

## 利用課題名 GPU を用いた大規模血液動力学計算

<sup>1</sup>松岡聡、<sup>2</sup>Massimo Bernaschi、<sup>2</sup> Simone Melchionna、<sup>2</sup> Sauro Succi、

<sup>2</sup>Mauro Bisson、<sup>1</sup>遠藤敏夫、<sup>1</sup>丸山直也、<sup>1</sup>額田彰

1 東京工業大学

2 Consiglio Nazionale delle Ricerche

邦文抄録(500 字程度)

生体中の流体のマルチスケールシミュレーションのためには、複雑形状を流動する粒子の運動を計算する必要があり、数億個オーダーの粒子の相互作用、粒子と流体との相互作用を計算する必要がある。一つの例として、人間の冠動脈の血流を赤血球のサイズの空間解像度を用いて、生理学的なヘマトクリット値(赤血球の体積率)の条件で計算する。本稿ではTSUBAME2 スーパーコンピュータ上で高いスケーラビリティを持つ血行力学シミュレーションの手法を提案し、600TFlops 以上の高性能を実現した。この結果は、新規数学モデル、計算アルゴリズム、コンピュータアーキテクチャ、チューニング技術の統合により、臨床的に意義のある生体内流れの予測が可能であることを示している。

Keywords: 血液動力学、流体計算、粒子法、マルチフィジックス、GPGPU

### 1. はじめに

輸送現象は生体における基本現象であり、筋肉の収縮、消化現象、細胞への栄養物の輸送、血液循環はその例である。血液は代表的な生体流体であり、代謝、免疫反応、組織修復などの基本的な生理的機能に必要な、生物学的物質を供給する。

生体内の現実的な血液と血管を計算することは困難な課題である。なぜならその計算モデルは、複雑な形状の血管の中を流れ、心拍によって不規則に流速や圧力が変化する流体の動きと、赤血球、白血球や他の粒子の挙動を統合する必要があるからである。

この数年で、大規模血行力学シミュレーションは大きな進歩を遂げた<sup>[1-3]</sup>が、現実的な形状やサイズの血管に対して、流体と赤血球などの粒子挙動を連立するシミュレーションには至っていなかった。流体の圧力による非局所的相関のために、大域的な形状が、特に動脈の壁面せん断応力に大きな影響を与える。壁面せん断応力は、動脈硬化につながる複雑な生体力学的変化を誘発するとされている。正確で信頼できる血行力学（壁面せん断応力）のシミュレーションは、循環器系疾患の進行を予測するための、新しい非侵襲的な手法となることが期待される。

本稿では、CT 血管造影から再構築された人間の冠動脈の、初のマルチスケールシミュレーションについて述べる。冠動脈は心筋に血液を供給し心臓全体に渡るネットワークを成す。シミュレーション規模は5cm 程度であり、解像度は赤血球の直径の約 8 $\mu$ m に対し 10 $\mu$ m 程度である。シミュレーション対象は図 1 に示すような、約 10 億ノードから成る流体と 1000 万~4

億 5000 万の粒子である。シミュレーションは、流体のための Lattice Boltzmann (LB)法と粒子のための特化された Molecular Dynamics (MD)法を統合した、MUPHY ソフトウェア(Multi PHYsics/multiscale)<sup>[4,5]</sup>により行う。このシミュレーションはTSUBAME2 スーパーコンピュータの 4000 GPU を用いて 500TFlops 以上の平均性能、90%以上の並列化効率を達成した。

