Vol.2013-ARC-207 No.7
Vol.2013-HPC-142 No.7
2013/12/16

マルチカラー接触判定格子を用いた粒子接触判定計算の OpenMP による並列化

加藤　淳也[1,A)]　河村　祥太[1]　竹田　宏[2]　片桐　孝洋[3]　堀端　康善[1]

概要：粒子解析においては接触判定計算が最も計算負荷が大きく，並列化による高速化が必要である．接触判定計算を並列化すると，複数のスレッド間で同時に接触情報を読み書きするので，排他制御が必要となる．排他制御のコストは高く，特に 8 スレッドを超える高スレッド実行で性能劣化を生じる．そこで本研究では，排他制御が不要な新方式である，マルチカラー接触判定法を提案する．FX10 を用いた性能評価の結果，従来法に対し，スレッド数 16 で約 5 倍の性能向上が出来ることを確認した．

1. はじめに

粉体計算や MPS(Moving Particle Semi-implicit)法による粒子解析においては，対象とする問題ごとに様々な運動方程式を立てて計算を行う[1]．その際，共通して行われる処理は，粒子接触判定計算である．粒子接触判定計算は，全体の時間の大部分を占めると言われており[1],[2][3],[4]，高速化が必須の処理である．特に近年の計算機においては，ノード内に 8 スレッド以上の並列実行可能な CPU が普及している．したがって，並列化による高速化が課題となっている．粒子接触判定計算においては，従来，接触判定計算の計算量を減らす方法が使われてきた．接触判定計算を全粒子について行うのではなく，接触する可能性のある近傍の粒子についてのみ物理量の計算を行う．この近傍の粒子を絞り込むために，接触判定格子を用いる方法[1]がある．また，キャッシュを有効利用することで並列化時の台数効果を上げるため，粒子に番号を振ったうえで，粒子を接触判定時の演算が局所化するように並び替える方法[5]がしられている．

本報告では，接触判定計算において更なる高速化のため，新しい接触判定計算の方法を提案する．粉体計算で用いられる離散要素法(Discrete Element Method; DEM)[6]で必要となる接触する各粒子に働く接触力の足しこみ計算において，従来は，作用-反作用の法則を用いて演算量の削減を行っている．この従来法をスレッド並列化する際，接触する相手の情報も同時に更新することになるが，この操作は他スレッドが更新する情報の書き換えにほかならない．従って，複数のスレッド間でデータの読み書きが競合し，排他制御が必要となる．この排他制御は OpenMP を利用する場合，critical 指示子や atomic 指示子を入れるのが普通である[1]．一方，critical 指示子等を入れない並列化では，接触時に相手の情報を更新しない方法により排他制御を回避してきた．しかしこの方法は，critical 指示子を入れる方法に対して，約 2 倍の演算量の増大になる．つまり，排他制御する方法では演算量が 1/2 になるメリットがある反面，排他制御によるスレッド並列化のオーバーヘッドが生じる．一方で，排他制御がない方法では演算量が増える反面，スレッド並列化時のオーバーヘッドが少ない．つまり，両者は欠点と利点があり，状況に応じて最適な方法が異なる．

そこで本研究では，演算量は critical 指示子による方法と同じく 1/2 にしつつも，排他制御を必要としない新しい方法を提案する．この方法を，**マルチカラー接触判定法**と呼ぶ．マルチカラー接触判定法で用いる接触判定格子のことを，**マルチカラー接触判定格子**と呼ぶ．本稿では従来の方法とマルチカラー接触判定法とのスレッド並列実行性能を比較することで，提案法の有効性を検証することを目的とする．

2. 接触判定計算

2.1 プログラム全体の流れ

本研究における接触判定計算までの，プログラム全体の手順を図 1 に示す．初めに，粒子を疑似乱数（以後，乱数）で発生させて粒子を登録する．次に粒子番号の並び替えを

†1 1.　法政大学大学院理工学研究科
Graduate School of Science and Engineering, Hosei University
†2 2.　株式会社アールフロー
R-flow Corporation, Ltd.
†3 3.　東京大学情報基盤センター
Information Technology Center, The University of Tokyo
A) junya.kato.3g@stu.hosei.ac.jp

