

生体と医療技術分野の大規模流体解析にむけて

小野謙二¹, 岩田 正子², 野田茂穂³

1 理化学研究所 ものづくり情報技術統合化研究プログラム

製品機能シミュレーションチーム チームリーダー

/ 北海道大学 大学院工学研究科 人間機械システムデザイン専攻 客員助教授

2 理化学研究所 ものづくり情報技術統合化研究プログラム 製品機能シミュレーションチーム

3 理化学研究所 情報基盤センター/ 信州大学 大学院

1. はじめに

近年、医療分野と工学分野の知識の融合により、人体における現象の把握、効果的な医療技術の開発、さらに新しい学術・技術分野の創出などを目指した研究活動が活発になっている。生体内における現象のとらえ方として、運動機能、形状、全身をめぐる血液の流れなどのマクロな視点、対極として細胞核内での遺伝情報の転写メカニズム、細胞内における反応・化学物質の輸送などのミクロな視点、さらにマイクロメートルオーダーのメゾ領域がある。現象面からみると、物質・熱輸送、生化学反応、構造応答などの個別の現象が複雑に組み合わせられ制御されている複合現象である。このようにスケールと多様な現象が絡まっている事象に対し、実現象の観察・実験と理論、さらにシミュレーションの3つのツールは、複雑な現象を解きほぐし、現象の理解とそこから得られる知識や技術に貢献している。特に、第3の方法としてシミュレーションには大きな期待が寄せられている。

流体や熱の移動現象は、生体の活動を支える重要な現象の一つである。中でも、血液の循環は細胞の活動に必要な多くの物質輸送に重要な役割を果たしている。血流に関連する問題を流体解析を用いて取り組んだ研究例は、数多く報告されている。しかし、血圧の発生メカニズムや血栓の形成過程など明らかになっていないことも多い[例えば、1]。また、超音波を用いて腎臓などに生成される結石を破碎したり、腫瘍を加温あるいは焼灼して治療する医療技術も流体応用の技術の一つである[2]。これらの例では、流動だけでなく、熱の移動現象、また組織や血管の構造的な応答も含めたモデル化を考慮しないと合理的な結果を得ることができない。このような解析は構造流体連成解析と呼ばれるが、汎用的な解法はなく対象とする問題分野毎に個別のプログラムを用いて解析しているのが現状である。

連成問題のプログラム開発は、それぞれの分野の現象および数値解法に精通しているだけでは難しく、総合的な知識が必要となる。今後、扱う問題が増え現象が細分化していくと、もはや一人だけのプログラム開発は困難になる。これは連成問題だけではなく、異なる時空間スケールの現象を扱うマルチスケール問題のプログラム開発においても同様の問題が発生する。研究者毎に自由なプログラムを開発して、インターフェイスだけを揃えるという開発スタイルでは効率が悪い。必然的に、専門領域の研究者グループによる協調的なシステム開発へと移行せざるを得ない。そこで、著者らは物理シミュレーションを対象としたプログラム開発のフレームワークを提案している[3]。これは、流体や構造、化学分野のシミュレーションプログラムの開発を効率的に支援する仕組みである。

形状の点から生体分野のシミュレーションの問題点を考えてみよう。我々が対象としている生体は非常に複雑な形状を有している。このような形状を対象に複雑なシミュレーションを実施するには、いかに形状の影響をシミュレーションに取り込むかという問題がある。現状では、物体の形状を三角形要素などで定義し、その表面要素に基づいて空間格子を生成する物体適合格子や非構造格子などの手法が主流である。これらの方法は確立した方法であり、適切に格子が作成できれば計算は可能である。しかしながら、形状が複雑になると格子が適切に生成される保証はなく、作成に多くの労力を要する問題点がある。この問題点は、生体のような複雑な形状をCTスキャンなどによりボリュームデータとして取得し、そこからメッシュを作成すると

いう過程で大きな困難を伴う。この問題点に対応する方法の一つは、ボクセルベースの解析手法の確立である[4]。これらの技術は、構造解析分野では、X-FEM[5]、被覆法[6]、流体解析分野では、カットセル[7]、VOF[8]、Immersed Boundary法[9]などが研究されている。