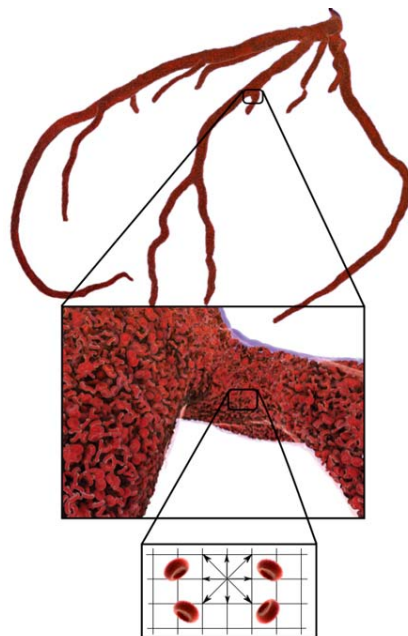


図 1 シミュレートされる冠動脈の形状の例。Lattice Boltzmann メッシュの中に赤血球(RBC)が存在する。

本研究は、高性能計算技術と物理/計算モデリングの双方の側面において特徴を持つ。非常に複雑な不規則形状の領域をTSUBAME2の4000GPUに均等に割り

当てる必要がある。この領域分割問題は流体計算部分だけであっても困難であるが、さらにこのマルチスケール法においては MD 部分の負荷のバランスも良好に保つ必要がある。

このような複雑かつ大規模形状のシミュレーションは既存研究では稀である。本研究では、大規模並列アーキテクチャの利用によって、過去最大規模のシミュレーションを、形状の規則性に依存せずに可能であることを示す。

## 2 マルチスケール生体流体力学

生物は主に二つの要素から成り立つ：水溶性の溶媒、血漿、サイトソルなどの液体と、溶媒の中を流れる生体粒子、細胞、タンパク質、DNA などの懸濁物質である。このような条件をシミュレートするために、MUPHY<sup>[4,5]</sup> ソフトウェアを開発してきた。

以前のバージョンの MUPHY と比べ、現在のフレームワークでは流体と粒子の挙動の数値的解法を扱うことができる。これにより MUPHY は、溶媒と溶質のライブラリを基にして、生体内流れ向けの計算基盤として利用することができる。機能の一部は以下の通りである：ニュートン/非ニュートンレオロジー特性の選択、分子流体力学から確率的摩擦動力学の再現のためのカーネルの選択、確率的変動の効果、分離した粒子または重合体と分子の形状の溶質の選択、多様な流体力学形状の懸濁物質、異なる粒子間力の選択、規模に対応する粒子・流体間結合機構、不規則境界の物質と組織への対応。このような選択肢により、生体流体は様々な手法でモデル化でき、生物学的・生理学的システムのマルチスケール・マルチフィジックスの挙動が理解できる。

MUPHY は 2 つの計算エンジンを持つ。一つは流体運動学上の流体の挙動を Lattice Boltzmann (LB) 法により扱う：衝突による手法により連続体中の流体の挙動を再現する (Navier-Stokes 方程式に基づく直接的巨視的な流体の挙動と対比的に)。二つ目のエンジンは Molecular Dynamics (MD) と技術的側面が共通する手法に基づき、Lagrangian 体の挙動を扱う。しかしながら懸濁物質の性質のためには通常の MD では対応できず、本質的な拡張が必要である。最後に、流体と粒子の結合のために運動モデルに基づいて設計されたカーネルが存在する。これは巨視的流体力学に基づく stresslet や境界積分法などと大きく異なる。これらの計算環境により、現実的な条件における生体流体の研究に利点をもたらす、時間に陽なシミュレーション技術を提供する。特に、1) 冠動脈のような不規則な境界形状に対応する、2) 粒子の動力学を用いるため、複雑な流動学を描写できる、3) メッシュ以下の空間解像度の流体力学上の相互作用を可能とする、4) 流体と粒子間の複雑な界面の記述を避ける。

後の二つはシミュレーション性能の加速と、特に物体が流体に及ぼす剛体力に関連した数値安定性の保証のために必須である。フレームワークは、剛体力についてシングルもしくはマルチ時間ステップアルゴリズムを採用する。ここで強調したいことは、LB 法が、力の急激な変化に耐性があり、1000 のオーダーのレイノルズ数のほふく流による濃厚サスペンションをシミュレート可能なことである。この利点により現実的な生物学的条件下において広範な物理現象に対応し、構造的関係に支配される連続体に同化できない、非局所的な流動学的反応の再現が可能となる。

### 2.1 Lattice Boltzmann 法と Molecular Dynamics 法

LB 法<sup>[6]</sup>は、離散速度で動く「流体粒子」の格子点  $x$ 、時間  $t$  における存在確率の singlet 分散の発展に基づく。流体粒子は物理的粒子の一団 (population と呼ばれる) の集団的動作を表現する。流体と個体の結合は回転並進カーネルに基づき、それらは流体中を動く剛体を、貫通できない物質、柔軟な小胞、またはその組み合わせとして扱う。

