課題番号 jh180034-NAH

MPF 法によるトポロジー最適化を用いた負荷バランスと ノード間通信低減を両立させる動的領域分割の開発

青木 尊之（東京工業大学）

概要　計算領域を局所的に細分化する AMR 法の大規模解析を高効率で実行するためには、負荷バランスを保ちつつ通信コストを最適化する領域分割が必要である。多結晶の組織形成過程のシミュレーションに用いられる MPF（Multi-Phase Field）法を領域分割に適用する。各結晶粒は界面エネルギーの現象を駆動力として成長し、粒界面の表面積を最小化するため、各結晶粒の体積（計算負荷）を均一化する項を導入することで、計算コストと通信コストを同時に最適化する領域分割を実現する。MPF 法による領域分割を高速化するために、必要な領域にのみフェーズフィールド変数を割り当てて計算を行う APT（Active Parameter Tracking）法と結晶粒界面近傍にのみ高解像度の計算セルを配置するセルベース AMR（Adaptive Mesh Refinement）法を導入する。ブロック構造 AMR 法を導入した界面移流計算の複数 GPU 計算に対し、提案する MPF 法による領域分割法を導入した。TSUBAME3.0 を用いて 64GPU までの強スケーリングを測定した。MPF 法による領域分割は空間充填曲線よりもスケーラビリティを改善し、最大で通信時間をモートン曲線の 60%に削減した。領域分割にかかる時間は全計算時間の数パーセントであり、ブロック構造 AMR の計算に対して MPF 法による領域分割のオーバーヘッドは十分に小さいことを確認した。

1. 共同研究に関する情報

(1) 共同研究を実施した拠点名

東京工業大学・学術国際情報センター

(2) 共同研究分野

■ 超大規模数値計算系応用分野

□ 超大規模データ処理系応用分野

□ 超大容量ネットワーク技術分野

□ 超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

青木 尊之 (東工大) : 研究総括・研究遂行の詳細な指示

高木 知弘 (京都工芸繊維大) : MPF 法によるトポロジー最適化

山中 晃徳 (東京農工大) : APT 法の GPU 計算実装

杉原 健太 (東工大) : レベルセット関数の H-J 再初期化 GPU 実装

黄 遠雄 (東工大): AMR を適用した移流問題の GPU 実装

渡辺 勢也 (東工大) : MPF 法の実装、計算負荷と通信コスト間のパラメータ調整

長谷川 雄太 (東工大) : 領域間通信の MPI 実装

松下 真太郎 (東工大) : 界面に適合する AMR 法による気液二相流の MPF による動的領域分割の適用

Xiangyu Y. Hu (ミュンヘン工科大学) : Centroidal Voronoi Tessellation 法による領域分割と比較

2. 研究の目的と意義

さまざまな現象の数値シミュレーションにおいて、計算領域の中の計算負荷が均一であることは稀であり、実問題では複雑な境界条件に囲まれていることが多い。ステンシル計算にとって、均一な直交格子は実行性能を出し易いが、非構造格子と比較するとメモリ使用量が多く、Time-to-Solution は長い。計算領域を局所的に細分化する AMR（Adaptive Mesh Refinement）法は直交格子を用いながら計算効率を大幅に改善（格子点数を大幅に削減）できるが、計算負荷が局所空間に集中する典型的な例である。近接相互作用に基づく DEM（Discrete Element Method）や SPH（Smoothed Particle Hydrodynamics）などの粒子法も粒子分布が空間に一様でなく、複数ノードによる大規模計算においては、全体の計算領域をどのように分割するかが最重要課題となっている。空間充填曲線を用いて分割を行えば、分割のコストも少なく、計算の負荷バランスを均一にすることができる。しかし、各領域が凸形状にならないために領域間通信が大きなオーバーヘッドになり、大規模計算ではスケーリングが悪化することが分かっている。一方、Metis のようなグラフ理論を用いた分割は各領域を凸形状にすることができるが、分割コストが大きいため、動的 AMR 法や粒子法のように計算負荷の空間分布が時間変化する問題（動的領域分割）には適用できない。

計算の負荷バランスと領域間の通信低減を両立するコストの低い、新たな動的領域分割法の開発が必

要である。本研究では、材料分野において多結晶・多相構造の組織形成過程の計算で使われている MPF（マルチフェーズフィールド）法を動的領域分割に適用する新しいアイディアを試みる。MPF 法では、各結晶粒は界面エネルギーを駆動力として成長し、結晶粒界面の表面積を最小化する。各結晶粒・各相を各ノードの計算領域と見なし、計算負荷を均一化するような制限を与えた各結晶粒・各相の競合成長により、計算負荷のバランス保ちつつ表面積を最小化する効果により通信コストを陰的に最適化する。

本研究では動的な AMR 法を適用したステンシル計算に対して MPF 法による動的領域分割を適用し、その有効性を確認する。エクサスケールのスパコンで使われる GPU を用いた大規模 AMR 計算を前提に、空間充填曲線などのこれまでの方法に対する優位性を示し、実 AMR アプリケーションに適用することで有効性の検証とスケーリングの確認を行う。

3. 当拠点公募型共同研究として実施した意義

研究代表者のグループはこれまで大規模計算（HPC 分野）の多くの実績があり、GPU コンピューティングの研究を進めてきた。本共同研究の体制は、フェーズフィールド法の大規模計算の共同研究により 2011 年に連名でゴードンベル賞を受賞した京都工芸繊維大学、MPF（マルチフェーズフィールド）法の計算で先進的な研究を進めている東京農工大の研究者、動的領域分割法で共同研究を進めているミュンヘン工科大学の研究者らで構成されている。JHPCN の枠組みを使い、HPC 分野の研究者と異なる専門分野の研究者が連携することで、計算の負荷バランスとノード間通信を両立する新しい動的領域分割を開発し、実問題へ適用して有効性を実証することを目指す。