行う．

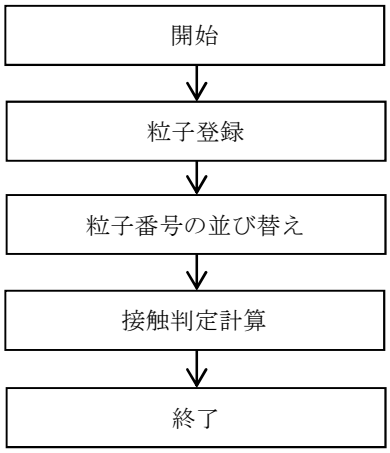

図 1 プログラム全体のフローチャート

2.2 接触判定格子と粒子登録

本研究では，解析対象となる三次元領域に任意の数だけ粒子を発生させる．粒子の位置は，乱数で決められる．各粒子について接触判定計算を行う．この接触判定計算には，接触する可能性のある粒子を判定するために，接触判定格子を用いる．図 2 に，接触判定格子のイメージを載せる．

図 2 では，計算領域を格子状に分割する．その後，格子内に入っている全粒子を，対応する判定格子上に登録する．格子幅を粒子直径と同じにすることにより，接触する可能性のある粒子を同一格子内と隣接格子内に存在する粒子に限定することができるため，計算量を減らすことができる．

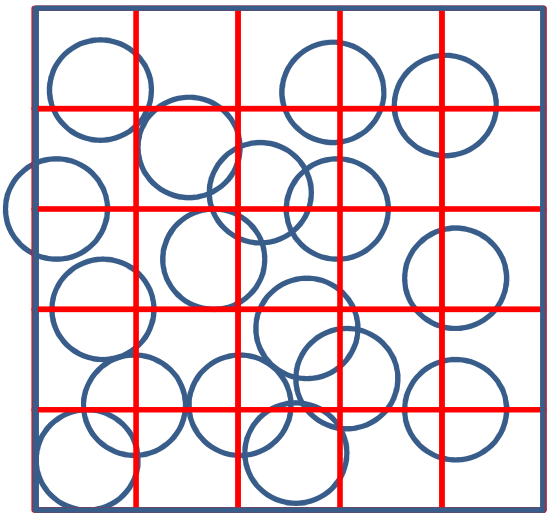

図 2 接触判定格子のイメージ

2.3 逐次計算

接触する粒子に対する接触力を更新する際，自分自身の情報だけでなく，相手の情報も同時に更新する．これは，作用-反作用の法則を用いて演算量を 1/2 に削減するための処理である．図 3 にプログラムの一部分を記載する．

```
<1>do n=ns, ne
<2>    do n1=n1s, n1e
<3>     …
<4>     fxyz(1,n)=fxyz(1,n)+fx
<5>     fxyz(2,n)=fxyz(2,n)+fy
<6>     fxyz(3,n)=fxyz(3,n)+fz
<7>     fxyz(1,n1)=fxyz(1,n1)-fx
<8>     fxyz(2,n1)=fxyz(2,n1)-fy
<9>     fxyz(3,n1)=fxyz(3,n1)-fz
<10>    …
<11> end do
<12>end do
```

図 3 接触判定計算における競合

図 3 では，n は接触力を計算している粒子番号，n1s～n1e は粒子番号 n が入っている格子に存在する粒子群についての最初の粒子番号から最後の粒子番号，fxyz は総接触力を収納している配列，そして fx, fy, fz は粒子番号 n に働く接触力を示している．図 3 からわかるように，自分の接触力である図 3<4>～<6>の fxyz(*,n)の更新と，相手の接触力である図 3<7>～<9>fxyz(*, n1)の情報を更新している．

