

マルチコア上での粒子法半陰解法における AoS 再配置を用いた近傍粒子探索

Neighbor search with AoS relocation for MPS method on multicore

齊藤 大輔†
Daisuke Saito

吉田 明正†
Akimasa Yoshida

1 はじめに

流体シミュレーション手法の一つである粒子法 (MPS 法) [1] は、界面の複雑な動きのような現象を扱うことが可能であり、様々なシミュレーションに用いられている。粒子法の高速度化研究としては、CPU による OpenMP を用いたループ並列処理 [2] や SIMD 並列処理 [3], GPU による CUDA を用いた並列処理 [4], 近傍粒子探索において各バケットをスレッドブロックで実行する方法 [5] などが存在する。

本稿では、半陰解法を用いた粒子法において、粒子の位置情報を動的に再配置することによる近傍粒子探索の高速化手法を提案する。粒子法においては、近傍粒子探索が計算時間の大部分を占め、一般的な近傍探索手法としてバケット法がある。標準的な実装では、バケット内粒子の位置情報の管理に線形リストが用いられており、近傍粒子探索時に頻繁な線形リストへのアクセスが発生する。そこで、本論文では、バケット内粒子の位置情報がメモリ上で連続配置されるように再配置を行い、線形リストへのアクセスを軽減する。

性能評価では 3 次元水柱崩壊問題を取り扱い、粒子法コードの並列化には OpenMP を用いた。Intel Xeon Platinum 8358 上での性能評価の結果、提案手法は高い実効性能が得られおり、その有効性が確認された。

2 粒子法半陰解法

2.1 粒子法の概要

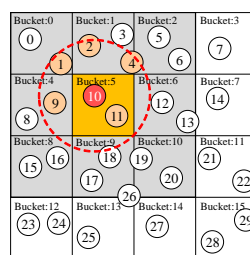
粒子法 (MPS 法) は、非圧縮性流体のシミュレーション手法の一つであり、ナビエ-ストークス方程式と流体の質量保存則を離散化して計算を行う。MPS 法には陽解法と半陰解法の 2 種類の解法が存在するが、本稿では半陰解法を対象とする。

2.2 バケット法を用いた近傍粒子探索

粒子法では、各粒子の影響半径内にある粒子との相互作用を求めながら計算する。一般的な近傍探索手法であるバケット法は、解析領域を立方体で敷き詰めるようにグリッド分割し、対象粒子の近傍粒子を求めるときに、近傍バケット内を探索する手法である。分割した 1 つの領域をバケットと呼ぶ。

粒子法では、バケット法を 3 次元空間で用いるが、説明のため 2 次元における粒子管理を図 1 に示す。図 1(a) において、10 番の対象粒子の影響半径内が赤い円で示した点線である。バケット法を用いない場合、近傍粒子の判定に解析領域内の全ての粒子との距離を比較する必要があるが、バケット法を用いる場合には、グレーのバケットのみを探索すれば良いので探索効率が向上する。

バケット法において図 1(a) の粒子を線形リストを用いて管理する場合、粒子情報は図 1(b), (c) のように 2



(a) バケット分割の例

0 2 5 7 ...
(b) bfst[BN]

1 -1 3 4 -1 6 -1 -1 ...
(c) next[N]

BN: バケット数, N: 粒子数

図 1 バケット法による粒子管理。

RelocatePosition[][]								RelocateReadableRange[]					
Bucket 0	X ₀	Y ₀	Z ₀	X ₁	Y ₁	Z ₁	0	0	0	0	0	0	0
Bucket 1	X ₂	Y ₂	Z ₂	X ₃	Y ₃	Z ₃	X ₄	Y ₄	Z ₄
Bucket 2	X ₅	Y ₅	Z ₅	X ₆	Y ₆	Z ₆	0	0	0	0	0	0	0
Bucket 3	X ₇	Y ₇	Z ₇	0	0	0	0	0	0	0	0	0	0
...

