

複雑境界形状の大規模固液混相流解析のための動的負荷分散を考慮した並列計算法

福田 朝生¹・福岡 捷二²

¹正会員 博(工) 中央大学研究開発機構准教授 (〒112-8551 東京都文京区春日 1-13-27)
E-mail: t-fukuda@tamacc.chuo-u.ac.jp

²フェロー 工博 Ph.D. 中央大学研究開発機構教授 (同上)
E-mail: sfuku@tamacc.chuo-u.ac.jp

本稿では、複雑境界形状の領域の大規模固液混相流解析に適した並列計算法 (DBA法) を構築した。この手法は、一つのプロセスに複数立方体計算領域ブロックを分担させかつ、プロセス間の計算負荷をバランスさせるように、動的にブロックのプロセスへの割り当てを変更するものである。構築した並列計算法の性能を確認するため、プロセスに単一ブロックのみを割り当てる並列計算法 (SB法) と DBA法を用いて粒子の沈降・落下の解析を行った。SB法では負荷の小さい領域に計算資源を固定化し、プロセス間の大きな負荷のアンバランスを生じるのに対し、DBA法はブロックのプロセスへの割り当てを動的に変更し負荷をバランスさせるため、SB法と比較しより高速に計算が行えることを確認した。

Key Words: parallelization technique, dynamic load balance, computation speed, complex boundary geometry, interface-resolved simulation

1. 序論

固体と流体の相互作用の高精度の解析法として、固体を剛体としてLagrange的に取り扱い、また剛体よりも小さな計算格子を用いて Euler 的に流体を解析する固液混相流解析 (以降 IR (interface-resolved) 法と呼ぶ) がある¹⁾²⁾³⁾。IR法は計算負荷が大きくこれまで直接現地を対象とした問題には用いられていない。しかし、これが実現すれば、様々な固液混相流の解析が可能となるため、その活用用途はきわめて広い。

図-1 は IR 法の土石流の予備計算であるが、連続体の土砂モデルでは説明が難しい巨石の浮き上がりが、IR法では特殊なモデル化なしに現れることを確認している。また、物体の密度を小さくすれば浮遊する流木の挙動も解析できる。これにより鋼製フレームの砂防堰堤と流木と巨石を含む土石流の連成解析によって巨石や流木の捕捉効果の検討や、巨石の衝撃力の見積もりも可能になる。また、土丹が露出する河道では、帯工などの横断工作物の袖の土丹が浸食する課題がある。その対策として屈撓性をもつ袋詰玉石などで河岸を保護することは重要であるが、水流を受ける袋詰玉石の挙動の適切な推定技術が無く流出してしまう事例が多い。この課題に対し図-2のように、著者らは、小球の連結によって網をモデル化し、その中に石礫を詰めた袋詰玉石のモデルを作成し、袋詰

玉石と洪水流の IR 解析によって流失しにくい袋詰玉石の配置を検討している。計算負荷の大きな IR 法によりこれらの現地を対象とした解析を実施していくためには、効果的な並列計算技術の開発が必要不可欠である。

著者らの既往の研究³⁾⁴⁾では IR 法で粒子と水の相互作用を説明するためには、粒子の大きさの概ね 1/4 程度以下の計算格子で解析する必要があった。図-3に示すオレンジの枠のブロックはおおよそ数メートルのスケールをイメージして示したが IR法の場合、数十センチの石礫を4分割するとこの大きさのブロック 1 個あるいは数個程度で数十 GB から百 GB 程度のメモリを確保する必要がある。

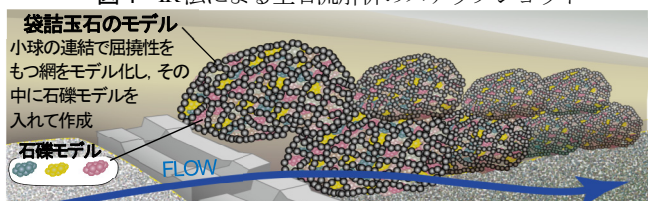
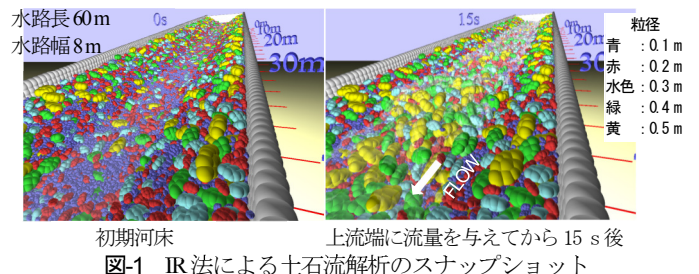