4. 前年度までに得られた研究成果の概要

新規課題なので該当しない。

5. 今年度の研究成果の詳細

5.1 MPF 法の領域分割への適用

MPF 法は多結晶粒の組織形成のシミュレーション手法であり、結晶粒界面のエネルギーの減少を界面駆動力として各結晶粒が成長していく。そのため、結晶粒と結晶粒の界面の表面積が最小となるような多結晶粒組織が最終的に形成される。MPF 法では、異なる N 個の結晶粒から構成される組織をシミュレーションする場合、結晶粒の存在を表す秩序変数 $\phi_i \in \{1,\ 2,\ \cdots, N\}$を定義する。結晶粒 i が存在する座標では $\phi_i = 1$、存在しない場合は $\phi_i = 0$とする。結晶粒と結晶粒の界面は有限の厚みがあると仮定し、$0 < \phi_i < 1$ の値を取る界面では異なる複数の結晶粒のフェーズフィールド変数が共存するため、各格子点において、各結晶粒のフェーズフィールド変数は以下の拘束条件を満たす。

$$\sum_{i=1}^{N} \phi_i = 1$$

任意座標で共存している結晶粒の数を N とし、一般的な粒成長で用いられるフェーズフィールド変数 ϕ_iの時間発展方程式は

$$\frac{\partial \phi_i}{\partial t} = -\frac{2}{n}\sum_{j=1}^{N}\omega_{ij}\left[\sum_{k=1}^{N}(\omega_{ik}-\omega_{jk})\left(\phi_k + \frac{\delta^2}{\pi^2}\nabla^2\phi_k\right)\right]$$

である。ここで、δは結晶粒の界面厚さ、ω_{ij}は同一の結晶粒を識別するパラメータであり、

$$\omega_{ij} = \begin{cases} 0 & (i = j) \\ 1 & (i \neq j) \end{cases}$$

と設定する。

一般的な粒成長では、界面エネルギーが最小となるように大きい結晶粒が優先的に成長し、小さい結晶粒は収縮・消滅するため、一般的なマルチフェーズフィールド法をそのまま領域分割に利用することはできない。結晶粒の数を指定したプロセス数に固定し、かつ各結晶粒で割り当てられた小領域の計算コストを均一化するために、隣接する小領域との計算コストの差に基づき界面を駆動させる補正項を導入する。領域分割のための MPF 法の時間発展方程式は

$$\frac{\partial \phi_i}{\partial t} = -\frac{2}{n}\sum_{j=1}^{N}\omega_{ij}\left[\sum_{k=1}^{N}(\omega_{ik}-\omega_{jk})\left(\phi_k + \frac{\delta^2}{\pi^2}\nabla^2\phi_k\right) + k(C_i - C_j)|\nabla\phi_i|^2\ \left|\nabla\phi_j\right|^2\right]$$

である。ここで、k はパラメータ、C_i は結晶粒 i で割り当てられた小領域の計算コストであり、

$$C_i = \frac{N_i^{\mathrm{block}}}{N_{\mathrm{ave}}^{\mathrm{block}}}$$

と見積もる。N_i^{block} は小領域 i のブロック数、$N_{\mathrm{ave}}^{\mathrm{block}}$ は一つの小領域に含まれるブロック数の平均値であり

$$N_{\mathrm{ave}}^{\mathrm{block}} = \frac{1}{N}\sum_{i=1}^{N} N_i^{\mathrm{block}}$$

と計算する。

離散化には 2 次精度中心差分を、時間積分には 1 次のオイラー法を用いる。

5.2 Active Parameter Tracking 法のよる MPF 法の高速化

領域分割に用いるフェーズフィールド変数ϕの数は領域分割の数に比例して増えるため、計算領域を 1000 ノードに領域分割する場合は 1000 個の変数ϕが必要になる。全領域に変数 ϕ のメモリを確保するとメモリ使用量が破綻するため、図 1 に示す、Active Parameter Tracking 法を導入する。同一格子点に存在

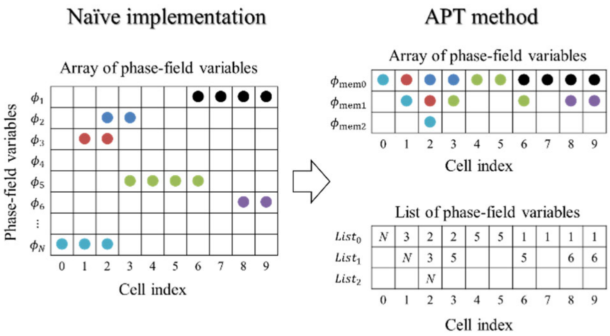

図 1　Active Parameter Tracking 法の概念図

しうる最大の変数 ϕ の数（図 1 では 3）のメモリを確保し、そのメモリに $\phi > 0$ の変数のみを保存する。これにより、メモリ使用量を大幅に削減することができる。

5.3 MPF 法へのセルベース AMR 法の導入

MPF 方程式の解は、相の界面で有限な厚さのプロファイルを持ち、AMR 法の最小リーフに対して設定する界面幅を全領域で用いる必要がある。大規模計算では、最小リーフと同程度の解像度のセルを用いても MPF 計算がかなり大規模になってしまう。そこで、AMR 格子とは別の MPF 方程式を計算するためだけの Narrow Band のような格子細分化を用いて、効率的な計算を行う。図 2 は 2 次元均一格子を 5 つの小領域に分割した例であり、右図はセルベース AMR 法を導入した MPF 法の結果で、色は APT 法のリストに登録されている変数ϕの数を示す（水色のセルには 1 つ、オレンジは 2 つ、赤は 3 つ）。2 つ以上の変数が存在している領域とその近傍にのみ高解像度の計算セルを割り当てることで、計算精度を維持したまま格子点数を大幅に削減する。