流体と固体の流体力学的作用は、 $i$  番目の粒子を中心とした伝達関数に従い形成される。関数は球もしくは楕円体対称であり compact support である。流体力学的形状はメッシュサイズより小さくなりうる<sup>[6]</sup>。流体と粒子の結合は  $N$  個の懸濁物質ごとに格子点  $\{x\}$  にまたがる畳み込み演算を必要とする。

基本的に流体力学的作用の計算量は系のサイズの三乗で増加するが、流体と固体が並行で時間発展するため、アルゴリズム上の利点が生じる。実際、LB 法の計算コストは格子点数  $M$  について線形である。ここで  $O(M)=O(Nc)$  であり、 $c = NM$  は溶質の密度の定数である。粒子間の直接力の計算のために link-cell 法を用いることにより、粒子力学の計算量は  $O(M)$  となる。この性質のために、溶媒を介した粒子間作用は局所的で陽的となり、LB-MD 結合のコストは格子数と懸濁物質数に比例する。しかしながらこの計算は、 $O(100)$  の格子点と不連続なメモリアクセスの高コストのために、最も時間を要する部分である。

## 3 MUPHY

MUPHY (MULTI PHYSICS/multiscale) ソフトウェアは元々 Fortran 90 で記述され、並列化のために MPI を用いていた<sup>[4]</sup>。不規則形状を柔軟かつ効率的に扱うために、MUPHY は間接アドレッシング法を用いる。これによりメモリ必要量を最小に抑えるだけでなく、良好な負荷分散と、プラットフォームに応じた最適な通信パターンの選択が可能となる。当初 MUPHY は、比較的低クロックの数千の PowerPC プロセッサを高速専用ネットワークで結合した IBM BlueGene アーキテクチャ<sup>[7]</sup>上で開発された。BlueGene/P 上で、我々が利用可能な最大構成 (294,912 コア) で良好なスケラ

ビリティをすでに示している<sup>[4]</sup>。この時数十 TFlops の性能を達成したが、PowerPC アーキテクチャの SIMD 的命令を利用できないなどの制限があった（これらの命令は連続アクセスを必要とするが、MUPHY ではデータアクセスパターンが不連続である）。一方で近年の GPU の計算性能の向上を受け、我々は GPU クラスタを対象とした MUPHY を開発した<sup>[4]</sup>。

複数 GPU を用いるソフトウェアは、GPU 内と GPU 間の、二レベルの並列性の記述を必要とする。通常のマルチコアシステム上ではハイブリッド方式 (OpenMP+MPI など) もしくは単純な分散メモリ方式 (MPI ライブラリが共有メモリでも効率的に動作することを期待して) にて実装する一方、複数 GPU システムにおいてはハイブリッド方式が唯一の選択肢である。考慮すべき点はそれだけではない：通常のマルチコア環境では並列度は数スレッド、ハイエンドのものでも数十スレッドである。GPU ではハードウェアを十分に稼働させるためには数百スレッドが必要であり、はるかに細粒度の並列性が必要である。さらに、GPU 上のグローバルメモリのデータを同マシンの別 GPU とやりとりする際には、CPU を介する必要がある（ただし最新の NVIDIA GPU と CUDA ドライバにより、条件によっては可能となる）。

GPU 間の通信のために CPU を介することによるオーバーヘッドは発生するものの、CUDA の stream の概念と非同期メモリコピーにより、GPU と CPU 間のデータ転送とカーネル (GPU 上で実行される関数) の実行をオーバーラップすることが可能である。さらに CPU 上の関数 (MPI 関数など) の実行は GPU とは並行に行われる。結果として、CPU は MPI 用のコプロセッサのような役割を果たす。

