

複雑境界形状の固液混相流解析のための

各プロセスに複数の計算領域ブロックを割り当てる動的負荷分散手法

Dynamic load balancing technique assigning plural computational domain blocks to every process for simulations of solid-liquid flows in complicated boundary geometry

○ 福田朝生, 中央大, 東京都文京区春日 1-13-27, E-mail: t-fukuda@tamacc.chuo-u.ac.jp
福岡捷二, 中央大, 同上, E-mail: sfuku@tamacc.chuo-u.ac.jp

This paper presents a new parallelization technique DBA-BC(dynamic block assignment method taking into account balancing block computation load and shortening communication times) 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 block computation load among processing elements and to shorten communication times by reducing the number of communication pattern. Simulations were performed on moving particles in the water by using the DBA-BC method and the DBA-B method that does not take into account shortening communication times. The results showed that DBA-BC method was able to shorten 6 % of computation times in every time step in comparison with that of DBA-B method.

1. 序論

水深と同スケールの巨石を含む土石流の運動は、工学的に連続体として扱い、現象を説明できるものもあるが、複雑な力学現象の厳密な取り扱いを欠いている。これらの水理現象の推定には、個々の粒子の運動を直接 Lagrange 的に解析し、粒子周りの流れ場を粒子よりも小さな計算格子を用いて Euler 的に解析する固液混相流解析法^{(1)(2) (3)(4)} (以降 IR (interface-resolved) 法と呼ぶ) が今後の有効な解析手法となると思われる。しかし IR 法についても課題がある。すなわち、これらの手法は計算負荷が大きくなり、適切な並列計算法を開発しなければ、現状では現地溪流を対象とした大スケールの土石流解析は困難となっている。

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 以下の計算格子で解析する必要があった。IR 法の場合、数十センチの石礫を 4 分割すると数 m 立法の計算領域ブロック 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。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、土石流の流路となる溪流の形状とは関係なく、直方体領域を計算対象とせざるを得ない解析法 (以降、直方体法と呼ぶ) では、流路とは関係のない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無駄な計算コストが生じる。このため直方体法では、土石流の流路となる部分だけを計算対象とすることができる計算法 (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。このため、直方体法は現地スケールの IR 法の解析に適用することが難しいといえる。

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。そして溪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、この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⁽⁵⁾⁽⁶⁾

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。

著者らの既往の研究においては、MPI と OpenMP のハイブリッド計算の枠組みで複雑境界形状の IR 解析に適した任意形状法を構築した⁽⁷⁾。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、各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、著者らが構築した手法 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。この手法では、ブロックの計算時間を各プロセスで均等化するようにブロックをプロセスに自動的に割り当てて、MPI 通信時の同期待ちに要する時間を削減することで高速計算を狙ったものである。しかし、この手法では実際の MPI の通信量の削減は十分考慮できていなかった。すなわち通信量の削減には、各プロセスがデータ通信を行わなければならないプロセス数 (本稿では通信パターン数と呼ぶ) をできるだけ少なくする必要がある。DBA 法では、これらを考慮せずにブロックのプロセスへの割り当てを決定していた。このため、ブロック数とプロセス数が比較的小さく、相対的に通信に要する時間が少ない場合には、ブロックの計算負荷をプロセス間で均等化することだけで計算時間を短縮できた。しかし、ブロック数とプロセス数が多くなると、通信パターン数が増加し通信時間が増加するため、全体の計算時間を短縮することが困難となっていた。これらの課題を解決するため、本研究では、ブロック数とプロセス数が多い場合でも適切に計算時間を短縮できるように、通信時間の短縮も考慮してブロックをプロセスに割り当てて計算時間を短縮する新しい並列計算法 DBA-BC (dynamic block assignment method taking into account balancing block computation load and shortening communication times)を構築した。また、著者らの⁽⁷⁾が既往の研究で構築した DBA 法は、「プロセス間のブロックの負荷分散 (balancing block computation load)」は考慮している。このため以降では既往の研究⁽⁷⁾で構築した DBA 法を DBA-B (dynamic block assignment method taking into account balancing block computation load) 法と呼ぶことにする。

本稿では最初に新しく構築した DBA-BC 法の計算法を説明す

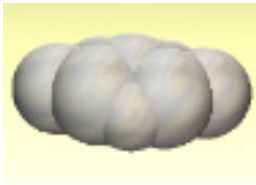


Fig. 1 Gravel particle model.