図 3 の処理について，図 3<1>のループ n を OpenMP 並列化する場合を考える．このとき，n について，粒子番号 0 番と粒子番号 1 番の粒子が，同一格子内にあるとする．また，粒子番号 0 番の粒子はスレッド 0，粒子番号 1 番の粒子はスレッド 1 で処理されるとする．

ここで，粒子番号 0 番の粒子の図 3 での処理は，n=0 および n1=1 に相当する．このとき，スレッド 0 において，fxyz(*,n1)の更新（ここでの n1 は 1）を行っている最中に，スレッド 1 において，fxyz(*,n)（ここでの n は 1）の更新を行う状況がありうる．すなわち，スレッド 0 とスレッド 1 の間で，fxyz(*,n)と fxyz(*,n1)が同じ配列の要素を指しており，同時に値を更新しようとするタイミングがありうる．この状態では，スレッド 0 もしくはスレッド 1 の結果のみが反映され，逐次の結果と一致しなくなる．したがって，図 3 の処理は，排他制御が必要である．

以降，図 3 の衝突判定計算において，スレッド間で値を同時に書き込むことを「競合」と呼ぶことにする．

2.4 競合の回避方法

図 3 をスレッド並列化した場合の競合について前節で説明した．ここでは，この競合の問題を回避する実装について述べる．従来，図 3 の競合を回避するには，OpenMP で提供される critical 指定子を用いたディレクティブである critical 文や atomic 文により，排他制御を行うことで競合の問題を制御してきた[6].

Vol.2013-ARC-207 No.7
Vol.2013-HPC-142 No.7
2013/12/16

また，critical 文を使わない方法として，図 3 の演算について，接触時に相手の情報を更新しないようにすることで，競合を回避してきた．ただしこの方法では，接触力計算に関して冗長計算を行うことから，演算量が 2 倍に増加する．

以上の 2 実装のプログラムを，図 4 に示す．

図 4<3>～<11>では，critical 文を使う従来法では，対象範囲を critical 文で囲み，排他制御を行う．図 4<12>～<15>の冗長計算をする方法では，相手の接触力の値を更新しないため，fxyz(*, n1)の式が存在しない．一方，図 4<16>～<23>の提案法であるマルチカラー接触判定法では，相手の接触力の更新である fxyz(*, n1)の式があるが，critical 文は存在しない．なお，マルチカラー接触判定法については，第 3 節で説明する．

```
<1> do n1=n1s,n1e
<2>   ･･･
<3>   if( critical 文を使う実装 ) then
<4>!$omp critical
<5>       fxyz(1,n)=fxyz(1,n)+fx
<6>       fxyz(2,n)=fxyz(2,n)+fy
<7>       fxyz(3,n)=fxyz(3,n)+fz
<8>       fxyz(1,n1)=fxyz(1,n1)-fx
<9>       fxyz(2,n1)=fxyz(2,n1)-fy
<10>      fxyz(3,n1)=fxyz(3,n1)-fz
<11>!$omp end critical
<12> else if(  冗長計算を使う実装) then
<13>   fxyz(1,n)=fxyz(1,n)+fx
<14>   fxyz(2,n)=fxyz(2,n)+fy
<15>   fxyz(3,n)=fxyz(3,n)+fz
<16> else if(マルチカラー接触判定法) then
<17>   fxyz(1,n)=fxyz(1,n)+fx
<18>   fxyz(2,n)=fxyz(2,n)+fy
<19>   fxyz(3,n)=fxyz(3,n)+fz
<20>   fxyz(1,n1)=fxyz(1,n1)-fx
<21>   fxyz(2,n1)=fxyz(2,n1)-fy
<22>   fxyz(3,n1)=fxyz(3,n1)-fz
<23> end if
<24>  ･･･
<25>end do
```