図-2 袋詰玉石と洪水流の IR法の解析例

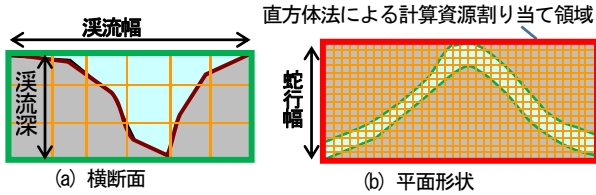


図-3 渓流形状と計算境界の関係

る。このメモリの大きさは多くのスパコンの1 計算機（ノード）の主メモリに匹敵する。したがって、渓流の形状に関係なく直方体領域を対象とせざるを得ない解析法（以降、直方体法と呼ぶ）では無駄な領域に計算機を複数台割り当てる必要が生じ、その計算機を確保するコストが生じる。このため直方体法では、渓流のみを対象とできる計算法（任意形状法と呼ぶ）と比較し計算コストが容易に数倍増加するため、直方体法での現地スケールのIR法の実現は難しいといえる。

また、土石流などを解析する場合、計算前に土石流の及ぶ領域を推定することが難しい。そのため溪岸の上方など土石流が及ぶか不明な領域も計算領域とせざるを得ない。領域を均等分割し、各計算資源に割り当てて並列計算をすると、土石流がほとんど及ばない領域を分担する計算資源は、毎ステップの計算を早々に終了してしまう。そして渓床など負荷の大きい領域を分担する計算資源の毎ステップの計算が終了して、領域境界のデータの受け渡しが可能となるまで、負荷の小さい計算資源は同期待ちとなる。このため計算資源全体を効率的に活用できない問題がある。これを回避するため、粒子数の空間分布に応じて自動的に各計算資源の負荷を分散させる技術が必要となる。しかし、動的負荷分散の研究では、直方体領域の粒子群の解析を対象とした研究例⁹⁾が多く、必ずしも複雑境界形状のもとで効果的な並列計算手法となっていない。このことから、現地スケールのIR法の実現のためには、任意形状法であることがまず重要であり、この枠組みの中で適切な動的負荷分散手法を開発することが重要である。

本研究では複雑境界形状の大規模 IR 解析に適した新たな任意形状法を構築した。この手法では、対象領域を複数の立方体領域ブロックに分割し、プロセスは複数のブロックの計算を分担する。以降では各プロセスが複数の領域ブロックを分担する並列計算法を PB(plural blocks)法とよび、単一ブロックのみを分担する並列計算法を SB(single block)法とよぶ。また、さらに本稿で開発した並列計算法 (DBA (dynamic block assignment)法) は、PB法の枠組みで動的負荷分散を考慮できるものである。本稿では新しく開発した DBA 法の計算法をまず説明する。構築したプログラムは MPI と OpenMP のハイブリッド計算であるが (図-4 参照)、本稿では主に MPI のプロセス並列部分について説明する。つづいて現地渓流を模した領域を対象に SB 法と DBA 法で解析し、両者の比較から DBA 法の有効性を示す。