### 3.1 領域分割

本研究のシミュレーション対象の領域は、図 1 に示すように非常に不規則であるため、計算資源間で領域を分割すること自身が大きな課題となる。以前の実験では、計算負荷を均等に分散させるために、SCOTCH グラフ/メッシュ分割ツール<sup>[2]</sup>の並列版である、PT-SCOTCH を用いた。このツールはグラフバイセクションアルゴリズムに基づき、計算領域の形状にかかわらず分割を行うものである。しかしながら形状の知識が無いと、分割数が増え、部分領域がより不規則になると、分割の性質が悪くなると分かった。境界面積が増え通信オーバーヘッドが増加する。より適した解は、グラフに基づく分割と flooding に基づく手法 (graph-growing 法とも呼ばれる) を、以下のように統合した場合に得られると分かった：格子はまず PT-SCOTCH により固定数 (256) の部分領域に分割される。そして各部分領域は flooding アルゴリズムによりさらに分割される。

MUPHY では通信パターンは実行時に判明する。各タスクは、そのタスクがシミュレーション中にアクセスする必要のある格子点を持った、隣接タスクを決

定する。シミュレーション中に、LB アルゴリズムにおいては非局所的な population の流れが存在し、分子のシミュレーションにおいては粒子の移送やドメイン間力の計算があることに注意が必要である。前処理段階においては主に MPI 集団通信が用いられ、それ以降の実行においてはほとんどの通信が以下のような形の対一通信である：ノンブロッキング通信関数が用いられ、受信処理が先に起動してから送信処理が起動される。その後各タスクは MPI\_wait 関数により、ノンブロッキング通信の終了を待つ。

Lattice Boltzmann 部分においては大域数値 (x, y, z 方向運動量など) の計算にのみ MPI 集団通信 (reduction) が用いられる。Molecular Dynamics 部分においてもほとんどの通信が対一である。ただし不規則的形狀であることにより領域分割が例外となり、その点は次節で述べる。

### 3.2 並列 Molecular Dynamics

多くの並列 Molecular Dynamics アプリケーションにおいてはシミュレーション形状は規則的であり、各タスクがほぼ同数の粒子を持つように単純な Cartesian 分割が用いられる。不規則的形狀の場合にこのような方法を用いると、LB のための分割と MD のための分割という、二つの異なる分割が生じる。その結果一つの部分領域が二つ以上のプロセッサに所属することとなり、粒子と流体の相互作用が複雑な通信パターンを持つ非局所的処理となってしまふ。そのため、我々は MD と LB の間で並列分割を一致させることにした。これにより各タスクは LB の演算、MD の演算、さらに粒子-流体間作用の演算をほぼ局所的に実行できる。本手法では LB 格子は、各粒子が所属する部分領域を特定するためにも用いられる：場所  $R$  にある粒子は、そのベクトルにもっとも近い整数ベクトルに対応する格子を持つ部分領域に所属する。粒子は領域間にほぼ均一に存在すると期待できるため、MD 部分の負荷分散も良好に保たれる。

不規則形状の領域に適した新規並列化戦略を開発した。なかでも課題となったのは、部分領域の境界部分の隣に位置し、領域内粒子とも領域間粒子とも作用するような粒子の特定である。我々はセルの概念<sup>[8]</sup>によりこれを解決する。ここでのセルは、作用のカットオフ距離以上の辺の長さを持つ立方体であり、計算される不規則形状を埋め尽くすものである (図 2)。この概念により、プロセッサが領域内/領域間粒子ペアを効率的に探索可能であり、領域境界に存在する粒子と領域間を移動する粒子の上位集合を交換することにより通信量を削減可能である。

MUPHY のもう一つの構成要素は流体-粒子間相互作用の計算部分である。各懸濁粒子は、流体力学的な力と、流体の巨視的な速度と、 $4 \times 4 \times 4$  サイズの格子に渡る渦度の影響を受ける。同様に格子点は、粒子の運動量輸送の影響を受ける。これらの非局所的な処理は、ある粒子を持つ各プロセッサが流体力学的な量を周りの領域と交換するために、複数の通信ステップを必要とする。

## 様式 グランドチャレンジ大規模計算制度

流体-粒子間結合は、粒子と隣接領域の境界セル中の格子ノードの間の作用を含む非局所的な処理である。このため結合計算においても領域間力の計算と同様、セルの概念を用いる。流体が粒子に及ぼす力やトルクの計算のために、各プロセッサは境界セル中の粒子の情報を近隣と交換する。最後に外部の粒子に関連する力やトルクは隣接プロセッサと交換され、受信側では外部からの影響が境界粒子に伝えられる。この手法は他の手法より効率的である。たとえば粒子の代わりに境界セル中の格子点を交換した場合、一ステップにおける通信回数を削減することができる。しかしこの手法では界面に近いすべての格子ノードの交換のために通信オーバーヘッドは大きくなる。