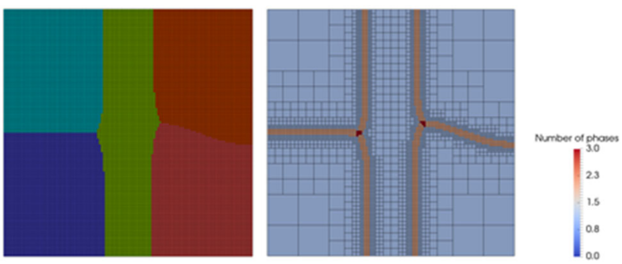

図 2　セルベース AMR 法を導入した MPF 法（右）による 2 次元均一格子の分割（右）

5.4 細分化格子の静的な領域分割でのパラメータ設定

5.4.1 2 次元細分化格子の領域分割

計算領域の左下に配置された円の表面に細かい格子を配置した 2 次元細分化格子の領域分割を、MPF 法の体積補正項のパラメータ k を変えて行い、負荷バランスと通信コストへのパラメータの影響を調べる。最も細かいブロックは128×128相当の解像度であり、最も粗いブロックの 16 倍の解像度である。

領域分割数を 16 とし、MPF 法のパラメータを $k \in \{100,\ 200,\ 400,\ 800,\ 1600\}$とする。MPF 法の最小格子の解像度は、細分化格子のブロック解像度と同じ 128×128 相当である。境界条件としてノイマン条件を課し、MPF 法の反復回数は 3000 とした。

パラメータを$k = 800$とした場合の領域分割の結果を図 3 に示す。MPF 法の結晶粒組織（左図）に基づき、細分化格子（右図）の分割が行えている。各小領域は表面積が小さくなる凸形状をしているのが確認できる。

各小領域のブロック数から見積もった負荷分散の誤差（左図）と全体のブロック数に対する小領域表面にあるブロック数の割合（右図）の時刻歴を図 4 に示す。パラメータkを大きくするほど負荷分散の誤差が小さくなり、$k = 1600$では、負荷分散の誤差が 3%

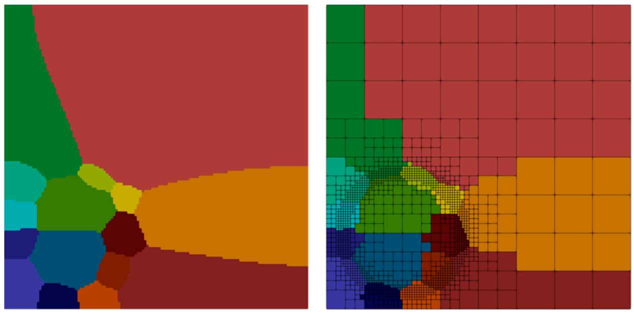

図 3　2 次元 AMR の MPF 法による分割（左：MPF 法の結晶粒組織、右：分割された細分化格子）

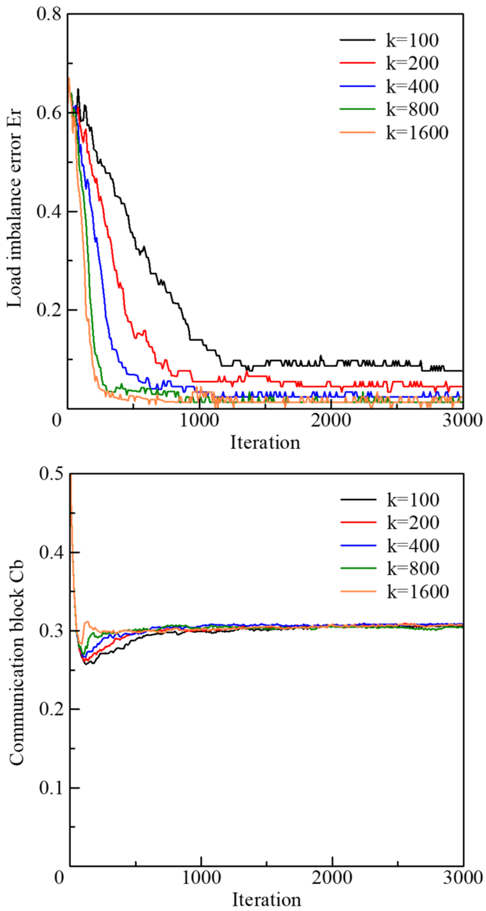

図 4　パラメータを変えた MPF 法の領域分割の結果（上：負荷分散の誤差、下：通信ブロックの割合）

以下の領域分割が行えている。また、パラメータが大きいほど収束するまでに必要なステップ数が少ないことも確認できる。一方、通信コストに関係する小領域表面のブロック数の割合は、パラメータによらずほぼ同じ値に収束している。

5.4.2 3 次元細分化格子の領域分割

1 辺が 1 の立方体の計算領域に半径が 0.2 の球形プロファイルを位置[0.25, 0.25, 0.25]に配置し、プロファイルの表面に高解像度を配置した細分化格子を、