る。そして、DBA-B 法と DBA-BC 法を用いて、土石流等を想定した水中の斜面上を粒子が移動する場を対象に解析し、両者の計算速度の比較から DBA-BC 法の計算性能の優位性を示す。

2. IR 法の計算法

本研究で用いる IR 法の数値解析法を説明する。液相の運動解析では、牛島ら²⁾の手法を参考に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解いている。乱流モデルとしてスマゴリンスキーモデルを用いている。以下に液相の解析の基礎式を示す。

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (1)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = g_i - \frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \{2(\nu + \nu_t) S_{ij}\} \quad (2)$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3)$$

$$\nu = \mu / \rho \quad (4)$$

$$\nu_t = (C_s \Delta)^2 \sqrt{2 S_{ij} S_{ij}} \quad (5)$$

ここに u_i は固相部分も考慮した計算格子内の質量平均の i 方向速度。 ρ および μ は計算格子内の体積平均密度および粘性係数である。 g_i は重力加速度、 P は圧力と SGS 応力の等方成分の和、 C_s はスマゴリンスキー定数、 Δ は計算格子サイズである。これらを直交格子差分法で離散化し、SMAC 法を用いて時間積分を行っている。なお、自由水面の変化は VOF 法⁸⁾で解析している。

石礫粒子は Fig. 1 に示すように小球を連結してモデル化しており、これらの粒子群の運動は剛体として解析している。粒子に作用する流体力は、液相の運動解析で解かれた混相流体の石礫部分に作用する力を体積積分することで求めている。また、接触力は通常の個別要素法と同様にバネ、ダッシュポット、摩擦スライダを用いて解析している。解析法の詳細は文献⁴⁾を参照されたい。

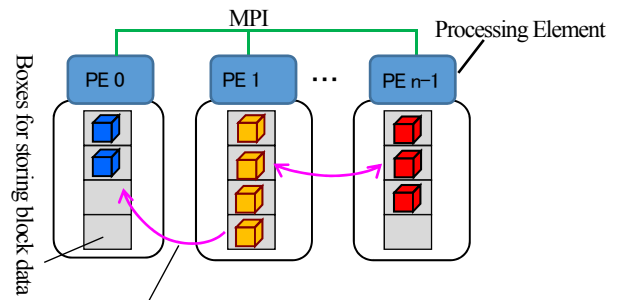
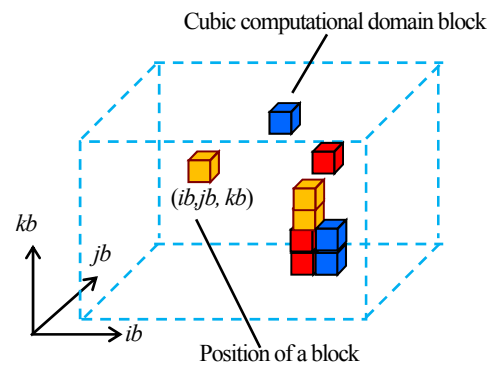
3. DBA-BC 法の構築

(1) 並列計算法のフレームワークとそのねらい

任意形状法である DBA 法 の概念図を Fig. 2 に示す。本並列計算法は、Fig. 2 上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おり、PB 法の枠組みの並列計算法である。そして DBA-B 法ではプロセス間のブロックの計算負荷バランスが均等化するように、また、DBA-BC 法はさらに通信時間を削減するように、ブロックのプロセスへの割り当てを動的に変更する。これにより計算時間全体の短縮を図るものである。

(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



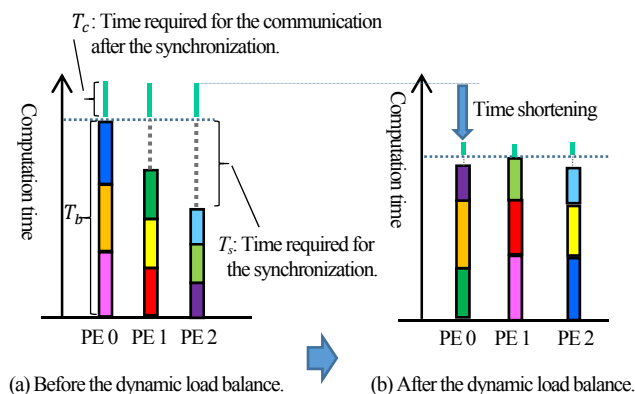
Dynamic block assignment to balance computation load among each processing element.

Fig. 2 Schematic diagram of DBA.

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。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ぬプロセスが生じる。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) DBA-B 法の計算時間短縮方法とその課題

まず、著者らが、既往の研究⁷⁾で構築した DBA-B 法の動的負荷分散について説明する。Fig. 3 には、本研究でプロセス毎の計算時間を推定するために用いた計算時間の模式図を示す。実際に計算時間は、複雑な種々の要素で決まるが、本研究では簡易的にこのようなモデルを用いる。このモデルでは全体の計算時間は、各プロセスのブロックの計算時間の最大時間 T_b と通信時間 T_c で構成される。このうちプロセス間のブロックの計算負荷の差が大きき場合、Fig. 3 の左図に示すように同期に要する時間 T_s が大きくなり同期待ちとなるプロセスの計算資源を有効に活用できなくなる。著者らが既往の研究⁷⁾で開発した DBA-B 法では、これらの同期に要する時間の削減ため、プロセス毎のブロックの計算時間をできるだけ均等化するようにブロックをプロセスに動的に割り当てるものであった。一方 Fig. 4(a)のようにプロセスはできるだ



Rectangles indicate computational times.
Different colors indicate different blocks.

