論文の内容の要旨

論文題目　大規模数値流体力学のための節点処理型有限要素法

氏名　藤澤 智光

有限要素法は、変分原理などによる数学的な裏付けを有する上に、非構造格子を用いた連続体の離散化が可能であることから複雑な幾何形状への適応性が高く、工学的諸問題における構造解析や流体解析において非常に有用な解析手法である。しかし、その最大の欠点は、メッシュの生成（プリプロセッシング）にあり、有限要素法の解析そのもの（メインプロセッシング）よりも遥かに時間を要することも珍しくない。近年は、自動メッシュ生成技術も大きな進歩を遂げてはいるものの、その多くは逐次的な処理を行っており、並列処理が可能なメッシュ生成ソフトウェアでも、本質的には逐次的なアルゴリズムに依存しているため、使用できるプロセッサーの個数は十数個のレベルに留まっている。

一方で、流体のような強い非線形性を有する対象に数値的な解析を行うためには、一般に膨大な計算量を要する。数値流体力学の産業への応用という観点からは、とくに、乱流と流体音響の定量数値シミュレーションへの期待が大きいが、この両者の数値計算は微細な計算格子を要することから計算量が膨大なものとなる。このような大規模な数値計算には、多数のプロセッサーを同時に用いて計算を行う並列解析が不可欠となり、実際、現在多くの数値流体解析は、PC クラスターや並列型スーパーコンピュータなどの並列環境で行われている。

ただし、並列環境における力学計算では、連立方程式の組み立てや求解のみを並列処理し、計算格子の生成に関しては逐次的に処理する場合がほとんどであるのが現状である。しかしながら、移動境界問題、大変形を伴う問題、亀裂が進展する問題などにおいては、解析の途中に頻繁に計算格子の再生成を行う必要があり、計算格子の生成において逐次的な処理を行うと、その部分が解析全体の深刻なボトルネックとなってしまう。

そこで、本研究では、大規模な数値流体力学解析を超並列環境で高い効率で実現するための手法の開発を目的として、とくに、メッシュ生成を各節点の周囲で局所的に行うことによって、プリプロセッシングとメインプロセッシングの両者を節点を単位として、シームレスに並列処理することのできる節点処理型有限要素法の開発を行った。

本論文は 8 章から構成される。第 1 章では序論を述べ、第 2 章と第 3 章で、本研究で提案する節点処理型有限要素法とそれを用いた並列計算法について述べた。第 4 章と第 5 章では、非圧縮流体解析および圧縮性流体解析に本手法を適用して並列計算を行った結果を示した。第 6 章と第 7 章では、本手法を用いた応用解析事例について触れた。最後に第 8 章で結論と今後の展開について述べた。以下に、第 1 章から第 7 章までの概要をまとめる。

第 1 章では序論として、研究の背景、従来の研究と、本研究の目的を述べた。流体のように強い非線形性を持つ対象を数値解析するには、モデルを導入して統計的に解析するアプローチと、大規模な計算機とそのアーキテクチャを高効率に稼動するためのソフトウェアを開発するアプローチの 2 つが必要であることを述べ、本研究では、近年の大規模コンピューティングの標準プラットフォームである並列型コンピュータの上で、高効率な有限要素解析を実現する計算手法の開発を目標としていることを述べた。

第 2 章では、本研究で開発した節点処理型有限要素法の基本と特徴について述べた。節点処理型有限要素法とは、有限要素メッシュを各節点の周囲に局所的に生成した上で、有限要素計算を節点を単位として処理することにより、粒子法的な有限要素解析を実現するものである。すなわち、境界が移動する問題や解適合解析（アダプティブ解析）などにおいて、節点を適切な位置に配置するだけで、品質のよい局所メッシュを生成して、効果的な有限要素解析を実現できる。また、確率論を応用することによって、局所メッシュの生成に用いる節点を並列に生成する手法についても述べた。さらに、二段階負荷分散法によって、並列計算におけるプロセッサー間の通信を削減し、力学の並列計算における効率を向上させる方法についても述べた。すなわち、通信を要しない局所メッシュ生成においては、節点番号による機械的負荷分散によって並列に処理し、その後、生成された局所メッシュ情報をグラフ情報として利用してグラフ分割を行った結果に基づいて、節点番号のリナンバリングと計算負荷の再分配を行う。これによって、力学計算における通信量を削減して、効率的な並列計算が可能となる。