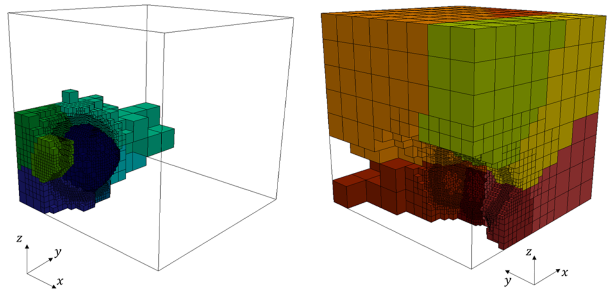

図 5 3 次元細分化格子の MPF 法による分割（左：領域番号 1 から 8、右：領域番号 9 から 16）

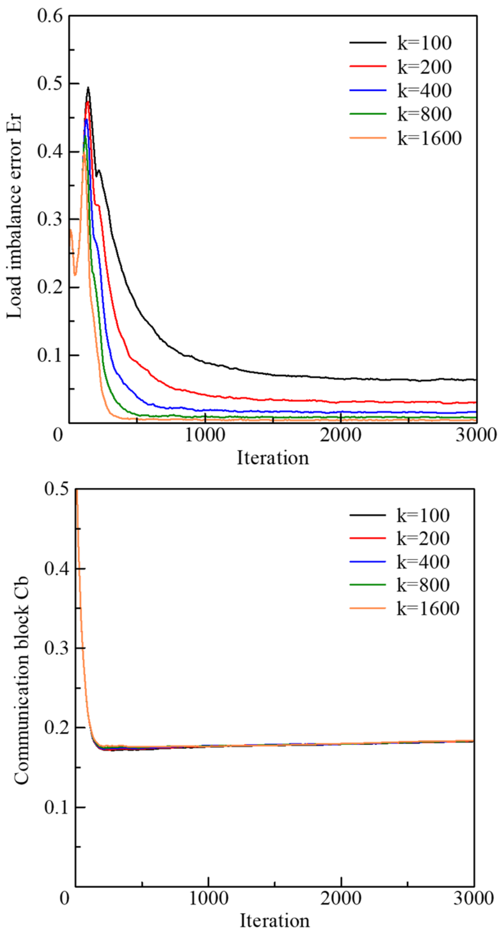

図 6 3 次元領域分割でのパラメータkの影響（上：負荷分散の誤差、下：通信ブロックの割合）

MPF 法の体積補正項のパラメータ k を変えて分割する。最も細かいブロックは$128 \times 128 \times 128$相当の解像度である。2 次元の領域分割の評価と同様に、領域分割数を 16 とし、MPF 法のパラメータを $k \in \{100, 200, 400, 800, 1600\}$とする。MPF 法の最小格子の解像度は、細分化格子のブロック解像度と同じ $128 \times 128 \times 128$ 相当である。境界条件としてノイマン条件を課し、MPF 法の反復回数は 3000 とした。3 次元細分化格子に対する MPF 法による領域分割の結果を図 5 に示す。各小領域が凸形状であることが確認できる。

負荷分散の誤差（左図）と通信が必要なブロック数の割合（右図）を図 6 に示す。2 次元の場合と同様に、パラメータkを大きくするほど負荷分散の誤差が小さくなり、収束するまでに必要なステップ数が少ないことが確認できる。このことから、体積補正項のパラメータを大きい値に設定することで、領域分割に係る計算コストを削減することができると考えられる。小領域表面のブロック数の割合は、パラメータによらずほぼ同じ値に収束している。

5.5 ブロック構造 AMR 法を導入した界面移流計算の複数 GPU 実装

5.5.1 保存形 Allen-Cahn 方程式による界面捕獲

気液二相流などの混相流の計算では、相界面の挙動を捉えるための界面捕獲手法が必要となる。格子上で大変形する界面を高精度かつ安定に捉えるために、界面にある程度の厚みをもたせた拡散界面を表現する Diffuse interface model（DIM）が注目されている。DIM の計算では、相界面近傍に高解像度の格子が必要であり、AMR 法の導入により計算コストが削減できる。

拡散界面の厚さを均一に保ち、高精度に界面の移流計算を行える DIM として、Conservative Allen-Chan 方程式を解く手法があり、界面プロファイルϕ^{AC}の時間発展方程式は

$$\frac{\partial \phi^{\mathrm{AC}}}{\partial t} + \nabla \cdot \left(\boldsymbol{u}\phi^{\mathrm{AC}}\right) = \bar{\gamma}\epsilon\nabla^2\phi^{\mathrm{AC}} - \bar{\gamma}\nabla\left[\phi^{\mathrm{AC}}\left(1-\phi^{\mathrm{AC}}\right)\frac{\nabla\phi^{\mathrm{AC}}}{|\nabla\phi^{\mathrm{AC}}|}\right]$$

である。ここで、$\boldsymbol{u}$は速度場、$\bar{\gamma}$とϵはパラメータである。

ここでは Conservative Allen-Chan 方程式を有限体積法で解く。時間積分には 2 段 2 次ルンゲクッタ法を用いる。空間の離散化には、左辺の保存方程式には 3 次 MUSCLE 法、右辺の拡散・逆拡散項には 2 次中心差分を用いる。

5.5.2 ブロック構造 AMR 法の複数 GPU 実装

保存形 Allen-Chan 方程式の計算に対し、界面に高解像度の格子を適合する AMR 法の複数 GPU 実装を行い、その動的領域分割に提案する MPF 法を適用する。

図 7 のように木構造を用いて計算領域を再帰的に分割し、木構造の末端のリーフに均一格子のブロックを割り当てる。界面のレベルセット関数を用いて

高解像度格子を割り当てる領域を決定する。

木データ構造の作成は再帰的な処理であり、GPU での処理が困難であるため、CPU で木構造の作成を行う。CPU 側で各ブロックの隣接情報などを作成し、その情報を GPU 側に転送する。一つのブロックに対して一つの CUDA Block を割り当て、ステンシル計算を行う。

複数 GPU 実装では、計算領域を MPI プロセス数の小領域に分割し、一つのプロセスが一つの小領域を担当して計算する。各 MPI プロセスに OpenMP7 スレッドと 1 台の GPU を割り当てるハイブリッド並列を行う。木構造の処理や領域分割の計算は各 CPU が OpenMP を用いて冗長に計算する。

