動的負荷分散を考慮した 流体中の任意形状物体運動群の並列計算法

DYNAMIC LOAD-BALANCING PARALLEL COMPUTATION METHOD FOR
MULTIPLE ARBITRARILY-SHAPED OBJECTS IN FLUIDS

丸山 紀尚¹・牛島 省²
Kisho MARUYAMA and Satoru USHIJIMA

¹ 学生員 京都大学大学院工学研究科 社会基盤工学専攻 (〒 615-8530 京都市西京区京都大学桂)

² 正会員 工博 京都大学教授 学術情報メディアセンター (〒 606-8501 京都市左京区吉田本町)

It is important to predict the movements of multiple solid objects driven by free surface flows in case that the flood and Tsunami occur. In this study, essential improvement was performed for our multiphase parallel computational method, in which multiple complicated-shaped solid objects are transported by the pressure and viscous effects of the predicted multiphase model. Our parallel computational method is based on the domain decomposition method. Thus, the improvement was achieved by dynamically adjusting only the “block-computational domain” according to the number of blocks included in them. On the other hand, the “fluid-computation domain” is fixed throughout the computation. As a result, we found 64-process simulation of water column collapse, including 12,800 objects in the water, exert 24.52 speedup compared to sequential execution.

KeyWords : *dynamic load balancing, parallel computation, solid object transportation, multiphase model*

1. はじめに

市街地を対象とした 3 次元大規模津波シミュレーションが近年盛んに^{1),2),3)}に行われている。例えば Pham Van Phuc²⁾ は 3 次元 VOF 法を用いて 6km×6km の計算領域の一部に仮想の市街地を設定し実現象 7 分間の津波遡上解析を行った。津波の 3 次元的な流れが解析でき、複雑な構造物に働く流体力の算定が可能である。しかし、市街地における氾濫流は多数の漂流物を伴うため、人や建物に衝突した時の被害の予測を行うには、流れを受ける複数の複雑形状物体の運動を解析する必要がある。その解法の 1 つとして多相場モデルによる MICS⁴⁾ があげられ、実験結果との比較^{4),5)} からその妥当性が示されている。

周囲流体の計算に加えて、複数の四面体から構成される物体群の衝突を伴う計算は非常に負荷が高い。そのため流体及び物体運動計算について、MPI を用いた領域分割法による解法の並列化^{6),7)}が行われ、計算領域内に物体が一樣に存在する場合について処理の高速化が確認されている。しかしながら、実際の漂流物は水面・水底など計算領域内の一部に集まることが多い。このような場合、領域分割法では各計算プロセスごと

に所有する物体数が異なり、負荷のインバランスから並列化効率が低下する⁷⁾。

メモリ分散型の並列システムにおいて、流れ場や電磁場から力を受ける粒子群の計算法としては、ブロック・サイクリック分割法や動的領域分割法、OhHelp⁸⁾、ハイブリッド分割法⁹⁾などがあげられる。ブロック・サイクリック分割法で領域を細かく分割すれば粒子数を均一にすることができるが、隣接プロセスとの通信量が増えてしまう。動的領域分割法で粒子数が均等になるように動的な領域分割を行った場合、結果として粒子数は均等に分割されるが、流体計算で負荷バランスが生じてしまう。OhHelp とハイブリッド分割法はともに流体計算と粒子計算の両方の負荷をバランスさせる方法であるが、主にプラズマシミュレーションで用いられる方法であるため、本研究のように粒子相互に直接的な相互作用を持つ場合に適用が難しい。

本研究では流体領域と物体領域を別々に取扱い、物体領域に関してのみ動的領域分割を行うことで、物体同士的位置関係が把握し易いように保ちながら、流体と物体の計算負荷をバランスさせる手法を提案した。静的な領域分割法との比較から本手法の検証を行った。