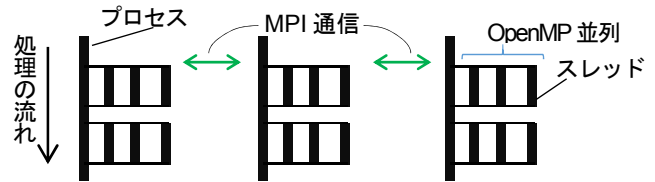


図-4 MPI と OpenMP のハイブリッド計算の並列計算処理の概念図

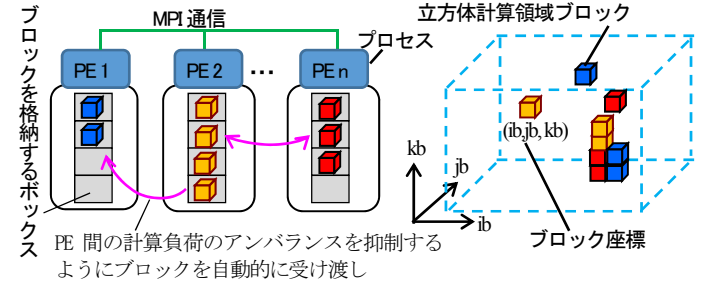


図-5 DBA 法の概念図

2. 動的負荷分散を考慮して各プロセスに複数の計算領域ブロックを割り当てた並列計算法

(1) 並列計算法のフレームワークとそのねらい

本研究で開発した任意形状法の DBA 法の概念図を図-5 に示す。本並列計算法は、図-5 右図のように空間を立方体で分割し、計算対象となる立方体のみをプロセス (Processing Element (PE)) に割り当てる。各プロセスは複数ブロックを計算できるようにしている。そして計算量が小さなブロックのみを受け持つプロセスは計算量の大きいブロックを多く受け持つプロセスとブロックを動的に交換するなどし、プロセス間の負荷のアンバランスを小さくし、計算資源を効果的に使用することで高速に計算することをねらっている。

(2) 主要なデータの取り扱い

ブロックは、ブロック座標 (ib, jb, kb) で管理され、各プロセスは全てのブロックの割り当て先のプロセスを把握している。各プロセスはブロックを格納できるボックスを用意し、ボックスの番号 nb と各プロセスが担当するブロックの ID を関係づけている。具体的に SB 法と PB 法のプログラム上の主要なループの記述について説明する。格子の x,y,z 方向のステンシル番号を i,j,k で表すとする。例えば流速の変数 u の値を求める際には、SB 法では一般に i,j,k それぞれのループを構成しその中で u(i,j,k) の値を求めることになる。これに対し PB 法では i,j,k のループの外にボックスの番号 nb のループを設けかつ u(i,j,k, nb) のように変数の引数に新たに nb を設ければ良い。このように計算の主要部分については、SB 法から PB 法に容易にプログラムを書き換えることができる。

なお、解析領域ブロック数がプロセス数で割り切れる場合、PB 法で最低限必要なメモリは SB 法と同一である。

割り切れない場合は、PB法ではブロックを格納するボックスを余分に1つ持たなければならないプロセスが生じる。この場合でも、SB法に対して最大で2倍以下のメモリでPB法の解析は実施可能である。

(3) 隣接ブロックとのデータの受け渡し

DBA法では図-5に示すように隣接するブロックは、同一プロセス内にある場合と別のプロセスにある場合がある。同一プロセス内のブロックの変数は、他のプロセスの進行に関係なく任意のタイミングで直接高速にアクセスできるメモリに格納される。一方、隣接するブロックが別のプロセスにある場合はMPI通信が必要となる。MPI通信はメモリアクセスと比較し低速であり、また、他のプロセスが送信するまでデータは受け取ることができないので同期が必要となる。このようにMPI通信には時間を要するためできるだけMPI通信を減らすことが高速化のために重要となる。本手法で各プロセスは、ブロック毎また送信の方向毎に分けてMPI通信を行わず、これらを1つにまとめて通信を完了させるアルゴリズムを構築している。以降ではブロックが同一のプロセスにない場合のブロック間のデータの通信方法について説明する。

図-6に示すブロックとプロセスの配置を条件とし各ブロックがブロック座標ibの正の方向と負の方向の2方向へデータを送信する際を例としてMPIの通信手順を説明する。説明のしやすさから2次元配置としているが、3次元配置でも同様の手続きとなる。

複数のブロックのデータを送信する際、ブロックの順序を送信・受信する両プロセスで把握しなければ、誤ってデータが受け渡しされてしまう。これを回避するブロック送受信データの整理方法を説明する。プロセス0からプロセス1へデータを送る例を説明する。まず図-6のブロックIDが小さい順に調べていきibが正の方向にデータを送信するブロックでかつプロセス0が送信側、プロセス1が受信側となる通信について、プロセス0は送信側のブロックを、プロセス1は受信側のブロックを抽出する(図-7参照)。その後同様にibが負の方向についても調べて抽出する。これにより図-7に示すように、送信または受信に関係するプロセスの両方で、個別のブロックおよび個別の方向毎に送信と受信がリンクするようにデータが整理される。このため、複数のブロックかつ複数方向の通信データを一度に送受信しても、データの取り違えを生じることなく正しく送受信ができる。