本稿では、生体と医療分野への適用を考慮した大規模なシミュレーションシステムを構築するための要素技術について、研究グループの研究内容を紹介する。

2. システムフレームワーク SPHERE

ソフトウェア開発の点から、シミュレーションプログラムを眺めてみると、従来、多くの研究者・エンジニアが物理シミュレーションのプログラムを開発し、利用してきた。それらのプログラムは、多くの人の手によって改良が加えられメンテナンスされているものも少なくない。これまでプログラム開発者は、入力・出力データ、プログラムの構成にいたるまで、それぞれのアルゴリズムを独自のやりかたでプログラミング・システム化してきたため、そのソフトウェアは本人以外理解が難しく、またユーザにとっても再利用が難しいという問題点があった。シミュレーションプログラムの利用価値を高めるためには、コードのポータビリティを始めとして、開発の効率化支援と共にメンテナンス性や利用環境なども考慮する必要がある。特に、大規模な解析となると並列化が必須となってくるが、並列コードの開発・デバッグは高度なスキルを必要とし、開発コスト・期間を引き上げる一因となっている。

SPHEREはこのような背景をもとに、並列化も含めシミュレーションプログラム開発の効率的な開発環境を提供すること、マルチフィジックス問題を扱うことも視野に入れ複数のコードの実行管理を行うことを目的として開発された。SPHEREは、ソフトウェア構造の統一化/標準化を意識し、様々な物理シミュレーションを包括的に扱うことを指向したシステムフレームワークである。

従来、提案するSPHEREのようなシステムフレームワークは、1990年代から米国の国研を中心に多くのライブラリ、ミドルウェアが提案されてきた。日本においても、90年代後半に大規模並列計算を意識したライブラリ実装の提言やミドルウェア開発が報告されている。これらは、いわゆるライブラリとミドルウェア的な利用を想定したフレームワークに大別される。詳細な例については、文献[10]を参照いただきたい。

SPHEREは、非定常物理シミュレーションを記述するフレームワークで、オブジェクト指向言語であるC++で記述されている。時間的に変化する物理現象の解析プログラムは、Fig.1のようにどれもほぼ同様の処理手続きで記述できる点に着目し、プログラムを前処理・メイン処理・後処理の3つに分類し、その制御構造を記述したひな形を基底クラスとして提供する。また、パラメータ設定、ファイル入出力、配列領域確保、変数宣言・初期化などの共通機能はクラスライブラリとしてまとめている。これらの検証済みの関数群を利用し、ソルバ開発者にとって本質的でないプログラミングを極力省くことにより、プログラムのコア部分の記述に集中することができる。また、記述するコードの量も低減できるので、開発の効率化が期待できる。実行環境としてはマルチプラットフォームで稼働し、ソースコードの高いポータビリティを提供する。現在の稼働環境として、IA32(Linux, Win2k/XP), PowerPC (MacOSX), Intel IA64(Linux), AMD Opteron (IA64互換, Linux), NEC SX series(SUPER-UX)での動作を確認している。

SPHEREは、XML形式で記述されたコンフィギュレーションファイルから、パラメータを読み取り設定する。XMLは標準的なファイルフォーマットなのでXMLを取り扱える多くのソフトウェア(データベースや他

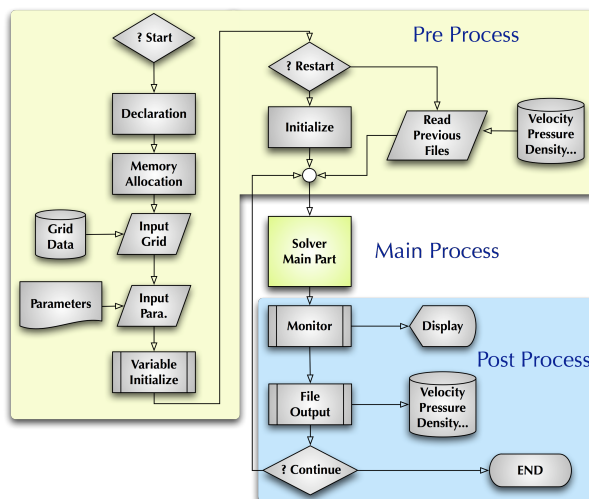


Fig. 1. Flow chart of a program and classification

のアプリケーション)でそれらを再利用できる利点があり、後に述べる問題解決環境としての重要な機能を担っている。XMLパーサーにはオープンソースライブラリであるlibxml2を利用して、XMLの構文解析を行っている。