第 3 章では、節点処理型有限要素法の核となる技術である局所メッシュ生成手法について詳細に述べた。フリーメッシュ法における衛星節点の選択手法である対角線比較法、およびフリーメッシュ法を 3 次元形状に適用するための局所メッシュ生成手法である局所逐次添加法について検討し、複雑形状を取り扱う際の問題点や、メッシュ生成の効率の低下につながる検索半径に関する問題を明らかにした。その上で、検索半径を用いずに、最小範囲の節点検索で高速かつロバストに局所メッシュを生成する包装法と多階層型バケットを応用した方法について述べた。この方法は、流体解析で必要となる分布密度に大きな差がある節点配置に対しても高い効率で局所メッシュを生成することができる。また、衛星要素の登録を行うことにより、単一のプロセッサーを使用した場合でも、逐次添加法など、従来の有限要素法のためのメッシュ生成手法に比肩し得る処理速度を達成した。

第 4 章では、節点処理型有限要素法を非圧縮性流体解析に適用した例を示した。非圧縮性流体の数値解析スキームとしては、移流現象と拡散現象を分離して取り扱うフラクショナルステップ法を用いた。具体的には、流速場については陽的に取り扱い、圧力場に関しては陰的に解く。一般には取り扱いが困難である非圧縮条件を、圧力に関するポアソン方程式を解くことに帰着させることによって容易に取り扱うことができる。一般に連立方程式を反復解法（陰解法）で解く場合は、プロセッサー間で多くの通信を要するために、並列処理における効率を確保することが困難である。そこで、本章では、節点ベースの負荷

分散法を用いた場合の反復解法の並列処理の効率について検討を加えた。

第 5 章では、節点処理型有限要素法を圧縮性流体解析に適用した例を示した。圧縮性流体の数値解析スキームとしては、主に二段階 TG 法を用い、参考として SUPG 法による結果も併せて示した。高速圧縮性流れでは、しばしば、物理量が不連続となる衝撃波を含み、その捕捉には計算格子を非常に細かくとる必要があるが、発生する場所を予測することは困難であるため、計算結果を計算格子にフィードバックさせながら解析を進めるアダプティブ解析を用いることが非常に有用である。しかしながら、従来の差分法や有限体積法、有限要素法を用いた解析では、計算格子の生成において逐次的な処理を行っているものが大部分であり、並列環境でアダプティブ解析を行うことが困難である。本研究で開発した手法を用いることで、超並列環境においても効率的なアダプティブ有限要素解析が可能となることを示した。

第 6 章と第 7 章では、節点処理型有限要素法を用いた応用解析事例について述べた。第 6 章では、解析途中に頻繁なメッシュの再生成が求められる亀裂進展問題を取り上げた。この問題は、直接に流体解析に結びつくものではないが、提案した局所メッシュ生成手法が亀裂のような特異な形状に対しても特殊な処理を要せずに、通常のなめらかな幾何形状と同様に取り扱えることを示した。第 7 章では、流体解析の応用例として、エアリード楽器の音響解析を取り上げた。流体力学に基づいて流体音を数値シミュレーションする方法は、近年のコンピュータの発達によって現実味を帯びてきたものであり、産業界からの期待も非常に大きい。ただし、流体音のシミュレーションには膨大な数値計算が必要となることから、アダプティブ解析などによる効率的な数値計算が重要である。本章では、将来の超並列環境における流体音のアダプティブ解析を念頭に、その実現方法について基礎的な検討を加えた。

以上から得られる結論は以下の通りである。

(1)各節点の周囲に局所的な有限要素メッシュを生成した上で、節点単位に有限要素計算を行うことで、プリプロセッシングとメインプロセシングの両者を節点を単位としてシームレスに並列処理できることを示した。すなわち、従来、有限要素法による解析が不向きとされてきた移動境界問題、大変形を伴う問題、亀裂進展問題やアダプティブ解析などにおいて、粒子法的な有限要素アプローチが可能であることを示した。

(2)包装法と多階層型バケットを応用した局所メッシュ生成手法を開発することにより、流体解析において必要となる分布密度に大きな差のある節点配置でも、最小限の検索で高速かつロバストに局所メッシュを生成する方法を示した。このことは、従来の有限要素法のためのメッシュ生成という観点からも、節点を単位とした超並列メッシュ生成が可能であることを示したことになる。

(3)局所メッシュ生成の基本となる節点を確率論を応用して並列に生成する方法を示した。メッシュフリー法においては、点の配置が解析精度を決定するため、品質のよい点配置を入力として用意する必要があるが、これは一般には難しい。本研究では、品質のよい局所

メッシュを生成することのできる重心ボロノイ分割の生成点を確率論を応用することによって生成できることを示した。また、この方法を用いれば、節点生成を並列に行うことができ、CAD モデルから節点生成、メッシュ生成、有限要素法における連立方程式の構成、その求解までのすべてのプロセスを並列環境で処理することが可能であることを示した。