次にプロセス間のMPI通信について説明する。本手法では、MPIの初期化にmpi_thread_multipleを指定し、複数スレッドを立ち上げて送受信を行い高速化を図っている。これを実現するブロッキング通信手法について説明する。

まず、ブロック座標ibが正の方向と負の方向へ送信する際の関連するプロセス間の送受信を表-1のように整理

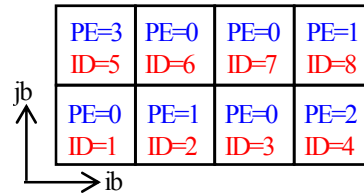


図-6 ブロックとプロセスの配置例

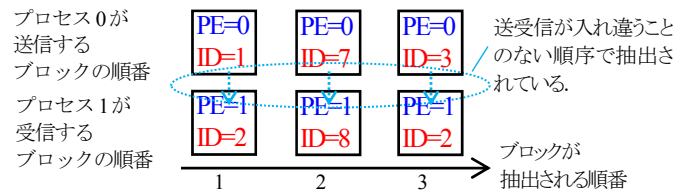


図-7 ibが正および負の方向へデータ送信する際のプロセス0の送信ブロックとプロセス1の受信ブロックの順番

表-1 ibが正および負の方向へデータ送信する際のプロセスレベルの送信受信表

送信側 PE	受信側 PE				
	0	1	2	3	...
0		○	○	○	...
1	○		-	-	...
2	○	-		-	...
3	○	-	-		...

表-2 送信受信順序設定用番号リスト

送信側	受信側				
	0	1	2	3	...
0	1	2	3	4	...
1	5	6	7	8	...
2	9	10	11	12	...
3	13	14	15	16	...

プロセス0: 送信 1(2)→送信 2(3)→送信 3(4)
→受信 1(5)→受信 2(9)→受信 3(13)
プロセス1: 受信 0(2)→送信 0(5)
プロセス2: 受信 0(3)→送信 0(9)
プロセス3: 受信 0(4)→送信 0(13)

※ 送信と受信の隣の数字はそれぞれ受信側、送信側のプロセス番号を示し、()内の数字は送受信の順番を定めるためのソート用として用いた表-2に示す送受信順序設定用番号を示す。

図-8 ibが正および負の方向へのデータ送信に関する各プロセスの送受信順序

する。この場合、例えばプロセス0とプロセス1がともに受信をコールする順序で通信作業を行えばデッドロックを起し、計算が止まってしまう。両プロセスが共に送信をコールしても同様である。そのため、ブロッキング通信で通信を完了させるためには全てのプロセスの送信と受信がリンクするような順番で、自身のプロセスの送信または受信の順番を設定する必要がある。送受信のデッドロックを回避するため表-2に示す送受信順序設定用の番号を用いる。この番号は列方向に優先的に増加する番号である。各プロセスは表-1の○となっている箇所の送受信の作業と表-2の同じ場所の送受信順序設定用の番号を関連させて持ち、この番号が小さい順に並ぶように作業をソートして送信、または受信の順番を定める。こ

これらの結果を図8に示す。プロセス0とプロセス1の両プロセスで、ソート番号(2)の通信であるプロセス0からプロセス1へのデータの送信が先に来ており、また、ソート番号(5)のプロセス1からプロセス0へのデータの送信が後ろに来ており、通信が1つ1つ完了していく送信と受信の順序になっていることがわかる。

各プロセスの送信と受信を複数スレッドを立ち上げて行う際のスレッドの作業順序について説明する。図9には3スレッドで送受信を処理する際の各スレッドの割り当てを示している。スレッド番号が順番になるようにラウンドロビン方式で割り当て、各スレッドでは図中の左から通信作業を行うことで、同一スレッドでは必ず表2の送受信順序算定用の番号の小さい方から作業を行うようになるためデッドロックを回避した送受信が行える。