SPHEREのプログラミングスタイルとしては、従来の手続き型言語を標準としている。もちろん、オブジェクト指向的なプログラムの構築も可能であり、最終的にはユーザのプログラミングスタイルに依存する。フレームワークがオブジェクト指向言語で記述されているにも関わらず、手続き型言語の記述スタイルを踏襲する理由は、物理シミュレーションのコードを書く場合、実効性能の高さからFortranが広く利用されてきた経緯、過去の資産の導入のしやすさ、オブジェクト指向プログラミングに対する要求スキルの高さなどの点を考慮してのことである。メイン関数はC++であるが、ここから呼ばれる関数として、C++の他にC、Fortran言語による関数・サブルーチンの利用が可能で、インターオペラビリティを考慮している。これにより、システムコールなどの細かな記述はC/C++で、実行効率が重要視される処理はFortranで記述することができ、言語の特性をうまく使い分けることができる。

並列化と連成問題の取り扱いは重要な機能の一つである。連成問題では、複数のソルバーを同時に起動・制御する必要があるが、SPHEREでは複数のソルバークラスをFig.2のようなイメージで登録している。プログラムの使用者は、実行時に実行させるソルバークラスを指定し、必要なパラメータを記述したXMLファイルを渡すことにより、任意のソルバーを実行することができる。連成時には複数のソルバーが実行されるが、各ソルバーの相互の情報をやり取りは、Fig.3のように弱連成形式と内部反復処理で取り扱う2つの方法を採用している。

SPHEREではデータクラスを導入して、これらのプログラミングを容易にしている。データクラスは、配列要素のメモリ確保や破棄、要素へのアクセスなどの配列要素に対する処理を司る独立したクラスである。領域分割に基づく並列処理では、並列化処理に必要な領域境界の同期処理と境界条件処理を引き受け、コード開発者からその煩雑な処理手続きの記述を隠蔽する。これにより、逐次プログラムから並列プログラムへの移行は、簡単なライブラリコールの追加のみにより記述可能になる。また、連成問題に必要なソルバークラス間のデータの相互参照に関わる複雑な処理も引き受ける。これは、異なるソルバークラスで使用するデータに対する相互参照の仕組みを組み込むことにより実現している。