小領域表面の計算セルではステンシル計算で隣接する小領域のデータが必要であるため、GPU 間の通信が発生する。そのために、各小領域の表面に隣接小領域のデータを保存するためのブロック（ハローブロック）を配置する。小領域表面のブロックを隣接小領域のハローブロックに転送することで GPU 間の袖領域通信を行う。一度の袖領域通信に対して複数回のステンシル計算を実行する Temporal Blocking を行うことで、通信時間を削減する。

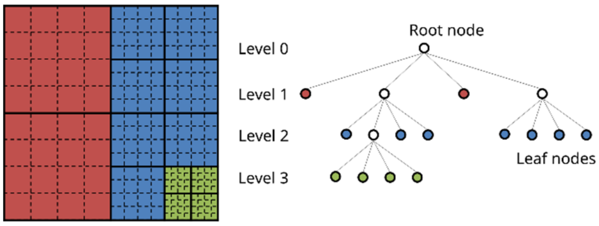

図 7　木構造を用いたブロック構造 AMR 法

AMR 法の並列計算では、動的な領域分割によるブロックのデータマイグレーションが発生する。他のプロセスへの転送が必要なブロックを判定し、ひとつずつ MPI Isend と Irecv を用いて通信を行う。

5.6 MPF 法による動的領域分割の性能評価

5.6.1 2 次元領域分割での性能測定

界面捕獲手法のベンチマークである Single vortex 問題に対し AMR 法を適用した 2 次元の保存型 Allen-Chan 方程式で解く。一辺の大きさが 1 の正方形の計算領域を設定し、半径が 0.2 の円形プロファイルを配置する。円形プロファイルは以下の速度場で移流する。

$$u(x,y,t) = 2\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos\left(\frac{\pi t}{T}\right)$$

$$v(x,y,t) = -2\sin(\pi x)\cos(\pi x)\sin^2(\pi y)\left(\frac{\pi t}{T}\right)$$

ここで、Tは周期であり、$T = 8.0$ sと設定する。

界面捕獲手法の計算に用いる AMR 格子には、各ブロックに16×16の均一格子が割り当てられる。最も粗いブロックの解像度は16×16相当であり、128×128セルに対応する。最も細かいブロックの解像度は128×128相当であり、2048×2048セルに対応する。高解像度の格子は界面近傍に割り当てられる。初期状態において、全ブロック数は 2305 で、計算セル数は 590,080 である。16GPU（Tesla P100）を用いて周期の半分にあたる$t = 4.0$ sまで計算する。計算領域の境界条件にはノイマン条件を用いる。AMR 格子の細分化と粗大化の処理は 500 ステップに一度行う。

領域分割に用いるマルチフェーズフィールド法の計算に128×128セルを用い、各 MPF プロセスが領域分割の計算を冗長に行う。パラメータは$k = 800$と設定し、初期状態における領域分割では、マルチフェーズフィールド法の反復計算を 2000 回行う。動的領域分割では、負荷バランスの誤差が 5%以下になるまで反復計算を行う。

2 次元 Single vortex 問題における動的領域分割の結果を図 8 に示す。どの時刻においても、MPF 法で分割された小領域は凸形状であることが確認できる。界面プロファイルの変化が激しい$t = 2.0$ sまでは、小領域の形状と位置も大きく変化している。界面プロファイルの変化が緩やかな$t = 2.0$ s以降では、小領域の形状の変化は小さく、時間が経過しても似た形状を保っている。小領域の形状と位置の時間変化が小さいことにより、ブロックの割り当てられるプロセスが変わることによるデータマイグレーションのコストが小さくなる。

2 次元 Single vortex 問題における負荷バランスの誤差と通信コスト（小領域表面のブロック数の割合）の時刻歴を図 9 に示す。MPF 法の領域分割の性能を比較するため、モートン曲線を用いた場合の結果（黒線）と比較する。MPF 法は設定した負荷バランス誤差 5%以内をキープすることが出来ている。MPF 法による領域分割を用いた場合の通信コストは、全時刻においてモートン曲線を用いた場合よりも小さく、MPF 法の領域分割は通信コストを削減できている。これは、MPF 法による領域分割では、小領域の形状が凸形状となり表面積が小さくなるため、通信量を削減できたと考えられる。なお、どちらの結果においても計算の進行に伴い通信量が増加しているのは、界面プロファイルの形状の変化によりブロック数が増加したためである。

動的領域分割において、一回の領域分割に要した MPF 法の反復回数を図 10 に示す。初期の分割では収束までに数百回の反復計算が必要であるが、動的

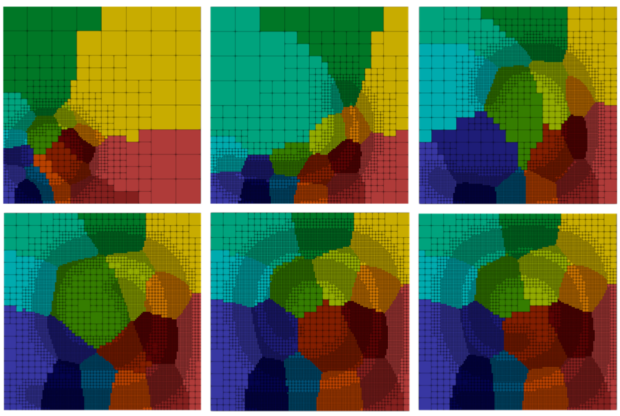

図 8　2 次元 Single vortex 問題に対する MPF 法による動的領域分割の結果（左上から$t = 0, 0.5, 1.0, 2.0, 3.0, 4.0$ s）