図 4 接触判定計算の競合
を回避するプログラム

3. マルチカラー接触判定法

この節では，提案法であるマルチカラー判定法を説明する．マルチカラー判定法は，マルチカラー判定格子を用いた，新しい接触判定計算の方法である．

マルチカラー判定格子とは，従来の接触判定計算に用いる接触判定格子を用いるが，接触判定計算のためのアクセスの仕方が異なる接触判定格子のことである．ここではまず，1 次元の場合における例を示すことで，マルチカラー接触判定法のアルゴリズムを説明する．

図 5 は，1 次元の解析対象領域に対するマルチカラー接触判定格子と格子番号を示している．

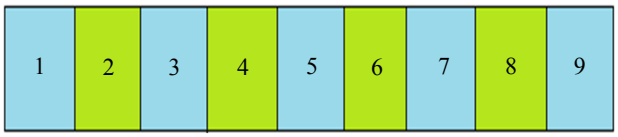

図 5　1 次元でのマルチカラー接触判定
格子のイメージ

図 5 では，従来の接触判定格子のデータ・アクセスパターンは，粒子の接触判定をする理由から，格子につけられた番号（格子番号）順にアクセスする．図 5 の例では，格子番号順に 1→2→3→4→5→…とアクセスしていく．このとき，OpenMP 並列化を考慮すると，接触判定格子の性質上，対象格子番号 i の両隣格子である，格子番号(i-1)と格子番号(i+1)には依存関係があるが，それ以外の格子には依存関係がないため，並列実行が可能である．

そこで，並列実行が可能である格子番号 i (i=1, 3, 5, 7, 9)の接触判定計算を並列に行う．その後，格子番号 j (j=2, 4, 6, 8)の接触判定を行う．このように，マルチカラー接触判定格子を用いた接触判定計算の方法を，マルチカラー接触判定法と呼ぶ．

マルチカラー接触判定法の適用の前提は，接触格子の格子幅を粒子直径，あるいはそれ以上に取ることである．そうすることで，同じ色の格子内にある粒子同士は接触することがない．そのためマルチカラー接触判定法では，自分と相手の接触力を更新する際に，原理上，競合が起きない．その結果，並列化をしても複数のスレッド間でデータの競合が起きない．

なお，マルチカラー接触判定法では，逐次の接触判定計算とは加減算の順序が異なる．ただし，数学上における演算結果の一致は保証される．

以上は 1 次元の解析対象に対するマルチカラー接触判定法の説明である．2 次元，3 次元に解析対象を広げる場合においても，マルチカラー接触判定法は，マルチカラー接触判定格子を 2 次元，3 次元に拡張することで拡張が可能となる．

本研究では，3 次元問題を扱う．マルチカラー接触判定

Vol.2013-ARC-207 No.7
Vol.2013-HPC-142 No.7
2013/12/16

法を 3 次元問題に適用するには，図 6 のように，3 次元にマルチカラー接触判定格子を拡張する．図 6 では，接触判定格子上のデータ依存を回避するため，8 色に色分けする．このことで，1 次元と同様の議論で，マルチカラー接触判定法が実現できる．

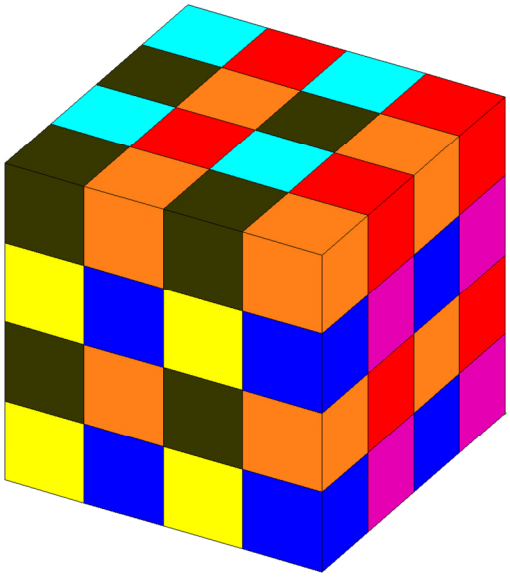

図 6　3 次元のマルチカラー接触判定格子のイメージ

4. 性能評価

4.1 計算環境・初期設定

本性能評価では，東京大学に 2012 年に導入された FX10 スーパーコンピュータシステム（富士通 PRIMEHPC FX10，以降 FX10 と呼ぶ）の 1 ノードを用いて行った．1 ノード内のコア数は 16 であるため，OpenMP による並列化は最大の 16 スレッドまで行った．