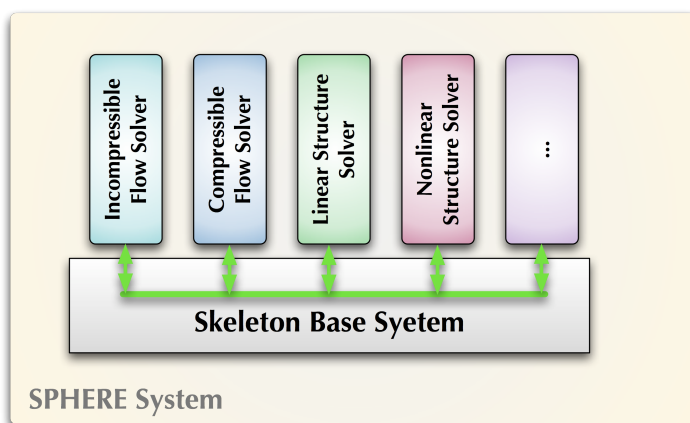


Fig. 2. Multiple solvers on SPHERE system

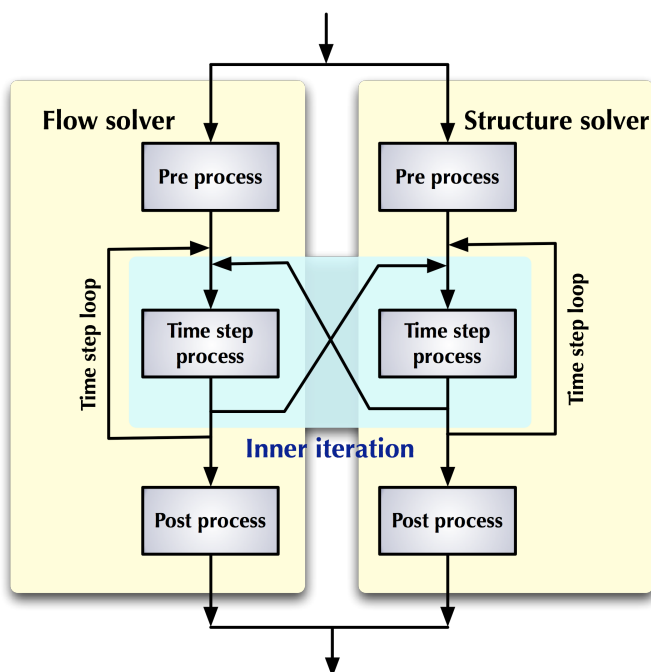


Fig. 3. Example of coupling procedure with inner iteration

3. シームレス・シミュレーション

シミュレーションでは、形状モデリング、メッシュ作成、シミュレーション、可視化などの一連のプロセスにおいて、複数のツールを併用することが多い。各プロセスの個々のアプリケーションは固有のデータ構造をもち、他アプリケーションとの連携には標準的なデータフォーマットを介して協調的に作業することが一般的である。ところが、内部データ構造と標準データフォーマットは必ずしも対応していない。このため異なるアプリケーションにデータを渡した場合に必要な情報が不足、あるいは、データの許容誤差の範囲で幾何形状のトポロジーが変化する問題が生じることがある。

シミュレーションを利用した設計においては、解析は一度だけではなく、解析結果を評価し形状を変更するという一連の処理を数回繰り返し、目的の機能や性能を達成していく場合がある。現状では解析の各処理における幾何形状の取り扱いが一貫しておらず、直接的に解析結果から形状へフィードバックをかけることができない。

上記の問題点を解決するため、複雑な幾何形状を表現するデータ構造と形状の操作方法を解析処理の各プロセスで共有するCAEフレームワークを構築している。また、そのフレームワーク上で有機的に動作するモデリング（プリ）、可視化（ポスト）アプリケーション、および格子生成の不要なボクセルベースの物理シミュレーションを開発中である。このプラットフォーム上では、解析の各アプリケーションは唯一の幾何形状データに対して共通の関数により形状を操作できるため、どの処理段階においても形状データを共有できる。つまり、解析サイクルの閉ループを構築することができる。また、解析結果と形状に対する変位を結びつける評価関数を介して、流体や構造の物理シミュレーションの中で直接形状変更を行うことにより、非常に複雑な形状に対する形状最適化の自動化への道が拓ける。我々は、この目的のため形状表現手法として、陰関数表現による物体形状の定義関数とその幾何形状の直接的な操作関数群をAPI(Application Programming Interface)として整備、構築している。

シームレスなシミュレーションを実現するにあたり、幾何形状データの取扱いは、そのシステムの中心を担う重要な要素技術である。著者らは、ボリューム表現を中心にFig. 4のようなシステム構築を目指している。この研究は、NEDOのプロジェクトとして実施している。

