

# 粒子法のマルチコア/AVX-512 命令による 並列処理のための疎行列データ構造

Sparse Matrix Data Structure for Parallelization of Particle Method by Multicore/AVX-512 Intrinsics

齊藤 大輔†  
Daisuke Saito

吉田 明正†  
Akimasa Yoshida

## 1 はじめに

粒子法 [1][2] は流体シミュレーション手法の一種であり、界面の複雑な動きのような激しい現象を扱うことができる。半陰解法を用いた粒子法 (MPS 法) の高速化においては、計算の大部分を占める連立一次方程式の求解の高速化が必要とされる。また、大規模計算を行うためには効率的な疎行列の格納が必要となる。

しかしながら、文献 [3] では、マルチコア/AVX-512 命令 [4] による粒子法の高速化が実現されているが、疎行列データ構造は導入されていなかった (対角要素を各行の先頭に配置する) を提案し、そのデータ構造により実装した粒子法プログラムの並列化を実現する。

性能評価では、30,976 粒子および 357,120 粒子の 3 次元水柱崩壊問題を取り上げ、拡張 ELL 形式を用いた粒子法プログラムにおいて、OpenMP/AVX-512 intrinsics による並列処理を実現した。Intel Xeon Platinum 8358 上での性能評価を行い、提案手法の有効性を確認する。

## 2 粒子法における疎行列データ構造

粒子法の 3 次元水柱問題において連立一次方程式求解に用いる疎行列は、粒子数 30,976 では非ゼロ要素が 0.093%、粒子数 357,120 では非ゼロ要素が 0.014% となるため、効率的なメモリ格納方法を用いることで大幅なメモリ削減と計算速度の向上が得られる。

### 2.1 ELL 形式による係数行列の格納

係数行列格納について今回は、ELL(Ellpack-Itpack) 形式 [5] を採用する。図 1 が ELL 形式での要素格納の図である。ELL 形式とは、非ゼロ要素のみを行ごとに左詰めで格納した行列の形式であり、 $N \times N$  の疎行列を 2 つの  $N \times M$  の ELL 行列 (Value 行列, Column Indices 行列) に格納する。ここで、 $M$  は疎行列の 1 行における最大の非ゼロ要素数である。ELL 形式を用いることで、アクセスする必要のないゼロ要素へアクセスする回数が減り、高速化を見込める。

### 2.2 拡張 ELL 形式による係数行列の格納

提案する拡張 ELL 形式では、疎行列の対角要素を先頭に配置することで計算効率の向上を図る。図 2 は、拡張 ELL 形式での要素格納を表している。図 3 は、ELL 形式の行列格納関数の処理の一部を切り出したものである。

粒子  $i$  に対する近傍粒子  $j$  がある場合、係数行列  $A$  の対角要素  $a_{ii}$  に重みを代入する処理が必要になる。標準的な ELL 形式を用いる場合は、実行前に対角要素が ELL 行列の何列目に保存されるかが確定していないため、確定後にインデックスの保存処理が必要となる。しかし、拡張 ELL 形式を用いる場合は、対角要素を先頭に配置するので保存位置情報が不要になり処理の高速化が見込

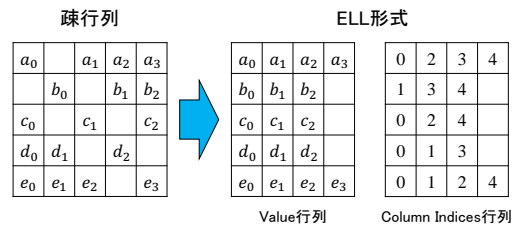


図 1 ELL 形式 .

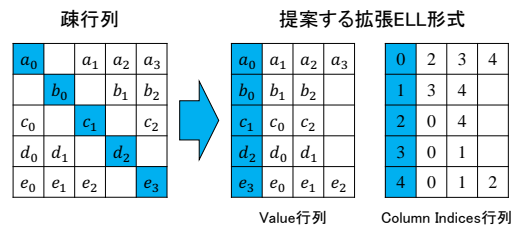


図 2 拡張 ELL 形式 .