Fig. 3 Schematic diagram of computation time before and after dynamic load balance

け空間的に隣接するブロックを受け持つようにすれば通信パターンは少なくて済むが、Fig. 4(b)のようにプロセスが空間的に隣接しないブロックを複数分担するようになると通信パターンが増加し通信時間 T_c も増加する。DBA-B法では、この増加する通信時間の削減は考慮できていなかった。このため、特にプロセス数やブロック数が多くなる場合は、通信パターン数が増加し動的なブロックのプロセスへの割り当てにより計算時間を短縮することができない課題があった。これに対し本研究では、通信時間 T_c の短縮も考慮した新しい並列計算法DBA-BC法を構築した。

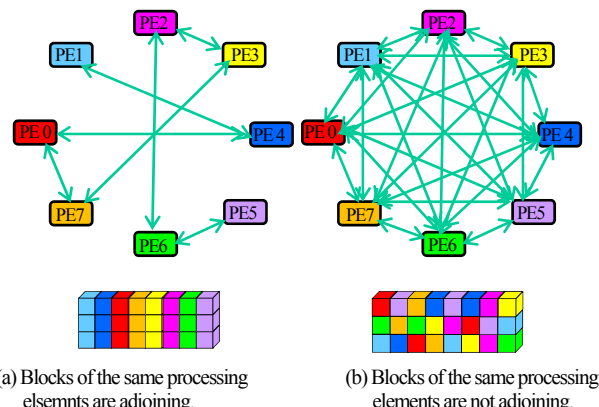
(4) DBA-BC法における通信時間の推定方法

DBA-BC法では、MPI通信時間の削減を自動的に行えるようにすることを指すが、このためにはMPI通信時間の計測が必要となる。MPI通信は、プロセス毎に異なったタイミングで実施されるため全体の計算時間の遅延要因となる通信時間を厳密に特定することは難しい。そこで本研究では、この通信時間を簡易的に計測できる時間で代表する。本研究の計算法では、プロセスはブロック毎に毎回通信することはない。プロセスは自身の受け持つブロックの通信データを一度取りまとめ、これらをまとめて通信する手順を踏んでいる。このまとめたデータを通信するプログラムの部分について、この通信が一番遅く開始したプロセスの時刻 $T_{c,start}$ を計測し、またこの通信が一番遅く終わったプロセスの時刻 $T_{c,end}$ も計測する。そしてこれらの差を通信時間 $T_c (= T_{c,end} - T_{c,start})$ として代表させた。

DBA-BC法では、ブロックの入れ替え後の通信時間 T_c を推定し、この推定値に基づいてブロックの入れ替えを行う。この通信時間は以下の簡易的な直線式で推定する。

$$T_c = a_0 + a_1 C_p \quad (6)$$

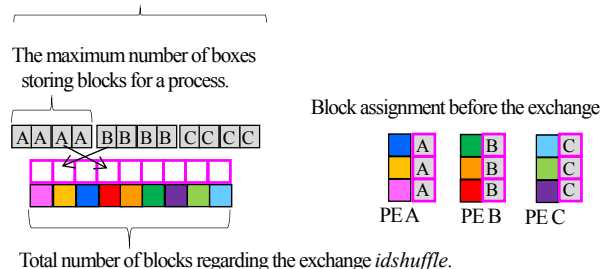
ここで C_p は、通信パターン数であり、各プロセスの通信相手となるプロセス数の総和である。Fig. 4の図で説明すると、一つの矢印は、矢印の両端のプロセスでそれぞれ通信パターン数としてカウントする。このため全通信パターン数は矢印の数の2倍である。また、 a_0 、 a_1 は係数である。DBA法では、ブロックのプロセスへの割り当て変更毎に一組の T_c と C_p の値を算定し、これらを標本として蓄積していき最小二乗法により係数 a_0 、 a_1 を逐次求める。



Different color of blocks indicate different processing elements.

Fig. 4 Communication patterns depending on the assignments of blocks to processes.

Total number of boxes of the processes regarding the exchange: $nboxshuffle$.



Total number of blocks regarding the exchange $idshuffle$.

Fig. 5 Given condition for the block exchange.

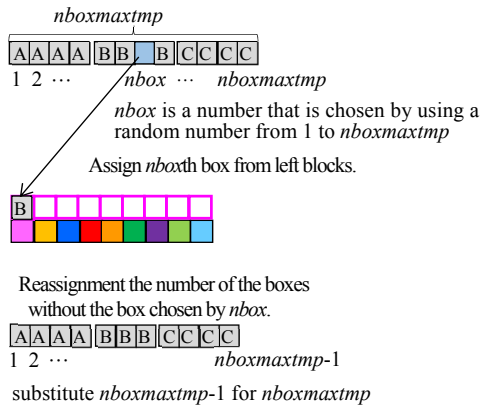
(5) プロセス間のブロックの入れ替えに関するアルゴリズム

ブロック配置問題のアルゴリズムの概念図をFig. 5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ブロックの計算時間を推算し、その最大値 T_b を求める。また、割り当て後のブロック配置を基に式(6)に用いる通信パターン数 C_p を求めてMPI通信時間の T_c を推算する。そして T_b+T_c が最も小さくなる組み合わせを選定する。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_t)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。

