n \quad (\text{座標のみ}) \quad (6a)$$

$$m = 6n \quad (\text{座標+速度}) \quad (6b)$$

$$m = 7n \quad (\text{座標+メタ+電荷}) \quad (6c)$$

$$m = 10n \quad (\text{座標+速度+メタ+電荷}) \quad (6d)$$

のいずれかに準拠する。

4. アンパッキング写像

受信側では逆写像

$$\{\mathbf{x}'_i, \mathbf{v}'_i, q'_i\}_{i=1}^n = \text{Unpack}(\mathbf{B}) \quad (7)$$

により \mathbf{B} を局所状態へ復元し, 必要ならば式 (4) の逆変換を適用して物理座標系に戻す。

5. 交換 (マイグレーション) 写像

粒子 i が他プロセッサへ移動する際の送信ベクトルは

$$\mathbf{E}_i = [\mathbf{x}_i, \mathbf{v}_i, \text{Meta}_i, q_i], \quad (8)$$

ここで Meta_i には ID, Type, Mask, Image が含まれる。受信側は \mathbf{E}_i を追加し局所粒子数

$$N_{\text{local}} \leftarrow N_{\text{local}} + 1 \quad (9)$$

とする。

6. 再始動データ写像

再始動ファイルでは粒子ごとに

$$\mathbf{R}_i = [\mathbf{x}_i, \text{Meta}_i, \mathbf{v}_i, q_i, \boldsymbol{\xi}_i] \quad (10)$$

を記録する。ただし $\boldsymbol{\xi}_i$ は fix により追加される拡張自由度ベクトルでサイズ可変である。ファイル行長

$$\ell_i = 12 + \sum_{\text{fix}} \ell_i^{(\text{fix})} \quad (11)$$

が可変長となる。

7. メモリ使用量評価

確保済みバイト数は

$$M = \sum_{A \in \mathcal{A}} \text{bytes}(A), \quad \mathcal{A} = \{\text{tag, type, mask, image, } \mathbf{x}, \mathbf{v}, \mathbf{f}, q\}, \quad (12)$$

で与えられる。ここで $\text{bytes}(A) = \text{sizeof}(A_{\text{elem}}) \times \text{card}(A)$ である。

8. 粒子生成写像

座標 \mathbf{x}^* と型 t が与えられたとき,

$$\mathbf{S} \cup \{(\mathbf{x}^*, \mathbf{0}, q = 0)\} \longrightarrow \mathbf{S}', \quad (13)$$

により新粒子を追加し,

$$N_{\text{local}} \leftarrow N_{\text{local}} + 1 \quad (14)$$

とする。

9. データファイル書式写像

データファイルへの書込み行は

$$(\text{ID}, \text{Type}, q, \mathbf{x}, \mathbf{I}), \quad \mathbf{I} = (I_x, I_y, I_z), \quad (15)$$

であり, $I_\alpha = \text{Image}_\alpha - I_{\text{max}}$ が周期画像フラグとなる。ハイブリッド様式では

$$\text{行} \leftarrow \text{行} \parallel (q) \quad (16)$$

の連結演算を追加する。

以上より, `AtomVecCharge` は, 式(1) - (16) で定義される諸写像を組合せることで MPI 並列 DEM シミュレーションにおける電荷付き粒子群の安定かつ高速なデータ管理を実現している。