める。つまり、図 3 における 7~12 行目の if 文の処理を省略することができる。

また、粒子法のプログラムでは係数行列を設定した後に、例外処理としてポアソン方程式のディリクレ境界条件に満たない粒子の対角要素にアクセスする必要がある。提案する拡張 ELL 形式では対角要素のインデックス情報が不要なため、標準的な ELL 形式と比較して、対角要素のインデックスを保存する配列を定義しないことによる粒子数  $N \times 4\text{byte}$  のメモリの節約と、配列読み出し処理の高速化が可能になる。図 3 では、30 行目にあたる処理を省略することができる。

## 3 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拡張 ELL 形式を用いたマルチコア/AVX-512 命令による並列処理コードの性能評価を行う。尚、圧力計算に用いる係数行列の格納には提案する拡張 ELL 形式を用いる。

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、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 である。

### 3.1 拡張 ELL 形式の有無の性能評価

逐次実行において連立一次方程式の係数行列に用いる拡張 ELL 形式の有無による結果を比較する。拡張 ELL 形式無しでは、疎行列を  $N \times N$  の行列として定義したものを使用する。ELL 形式のみを比較するために OpenMP や AVX-512 命令は用いない。

† 明治大学総合数理学部ネットワークデザイン学科  
Department of Network Design, School of Interdisciplinary  
Mathematical Sciences, Meiji University

```

01: for (i=0; i<NumberOfParticles; i++) {
02:     int ELLInd = 0; // ELL行列に代入するためのカウンタ
03:     int ELL_DiagonalInd = 0; // 対角要素がELL行列の何れ目か
04:     double diagonal = 0; // 対角要素を入れておくための変数
05:
06:     for (j=0; j<NumberOfParticles; j++) {
07:         // 対角要素の時、ELL行列におけるインデックスを保存する
08:         if (j == i) {
09:             ELL_DiagonalInd = ELLInd;
10:             ELLInd += 1;
11:             continue;
12:         }
13:
14:         // 粒子jが近傍粒子だった場合に、重みcoefficientIJを計算(省略)
15:
16:         // ELL行列への代入
17:         Mat_ELLVal[i*ELL_WIDTH + ELLInd] = (-1.0)*coefficientIJ;
18:         Mat_ELLCol[i*ELL_WIDTH + ELLInd] = j;
19:         // カウンタの更新と対角要素変数の加算
20:         ELLInd += 1;
21:         diagonal += coefficientIJ;
22:     }
23:
24:     // ELLVal行列に対角要素の値を保存する
25:     Mat_ELLVal[i*ELL_WIDTH + ELL_DiagonalInd]
26:         = diagonal + (COMPRESSIBILITY)/(DT2);
27:     // ELLCol行列に対角要素のインデックスを保存する
28:     Mat_ELLCol[i*ELL_WIDTH + ELL_DiagonalInd] = i;
29:     // ELL行列の対角要素のインデックスを配列に保存する
30:     ELL_diagonalIndex[i] = ELL_DiagonalInd;
31: }
    
```

図3 標準的な ELL 形式を用いた係数行列の生成関数 .

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	icc 2021.6.0

図4が逐次実行の結果である。グラフ上の緑色部分は係数行列格納の関数の実行時間を、水色部分はCG法の実行時間を、青色部分はその他の実行時間を示す。係数行列格納に拡張 ELL 形式を用いることで、主に CG 法の実行時間が短縮され CG 法単体の比較において、ELL 無しでは 9358.3[s] だった実行時間が 25.6[s] となり 365.6 倍の速度向上が得られた。係数行列格納の関数 (SetMatrix Method) においては、ELL 無しでは 672.2[s] だった実行時間が 305.6[s] となり 2.20 倍の速度向上が得られた。粒子法全体では、4.12 倍の速度向上が得られた。