(a) AoS 形式で再配置した粒子位置の配列 (b) 粒子位置を読み込める範囲

図 2 AoS 形式で再配置した粒子位置の配列。

つの配列に格納される。説明のため、図 1(a) のバケット番号 0 ~ 3 までの粒子を格納している。ここで、図 1(b) の bfst[n] は n 番目のバケット内の先頭粒子番号を、図 1(c) の next[i] は粒子 i の所属バケット内の次の粒子番号を格納する。配列 next における -1 は次の粒子が存在せず、粒子 i が所属バケット内の最後の粒子であることを意味する。

3 AoS 再配置を伴うデータレイアウト最適化

粒子法のデータレイアウトに関しては、AoS 形式や AoSoA 形式 [3] が考えられるが、本稿では AoS を対象とする。

3.1 AoS によるデータレイアウト

AoS (Array of Structures) 形式とは、3 次元情報の x, y, z 成分を構造体として連続して配置する方法である。粒子法においては、粒子の位置、速度、加速度などが AoS 形式で格納される。

3.2 AoS 再配置を用いた近傍粒子探索

提案手法は、バケット法に用いるバケット所属粒子を管理する線形リストを利用して、近傍探索に用いる粒子の位置データを AoS 形式で再配置することで、圧力、重力、加速度などを計算する際の線形リストへのアクセス回数の軽減をし、計算時間を短縮する。

提案手法により粒子の 3 次元位置情報を AoS 形式で再配置した配列は図 2(a) である。格納された粒子は同様に図 1(a) に対応している。また、図 2(b) が粒子の位置データを再配置した各バケットの粒子数である。再配置した配列を用いて近傍粒子探索を行うコードを図 3 に示す。

† 明治大学大学院先端数理科学研究科ネットワークデザイン専攻
Network Design Program, Graduate School of Advanced
Mathematical Sciences, Meiji University

```

01: #pragma omp parallel for schedule(dynamic,64)
02: for (i=0; i<NumberOfParticles; i++) {
03:   (粒子 i が所属するバケットの近傍バケット番号 bn を求める処理)
04:   for (int j=0; j<RelocateReadableRange[bn]; j++) {
05:     // 再配置後の近傍候補粒子のインデックスを求める
06:     int k = jbn*RELOCATE_ARRAY_WIDTH + j;
07:     double x = RelocatePosition[k*3] - Position[i*3];
08:     double y = RelocatePosition[k*3+1] - Position[i*3+1];
09:     double z = RelocatePosition[k*3+2] - Position[i*3+2];
10:     double distance = sqrt(x*x + y*y + z*z);
11:     if (distance < InfluenceRadius) {
12:       (近傍粒子の場合に、相互作用の計算を行う)
13:     }
14:   }
15: }

```

図 3 AoS 形式での再配置を用いた近傍粒子探索コード。

従来のバケット法における線形リストを用いた粒子の位置データ読み出しは、アクセスしたバケット内粒子の同じバケットに所属する次の粒子を辿るようにして粒子の位置データにアクセスをしている。この場合、1つのバケットに存在する粒子の位置データはメモリ上で連続して並んでいない状態となる。また、シミュレーション時間の経過につれ、非連続性が増してしまふ。提案する再配置を用いた場合には、1つのバケットに帰属する粒子の位置データがメモリ上に連続して配置されるので、アクセス時間を短縮することができる。再配置にかかる時間増加よりも、再配置後の近傍探索時間の短縮が上回れば高速化が達成される。

4 Intel Xeon Ice Lake 上での粒子法の性能評価

本性能評価では、Intel Xeon Platinum において、半陰解法を用いた粒子法における AoS 再配置を用いたマルチコアによる並列処理コードの性能評価を行う。なお、圧力計算に用いる係数行列の格納には拡張 ELL 形式 [3] を用いる。

評価プログラムとして 3 次元水柱崩壊問題を取り扱い、粒子数 30,976 において 0.003 秒刻みで 667 フレーム (2 秒分) のシミュレーションを行う。連立一次方程式の求解については CG (Conjugate Gradient 法) を採用し、コンパイラの最適化オプションについてはデフォルトの -O2 で行う。評価に用いたマシンは表 1 の通りである。

表 1 性能評価マシン。

マシン	Xeon Ice Lake Server
プロセッサ	Intel Xeon Platinum 8358
コア	32cores, 2.6GHz
メモリ	256GB
OS	Ubuntu 20.04.4
処理系	Intel oneAPI Toolkits icc 2021.6.0