2. 数値解析手法

(1) 多相場モデルの基礎式と気液混相流の計算

本研究では一流体モデルと同様の考え方により，固気液相から構成される場を近似的に扱う多相場モデル⁴⁾を用いる．モデルの概要を図-1に示す．

気相と液相のみから構成される以下の基礎式を，有限体積法に基づきコロケート格子上で離散化し，MAC系の解法により気液相に対する密度，流速および圧力の計算を行う．計算法は既報⁴⁾と同様である．

$$\frac{\partial \rho_f}{\partial t} + \frac{\partial}{\partial x_j}(\rho_f u_j) = 0 \quad (1)$$

$$\frac{\partial u_j}{\partial x_j} = 0 \quad (2)$$

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = f_i - \frac{1}{\rho_f} \frac{\partial p}{\partial x_i} \\ + \frac{1}{\rho_f} \frac{\partial}{\partial x_j} \left[\frac{\partial}{\partial x_j}(\mu_f u_i) + \frac{\partial}{\partial x_i}(\mu_f u_j) \right] \end{aligned} \quad (3)$$

ここで t は時間， x_i は3次元直交座標系の座標成分， f_i は外力加速度の x_i 成分である．また，速度ベクトルの x_i 成分 u_i ，密度 ρ_f ，圧力 p ，粘性係数 μ_f は気液相で平均された値である．

図-1 計算手順

(2) 任意形状物体の表現方法

物体は任意形状の剛体として扱うためT型固体モデル¹⁰⁾で表現する．すなわち，通常の計算では四面体要素の集まりとして計算し，衝突の計算では物体の表面に配置された衝突判定球について個別要素法 (DEM) を用いて計算する．衝突判定時などの探索格子として，領域全体に物体スケール幅の物体セルを配置する．ここで物体スケール l_{blk} は，物体の重心点と物体 k を構成する四面体頂点 P_k の距離が d_{P_k} である時， $l_{blk} \equiv \max_{k \in B} \{ \max_{P_k} \{ 2d_{P_k} \} \}$ とする．物体と流体との相互作用を適切に扱うために，物体スケールに比べて流体セル幅を細かくとる必要がある．一方，物体を構成する四面体要素に対しては，流体セルから生ずる制約条件は無く，物体形状を適切に表現できる四面体要素分割を行えばよい．本研究では形状を正確に表現するために四面体要素数を多く設定して計算する．図-2の概略図のように基本的には以下のデータ量の大小関係が成り立っている．

$$(\text{物体構成四面体数}) > (\text{物体を包む流体セル数}) \quad (4)$$

$$(\text{全流体セル数}) > (\text{全物体セル数}) > (\text{物体数}) \quad (5)$$

(3) 物体運動計算法

物体運動計算法は既報⁴⁾と同様であるが，ここでは2章(5)節の説明のため，物体と流体の相互作用に関わる部分である図-1のA，B，C，Dについて解説する．

a) 固気液混相流の密度・粘性係数の算出 (計算A)

セル内の密度 ρ と粘性係数 μ を次式より求める．

$$\psi = \left(1 - \sum_{k \in C} \alpha_k \right) \psi_f + \sum_{k \in C} \alpha_k \psi_k \quad (6)$$

ここで ψ は体積平均されたセル内の物性値， ψ_f は流体の物性値， ψ_k は物体 k の物性値， α_k はセルC内における物体 k の体積割合である．物性値は物体ごとに異なってもよい．

図-2 物体を構成する四面体，流体セル，物体セル

b) 流体力の評価 (計算 B)

物体に作用する流体力 $\mathbf{F}_k$ は、セル C 内における物体に作用する流体力 $\mathbf{F}_{Ck}$ の総和をとることで得られる。 $\mathbf{F}_{Ck}$ は既報⁴⁾と同様に浮力項、圧力項、拡散項を用いて評価する。物体に作用する流体力モーメント $\mathbf{N}_k$ も同様に、計算セル C における流体力モーメント $\mathbf{N}_{Ck}$ の総和をとることで得られる。次式を用いて流体力の評価を行う。

$$F_{Cki} = \alpha_k \sigma_k V_C \left[-\frac{\rho - \rho_f}{\rho} g \delta_{3,i} - \frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left\{ \frac{\partial}{\partial x_j} (\mu u_i) + \frac{\partial}{\partial x_i} (\mu u_j) \right\} \right] \quad (7)$$

$$\mathbf{F}_k = \sum \mathbf{F}_{Ck} \quad (8)$$

$$\mathbf{N}_{Ck} = \mathbf{r}_{Ck} \times \mathbf{F}_{Ck} \quad (9)$$

$$\mathbf{N}_k = \sum \mathbf{N}_{Ck} \quad (10)$$

ここで σ_k は物体 k の密度、 V_C は流体セルの体積、 g は重力加速度、 ρ 、 μ は多相場の密度、粘性係数、 $\mathbf{r}_{Ck}$ は物体重心点からセル中心点に向かうベクトルである。

c) 流体セルの物体体積分率の算出 (計算 C)