具体的なアルゴリズムをFig. 6に示す。図のように試行回数 n_t のループの内側に、入れ替えに関するブロックの総数($idshuffle$)のループを設ける。ここで乱数を用いてプロセスの持つボックスをブロックに割り当てていく。試行回数ごとに1つのボックスとブロックの組み合わせが決定される。決定された組み合わせに基づいてブロック計算時間が最大となるプロセスのブロックの計算値 T_b とMPI通信に要する時間 T_c を推算し、 T_b+T_c を求める。1つ前の試行回数まで求められた T_b+T_c の推定値より小さければ時間短縮が可能な組み合わせとしてブロックのプロセスへの割り当てを記憶する。これを繰り返すことでより時間短縮が可能な組み合わせを探索し、時間短縮が可能であれば最終的に得られた組み合わせで実際にブロックとプロセスの組み合わせを変更する。

Before the block exchange algorithm, the block computation times of each process were measured. Then, the maximum block computation time among all processing elements T_b was determined. The communication time T_c was also measured. The criteria of the time $t_{shuffle}$ for the block exchange was substituted by $T_c + T_b$.

do Loop of the number of the trial: n_t
 $n_{boxmaxtmp} = n_{boxshuffle}$
do Loop of the total number of blocks regarding the exchange: $id_{shuffle}$



enddo The end of $id_{shuffle}$ loop

An example of a block assignment which is given by the above $id_{shuffle}$ loop.



1. Computing $T_c + T_b$ based on this block assignment
2. If computed $T_c + T_b$ is less than $t_{shuffle}$, $t_{shuffle}$ is substituted by $T_c + T_b$, and the given block assignment is memorized as the provisional assignment that is able to shorten the computation time.

enddo The end of n_t loop

Fig. 6 Algorithm that assign blocks to processes to shorten computation time.

4. 計算性能の確認

(1) 計算条件

本章ではDBA-B法とDBA-BC法の比較によりDBA-BC法の計算性能を分析する。計算対象とした問題は、Fig. 7に示すように水中で土砂が、斜面に沿って移動する問題である。2m×2m×2mの立方体の計算領域ブロックを192個使用して計算領域を設定した。斜面を構成する底部の3つのブロックにはFig. 1に示す粒子径0.2mの粒子をランダムに投入した。なお、粒子径は、同一体積の球の直径とした。投入した石礫粒子は30,000個である。Fig. 7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計算ノードはintel社製Xeon E5-2697 v2 (12コア)のCPUを2基積んでおりメモリは128GBである。DBA-B法およびDBA-BC法ともに4ノード計96コアを使用した。16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12個用意したため、プロセスは常に12(=192/16)個のブロックを受け持つ。

Fig. 8には、DBA-B法とDBA-BC法の初期のブロックのプロ

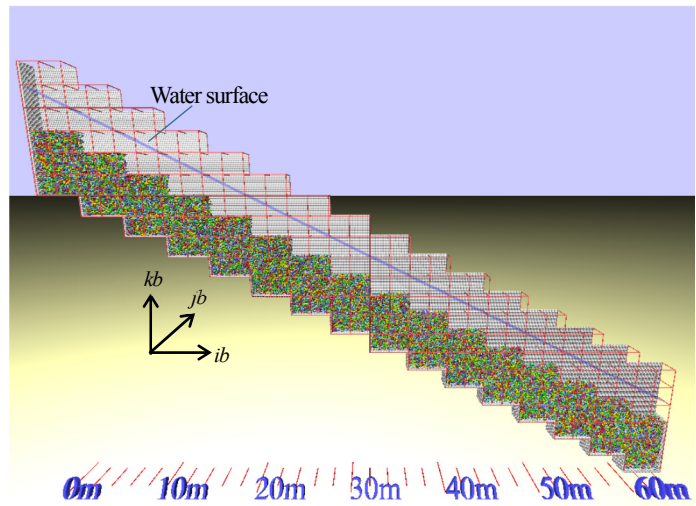


Fig. 7 Initial condition of the gravel bed and water surface.

Table 1 Simulation parameters.

ρ^w : Density of water	1,000	kg/m ³
ρ^p : Density of particles	2,650	kg/m ³
μ^w : Fluid viscosity	8.9×10^{-4}	Pa·s
C_s : Smagorinsky constant	0.173	
D : Particle size	200	mm
Δt^f : Time step for the fluid simulation	1.0×10^{-5}	s
Δt^p : Time step for the particle simulation	4.0×10^{-7}	s
The number of gravels		30,000
The number of small spheres composing gravels		270,000

セスへの割り当てを示している。図中の番号は、プロセスの番号である。初期では、同一プロセスのブロックが隣接するケースが少なくなるように配置し、通信パターン C_p が多くなる条件としている。これにより、DBA-BC法が自動的に通信パターン数を減らし通信時間を短縮できるか確認する。本解析は16プロセスで実施したため、通信パターン数 C_p の最大値は240(=15×16)である。これに対し初期配置の通信パターンは238であり通信パターン数はほぼ最大となっている。