(4) 動的負荷分散

DBA法の動的負荷分散について説明する。図10にプロセス毎の経過時間の模式図を示す。MPI並列の場合、図10左図に示すようにブロック毎の主な計算のほかに通信全体に要する時間がある。このうちプロセス間の負荷の差が大きい場合、同期に要する時間が大きくなり計算資源のロスが大きくなる。本研究で提案する動的負荷分散は図10右図に示すように、できるだけプロセス間の負荷のアンバランスを小さくするようにブロックとプロセスの割り当てを自動的に再配置するものである。

ブロック配置問題のアルゴリズムの概念図を図11に整理する。構築したアルゴリズムは、もともと各プロセスが持っていたブロックに対し、プロセスがブロックを分担するために用意しているボックスを組み合わせを変えて割り当てる。そして事前に計測していたブロック毎の計算時間を用いて、割り当て後の各プロセスの時間を推算し、その最大時間が最も小さくなる組み合わせを設定する。本研究ではこのブロックの入れ替え戦について全ての組み合わせを厳密には検討せず、乱数を用いていくつかの試行(試行回数: n_i)を行い、その中の組み合わせの最適な組み合わせで入れ替えることとした。

3. 計算性能の確認

本章ではSB法と本研究で開発したDBA法との比較を通してDBA法の計算性能を分析する。

(1) IR法の数値解析法

IR法の数値解析法を説明する。液相の運動解析では、固相を密度の異なる流体として全体を非圧縮性流れとして差分法で解いている。種々の形状の石礫粒子群は小球を連結してモデル化し、これらを剛体として解いている。解析法の詳細は文献³⁴⁾を参照されたい。

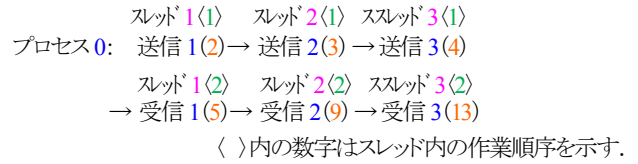


図9 送受信のスレッドへの割り当て順序(プロセス0の例)

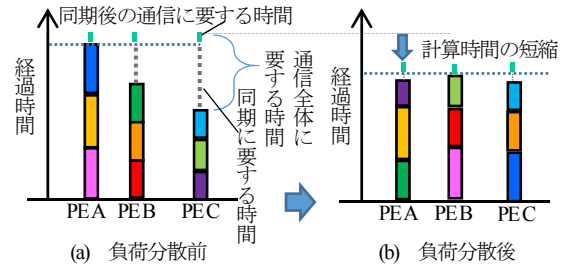
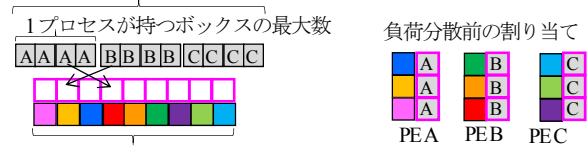


図10 負荷分散前後のプロセス毎の計算時間の例
 四角は各ブロックの作業時間を示し、色の違いはブロックIDの別を示す。

図10 負荷分散前後のプロセス毎の計算時間の例

入れ替えに関連するプロセスのボックスの総数



入れ替えに関連するブロックの総数

図11 ブロック入れ替えアルゴリズムの概念図

(2) 計算条件

図12は土石流の大規模IR解析をDBA法を用いて行う際のイメージを示す。本検討では計算性能の確認段階であるため広域は対象せず、図中に示すように、ある横断に関係するブロック群を想定して計算性能の検討を行った。図13には計算性能を確認するために行った石礫粒子沈降・落下計算の初期状態を示す。5m×5m×5mの立方体領域ブロックを24ブロック使用して計算領域を設定した。ある横断をイメージして溪床溪岸に相当するブロックに粒子が集中するようにし、また、ブロック内ではランダムに粒子の初期位置を設定した。投入した石礫粒子は15万個である。図13からわかるように本手法は、任意の地形に合わせてブロックを配置できる。このため図11の破線で示すように、複雑地形に合わせて計算領域を設定することも可能である。