表 1. FX10 構成

項目		仕様
ノード	理論演算性能	236.5GFlops
	プロセッサ数（コア数）	16
	主記憶容量	32GB
プロセッサ	プロセッサ名	SPARC64 IX-fx
	周波数	1.848GHz
	理論演算性能（コア）	14.78GFlops
ソフトウェア	OS	Red Hat Enterprise Linux Server 6.1
	コンパイラ	Fujitsu Fortran Compiler
	コンパイラオプション	-Kfast,openmp -Ksimd=2

今回の性能評価では，粒子径は 0.002，粒子数は 6250 万，ステップ数は 20 に設定した．また判定時に各粒子は接触数の情報を更新するが，3 次元の接触判定格子を利用するため，1 粒子あたりの接触数情報の上限は 12 とする．

提案手法の有効性を見るために，表 2 のように 5 つのプログラムを用意した．

表 2. Method による実装方式の違い

method	実装方式
Method 1	粒子並列
Method 2	粒子並列（冗長計算）
Method 3	接触判定格子並列
Method 4	接触判定格子並列（冗長計算）
Method 5	マルチカラー接触判定格子並列

ここで，粒子番号で最外ループを回し，その最外ループを並列化する方法を粒子並列と呼ぶ．一方で，接触判定格子番号で最外ループを回し，その最外ループを並列化する方法を接触判定格子並列と呼ぶ．

この時，表 2 では，並列化時のデータの競合を防ぐために critical 文を入れた粒子並列のプログラムを method1 とする．粒子並列において，相手の情報を書き込まず，計算量が 2 倍になるプログラムを method2 とする．また，接触判定格子並列化時のデータの競合を防ぐために critical 文を入れたプログラムを method3 とする．接触判定格子並列において，相手の情報書き込まず，計算量が 2 倍になるプログラムを method4 とする．以上の method1～method4 は従来法である．最後に，今回の提案手法であるマルチカラー接触判定格子を用いて並列化するプログラムを method5 とする．

4.2 OpenMP による並列化

FX10 の 1 ノードを占有し，OpenMP により最大 16 スレッドで並列実行したときの計算時間と台数効果について性能を評価する．ここで台数効果とは，並列処理の効果を示す一般的な指標である．式(1) のように「逐次実行における計算時間」に対する「並列実行における実行時間」の比で表される．

$$S = \frac{\text{逐次実行に要する計算 時間}}{\text{並列計算に要する計算 時間}} \qquad (1)$$

なお，式（1）における計算時間とは，接触判定計算に要する計算時間 T_c と粒子の接触力計算時間 T_f を合計した総計算時間 T_{ALL} の事を指す．

4.3 接触判定計算の性能評価

4.3.1 計測時間の比較

Vol.2013-ARC-207 No.7
Vol.2013-HPC-142 No.7
2013/12/16

各 method において，逐次計算と並列計算（スレッド数 2，4，8，および 16）での計算時間の比較を行う．ここで計算時間とは，5 ステップ目以降の計算時間を測定し，その値を平均したものである．計算時間の結果を，表 3 に示す．

表 3. 各 method の実行時間（秒）

THREADS	Method 1	Method 2	Method 3	Method 4	Method 5
1	81.46	58.24	34.13	56.00	36.38
2	57.22	58.74	56.49	58.44	37.36
4	48.48	39.90	47.82	39.04	24.02
8	54.62	31.79	54.48	30.59	18.15
16	81.09	29.15	81.33	27.87	16.14