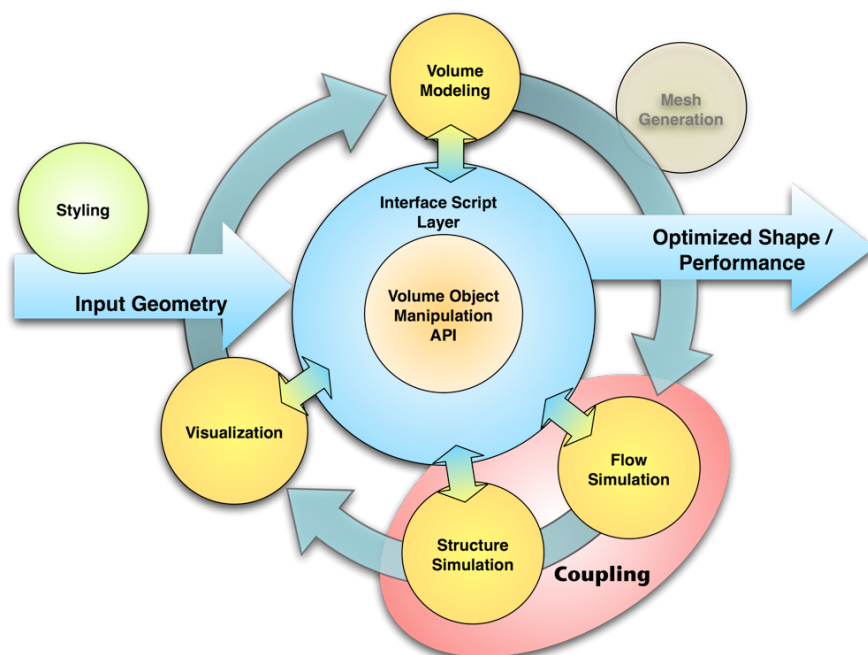


Fig. 4. Circulation of Processes on Physical Simulation

物体形状の表現方法には、大別して境界表現とボリューム表現がある。境界表現は物体の輪郭を面データで記述していく方法で、少ない情報量で形状を表現できる利点があるが、断面切断や穴開けなどのトポロジー変化を伴う形状操作には弱い。一方、ボリューム表現は、物体の形状だけでなく内部属性についてもデータを保持するためデータ量は増加するが、内部構造・属性・形状を記述できるためトポロジー変化に耐性があり、頑健なモデリングが可能となる。著者が所属するVCADプロジェクトにおいては、このボリュームベースのモデリングツールが開発されている。その基盤となっているのはKitta Cube と呼ばれる形状表現手法であるが、近年は陰関数による形状表現手法も導入されつつある[11]。

陰関数(Implicit Surface/Function)による形状表現方法は、近年コンピュータグラフィックスの分野で精力的に研究されている。その代表的な方法にRadial Basis Function(RBF)を用いて、点群やポリゴンデータの座標情報を拘束条件として連立一次方程式を解くことにより、スカラ情報で形状を表現する方法がある。また、連立一次方程式の解を求めやすくしたCompactly Supported RBF(CSRBF)などの応用も考案されたが、形状を少しでも変更すると大域的な連立一次方程式を解かねばならないため、計算時間がかかるという問題点がある。したがって、大規模なデータに対しては実用的な方法ではない。また、符号付き距離関数に基づくLevel Set法が幾何形状の処理に応用されているが、これもやはり微分方程式を解く必要があるため、モデル規模が大きくなると計算時間がかかる問題点が現れる。これに対して、局所的な形状関数とそれに付随する重み関数の組からグローバルな陰関数を生成する手法(Multilevel Partition of Unity, MPU)が大竹ら[12]から提案されている。この方法では、陰関数の生成時間が非常に短く、実用上の形状近似精度も十分であることが検証されている。また、形状の特徴や、誤差評価に基づく細分化、欠損データの修復機能などにも対応可能である。したがって、実用性を考慮し、本研究開発では、形状表現関数としてMPUをカーネルとして選択し、形状操作関数を開発整備している。ボリューム表現による形状の定義・作成・操作などの基盤技術はVCADプロジェクトのVCADモデリングで研究開発を行っている。

次の問題点は、形状データを流体シミュレーションへとどのようにつなぐかである。流体シミュレーションにおけるメッシュ生成の問題点は、形状表現の問題と関連して冒頭で述べた。ボクセルベースの流体解析手法は、従来の解析手法のように格子生成を行わず、スキームの中に形状近似と形状効果を反映する仕組みを導入している点が大きく異なる。形状近似のレベルとして、Fig. 5のようなブロック近似、ポロシティ近似、カットセル、境界埋め込み近似などの近似レベルに分類できる[13]。それぞれの形状表現レベルで形の影響を適切に組み入れる方法を採用している。

開発した陰関数のAPIは、所望の形状近似のスキームに必要な幾何情報を与えることができ、形状データからソルバー側への単一のインターフェイスとなっている。この形状表現関数との直接的なインターフェイスを介し、計算スキームの中でその情報を利用することにより、形状情報とのダイレクトカップリングを実現する[14]。これは、格子生成のステップを廃した流体解析の完全自動化に必須な要素技術である。

シームレスなシミュレーションの一例として、自然物であるザリガニの形状を取得し、そのデータを用いたシミュレーションの結果を示す。形状データはスライサーにて取得し、ボクセルデータに変換されている。Fig. 5はこのボクセルデータを流体ソルバーに渡し、自然対流場を計算した結果である。テスト計算のため粗いボクセルで近似しているが、ボリュームデータから直接計算できることが確認できた。