計算性能の検討で使用した中央大学の計算ノードはintel社製XeonE5-2697v2(12コア)のCPUを2基積んでおりメモリは128GBである。SB法およびDBA法ともに3ノード計72コアを使用した。SB法ではプロセスは1つのブロックしか受け持たないため、ブロック数と同一の24プロセス(プロセスあたり3コア)を使用している。これに対しDBA法では12プロセス(プロセスあたり6コア)を使用し、ブロックを受け持つことができるボックスを各プロセス2つにしたため、プロセスは常に2つのブロックを受け持つ。また、図中のPEはDBA法の初期のプロセス番号を示しており、SB法のPEはID-1である。DBA法の初期条件は、粒子の密なブロックを受け持つプロセスと疎な領域を受け持つプロセスに概ね分け、負荷のアンバランスが大きくなる条件

とした。そして、DBA法が粒子配置に応じて動的にブロックのプロセスへの割り当てを変化させ、プロセス間の負荷を分散できることを確認した。

この落下問題に対し、動的負荷分散がより困難な条件となる粒子移動の活発な初期の0.5s間（計算時間ステップ $dt=10^{-4}s$ を 5,000 ステップ分）の計算を行った。なお、粒子の剛体計算では衝突を安定に解くため、さらに 50 分割した時間ステップを用いている。DBA 法では 50 ステップ毎にブロックとプロセス割り当ての入れ替えアルゴリズムで時間短縮検討している。このアルゴリズムの中に用いた入れ替え時の試行回数 n_t は 5,000 回とし、また、当該時刻でプロセスが受け持つブロックの合計の計算時間を調べ、計算時間が大きい順の上位 2 プロセスと下位 2 プロセスの計 4 プロセスを入れ替え対象とした。ブロックの入れ替え作業は、現状ではハードディスクにブロックのデータを書き出し、プロセスへの割り当てを変えて読み込むことで行っており、この入れ替え作業には時間を要する。このため、時間短縮率が小さな入れ替えを抑制するため、毎ステップ計算時間の各プロセスの最大値について 5% の時間短縮が見込まれる際にだけ入れ替えを実施することとした。なお、 n_t の設定などいくつかの入れ替えに関係するパラメータは一次設定として与えたものであり、必ずしも最適化されたものではない。種々の問題に対するこれらの値の感度分析などは今後の課題である。

(3) DBA 法の性能評価

図-14 は 10 ステップ毎に、毎時間ステップの平均計算時間と累加時間を示したものである。DBA 法では、150 ステップまでの間に 3 回 5% の時間短縮が見込まれたためブロックの入れ替えが行われ、この入れ替えに時間を要しているが、それ以降は 5% の時間短縮が見込まれず入れ替えは行われていない。図-14 より、初期では SB 法の方が毎ステップあたりの計算時間が短いですが、100 ステップ以降は DBA 法の方が計算時間が短くなっている。これにより検討の最終段階の 5,000 ステップの累加時間では SB 法では 43,304 s に対し、DBA 法では 33,452 s であり、同一の計算機を用いているにも関わらず DBA 法は SB 法に対し 23% もの計算時間の短縮を実現している。

この高速化の要因を分析する。図-15 は、SB 法および DBA 法のプロセス毎の計算時間とブロック毎の計測時間を図示している。ブロック毎の時間は、ブロック毎の主要ループの時間を計測し、サブルーチン全体の時間に合うように比率を乗じて求めた。まず、SB 法と初期の DBA 法について両者のプロセス 0 を見ると、DBA 法は粒子の多いブロックを 2 つ持っているため計算時間を要しているが、1 ブロックでは DBA 法の方が高速に計算が行われている。これは SB 法ではプロセスあたり 3 コアで計算するのにに対し DBA 法ではその倍の 6 コアである