式 (6) や式 (7) の α_k を求めるため、サブセル法¹⁰⁾を用いて流体セルに含まれる物体の体積分率を求める。

d) 流速の修正 (計算 D)

物体の運動量を流れ場に反映するために、気液計算で得られた流速 $\mathbf{u}$ と物体部分の速度 $\mathbf{v}_{Ck}$ の質量平均である次式から流速の修正を行う。

$$\mathbf{u}_C = \frac{1}{M_C} (M_f \mathbf{u} + \sum_k \sigma_k \alpha_k V_C \mathbf{v}_{Ck}) \quad (11)$$

ここで $\mathbf{u}_C$ は多相場の流速ベクトル、 M_C 、 M_f はセルの質量、気体および液体部分の質量、 $\mathbf{v}_{Ck}$ は流体セル C の位置における物体 k の速度ベクトルであり、次式の通り重心速度 $\mathbf{v}_k$ と角速度 $\boldsymbol{\omega}_k$ から求められる。

$$\mathbf{v}_{Ck} = \mathbf{v}_k + \boldsymbol{\omega}_k \times \mathbf{r}_{Ck} \quad (12)$$

(4) 並列計算手法

flat-MPI を用いた領域分割法による並列計算を行う。本手法では、流体運動と物体運動の双方の負荷をバランスさせるため、各プロセスが計算する流体領域と物体運動領域を別々に取り扱う。それぞれ流体サブドメインと物体サブドメインと呼ぶ。この手法では両サブドメインとも 1 つのプロセス内で分割されるようなことが無いため、物体同士の位置関係を把握しやすい。流体サブドメインは 3 次元の計算領域を各方向に等分割して得られる。流体セルは均一セルなので各プロセスが持つ流体セル数は等しくなる。物体サブドメインは、計算領域全体に配置された物体セルに物体の計算処理のコストをマッピングし、その空間情報をもとに各プ

ロセスの物体運動計算の負荷がバランスするように領域を分割する。物体計算処理のコストは、物体の形状の種類や衝突の有無などで決めるが、本計算では単純に、物体重心が存在する物体セルに 1.0 を加算し、各物体サブドメインのコスト (物体数) が均等になるように、領域分割を行っている。また、物体領域の分割は物体セル幅の単位で行う。図-3 のように分割された各ドメインを複数のプロセスにより演算処理を同時に進めることで、計算速度を向上させる。

物体は計算中に偏在する可能性があるため、計算中に物体処理コストの偏りを判定し、偏りがあると判定した時は物体領域の再分割を行う。物体処理コストの偏り判定方法は以下の式である。

$$C_{max} > \frac{C_{sum}}{N} (1 + \gamma) \quad (13)$$

ここで C_{max} は各物体サブドメイン内のコストの総和の最大値、 C_{sum} は全体の計算領域内のコストの総和、 N はプロセス数、 γ は非負のパラメータであり、本計算では $\gamma = 0.1$ とした。再分割後、 C_{max} が増加してしまう場合や分割前と同じ分割領域になる場合は再分割を中止する。

剛体シミュレーションでよく行われるように、物体の種類ごとに設定される物体の形状・材質データと、重心点位置や重心点速度、物体の種類番号などの物体の状態データを分離して扱う。形状・材質データは全ての計算プロセスが所有し、状態データは領域分割法によるデータ分散が行われている。

(5) 並列計算における通信処理

領域分割法には隣接プロセスと袖領域中の情報を交換する必要がある。本研究では、袖領域の大きさは、流体サブドメイン、物体サブドメインどちらも 1 回の繰返し計算に必要なだけの流体セル数、物体セル数として、毎回通信を行う。

また本手法では、自身の流体サブドメインと物体サブドメインの不一致から、物体と流体の相互作用処理 (図-1 中の A, B, C, D) に関する通信が必要になる。