また、初期配置ではプロセス間のブロックの計算時間のアンバランスが生じるように13, 14, 15番の3つのプロセスは、初期では粒子が多い下層3段目までのブロックを受け持たないようにした。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、プロセス間のブロックの計算負荷を分散できるか確認した。本研究では計算性能の確認が主目的であるため大規模な長期計算は行わず、初期の1,000ステップ(0.1s間)だけの計算を行った。計算に用いた各種パラメータをTable 1に示す。

時間短縮を図るブロックのプロセスへの割り当ての入れ替えアルゴリズムは10ステップ毎に実施した。このアルゴリズムの中で入れ替え対象として選ぶプロセスは、DBA-B法の場合は計算時間が大きい順の上位1プロセスと下位1プロセスの計2プロセスとした。また、DBA-BC法では、同様に10ステップ毎に入れ替えは検討するが、入れ替え対象となるプロセスの抽出方法を2パターンとし、10ステップ毎にこれらを交互に変更している。1つ目の

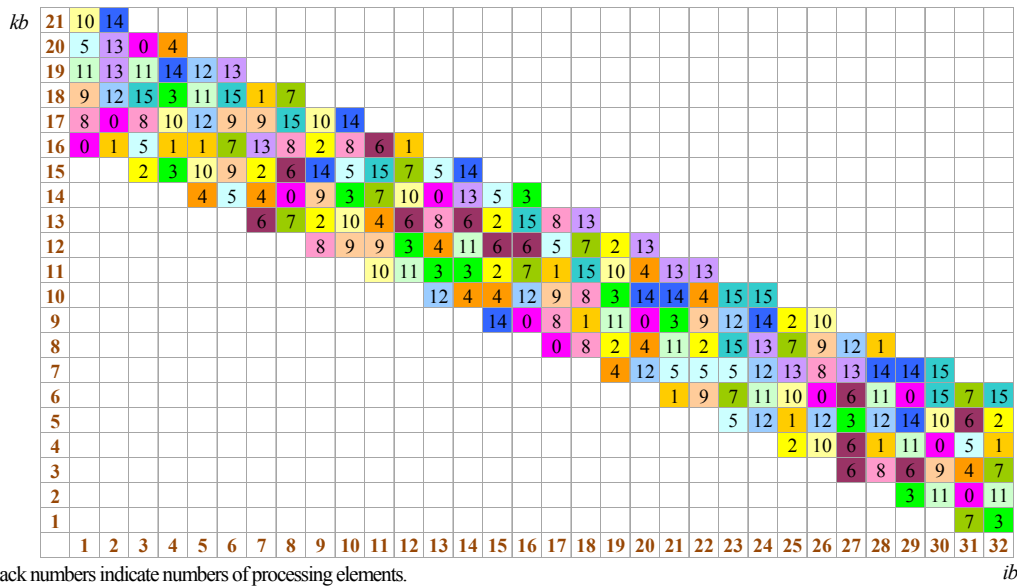


Fig. 8 Initial assignments of blocks to processing elements.

パターンは、DBA-B 法と同様である。2 つ目のパターンはランダムに 2 プロセスを抽出している。通信パターンの削減は必ずしもブロックの計算負荷の大きなプロセスだけに着目すべきではなく、全体として通信パターン数を削減することが重要と考えられる。このため、DBA-BC 法では、入れ替え対象プロセスの選定方法として 2 パターン目を追加した。

ブロックの入れ替えに関するアルゴリズム (Fig.6 参照) における入れ替え時の試行回数 n_i は 1,000,000 回とした。 n_i の設定などいくつかの入れ替えに関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。

(2) DBA-BC 法の性能評価

Fig.9 に DBA-B 法および DBA-BC 法それぞれの計算時間ステップ毎の 1 ステップの計算時間を示す。本手法では、ブロックの入れ替えはブロックデータを一度ハードディスクに書き出し、割り当てプロセスを変更して読み込むことを行っている。このため、ブロックの入れ替えには時間を要している。図中の毎時間ステップの時間が 3.5s 以上となっている時間ステップは、ブロックの入れ替えを含むステップである。

本来計算性能の評価は、ブロックの入れ替え時間を含む全体の計算時間で行うべきである。しかし、実際の大規模な土石流の解析は、最低でも数週間程度の解析が必要と推定しており、大規模な計算を行う前の動的負荷分散法構築の準備的研究である本研究ではこれらの長時間の計算を行うことは難しい。さらに土石流の解析では、移動する土砂に対して無視しえない静止した広範囲の地形を形成する土砂を解析しなければならない。このような条件では、静止している現地溪流地形に応じて、ブロックのプロセスへの割り当てを最適化することが、土石流発生後の計算時間の高速化にも大きく影響を及ぼし続け重要になると考えられる。このことから、1 日程度を要してもまず地形に応じたブロックのプロセスへの割り当ての最適化によって毎時間ステップの計算時間を高速化することが、全体の計算時間の短縮を考える上で重要と考えられる。そのため、本研究では全体の計算時間ではなく、毎時間ステップの計算時間をどの程度短縮できるかに着目して DBA-BC 法の計算性能の考察を行う。なお、1,000 ステップの累加の計算時間は、DBA-B 法と DBA-BC 法でそれぞれ、21,288s, 17,631s