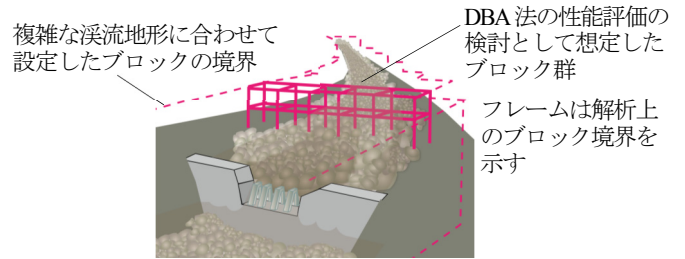


図-12 複雑境界形状となる現地溪流で DBA 法を用いて土石流の大規模 IR 解析を行う際のイメージ

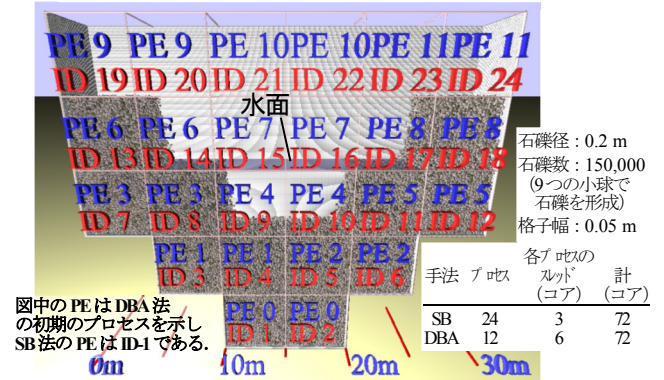


図-13 粒子沈降・落下計算の初期条件および計算に用いた諸量

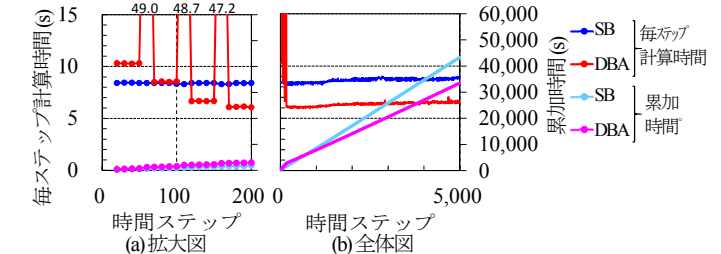


図-14 時間ステップ毎の計算時間の変化

ためである。したがって DBA 法で、2 つあるブロックのどちらかを負荷の小さいブロックと入れ替えることができれば、SB 法よりもプロセスの最大時間を小さくすることができる可能性がある。プロセスに複数のブロックを分担させる PB 法はここにメリットがあり、また、この後の負荷分散まで動的に行うものが DBA 法である。

図-15 の負荷分散後の DBA 法では、初期と比較し負荷が分散されており、かつ最も時間を要しているプロセスは SB 法よりも計算時間が短くなっている。これにより、当初のねらい通りの高速化が実現したといえる。また、図-16 には、DBA 法の最終段階のブロックとプロセスの割り当てを示す。ほとんどのプロセスが粒子の多いブロックと少ないブロックの両方を分担するようになっており、結果として粒子数の空間的な偏りが適切に考慮されて、負荷分散がなされたことがわかる。これらの結果より、DBA 法は複雑な境界条件下でも効果的にプロセス間の負荷を分散でき、SB 法より高速に計算を行うことができると考えられる。

(4) 現地スケールの IR 解析の計算規模の概算

砂防堰堤に作用する土石流の衝撃力を DBA 法を用いた IR 法で推定する場合の計算規模を概算する。土石流の現象が概ね 60 s 程度とし、堰堤上流 200 m の区間（本

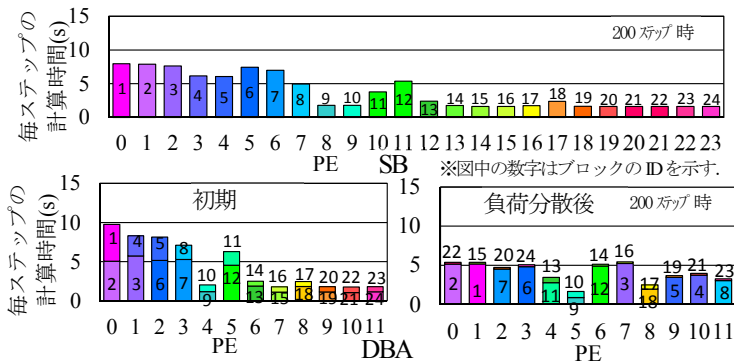


図-15 プロセスとブロック毎の計算時間

検討の 40 倍(200/5) を解析しようとする概ね 13 万ノード時間 (=60×33,452/0.5×40×3ノード/3,600) の計算資源が必要となる。九州大学スパコンの利用費から概算すると 100 万円程度の利用費となる。実験は大がかりな施設が必要となることなども考慮すると、精度の高い IR 解析は有効な技術であり、今後 DBA 法で現地土石流の IR 解析などが実現可能であることを示していくことが重要である。また、試算したコストから判断すると、解析は必ずしも容易には実施できるものではないことも事実である。このことから、DBA 法のコスト低減の意味は非常に大きく、現地 IR 解析の実現において DBA 法はきわめて重要な技術であるといえる。