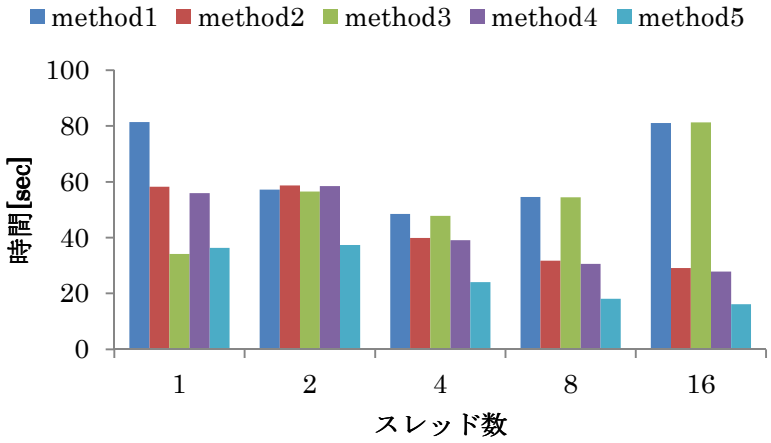

図 7 各 method における接触力判定計算時間

4.3.2 台数効果の比較

各 method において逐次計算と並列計算（スレッド数 2，4，8，および 16）での台数効果の比較を行う． 計算時間を，以下の表 4 に示す．また，method5（提案法）に対する，従来法の各 method の比較を図 8，9 に示す．

表 4. 台数効果

THREADS	method1	method2	method3	method4	method5
1	1.00	1.00	1.00	1.00	1.00
2	1.42	0.99	0.60	0.96	0.97
4	1.68	1.46	0.71	1.43	1.51
8	1.49	1.83	0.63	1.83	2.00
16	1.00	2.00	0.42	2.01	2.25

図 8 は，critical 指示子を用いた method1，method3 と，マルチカラー接触判定法を用いた method5 の台数効果の比較である．

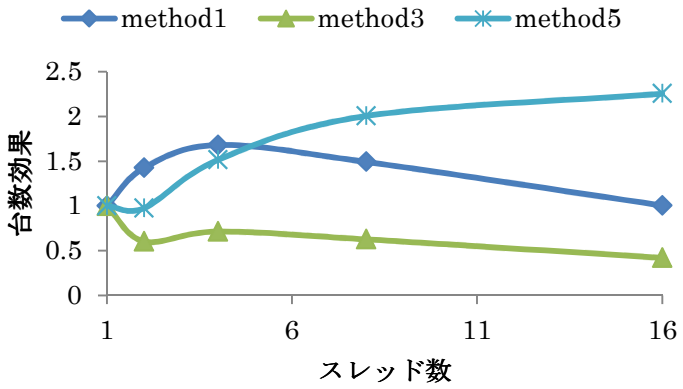

図 8 method5(提案法)と，従来法である method1 と method3 との台数効果の比較．

図 8 より，method1，method3 の両方で，4 スレッドを超えると台数効果の劣化が確認できる．特に 8 スレッドを超える高スレッド実行で性能劣化の発生を確認できた．これは，排他制御にかかるコストの高さが原因であると考えられる．