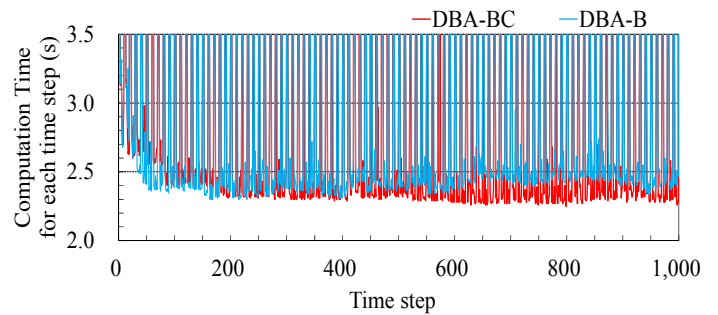


Fig. 9 Temporal variation in computation time for each time step.

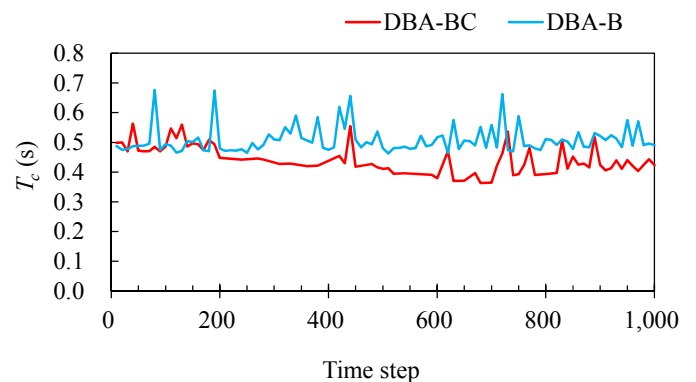
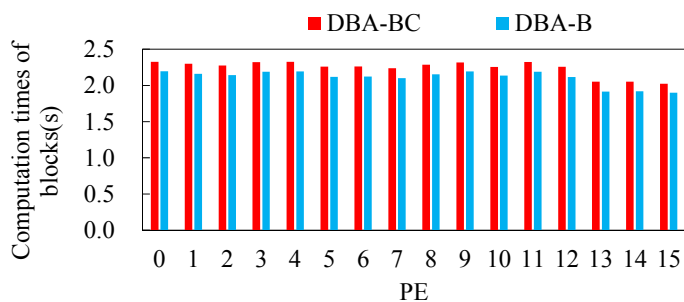


Fig. 10 Temporal variation of communication times.

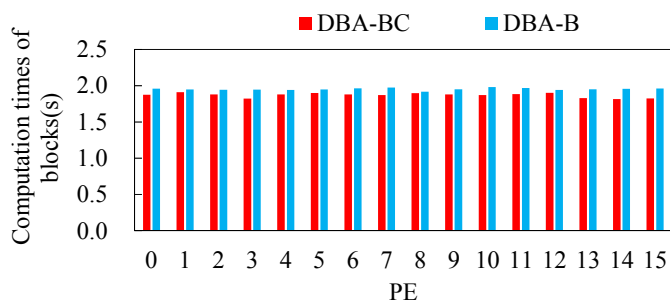
であった。

Fig.9 より、毎時間ステップの計算時間は DBA-B 法および DBA-BC 法の両方で、100 ステップ程度まで大きく減少している。また、その後は、計算時間は、緩やかに遷移していくが、計算の最終段階では、DBA-B 法と比較して DBA-BC 法の方が毎時間ステップの計算時間を短縮できている。両者の差が付き始めた 600 ステップ以降のブロックの入れ替えに関与しないステップの、毎時間ステップの平均時間は、DBA-B 法では 2.48s に対して、DBA-BC 法は、2.34s であり DBA-BC 法は、DBA-B に対し 6%ほど毎時間ステップの計算時間を短縮できている。

これらの計算時間の変化傾向の要因を詳細に分析する。 Fig. 10



(b) Initial stage.



(b) After 100 time steps.

Fig. 11 Block computation time of every processing element.

には、DBA-B 法および DBA-BC 法の各時間ステップで計測された通信時間 T_c の時間変化を示す。この図より、200 ステップ以降では、DBA-BC 法の方が通信時間を短縮できている。また、その短縮時間の大きさは、Fig. 9 に示す毎時間ステップの DBA-B 法と DBA-BC 法の差にほぼ近い。このことから、200 ステップ以降で DBA-BC 法の方が DBA-B 法と比較して計算時間を短縮できた要因は、今回新たに構築した通信時間短縮のアルゴリズムの効果といえる。