### 3.2 拡張 ELL 形式を用いた OpenMP と AVX-512 による階層的並列処理の性能評価

図5に拡張 ELL 形式を用いた係数行列格納に加えて、OpenMP によるループ並列処理と AVX-512 を用いた SIMD 並列処理を適用した結果を示す。32 スレッド実行では、全体の実行時間が 179[s] となり、逐次実行時間 12,703[s] と比較して 70.9 倍の速度向上が得られた。CG 法の実行時間については 2.46[s] となり、逐次実行時間 9,358.3[s] と比較して 3804.4 倍の速度向上が得られた。また、30,976 粒子よりもさらに粒子数を増やし、357,120 粒子で計測した結果が表2である。ELL 形式を用いることで、大規模な問題でも計算を行えることが確認された。

表2 粒子法の並列処理時間 .

水柱崩壊モデルの粒子数	32-Threads/AVX-512 実行
30,976[個]	179[s]
357,120[個]	31,551[s]

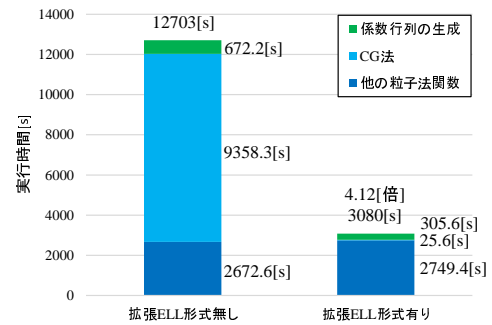


図4 拡張 ELL の有無による逐次実行 (30,976 粒子) .

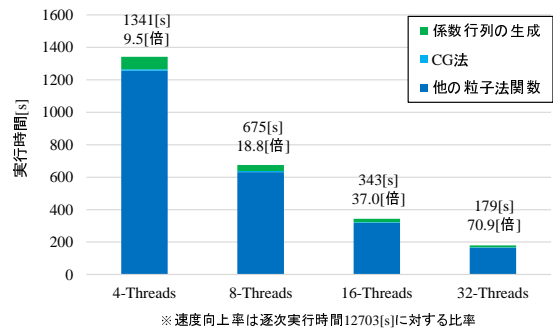


図5 拡張 ELL を伴うループ/SIMD 並列処理 (30,976 粒子) .

これらの結果から、本稿で提案するマルチコア/AVX-512 命令による並列処理を実現するための拡張 ELL 形式の有効性が確認された。

## 4 おわりに

本稿では、半陰解法を用いた粒子法においてマルチコア/AVX-512 命令による並列処理を実現するために、従来の疎行列格納に用いられる ELL 形式を拡張し、対角要素を各行の先頭に配置するデータ構造を提案した。

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、ELL 形式を採用することで逐次実行比で 4.12 倍の速度向上が得られた。また、マルチコア/AVX-512 命令を併用したコードでは、逐次実行比で 70.9 倍の速度向上が得られた。以上の結果から、粒子法における拡張 ELL 形式を用いたマルチコア/AVX-512 命令による並列処理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] プロメテック・ソフトウェア株式会社. “粒子法 CFD ソフトウェア Particleworks の最近の産業応用事例”, [https://www.pcluster.org/ja/event/data/ws20210910\\_7-pccc-oss\\_tokura.pdf](https://www.pcluster.org/ja/event/data/ws20210910_7-pccc-oss_tokura.pdf), 2021.
- [3] 齊藤大輔, 吉田明正. “半陰解法を用いた粒子法におけるマルチコア/SIMD 命令による高速化”, 情報処理学会研究報告 Vol.2022-HPC-185 No.5, 2022.
- [4] 北山洋幸. “512 ビット・ベクトルプログラミング入門 AVX-512 命令を駆使しよう”, カットシステム出版, 2021.
- [5] Masatoshi Kawai, Kengo Nakajima. “Low/Adaptive Precision Computation in Preconditioned Iterative Solvers for Ill-Conditioned Problems”, HPCAsia2022 Pages 30-40.