冗長計算を用いた method2, method4 とマルチカラー接触判定法を用いた method5 の台数効果の比較を，以下の図 9 に示す．

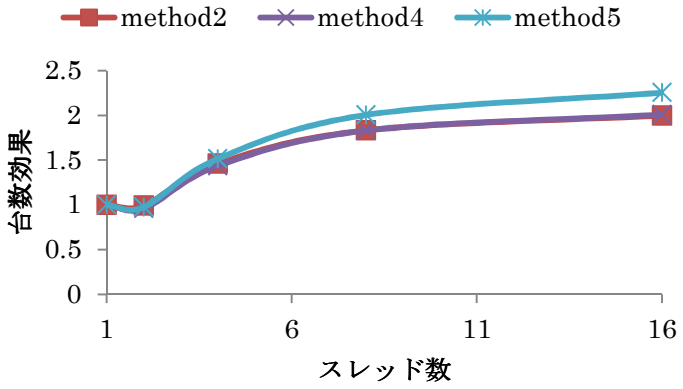

図 9 method5（提案法）と従来法の method2，method4 との台数効果．method5 の 2 スレッドでの実行時間を 1 としている．

図 9 より，method2，method4 では排他制御を用いていない理由から，図 8 での method1，method3 のような性能の劣化がない．また，排他制御や冗長計算を行わないマルチカラー接触判定法を用いた method5 は，従来法である method1~method4 の手法よりも良い台数効果が確認できる．

また表 3 から，16 スレッド時において，提案法を利用することで最大で 5.03 倍（method3 の 81.33[秒]に対する，method5 の 16.14[秒]）の速度向上が得られることがわかる．

5. まとめ

本研究は，粉体解析等で用いられる接触判定計算において，マルチカラー接触判定格子を用いた新しいスレッド並列化手法である，マルチカラー接触判定法を提案した．また，いくつかの従来法に対する性能評価を行った．

我々の先行研究[3]では格子並列を採用しているが，計算の効率化のために，粒子番号のリナンバリングを行っている．本研究で提案したマルチカラー接触判定法では，このリナンバリングに加え，格子のアクセスの仕方を工夫することで，スレッド並列化時にデータ競合が起こらないようにしている．そのため，従来法に対しより高い並列化効率が期待できる．また，我々の先行研究[4]では，粒子分布に偏りがある場合についても性能評価を行っている．先行研究[4]では，粒子番号ごとに計算する粒子並列との性能評価も行っている．これらの従来法と比較して，本研究で提案するマルチカラー接触判定法は，計算時間と台数効果の両方において，従来手法より優れた結果を得た．

今回の性能評価では，マルチカラー接触判定法は 16 スレッドでの台数効果が従来法より優れていた．しかし、理想値である 16 倍にはまだ及ばない．この理由は，初期粒子の配置に乱数を用いているので計算負荷に偏りがあるかもしれないこと，接触判定計算のための計算量が 16 スレッド実行時には少なすぎること，などが考えられるが，この原因分析は今後の課題である．また，性能プロファイラによると，提案法の L1 キャッシュ・ミスヒット率について，提案法は従来法よりも改善されるものの，約 85%と高かった．キャッシュ・ミスヒット率を改善するための方法の提案も今後の課題である．

6. 謝辞

本研究の一部は，学際大規模情報基盤共同利用・共同研究拠点，および，革新的ハイパフォーマンス・コンピューティング・インフラの支援による．

7. 参考文献

1) 酒井幹夫編著 : 粉体の数値シミュレーション，丸善出版 (2012)
2) Yusuke Shigeto, Mikio Sakai : Parallel Computing in Computational Granular Dynamics by Using Multi-Core Processors -Practical Usage of the DEM in Complicated Powder Systems in Industries-, J. Soc. Powder Technol, Japan, 47, pp.707-716 (2010)
3) 和田直樹, 高木翔, 岡大樹, 竹田宏, 片桐孝洋, 堀端康善: 粒子接触判定計算の OpenMP による最適化，情報処理学会研究報告，Vol.2012-HPC-136, No.3(2012)
4) 高木翔,和田直樹, 岡大樹, 竹田宏, 片桐孝洋, 堀端康善: 粒子分布を考慮した粒子接触判定計算の MPI および OpenMP による並列化，vol.2012-HPC-137-34(2012)
5) 西浦泰介，阪口秀: GPU を用いた DEM の高速化アルゴリズム，Transactions of JSCES, paper No.20100007 (2010)
6) P. A. Cundall and O. D. L. Strack, A Discrete Numerical Model for Granular Assemblies, Geotechnique, 29, pp. 47-65, 1979