また、Fig. 10 より、200 ステップまでは、両者の通信時間の差はない。しかし Fig. 9 では、初期から 200 ステップまでの間に計算時間は大きく減少している。したがって、この部分の計算時間の短縮は通信時間の短縮が要因ではない。初期からの計算時間の大きな低下は、DBA-B 法と DBA-BC 法の両方で考慮しているブロックの計算負荷のプロセス間の分散である。Fig. 11 には、各プロセスのブロックの計算時間を示す。ブロックの入れ替え前の初期段階と、ブロック入れ替え後の 100 ステップ後の結果を示した。初期では、13 番から 15 番目のプロセスは土砂が多数ある下層の 3 つのブロックを受け持たないようにした。このため、これらのプロセスのブロックの計算時間は、他のプロセスよりも小さくなっている。一方、ブロックを数回入れ替えした 100 ステップ後では、プロセス間のブロック計算に伴う負荷はほぼバランスし、かつ、最大のブロックの計算時間 T_b は、初期と比較して 100 ステップ後の方が低下している。したがって、Fig. 9 の 100 ステップまでの計算時間の短縮は、この各プロセスのブロックの負荷分散の結果と考えられる。これより、DBA-B 法と DBA-BC 法は、各プロセスのブロック負荷分散を適切に行うことができ、またこれによる計算時間短縮の効果は大きいといえる。

続いて、前述のように Fig. 9 では 200 ステップ以降で、DBA-B 法と DBA-BC 法の計算時間に差が生じている。そしてこの要因は DBA-BC 法で新たに導入した通信時間の短縮アルゴリズムの効果と考えられ、このアルゴリズムの動作状況の詳細を確認する。通信時間短縮を実現する上で考慮した通信時間の推定式(6)は、解析中に求めた複数の通信時間 T_c と通信パターン C_p の組のデータを

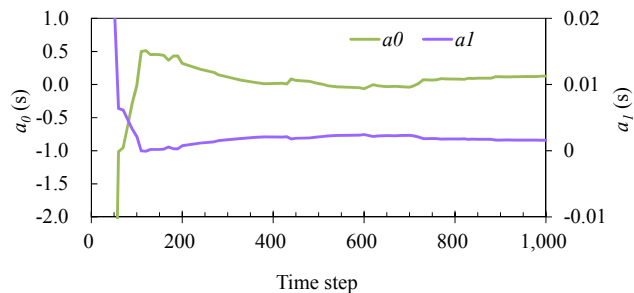


Fig. 12 Temporal variation of coefficient used in prediction of communication times (Eq. 6)

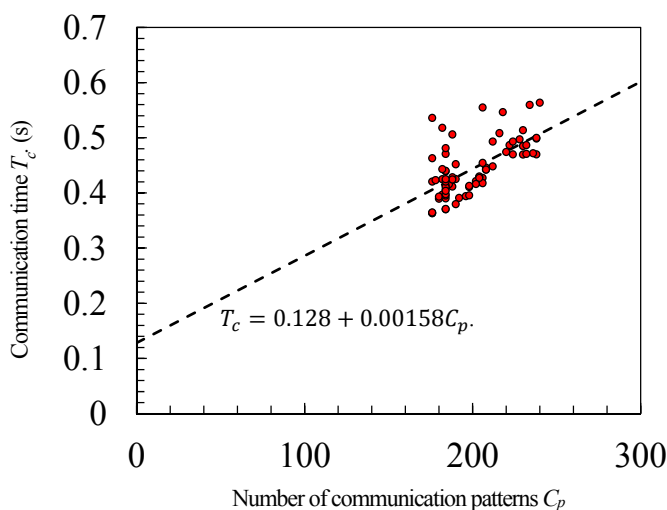


Fig. 13 Relationship between the number of communication patterns and communication times and their estimated approximation at the last stage.

を用いて最小二乗法により係数を求めている。Fig. 12 には、これらの係数が時間的にどのように推定されたかを示す。この図より、式(6)の係数は、200 ステップ程度までは、かなり変動しているが、それ以降はほぼ安定した値をとっていることがわかる。また、通信パターン C_p に係る係数 a_1 は、値が落ち着いてからは正の値をとっている。すなわち、通信パターン数が少なくなるほど時間短縮ができることを推定できている。

Fig. 13 には、解析の最終段階までで蓄積された通信パターン C_p と通信時間 T_c の関係をプロットで図示し、また、最終段階の推定式を破線で示している。この図より、推定式より上方に離散的にプロットがいくつか見られるものの、大部分のプロットは C_p が小さいほど T_c も小さくなる傾向となっており、推定式はこの傾向をよく捉えているといえる。

Fig. 14 には、これらの推定式を用いてブロック入れ替えが行われた後の解析最終段階 (1000 ステップ時) の DBA-BC 法のブロックのプロセスへの割り当てを示している。この図より、Fig. 8 に示す初期の割り当てと比較して最終段階では、同一プロセスのブロックが、より隣り合うように配置されていることがわかる。

Fig. 15 には、DBA-B 法および DBA-BC 法の両者の通信パターン数の時間変化を示す。DBA-B 法は、通信パターン数の削減を考慮できないため、ほぼ 16 プロセス使用時の最大の通信パターン数である 240 付近となっている。一方、DBA-BC 法は、通信パターン数の削減を考慮しているため、最終的に 180 程度まで減少しており、DBA-B 法と比較すると 60 程度も通信パターン数を削減で

