

Title	分離型解法による非圧縮性流体解析に適した並列アルゴリズムの開発
Author(s)	原田, 隆
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1008">http://hdl.handle.net/10119/1008</a>
Rights	
Description	Supervisor:松澤 照男, 情報科学研究科, 修士

# 分離型解法による非圧縮性流体解析に適した 並列アルゴリズムの開発

原田 隆

北陸先端科学技術大学院大学 情報科学研究科

1997年2月14日

キーワード: Finite Element Method, Incompressible viscous flows, Parallel Computing, Splitting Method.

現在において、より現実に近いモデルを解くために大規模な数値解析が求められている。これまでの大規模な流体解析は差分法解析で行なわれてきた。しかし、海岸や湖沼などの潮流現象や、気象などの大気の循環、飛行機まわりの流れの解析、建築物などの問題は一般的に境界の形状が複雑なものが多い。一方、有限要素法による流体解析は、任意の有限要素(三角形、四角形、四面体、六面体など)による形状近似や構造解析との整合性がよいなどの利点がある。しかし、数值的、安定性、解析精度、計算速度などの点から、課題が残されている。最近、超並列コンピュータの発達により、大規模解析、複雑な現象の解析などが可能になってきている。そのため、有限要素法解析による大規模な流体解析の並列化の研究が進められている。

かつて鎌形は、流れ関数渦度法を用いた有限要素法の並列化を超並列計算機CM-5上でSIMD型アルゴリズムで実現した。しかし、流れ関数渦度法を用いた解析法は3次元に拡張できない。さらに境界条件の取り扱いが流速と圧力を未知変数とする原始変数法の解析方法とくらべて、直接的ではなく複雑な与え方をするので最近の研究例が少ないのが現状である。

そこで、原始変数法のうち比較的簡単に高い精度が得られる分離型解法を用いた。分離型解法は時間段階 $\Delta t$ を1ステップ進めるためにさらに3段階の計算過程に分解して計算する方法である。さらにこの解法は、非線形である移流項と非圧縮の連続の式を分けて解くため、比較的数値が安定して求められるなど利点がある。しかし、大規模な連立1次方程式を3度解くことになるため、計算負荷が非常に高い。これを解消するために分離型解法に基づく有限要素解析法の並列化を行なう。基本的な有限要素解析法の並列化は鎌形が開発したSIMD型のアルゴリズムを踏襲する。

現在、並列計算機上での連立方程式の解法は、反復法やブロック化したガウスの消去法など数多く議論されていて、CM-5上でもその成果が計算ライブラリなどに搭載されている。しかし、有限要素解析法で作られる大規模行列は、非対称疎行列である。そのうえ、要素の数値の大部分が0なためにインデックスを用いて行列を小さくすることになり、そのままでは計算ライブラリが使えないことが多い。そのため M.Behr らや deněk Johan らによる有限要素法による非圧縮性粘性流体解析を並列計算機でおこなった研究ではSIMD型プログラミングで連立方程式の解法を作成しているのが現状である。連立方程式の解法である直接法や前処理つき反復法をSIMD型プログラミングで行おうとすると、どうしても逐次処理の部分が残ってしまい、思ったほど効率が上がらないことが多い。

そこで、超並列計算機CM-5のプログラミングモデルの一つであるグローバル/ローカル・プログラミングを採用する。

本研究では、分離型解法による有限要素法解析にグローバル/ローカル・プログラミングを検討することで有限要素法解析の高速化と効率化を試みて、その有効性を確認するのが目的である。

グローバル/ローカル・プログラミングとは、SIMD型プログラミング言語のグローバルプログラムをメインルーチンとして実行させ、メッセージパッシング等を使用し、ノードごとに別々の処理をするを実行するMIMD型ローカルプログラムをサブルーチンとして呼び出すことできる、ハイブリッドな並列プログラミングのことである。本研究では、有限要素法解析処理のうち、全体行列の重ねあわせと連立方程式の解法としてSOR法をグローバル/ローカル・プログラミングで行ない、SIMD型プログラムと比較検討した。

その結果、全体行列の重ね合わせの場合はSIMD型プログラミング、SOR法はグローバル/ローカル・プログラミングのほうがより高速な処理ができることがわかった。これによって、適切な処理をMIMD型ローカルプログラムにするグローバル/ローカル・プログラミングでより、一層高速な処理が可能になることがわかった。