図-3 流体サブドメインと物体サブドメインの概略

その際には通信を効率的に行うために、式 (4) 式 (5) の関係に基づき、通信負荷を減らすこと重要である。流体情報を持つプロセス (流体側プロセス) と物体情報を持つプロセス (物体側プロセス) のどちら側で計算するか、どの変数を通信するかについて検討を行う。

a) 計算 A について

式 (6) の計算を物体側プロセスが行う場合、不一致領域において物体を含む流体セル分の ψ_f の受信と計算後の ψ の送信が必要である。一方、流体側プロセスが行う場合は、物体を含む流体セル分の α_k と物体個数分の ϕ_k を求めるための物体種類番号の受信が必要である。本手法では流体側プロセスが計算を行う。

b) 計算 B について

物体側プロセスの計算では、式 (7) から ρ_f , p , ρ , μ の受信が必要である。一方、流体側プロセスが計算する場合は、 α_k , σ_k を求めるための物体種類番号、式 (9) の $\mathbf{r}_{Ck}$ の計算に必要な物体の重心点座標の受信と、 $\mathbf{F}_k$, $\mathbf{N}_k$ の送信が必要である。物体種類番号、重心点座標、 $\mathbf{F}_k$, $\mathbf{N}_k$ は不一致領域に存在する物体個数分のデータでよい。本手法では流体側プロセスが計算を行う。

c) 計算 C について

流体側プロセスで流体セルの物体体積分率 α_k の算出を行うには、物体節点座標が必要である。一方、物体側プロセスで計算するには流体セル座標が必要である。流体セルは均一格子であるため物体側プロセスは流体セル座標を通信しなくとも把握していることや、式 (4) に示すデータ量の観点から、物体節点座標の通信よりも計算後の α_k を送信する方がよい。本手法では物体側プロセスが計算する。

d) 計算 D について

式 (11) 式 (12) の通り、物体側プロセスでの計算には M_C , M_f , $\mathbf{u}$, V_C の受信と $\mathbf{u}_C$ の送信が必要である。流体側プロセスの計算では $\mathbf{v}_k$, $\boldsymbol{\omega}_k$, 重心点座標, α_k , 物体種類番号の受信が必要になる。本手法では流体側プロセスが計算する。

以上まとめると、図-4 のように物体プロセス側は不一致領域に存在する物体を含む流体セル分の物体体積分率と物体識別番号、物体個数分の物体重心座標、重心速度および重心角速度の送信が必要で、計算 C 後に行う。流体プロセス側は物体個数分の流体力と流体力モーメントの送信が必要で、計算 B 後に行う。

3. 計算手法の適用性

本手法の適用性の検証を行う。実際の漂流物シミュレーションで見られるように、密度差により水面や水底に移動する場合や、流れにより領域の一部に集中する場

図-4 物体サブドメインと流体サブドメインの間で行う不一致領域の通信

合などを想定して 1000 個の物体の水中落下計算と、水柱崩壊を受ける 12800 個の物体輸送計算を行う。固気液多相場の計算法は逐次計算下において実験結果との比較^{4),5)} から妥当性が示されているため、ここでは物体が偏在する場合の計算時間について検討する。流体サブドメインと物体サブドメインを常に均等な間隔に区切る方法を静的領域分割法、本手法の動的負荷分散法を動的領域分割法と呼び、両者の物体計算時間 T_{blk} で比較を行った。ここで物体計算時間とは固相を含む際に必要となる計算部分 (図-1 中の緑色の計算範囲) を測定したもので、動的領域分割法では流体サブドメインと物体サブドメインの不一致領域に関する通信と通信処理時間 T_{com} も含める。再分割は 50step に 1 回の再分割判定を満たす時に行われ、再分割の判定と再分割に要した時間 T_{rez} も測定する。

計算には京都大学の Cray XE6 を用いた。Cray XE6 は各ノードに 2 個の AMD 社製 16 コア Opteron プロセッサを搭載しており、Gemini インターコネクトでつながれている。