同様の計算プロセスは、医療画像に基づいた血流シミュレーションにも適用可能である[15]。Fig. 6はCTデータに基づいたシミュレーション結果である。この研究では、CTスキャンで得られたデータから表面データを作成せず、直接ソルバーに必要なデータを抽出している。このため、ボリュームデータから表面を抽出する場合に起こる問題を回避でき、高いロバスト性をもつ特徴がある。

4. pFTTデータ構造

ボクセルデータ構造として、単純な単一構造の直交格子、数種類の分割レベルの単一格子を階層構造にするもの(Multi-Level)、8分木(Octree)構造などがあり、後者の二つは局所的な細分化により形状近似度を

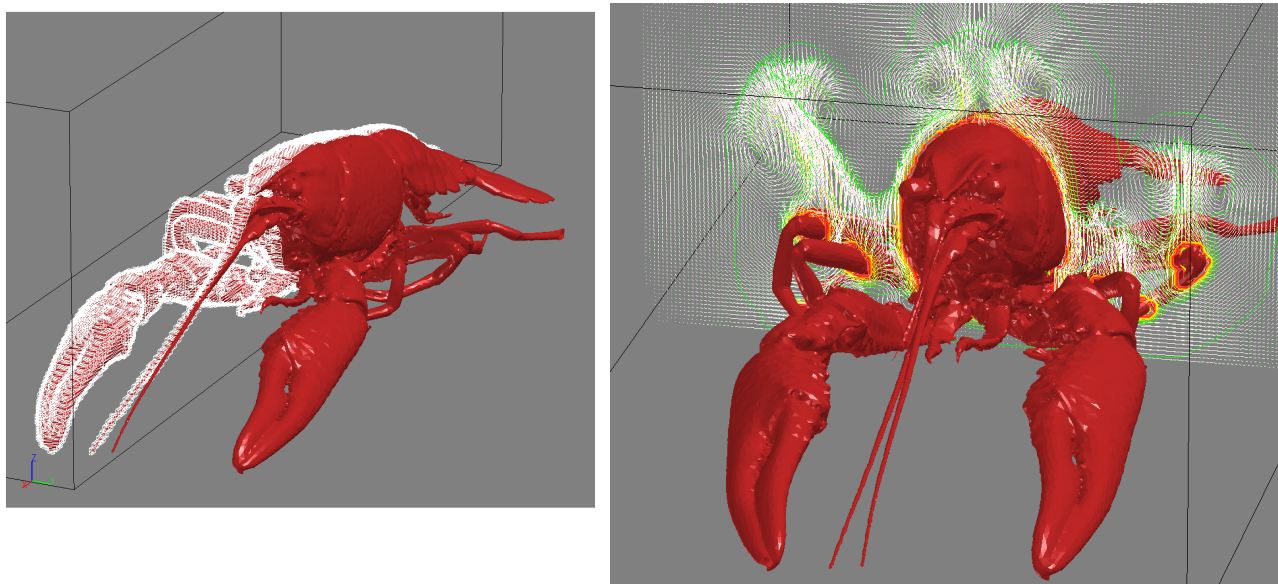


Fig. 5 Voxel data and computed vector field.

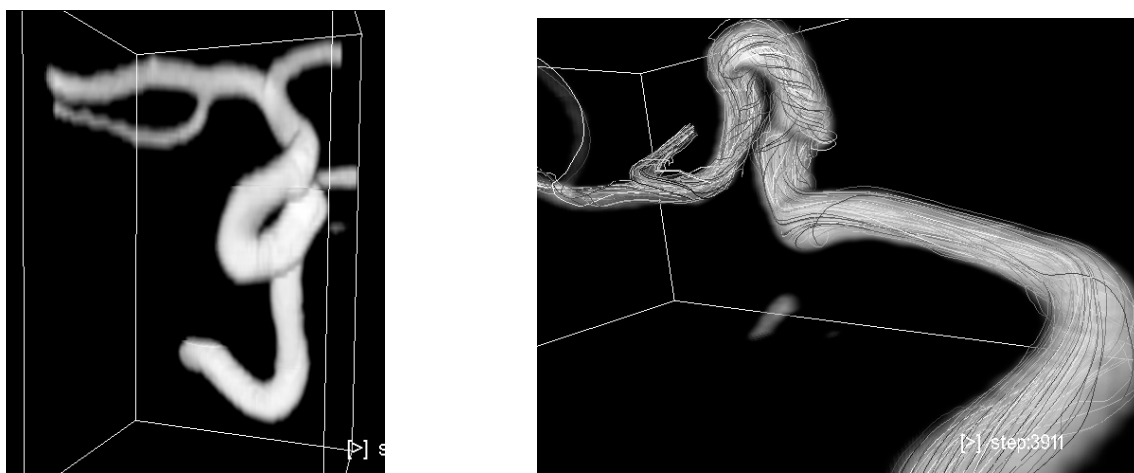


Fig. 6 VOF value rendered by volume rendering(left) and computed stream lines(right).