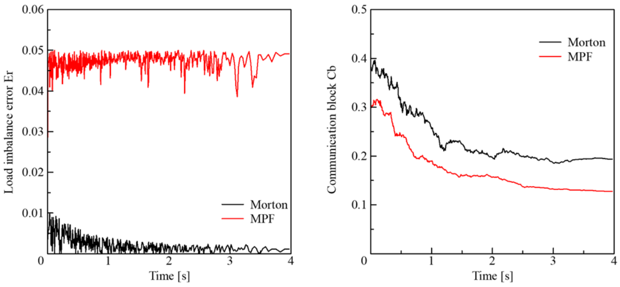

図 9 2 次元動的負荷分散における負荷バランス誤差（左）と通信コスト（右）の時刻歴

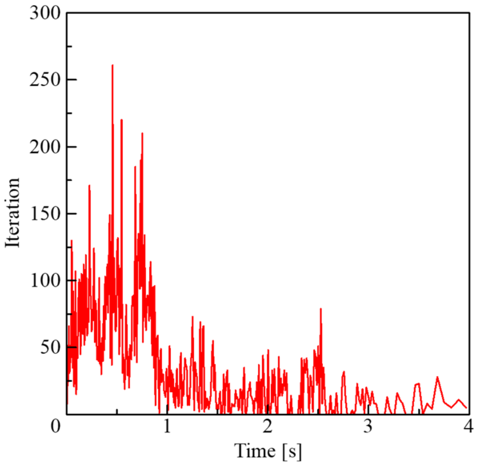

図 10 2 次元動的負荷分散における一度の領域分割に必要な MPF 法の反復回数

領域分割では、前の分割結果を初期値として新しい細分化格子の分割を行うことで､100 回程度の反復計算で十分に最適化された分割の結果が得られた。特に、界面プロファイルの変化が緩やかな$t = 2.0$ s以降では、数十回の反復で分割を行えている場合が多い。一回の領域分割にかかる時間は平均で 0.06 秒であり、ブロック構造 AMR の全計算時間の約 3.2%である。

5.6.2 3 次元領域分割での性能測定

一辺の大きさが 1 の立方体の計算領域を設定し、半径が 0.2 の球形プロファイルを配置し、一様速度場で移流させる。並列数を 32GPU とする。AMR 格子では､各ブロックに$16 \times 16 \times 16$セルが割り当てられる。最も粗いブロックの解像度は$8 \times 8 \times 8$相当であり、均一な$128 \times 128 \times 128$セルに相当する。最も細かいブロックの解像度は$128 \times 128 \times 128$相当であり、均一な$2048 \times 2048 \times 2048$セルに相当する。

領域分割に用いるマルチフェーズフィールド法の計算に$128 \times 128 \times 128$セルを用い、各 MPF プロセスが領域分割の計算を冗長に行う。パラメータは$k = 1600$と設定する。動的領域分割では、負荷バランスの誤差が 5%以下になるまで反復計算を行う。

負荷バランスの誤差と通信コスト（小領域表面のブロック数の割合）の時刻歴を図 11 に示す。3 次元計算においても､設定した負荷バランス誤差 5%以内をキープすることが出来ている。マルチフェーズフィールド法を用いることで、モートン曲線よりも通信コストを約 83%に削減できている。1 回の領域分割に要した MPF 法の反復回数を図 12 に示す。21 回程度の反復回数で領域分割を行えている｡3 次元計算では一つの小領域に含まれるブロック数が多くなるため、図 10 の 2 次元計算の場合よりも必要な反復回数が少なくなった。一回の領域分割にかかる時間は平均で 1.84 秒であり、ブロック構造 AMR の全計算時間の約 1.2%である。

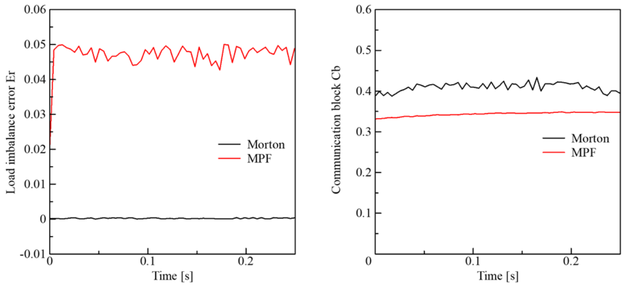

図 11 3 次元動的負荷分散における負荷バランス誤差（左）と通信コスト（右）の時刻歴

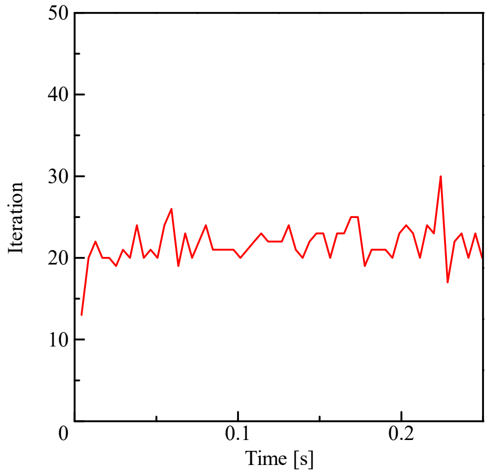

図 12 3 次元動的負荷分散における一度の領域分割に必要な MPF 法の反復回数

5.6.3 強スケーリングの測定

平成 30 年度の後半では、東工大の TSUBAME3.0 での強スケーリングの測定を行い、MPF 法による動的負荷分散を用いたブロック構造 AMR 法の実行性能を、空間充填曲線（モートン曲線）による場合と比較した。

気液二相流などの実際の流体計算では、液滴の飛び散りや気泡の発生などにより高解像度格子が必要な領域は広く分布しており、平成 30 年度の前半で試していた単一の球形プロファイルのような一箇所に局所的に集まることは稀である。そこで、強スケーリングの測定では、計算領域に複数の球形プロファイルをランダムに配置することで、実際の流体計算のように細い格子が計算領域に広く分散した条件を設定した。