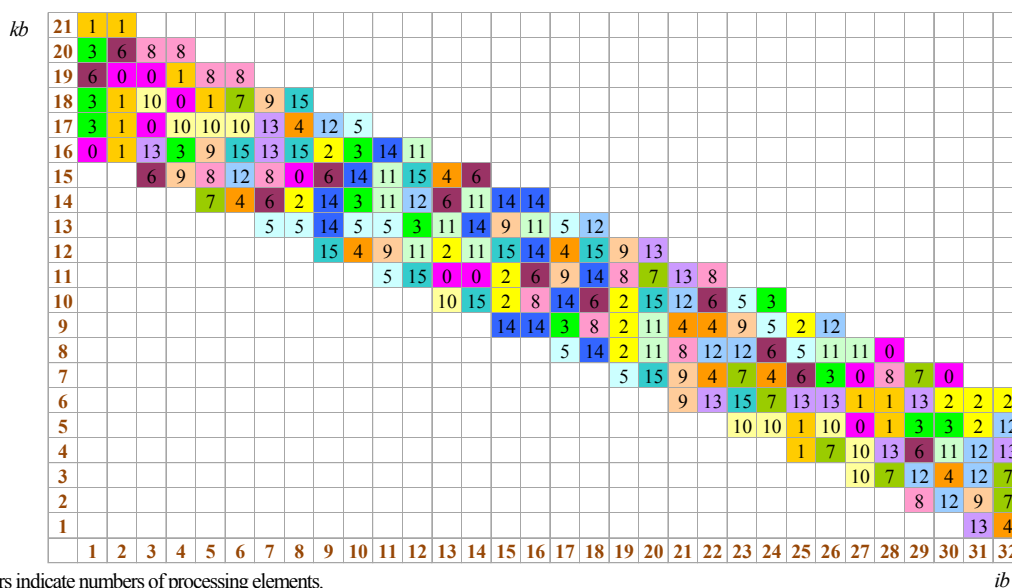


Fig. 14 The last assignments of blocks to processing elements in the DBA-BC simulation.

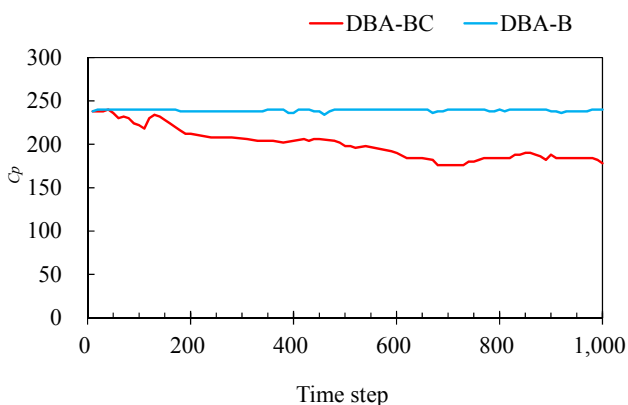


Fig. 15 Temporal variation of the number of communication patterns.

きています。これらの分析より DBA-BC 法は、同一プロセスのブロックが隣接するように配置されることにより、通信パターン数を減らし、結果として通信時間が削減し、DBA-B 法と比較して、毎ステップの計算時間が短縮できたといえる。

一方で本解析では 16 プロセスを用いているが、端から鉛直方向に順に同一プロセスのブロックを埋めた場合、通信パターン数は最小値の 30 となる。この場合、通信パターン数も削減でき、かつ粒子が多いブロックを各プロセスが同様に分担することになるため、プロセス間のブロックの計算負荷の差も小さくすると想定される。これに対し DBA-BC 法は最終段階でも通信パターン数は 180 程度となっており、最小値との差は大きい。DBA-BC 法は通信パターン数の削減について一定の効果が確認されたが、この通信パターン数削減のアルゴリズムについては改善の余地は大いに残されているといえる。

5. 結論

本研究では、複雑境界条件の大規模固液混相流解析に適した並列計算法 DBA-BC 法を構築した。この手法は、各プロセスのブロックの計算負荷がバランスするようにブロックをプロセスに割り当て、さらに同一プロセスのブロックが隣接するようにブロックをプロセスに割り当て通信パターン数を削減し、通信時間の削減を図るものである。この DBA-BC 法と、通信パターン数の削減を

考慮することができない DBA-B 法を用いて、水中の斜面上の土砂の運動を解析し両者の計算時間を計測した。その結果、DBA-BC 法は、並列計算法の構築のねらい通り、同一プロセスのブロックは DBA-B 法の結果と比較して隣接する割合が高くなり、これにより計算の最終段階 (1,000 ステップ時) の毎ステップの計算時間は、DBA-BC 法は DBA-B 法と比較して、本検討の場合約 6% の時間短縮を実現した。今後は、現地スケールの IR 解析を行い、より大規模な条件で DBA-BC 法の計算性能を確認する必要がある。

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,(2000), pp.120-127.
- (2) 牛島省, 山田修三, 藤岡奨, 福津家久, "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, 1 (2006), pp.100-110.
- (3) Fukuoka, S. and 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., 72 (2014), pp.84-96.
- (4) Fukuda, T. and 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, 4, (2014), pp.I_835-I_840.
- (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).
- (7) 福田朝生, 福岡捷二, "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", 土木学会論文集 B1(水工学), Vol.75, 5 (2018),掲載決定
- (8) Hirt, C.W. and Nichols, B.D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys., 39 (1981), pp.201-225.