4. 結論

本研究では、動的負荷分散を考慮して各プロセスに複数の計算領域ブロックを割り当てる複雑境界形状の大規模固液混相解析に適した DBA 並列計算法を開発した。解析対象は、溪流横断形状を模した空間的偏りのある粒子配置を初期条件とした、粒子の 0.5 s 間の沈降落下である。SB 法と DBA 法で解析を行い、DBA 法ではプロセス間の負荷のアンバランスが大きな状態から解析を実行した。解析の結果、粒子の移動が活発な中、0.5 s 間と短い時間で DBA 法は動的なブロックのプロセスへの割り当てにより急速に負荷をバランスさせ、SB 法に対して 23%も計算時間を短縮しており、効果的な並列計算法であることを

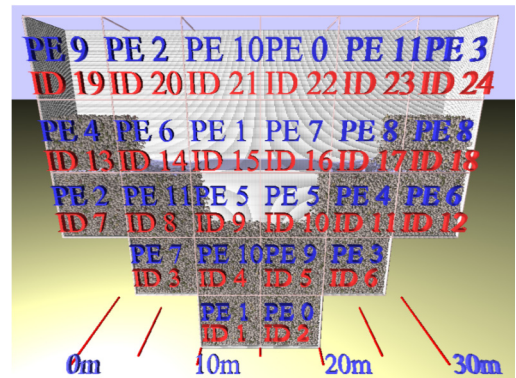


図-16 負荷分散後のプロセスとブロックの割り当て関係を確認した。今後はより大規模な問題を対象としDBA法の性能を確認する必要がある。

謝辞：本手法の開発など研究の一部は九州大学情報基盤研究開発センターのスーパーコンピュータを利用した。

参考文献

- 1) 梶島岳夫, 滝口智志, 浜崎洋至, 三宅裕：渦放出を伴う粒子を含む鉛直平行平板間の乱流構造, 日本機械学会論文(B編), Vol.66, 647, pp.120-127, 2000.
- 2) 牛島省, 山田修三, 藤岡奨, 瀬津家久：3次元自由水面流れによる物体輸送の数値解法(3D MICS)の提案と適用性の検討, 土木学会論文集 B, Vol.62, No.1. pp.100-110, 2006.
- 3) Fukuoka, S., Fukuda, T.: Effects of sizes and shapes of gravel particles on sediment transports and bed variations in a numerical movable-bed channel, *Adv. Water Resour.*, Vol.72, pp.84-96, 2014.
- 4) Fukuda, T., Fukuoka, S.: Interface-resolved large eddy simulations of hyperconcentrated flows using spheres and gravel particles, *Adv. Water Resour.* <http://dx.doi.org/10.1016/j.advwatres.2017.10.037>
- 5) 丸山紀尚, 牛島省:動的負荷分散を考慮した流体流の任意形状物体運動群の並列計算法, 土木学会論文集 B1(水工学), Vol.70, No.4, pp.I_835-I_840, 2014.
- 6) 青木尊之：動的負荷分散による GPU スパコンを用いた粒子法の大規模シミュレーション手法の開発, 学際大規模情報基盤共同利用・共同研究拠点 平成 26 年度共同研究最終報告書, 2015.

(2018.5.31 受付)

PARALLELIZATION TECHNIQUE USING DYNAMIC LOAD BALANCE FOR LARGE-SCALE SOLID-LIQUID MULTIPHASE SIMULATIONS IN COMPLEX BOUNDARY GEOMETRIES

Tomoo FUKUDA and Shoji FUKUOKA

This paper presents a new parallelization technique (DBA method) that is effective for large scale solid-liquid multiphase simulations in complex boundary geometries. This technique assigns plural blocks to every processing element and dynamically changing the assignment to balance computation load of processing elements. Simulations were performed on falling particles in the water and air by using the DBA method and the SB method that assigns a single block to every processing element. Computation efficiency of DBA method was examined from these simulations. The results showed that the SB method statically assigned computation resources to blocks of small computation load that made great load unbalance, while the DBA method reduced the load unbalance by dynamical changing block assignments to the processing elements and became faster than the SB method.