立方体の計算領域[0, 1]×[0, 1] ×[0, 1]に乱数を用いて半径が 0.05 の球形プロファイルを 45 個配置した。球形プロファイルは以下の回転速度場に従い移流する。

$$u(x,y,z,t) = -2\pi(y-0.5)$$
$$v(x,y,z,t) = 2\pi(x-0.5)$$
$$w(x,y,z,t) = 0$$

強スケーリングの測定に用いるブロック構造 AMR 格子では、各ブロックに$16\times16\times16$セルを割り当てる。最も粗いブロックの解像度は$8\times8\times8$相当であり、均一な$128\times128\times128$セルに相当する。最も細かいブロックの解像度は$128\times128\times128$相当であり、均一な$2048\times2048\times2048$セルに相当する。初期時刻におけるブロック数は 181,441 であり、743,182,336 格子セルである。GPU 数を 16, 32, 64 と変えて測定を行った。動的な格子細分化と領域分割は AMR 計算の 500 ステップ毎に 1 回行っている。これは、界面プロファイルが 1 ブロック分の格子（今回は 16 格子）を移動したときに、細分化を実行する頻度に相当する。MPF 法の体積補正項のパラメータは$k = 1600$とし、負荷分散の誤差が 5%以下になるように領域分割を行った。

強スケーリングの測定結果を図 13 に示す。1 秒間に更新（時間積分）されたセル数を用いて実行性能を評価した。MPF 法による負荷分散を用いることで、実行性能をモートン曲線の 1.07 倍（16GPU）、1.11 倍（32GPU）、1.31 倍（64GPU）に向上させることができた。16GPU から 64GPU での強スケーリングは、MPF 法を用いた場合は 70.6%、モートン曲線を用いた場合は 57.8%である。本研究で提案する MPF 法による負荷分散は強スケーリングを改善することを確認することができた。

強スケーリングの測定での実行時間の内訳を図 14 に示す。ステンシル計算の時間（青）はどの条件においても MPF 法がモートン曲線よりも短い結果となった。モートン曲線は各小領域のブロック数をほぼ完璧に均一にできるため、理想的には MPF 法よりもモートン曲線を用いた場合のほうが計算時間は短くなるはずである。本研究の AMR コードの複数 GPU 実装では、袖領域の通信に Temporal Blocking 法を導入しており、各 GPU は小領域内のブロックだけでなくハローブロックの計算も行う必要がある。MPF 法は表面積の最小化により各小領域のハローブロックの数を削減できたため、計算時間がモートン曲線よりも短くなったと考えられる。

GPU 間の袖領域通信の時間（赤）はモートン曲線よりも MPF 法を用いた場合が短く、MPF 法は通信コストを削減することが確認できた。64GPU の条件では、MPF 法を用いたときの通信時間はモートン曲線の場合の 60.9%であった。

MPF 法の領域分割にかかる時間（紫）は 16GPU の場合で 0.84 秒、32GPU で 1.12 秒、64GPU で 5.9 秒であり、並列数の増加に伴い長くなることが確認できた。これは、並列数の増加に伴い、フェーズフィールド変数の数が増加すること、一つの小領域に割り当てられるブロック数が少なくなり MPF 法の収束性が悪化し反復回数が増えることが原因である。平成 30 年度の前半では、MPF 法の計算負荷がオーバーヘッドとして大きくなり、並列化や GPU 化を行う必要があるのではないかと懸念された。今回の強スケーリングの測定では、MPF 法の計算に最も時間がかかった 64GPU の条件において、MPF 法のオーバーヘッドは全計算時間の 1%以下であった。 問題設定によっては（例えばブロック数が少ない小規模な問題に対し多くの並列数を設定する）MPF 法の収束性が著しく悪化することも確認できており、そのような場合には、補正項のパラメータや許容する負荷分散の誤差を調整して反復回数を低減させる必要がある。

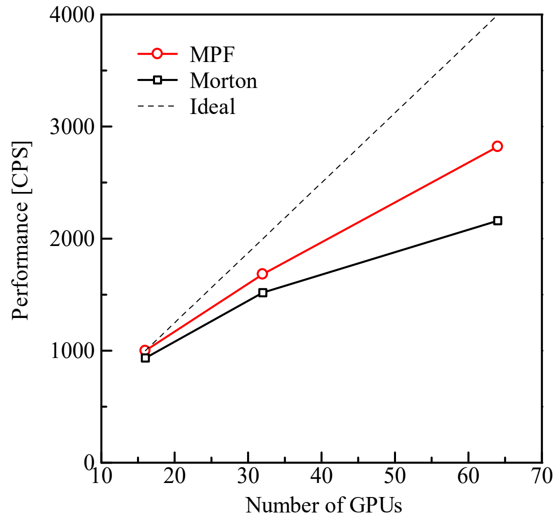

図 13 MPF 法とモートン曲線を用いた場合の強スケーリングの測定結果

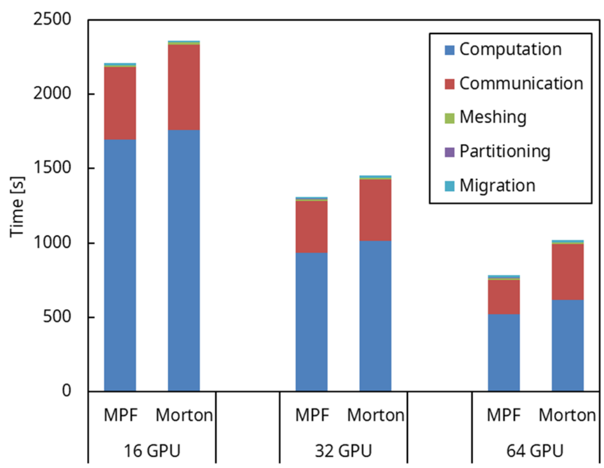

図 14 強スケーリングの測定の計算時間の内訳

6. 今年度の進捗状況と今後の展望

ブロック構造 AMR 法の複数 GPU 計算における袖領域通信のコスト削減に向けて、MPF 法に基づく領域分割による負荷分散手法の開発を行い、TSUBAME3.0 を用いて領域分割の性能を評価した。