4.1 AoS 再配置の有無の性能評価

提案した AoS 形式での再配置の有無による 1 スレッド実行での比較結果を表 2 に示す。なお、圧力勾配計算に用いる CG 法に関しては今回の AoS 形式での再配置によって高速化されないため、CG 法の実行時間と、その他の近傍探索計算を含む実行時間に分けて性能評価を行う。

CG 法を除いた実行時間においては、線形リストのみ (従来) を用いた場合に 128.5[s]、線形リストと AoS 再配置を用いた場合に 110.1[s] となった。線形リストと AoS 再配置を用いることで、1.17 倍の速度向上が得られた。全体の実行時間においても、1.13 倍の速度向上が得られ、提案する AoS 再配置の有効性が確認された。

4.2 AoS 再配置を用いた OpenMP によるループ並列処理の性能評価

AoS 再配置に加えて、マルチコアによるループ並列処理を適用した結果を図 4 に示す。グラフは、各スレッ

表 2 シングルスレッド実行での再配置の有無 (30,976 粒子)。

適用した最適化	CG 法	CG 法以外	全体
線形リスト	25.4[s]	128.5[s]	154.0[s]
線形リスト + AoS 再配置	25.4[s]	110.1[s]	135.6[s]
速度向上率	1.00[倍]	1.17[倍]	1.13[倍]

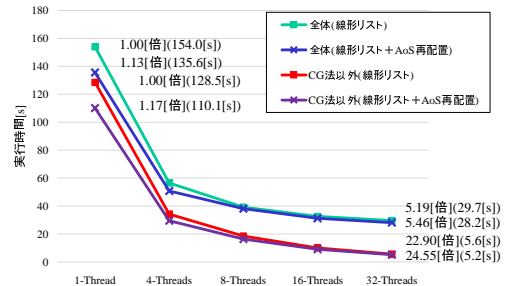


図 4 AoS 再配置を伴うループ並列処理の実行時間 (30,976 粒子)。

ドにおける全体の実行時間と CG 法以外の実行時間を、AoS 再配置の有無により比較している。提案手法による全体の実行では、32 スレッド実行においてシングルスレッド比で最大で 5.46 倍の速度向上が得られ、CG 法以外の実行では、最大で 24.55 倍の速度向上が得られた。これらの結果から、バケット内粒子の 3 次元位置情報の AoS 形式での再配置を用いたマルチコア並列処理の有効性が確認された。

5 おわりに

本稿では、半陰解法を用いた粒子法において近傍粒子探索におけるマルチコア並列処理を実現するために、従来の線形リストを用いた近傍粒子探索に加えて、バケット内粒子の AoS 形式での再配置手法を提案した。Intel Xeon Platinum 上で 3 次元水柱崩壊問題を用いて行った性能評価では、AoS 再配置を採用することでシングルスレッド実行で 1.13 倍の速度向上が得られた。また、マルチコア並列処理を併用したコードでは、逐次実行比で 5.46 倍の速度向上が得られた。以上の結果から、粒子法におけるマルチコア上での AoS 再配置を用いた近傍粒子探索の有効性が確認された。

参考文献

- [1] 越塚誠一, 柴田和也, 室谷浩平. “粒子法入門 流体シミュレーションの基礎から並列計算と可視化まで C/C++ソースコード付”, 丸善出版, 2014.
- [2] M. N. A. Alamsyah, A. Utomo, P. H. Gunawan. “Analysis OpenMP performance of AMD and Intel architecture for breaking waves simulation using MPS”, Journal of Physics: Conference Series, 2018.
- [3] 齊藤大輔, 吉田明正. “粒子法のマルチコア/SIMD 並列処理におけるデータレイアウト”, 情報処理学会研究報告 Vol.2023-HPC-188 No.5, 2023.
- [4] Shuai Zhang, Wenjin Gou, Yuqi Wang, Jifa Zhang, Yao Zheng. “Direct numerical simulation of atomization by jet impact using moving particle semi-implicit method with GPU acceleration”, Springer Computational Particle Mechanics, 2022.
- [5] Takaaki Miyajima, Kenichi Kubota, Naoyuki Fujita. “An optimization of search for neighbour-particle in MPS method for Xeon, Xeon Phi and GPU by using directives”, 1st International conference series on HPC technologies in Asia Pacific region, 2018.