最後に、粒子が流体へ及ぼす運動量輸送については、境界上の粒子の情報が交換される。この点は流体から粒子への影響の場合と同様だが、二回目のデータ交換は不要である。

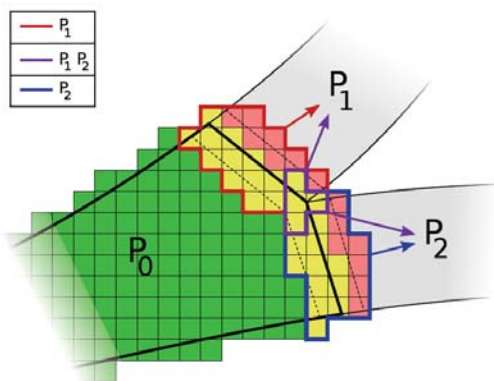


Fig.2 境界セルと内部セルを不規則領域に敷き詰めた様子。隣接領域とのデータ交換を限定することができる

## 4. 実験結果

これまでに述べたシミュレーションについて、全体実行時間と、計算と通信の内訳について測定する。シミュレーション対象は以下の通りである：流体については約10億の格子ノードから成る領域であり、約3000億ノードの bounding box に収まる。粒子は4億5000万の赤血球(RBC)である。流体-粒子結合のパラメータについては、以下のように設定した：各赤血球は小球の最小・最大主要方向について  $4\mu\text{m}$  および  $8\mu\text{m}$  の流体力学的形状を持つとし、ヘマトクリットは 58%とした。

多くの演算は単精度浮動小数で行われ、一部の reduction 処理は倍精度で行われた。LB 計算においては各 GPU スレッドは複数格子ノードの更新処理を行い、そのノード数は利用した GPU 数に依存する(スレッド設定は各 GPU 固定とした)。たとえば 512GPU の場合各スレッドは 8 格子ノードを担当する。MD 計算においては、粒子間作用の演算は粒子ごとに行われる。ここでスレッドグリッドは粒子の配列にそのまま対応するスレッドはそのグローバル ID に応じて粒子に対応し、作用ペアの探索は各スレッド独立に行われる。各スレッドは隣接セルを調べ、各作用ペアについて力

## 利用成果報告書 (英語版)

への寄与を計算する。この実験により、生理学的なレベルのヘマトクリットにおけるコードの信頼性の基礎試験を行う。

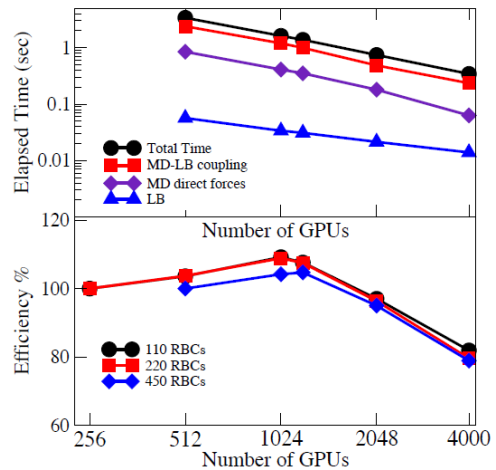


Fig.3 上部: 時間ステップあたりの経過時間. 下部: 並列化効率.

図 3(上部)では、シミュレーションステップごとの実行時間と、LB 部分、MD 部分の内訳時間を示す。いくつかの知見を述べる：MUPHY の Lattice Boltzmann 部分の 1GPU 上の性能は、高度に最適化された他の CUDA LB カーネルと一致する。GPU 数を増やした場合には実行時間は大幅に削減されており、4000GPU の場合には 256GPU の 12.5 倍の性能向上が見られ、並列化効率は約 80% に相当する。MD における直接力の計算は効率 95% に相当する。LB 部分は非常に効率的であり、負荷は常に 4% 以下である。