平成 30 年度の前半では、MPF 法を用い固定した細分化格子の領域分割を行い、体積補正項のパラメータの負荷分散誤差と通信コストへの影響を確認した。動的領域分割の評価を行うために、保存形 Allen-Cahn 方程式を用いた与えられた速度場での界面移流

計算に対し、界面に適合するブロック構造 AMR 法の複数 GPU 実装を行った。OpenMP や APT 法、セルベース AMR 法を用いて MPF 法の単一ノードのチューニングを行い、界面追跡の AMR コードに MPF 法による領域分割を導入した。円形や球形などのシンプルなプロファイルを移流させる問題に対し、MPF 法で動的領域分割を行い、従来の空間充填曲線と比較した。動的負荷分散の問題に対し、MPF 法は負荷分散を指定した誤差以下に保つことができた。MPF 法で分割された小領域は凸形状であり、モートン曲線よりも小領域表面のブロック数を削減することを確認した。

平成 30 年度の後半では、TSUBAME3.0 を用いて強スケーリングを測定した。MPF 法は通信コストを削減することでモートン曲線よりも強スケーリングを改善できることを確認した。当初の研究計画では、FY2018 Q4 に別課題（jh180035-NAJ）「界面に適合する AMR 法を用いた非圧縮性気液二相流の完全陽解法計算と GPU 実装」で開発を進める気液二相流の AMR コードに MPF 法による負荷分散を導入する予定であった。しかし、気液二相流の計算スキームの大幅な変更を行い、スキームの検証などに時間がかかったため準備が整わなかった。

今後の予定として、開発を進めている気液二相流コードなどの AMR アプリケーションに MPF 法による負荷分散を実装し、実アプリケーションでの有効性を検証する予定である。

7. 研究成果リスト

(1) 学術論文

[1-1] 渡辺勢也，*青木尊之*，長谷川雄太，河原淳，橋本博公: 格子ボルツマン法による物体を含む自由界面流れの大規模シミュレーション，日本混相流学会誌「混相流」，Vol. 33，No.1，pp 55—62，2019 年 3 月

[1-2] Seiya Watanabe, *Takayuki Aoki*, Tomohiro Takaki : A Domain Partitioning Method Using a Multi-phase-field Model for Block-based AMR Applications, Parallel Computing（投稿中）

(2) 国際会議プロシーディングス

(3) 国際会議発表

[3-1] Seiya Watanabe, *Takayuki Aoki*, Yuta Hasegawa: Simulations of Free Surface Flows Interacting with Solid Objects by Lattice Boltzmann Method Using Multiple GPUs, ECCM-ECFD 2018, Glasgow, UK, June 12, 2018

[3-2] Seiya Watanabe, *Takayuki Aoki*, Yuta Hasegawa: GPU-accelerated Lattice Boltzmann Simulations with Octree-based Dynamic AMR Method, DSFD 2018, Worcester, USA, June 28, 2018

(4) 国内会議発表

[4-1] 渡辺勢也, *青木尊之*, 黄遠雄, 長谷川雄太, 高木知弘: マルチフェーズフィールド法に基づく領域分割を用いた AMR 計算の動的負荷分散, 第 23 回計算工学講演会, 京都, 2018 年 6 月 8 日

[4-2] 渡辺勢也, *青木尊之*, 長谷川雄太, 河原淳: 多数の瓦礫を含んだ津波が構造物へ与える影響の数値解析, 混相流シンポジウム 2018, 仙台, 2018 年 8 月 9 日

[4-3] 渡辺勢也, *青木尊之*, 長谷川雄太: AMR 法を導入した格子ボルツマン法による大規模流体構造連成解析, 日本流体力学会年会 2018, 大阪, 2018 年 9 月 6 日

[4-4] 渡辺勢也，*青木尊之*，長谷川雄太: 動的 AMR 法を導入した格子ボルツマン法の複数 GPU による大規模計算，第 31 回計算力学講演会，徳島，2018 年 11 月 23 日

[4-5] 渡辺勢也，*青木尊之*，松下真太郎，Christian Conti，高木知弘: AMR 法の動的領域分割へのマルチフェーズフィールド法の適用，第 31 回計算力学講演会，徳島，2018 年 11 月 23 日

[4-6] 渡辺勢也，*青木尊之*，長谷川雄太: GPU スパコンにおける動的負荷分散を用いた AMR 法による大規模 LBM 計算，第 32 回数値流体力学シンポジウム，東京，2018 年 12 月 13 日

(5) その他（特許、プレス発表、著書等）

[5-1] *Takayuki Aoki*, Shintaro Matsushita, Seiya Watanabe: Explicit CFD solvers for free-surface flows and gas-liquid two-phase flows, Taiwan-Japan Workshop on Mechanical and Aerospace Engineering, Taiwan, National Chiao Tung University, Oct 6, 2018

[5-2] Seiya Watanabe, *Takayuki Aoki*: Large-scale Lattice Boltzmann Simulations with Adaptively Refined Mesh on Multiple GPUs, Taiwan-Japan CFD workshop, National Tsing Hua University, Hsinchu, March 8, 2019.

[5-3] 渡辺勢也，Phase-Field Student Award，第 31 回計算力学講演会，徳島，2018 年 11 月 24 日（受賞）

[5-4] 渡辺勢也，*青木尊之*，長谷川雄太，第 32 回数値流体力学シンポジウム ベスト CFD グラフィック・アワード 静止画部門 最優秀作品賞, 第 32 回数値流体力学シンポジウム, 東京，2018 年 12 月 12 日（受賞）