(1) 1000 個の物体の水中落下

1000 個の物体の水中落下計算を行った。計算領域は $1.0\text{m} \times 1.0\text{m} \times 2.0\text{m}$ であり、物体数は 1000、流体セル数は $50 \times 50 \times 100 = 250000$ 、周囲流体には水の物性値を与えた。物体は回転楕円体形状で、長径は 0.1m 、短径は 0.0415m 、物体密度を $2.65 \times 10^3 \text{ kg/m}^3$ 、1 つの物体を表現する四面体要素数は 121、衝突判定球数は 84 とした。時間刻みは $\Delta t = 2.0 \times 10^{-3} \text{ s}$ で、1600step の $t = 3.2\text{s}$ まで計算を行った。1 回の流体計算に、物体の衝突判定・衝突力計算と移動計算を 10 回行う。全体領域を $1 \times 1 \times 2$ に分割し、2 プロセスで計算した。ノード間通信を行わないため 2 プロセス間の通信コストは小さい。

図-5 に、初期状態から $t=3.2\text{s}$ までの計算結果を示す。青の枠線は全体の計算領域を表し、白線は動的領域分割法における物体サブドメインの境界を表す。

図-6 は各計算 step に要した物体運動の計算時間 T_{blk} である。400step までは静的領域分割法の計算時間が減少し、その後増加した。一方、動的領域分割法では約 1000step までは横ばいに進み、その後増加した。400step 付近は静的領域分割法の方が高速になり、計算の後半では静的領域分割法と動的領域分割法の間隔が狭まった。

図-7 は各計算 step でプロセスが保有する物体数の最大値 N_{pmax} を表す。図中の黄色の点は再分割が行われた箇所であり、再分割後に物体数の偏りが解消されているのが分かる。動的領域分割法では、1000step までは概ね物体数が均等である。1000step 以降に物体数が偏ってしまった原因は、図-8 から分かるように、1050step で下プロセスの物体サブドメインの高さが最小高さである 1 物体セル幅になってしまったためである。しかしながら、動的領域分割法では最初の 100step 間や 800step 付近で、静的領域分割法の 1.7 倍以上高速化され、本手法が非常に効果的に機能していると考えられる。

表-2 に示すように、実時間 $t=3.2s$ の計算にかかった処理時間は、静的領域分割法で流体計算に約 1.342 時間、物体運動計算に約 0.898 時間、動的領域分割法では流体計算に約 1.339 時間、物体運動計算に約 0.677 時間、物体運動計算時間のうち、不一致領域の通信と通信処理に要した時間は 68.2 秒であった。物体サブドメインの再分割回数は 14 回であったが、その処理に要した時間は 11.9 秒であった。1000step 間に要した物体計算時間について逐次計算との比較からスピードアップを求めると、静的領域分割法では 1.12、動的領域分割法では 1.56 となり、1.39 倍ほど並列化効率が向上した。

(2) 水柱崩壊を受ける 12800 個の物体輸送計算

次に計算プロセスを増やし、水柱崩壊を受ける 12800 個の物体輸送計算を行った。計算領域は、 $4.0m \times 4.0m \times 1.6m$ であり、物体数は 12800、流体セル数は $200 \times 200 \times 80 = 3200000$ 、周囲流体には水と空気 の物性値を与えた。物体は先ほどと同じ形状、同じ大きさであり、物体密度は水と同じ密度とした。時間刻み $\Delta t = 3.0 \times 10^{-3}s$ で、2000step の $t = 6.0s$ まで計算を行った。1 回の流体計算に、物体の衝突判定・衝突力計算と移動計算を 20 回行う。全体領域を $4 \times 4 \times 4$ に分割し、64 プロセス (2 ノード) で計算した。

図-9 に初期状態から $t=6.0s$ までの計算結果を示す。

表-1 実時間 3.2s の計算にかかった物体運動計算時間

領域分割法	T_{fld} [s]	T_{blk} [s]	T_{com} [s]	T_{rez} [s]
静的	4831.4	3232.3	—	—
動的	4821.4	2438.3	68.2	11.9

図-5 水中落下の計算結果

図-6 各計算 step ごとにの物体運動計算時間
図-7 各プロセスが保有する物体数の最大値

図-10 は各計算 step の T_{blk} 、図-11 は各計算 step の N_{pmax} を表す。概ね、物体数と物体運動の計算処理時間は比例している。動的負荷分散法では常に T_{blk} 、 N_{pmax} が下回っているが、領域内で 3 次元的な偏りが生じているため、本計算のように物体サブドメインを直方体領域に分割する場合、物体数の偏り解消の効果が小さかったと考えられる。