またこの結果は非同期通信と通信・計算のオーバーラップの組み合わせにより GPU 間の直接データ交換ができない問題を緩和できていることを示す。

図3(下部)は全体の並列化効率を示す。その基準は、利用可能なメモリの範囲でシミュレート可能なヘマトクリットレベルの場合には 256GPU であり、それより高レベルの場合は 512GPU である。1024GPU までは線形を超えた性能向上が起こっており、また利用した最大 GPU 数の場合の並列化効率は 80% 程度であると分かる。この高いスケーラビリティは、複数の枝を持つ冠動脈向けに提案したグラフ分割と flooding 法のハイブリッド方式により実現できたと考えられる。1200 領域を用いるテストケースでは赤血球の分布には広い分散があったが、MD/LB 部分の実行時間は小さかった。これは部分領域間の境界が小さく計算と通信の共有が最適化されたためである。

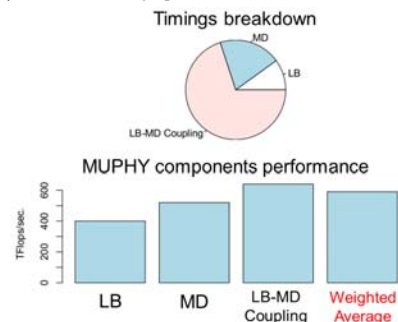


図 4: 各計算コンポーネントの計算時間の割合とそれぞれの性能.

TSUBAME2の1334ノード(4000MPIタスク)を用いた場合の、(重みつき)平均性能は、600TFlops弱となる。図4にその内訳を示す。この成果を実用にもたらずとすると、このTSUBAME2全体を用いることにより、完全な心拍の、マイクロ秒の時間解像度における、赤血球を全て含んだシミュレーションを、48時間で行うことが可能である。

## 5. おわりに

世界トップクラスのGPUクラスタを用いた、心臓規模の冠動脈の計算生体流体力学シミュレーションを初めて行った。シミュレートされた血行力学系は赤血球レベルの空間解像度を持ち、人間の冠動脈の複雑な形状から成る。計算された環境は10億の流体ノードと、並行に流れる数億の赤血球を含む。TSUBAME2スーパーコンピュータの持つ4000GPUを用いて、約600TFlopsの性能と90%以上の並列化効率を実現し、4億5000万の赤血球の存在下で約2兆回/秒のlattice更新を行った。

この成果は、高性能計算技術と物理的計算モデルの双方における新規開発によるものであり、我々の知る限り、理想化されない形状に対応した初の実装である。この研究は、現実の生体流体力学の研究に対するコンピュータシミュレーションの可能性を大幅に進歩させるものであり、心臓血管の臨床における応用を可能にするものである。

## 参考文献

- [1] D.A. Vorp, D.A. Steinman, C.R. Ethier, *Comput. Sci. Eng.*, pp. 51 (2001).
- [2] A. Quarteroni, A. Veneziani, P. Zunino, *SIAM J. Num. Analysis*, 39, 1488 (2002).
- [3] L. Grinberg, T. Anor, E. Cheever, et al., *Phil. Trans. Royal Soc. A*, 367 1896 2371 (2009).
- [4] M. Bernaschi, S. Melchionna, S. Succi et al., *Comp.Phys. Comm.*, 180, 1495, (2009).
- [5] S. Melchionna, M. Bernaschi, S. Succi et al, *Comp.Phys. Comm.*, 181, 462, (2010).
- [6] S. Melchionna, *Macromol. Theory Sim.*, DOI: 10.1002/mats.201100012 (2011).
- [7] M. Bernaschi, M. Fatica, S. Melchionna, S. Succi and E. Kaxiras, *Concurrency and Computation: Practice and Experience*, DOI: 10.1002/cpe.1466 (2009).
- [8] M. Bisson, M. Bernaschi, S. Melchionna, *Commun. Comput. Phys.*, 10, 1071 (2011).
- [9] S. Melchionna, *J. Comput. Phys.* 230, 3966 (2011).
- [10] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*, Oxford University Press, USA (2001).
- [11] A. Peters et al., *Proceedings of Supercomputing 2010*, New Orleans, 2010.
- [12] <http://www.labri.fr/perso/pelegrin/scotch>