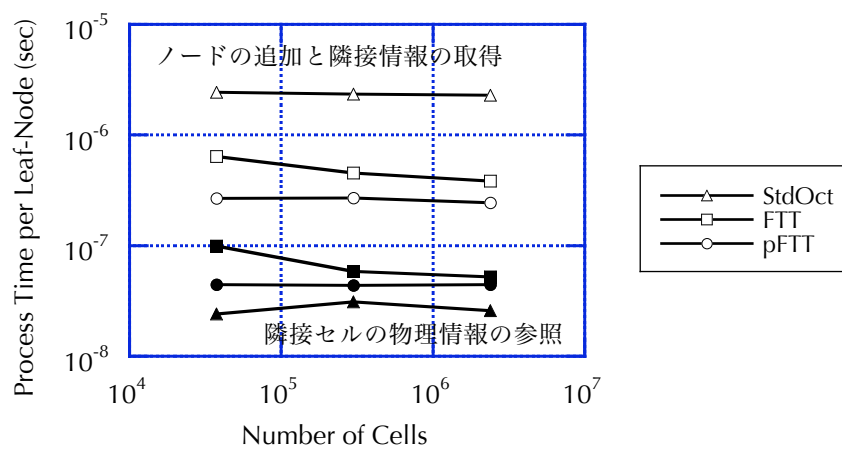


Fig. 7 Comparison of cost to get neighbour information.

向上できる．特に，計算結果に応じて局所的に格子密度を変化させるAdaptive Mesh Refinement は空間スケールオーダーが大きく異なる計算対象を効率的に扱うためには必須の要素技術となる．著者らの研究グループでは，メモリ使用量と木構造の操作時間のバランスに優れるFTT(Fully Threaded Tree)データ構造に系譜探索 (Pedigree search) を導入したデータ構造 (pFTT) を開発[16]し，これを利用しやすいAPIとして開発している．

Khokhlov[17]により提案されたFully Threaded Tree(FTT)は標準的なOctreeデータ構造に比べて使用メモリ量が少なく，ノードの追加削除の並列処理が可能である．FTTとpFTTは分割された8セルをまとめてノードの管理を行うため，1ノードあたりに必要なメモリ量は，表1のようにStdOctの44%となり効率的なことがわかる．KhokhlovのFTTの実装では32バイトであるが，我々の実装は効率面での改善を図るため40.5バイトであり，pFTTではPedigreeの変数を持っているが，FTTとほぼ同じメモリ使用量である．木構造の操作として，①ノードの追加と隣接情報の取得，②隣接セルの物理情報の参照（熱伝導方程式のラプラス演算子の離散化に必要な7点の物理量の参照時間の平均値），③SOR反復の3種類の時間コストを比較した．Fig. 5の白抜きの印は①，黒印は②を表す．StdOctは①のコストが大きく，②の参照コストが低い．pFTTは①のコストは小さく，動的にノードの追加削除が発生する場合には有利である．一方，②の時間はStdOctの約1.7倍程度かかっている．③の評価計算における一反復あたりの時間はセル数に依存せずほぼ一定で，StdOctで約 2.5×10^{-7} 秒，FTT，pFTTは共に約 4.3×10^{-7} 秒で，StdOctの約1.7倍であった．実際のアプリでは，反復回数は多くなるのでこの評価問題は，浮動小数点演算とロード/ストアの比から考えると，明らかにメモリバウンドであり，②の参照時間の影響が大きい．ループ中の演算数が多くなれば，StdOctとFTT/pFTTとの比は小さくなり，pFTTのメリットが出てくると考えられる．