表-2 に示すように、実時間 $t=3.2s$ の計算にかかった処理時間は、静的領域分割法で流体計算に約 1.799 時間、物体運動計算に約 0.619 時間、動的領域分割法では流体計算に約 1.784 時間、物体運動計算に約 0.541 時間、物体運動計算時間のうち、不一致領域の通信と通信処理に要した時間は 254.1 秒であった。物体サブドメインの再分割回数は 33 回であったが、その処理に要した

図-8 1050step と 1600step の計算結果

図-9 水柱崩壊の計算結果

時間は 19.5 秒であった。2000step 間に要した物体計算時間について逐次計算との比較からスピードアップを求めると、静的領域分割法では 21.45，動的領域分割法では 24.52 となり，1.14 倍ほど並列化効率が向上した。

4. おわりに

本研究では，物体が偏在する場合の動的負荷分散法として物体サブドメインのサイズを動的に調整する方法について検討した。物体群の水中落下と水柱崩壊の数値解析を行った結果，物体運動計算の高速化や並列化効率の向上が確認された。今後は実際の市街地において漂流物を含む津波シミュレーションを行いたい。また，3 次元的な偏りを解消する方法として，物体サブドメインを複雑な形状にすることや，共有メモリを利用したハイブリッド並列にすることを検討している。

図-10 各計算 step ごと の 図-11 各プロセスが保有する物体運動計算時間
物体運動計算時間 物体数の最大値

表-2 実時間 6.0s の計算にかかった物体運動計算時間

領域分割法	T_{fld} [s]	T_{blk} [s]	T_{com} [s]	T_{rez} [s]
静的	6474.9	2227.6	—	—
動的	6422.2	1948.5	254.1	19.5

参考文献

- 1) 野中哲也, 本橋英樹, 吉野廣一, 原田隆典, 川崎浩司, 馬越一綱也, 管付紘一: 京コンピュータによる橋梁を含む広域 3 次元津波シミュレーション, 性能に基づく橋梁等の耐震設計に関するシンポジウム講演論文集, Vol. 16, , 2013.
- 2) Punc, P. V., 長谷部雅伸: 三次元 VOF 法を用いた大規模津波シミュレーション, 清水建設研究報告, Vol. 90, , 2013.
- 3) 藤本啓介, 浅井光輝, 一色正晴: 測量情報を基にした地形モデルを用いた ISPH 法による津波遡上解析, 計算工学講演会論文集, Vol. 18, , 2013.
- 4) 牛島省, 福谷彰, 牧野統師: 3 次元自由水面流中の接触を伴う任意形状運動に対する数値解法, 土木学会論文集, Vol. 64/II, pp. 128-138, 2008.
- 5) 牛島省, 山田修三, 瀬津家久: 3 次元 MICS による波動流れにおける物体輸送の並列数値計算, 海岸工学論文集, Vol. 52, , 2005.
- 6) Ushijima, S., Kishimoto, K., Yamashita, H., Nagai, K. and Fujioka, S.: Direct Numerical Simulation for Free-Surface Flows through Porous Media with Multiphase Modeling, *International Symposium on Multiphase Process Engineering (MMPE)*, 2011.
- 7) 山下遼, 藤岡奨, 牛島省: 自由水面流中の接触を伴う複雑形状物体運動の並列計算法, 応用力学論文集, Vol. 9, , 2012.
- 8) H.Nakashima, , Y.Miyake, , H.Usui, and Y.Omura, : OhHelp: A Scalable Domain-Decomposing Dynamic Load Balancing for Particle-in-Cell Simulations, *In Proc. 23rd Intl. Conf. Supercomputing*, 2009.
- 9) 上岡功治, 村田健史, 上田裕子, 岡田雅樹, 大村善治, 松本紘: プラズマ粒子シミュレーションコードの並列化モデルと速度向上率の評価, 情報処理学研究報告, 2001.
- 10) 牛島省, 福谷彰, 牧野統師, 瀬津家久: 3 次元流体中を運動する接触と変形を考慮した任意形状固体モデルの数値解法, 応用力学論文集, Vol. 10, pp. 139-146, 2007.

(2013.9.30 受付)