

# ”速度ランプ温度モデル”の数式解説 (compute\_temp\_ramp.cpp)

Open DEM Japan

2025年7月2日

粒子群  $G$  に対し、位置座標  $\xi$  に沿って線形に変化する目標速度

$$\mathbf{v}_{\text{ramp}}(\xi)$$

を引いた「熱速度」を用いて温度および運動量テンソルを評価する実装が `compute_temp_ramp.cpp` である。以下ではプログラムの計算過程を連続した数式として示す。

■1. 速度ランプの定義 まず測定対象となる速度成分を

$$\kappa \in \{x, y, z\}, \quad \mathbf{e}_\kappa \text{ は対応する単位ベクトル}$$

で表す。ランプを定義する位置座標成分を  $\xi \in \{x, y, z\}$  とし、その区間を

$$\xi_0 < \xi < \xi_1, \tag{1}$$

速度境界値を

$$V_0, V_1 \quad (\text{対応する単位は速度}) \tag{2}$$

とする。

各粒子  $i$  の位置  $\xi_i$  における無次元係数

$$f_i = \frac{\xi_i - \xi_0}{\xi_1 - \xi_0}, \quad 0 \leq f_i \leq 1 \tag{3}$$

(式中では外挿防止のため  $\max(0, \min(f_i, 1))$  とクリップされる) が与えられる。速度ランプは

$$\mathbf{v}_{\text{ramp},i} = V_0 \mathbf{e}_\kappa + f_i (V_1 - V_0) \mathbf{e}_\kappa. \tag{4}$$

■2. 熱速度 粒子固有速度  $\mathbf{v}_i$  からランプを差し引いた熱速度を

$$\mathbf{c}_i = \mathbf{v}_i - \mathbf{v}_{\text{ramp},i}. \tag{5}$$

■3. 熱運動エネルギーと温度 質量  $m_i$  を用いた熱運動エネルギーは

$$K = \frac{1}{2} \sum_{i \in G} m_i \mathbf{c}_i \cdot \mathbf{c}_i. \tag{6}$$

自由度は

$$\text{DOF} = dN - n_{\text{extra}} - n_{\text{fix}}, \tag{7}$$

ここで  $d$  は次元数 (2 または 3),  $N = |G|$ 。

LAMMPS 単位系からエネルギー単位への換算係数を  $\lambda = \text{mvv}2\text{e}$  とすれば, 瞬時温度  $T$  は

$$T = \frac{2 \lambda K}{\text{DOF } k_B} = \frac{\lambda}{\text{DOF } k_B} \sum_{i \in G} m_i \mathbf{c}_i \cdot \mathbf{c}_i. \quad (8)$$

■4. 運動量テンソル (第 2 モーメント) 熱速度二次モーメント

$$\mathcal{K}_{\alpha\beta} = \lambda \sum_{i \in G} m_i c_{i\alpha} c_{i\beta}, \quad (\alpha, \beta) \in \{x, y, z\}, \quad (9)$$

は対称テンソルであり,

$$(\mathcal{K}_{xx}, \mathcal{K}_{yy}, \mathcal{K}_{zz}, \mathcal{K}_{xy}, \mathcal{K}_{xz}, \mathcal{K}_{yz})$$

の 6 成分が `compute_vector` として返される。

■5. バイアス除去・復元関数 粒子  $i$  に対し

$$\mathbf{v}_i \mapsto \mathbf{v}_i - \mathbf{v}_{\text{ramp},i}$$

とする操作が `remove_bias()`, 逆操作

$$\mathbf{v}_i \mapsto \mathbf{v}_i + \mathbf{v}_{\text{ramp},i}$$

が `restore_bias()` である。全粒子版は `remove_bias_all()` と `restore_bias_all()`。必要なランプ速度  $\mathbf{v}_{\text{ramp},i}$  はすべて式 (??) に基づき座標から即時計算されるため, 追加メモリは

$$3 N_{\text{local}} \times \text{sizeof}(\text{double})$$

程度に抑えられる (`memory_usage()` 参照)。

■6. まとめ `compute_temp_ramp.cpp` は, 位置依存速度ランプ (??) を用いて熱速度 (??) を定義し, 温度 (??) と運動量テンソル (??) を評価することで, せん断流や速度勾配を含む系の「乱れ (熱) 運動」を定量化する。自由度補正 (??) と単位換算  $\lambda$  は `compute_temp_com.cpp` と同様であり, 並列環境では `MPI_Allreduce` により各プロセスの部分和を集約してスカラーやテンソルが得られる。