Table 1 メモリ使用量（6物理量の場合，1セルあたり）

Implement	StdOct	FTT	pFTT	Khokhlov
Total	92	40.5	41.5	32
Overhead	68	16.5	17.5	8

6. まとめ

生体を対象にした流体計算は，連続相の空間スケールでさえ，大きなスケール差がある．また，複合した現象であるため，連成解析が必要となる．これらの要求に応えるには，既存の計算システムやスキームだけでは限界があり，形状表現やデータ構造，シミュレーションシステムまでを統合的に考えていく必要がある．このため，本稿で紹介したような要素技術を開発している．今後，開発したフレームワークの熟成，またこれを利用したソルバプログラムの開発を進めていく．また，多くのプログラム開発者に利用していただくため，理化学研究所からオープンソースでの配布を予定している．

謝辞

本研究の一部は，新エネルギー・産業技術総合開発機構(NEDO)の平成16年度産業技術研究助成事業の補助を受けています．

参考文献

- [1] 劉浩，岩瀬英仁，片岡則之，山本徳則，山口隆平，梶谷文彦，姫野龍太郎，イメージベースト血管モデルを用いた腎動脈分岐部の血流動態の解析、日本機械学会論文集（A編）、Vol.70, No.697, pp.1247-1253(2004).
- [2] 加藤 光章，吉澤 晋，小野 謙二，松本 洋一郎，“集束超音波医療における超音波伝播挙動の数値解析”，第19回数値流体力学シンポジウム講演要旨集，C8-2, p.186, (2005).

- [3] 小野 謙二, “SPHERE::物理シミュレーションプログラム開発のフレームワーク”, 計算工学, Vol. 10, No.4, pp. 5-10, (2005).
- [4] 小野 謙二, “設計における直交格子法の利用”, なかれ・日本流体力学会誌, 第21巻・第1号・(2002)16-25.
- [5] 長嶋利夫・大本洋平・谷周一: 拡張有限要素法 (X-FEM) による二次元熱応力解析, 日本機械学会論文集70-691A, pp.383-390 (2004).
- [6] 藤井、鈴木、大坪 「ボクセル有限要素法を用いた構造物の位相最適化」 日本計算工学会論文集 Vol. 2 pp 87-94 (2000).
- [7] Quirk, J.J., “An Alternative to Unstructured Grids for Computing Gas Dynamic Flows Around Arbitrarily Complex Two-Dimensional Bodies,” Computers Fluids, Vol. 23, No. 1, pp.125-142, (1994).
- [8] Hirt, C.W., et.al., “Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries,” J. Comput. Phys. **39**, pp. 201-225, (1981).
- [9] Peskin, C.S., et.al., “A Three-Dimensional Method for Blood Flow in the Heart I. Immersed Elastic Fibers in a Viscous Incompressible Fluid,” J. Comput. Phys. **81**, pp.372-405, (1989).
- [10] 小野謙二, 玉木剛, ” SPHERE -シミュレーションの開発・実行支援環境-, 電気学会, 静止器・回転機合同研究会資料, SA-06-14/RM-06-14, pp.73-78, (2006).
- [11] 加瀬 究, ”ボリウムCADの開発とVCADフレームワーク”, 理研シンポジウム (第5回) 予稿集, pp.50-64, (2004).
- [12] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H. P. Seidel : Multi-level Partition of Unity Implicits, In ACM TOG (Proc. SIGGRAPH 2003), **22**-3, pp. 463-470 (2003).
- [13] 小野 謙二, 白崎 実, 大竹 豊, “ボリウムデータを用いた流体解析システムの可能性”, シミュレーション, 第23巻第4号, pp. 286-291, (2004).
- [14] 白崎 実, 小野 謙二, 大竹 豊, “陰関数と直交格子ソルバー群のカップリング”, 第18回数値流体力学シンポジウム講演要旨集, p. 233, (2004).
- [15] 野田 茂穂, 姫野龍太郎, 小野 謙二, 石川 正昭, “医療画像に基づく脳血管血流シミュレーション“, 第16回バイオフロンティア講演会講演論文集, pp. 139-140, (2005).
- [16] 小野謙二, 岩田正子, 玉木剛, ” データ構造と実装の違いによるOctreeの性能評価”, ハイパフォーマンスコンピューティングと計算科学シンポジウムHPCS2006論文集, p.37, (2006).
- [17] Khokhlov, A. M., J. Comput. Phys. **143**, pp. 519-543, (1998).