日本流体力学会　年会2002
C123

並列非構造格子ソルバーの検証

Verification of Parallel Unstructured Grid CFD

○小泉哲平（東北大工），藤田健（東北大工），小寺正敏（航技研角田），

中橋和博（東北大工），岩宮敏幸（航技研），中村孝（航技研）

Teppei KOIZUMI*, Takeshi FUJITA*, Masatoshi KODERA**,
Kazuhiro NAKAHASHI*, Toshiyuki IWAMIYA*** and Takeshi NAKAMURA***
*Dept. of Aeronautics and Space Eng., Tohoku University, Sendai 980-8579, Japan
**Kakuda Space Propulsion Laboratory (NAL), Miyagi 981-1525, Japan
***National Aerospace Laboratory of Japan, Tokyo 182-8522, Japan

Development of unstructured grid generation makes it possible to generate the mesh for a very complex configuration in short time. But it takes so long time to solve NS equations for huge size computation containing over 10 millions nodes, that efficient and accurate parallel-unstructured CFD solvers have developed using Message Passing Interface (MPI) library. In the first instance, parallel computation in Overset Unstructured Grid Method has developed, which reduces computation time. Next, we reduce computation time and CPU load by parallel-unstructured CFD solver using domain decomposition method. The domain decomposition is based on METIS developed at the University of Minnesota. In this paper, verification for parallel-unstructured Euler/NS solvers is described.

1.　はじめに

非構造格子による数値計算は，複雑形状物体における格子生成や解適合格子生成の容易さから，これまで多く利用され，また移動物体周りの流れの計算に有効な非構造格子オーバーセット法に見られるように，その活用の場が広がっている．そして今日の大型計算機の飛躍的な発達にともない，3 次元複雑形状のような大規模計算に対する要求が強まる中で非構造格子への期待は非常に大きい．

しかし非構造格子法の問題点として構造格子法など他の手法に比べ計算メモリーを多く必要とする点が挙げられる．例えば，粘性流れを計算するような場合には，境界層を精度よく解析するため壁近傍を構造格子的なプリズム状の格子で構成し，その他の領域は四面体格子で構成するハイブリッド格子が必要となる．このように粘性流れの計算や 3 次元複雑形状の計算などにおいては，計算メモリーの増加や演算量の増大という問題が生じる．

ここ何年かの間に，並列計算機は非常に身近なものとなり，様々な高性能並列計算機が製品化され，価格性能比も飛躍的に向上してきた．また並列ライブラリも充実し，並列プログラムは実用化の時代に入ったといえる．並列処理には多くの利点があり，近い将来，日常的な計算手法になるであろう．並列処理の大きな利点として以下のことが挙げられる．まず CPU の数に比例して実行速度を向上できる点である．計算量を並列処理するという高速化手法は様々な場合に利用できて計算時間の短縮につながる．次に，一台ではメモリー容量不足で扱えない大規模計算を解くことができる点である．これは領域分割法と呼ばれ，解析対象となるいくつかの部分領域に分割し，各部分領域ごとに解析を行う省メモリーな計算手法として，DNS や三次元複雑物体の内外部流などの流れ場など非常に大きな格子点数を用いる数値シミュレーションを可能にすることができる．以上の点から並列処理を非構造格子ソルバーに施すことは非常に有効であると考えられる．

本研究では複雑形状物体の格子生成に有利な非構造格子用の計算ソルバーを並列化することを目的とする．今回は非構造格子オーバーセット法の計算を並列化すること，また計算領域を分割した 3 次元矩形翼の粘性計算の並列化を行うことでその有効性を検証する．

2.　非構造格子オーバーセット法の並列化

2. 1　並列手法

オーバーセット法 [1]は移動する物体を子格子，一様流中の準定常物体を親格子で生成し，それらを重ね合わせる手法であるが，内挿境界面を自動で検索する際に相手格子の情報が必要なためメモリーを並列化により省略することはできない．しかし単一のソルバーを同一ステップで親格子と子格子で相互に計算するため，この計算ルーチン部分を並列化することにより計算時間を最大で半分程度短縮することができる．

今回はヘリコプター全機の流れの計算により計算速度の向上を検証する．ヘリコプターの形状は，胴体の格子と回転するローターの格子をそれぞれ生成し，重ね合わせて定義した．格子点数は，胴体格子とローター格子それぞれ 161,128 と 168,821 である．

計算手法

ヘリコプター全機の流れの計算における支配方程式は 3 次元 Euler 方程式 [2]とし，時間積分には LU-SGS 陰解法，流束の計算には HLLEW 法を用いた．並列化した部分は流束計算の部分と内挿点を検索する部分である．

2. 2　計算結果

計算は 2CPU を搭載した alpha マシンを myrinet で接続したクラスターマシンで行った．ローターが一回転するのに要する CPU 時間を並列化した場合としない場合で比較した．Table.1 にその結果を示す．CPU 時間で約 40%の短縮に成功し

た．Fig.1 に等圧線図および表面圧力分布を示す．

Table.1 1CPU と 2CPU の時間比較

	1CPU	2CPU
ユーザータイム	59325.80s	35058.27s
リアルタイム	16h31m	9h45m

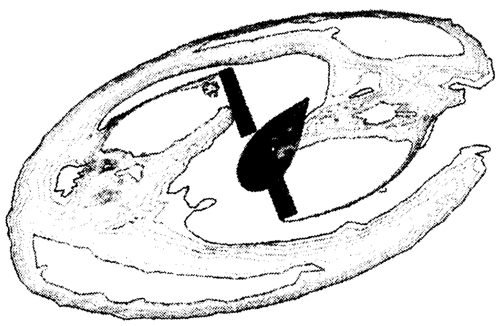

Fig.1 ヘリコプターの等圧線図

3. 領域分割法による並列化

3.1 領域分割法

非構造格子の分割プログラムは，ミネソタ大学が提供するフリーウェア MeTiS[3]をベースとしたものを用いた．MeTiS により分割された領域の境界に情報交換のためのセル一層分の境界を定義し，この境界層で情報の受け渡しを行う．この手法により 3 次元矩形翼の領域分割を行った．翼表面を分割した様子を Fig.2 に示す．

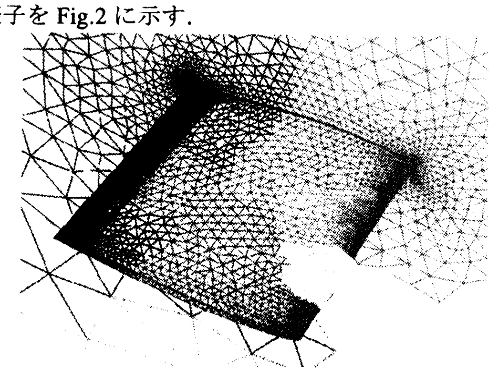

Fig.2 翼面分割図

3.2 計算手法

計算は 2CPU を搭載した alpha マシンを myrinet で接続したクラスターマシンで行った．今回は粘性計算を可能にするため，既存の四面体格子対応の MeTiS を四面体要素，プリズム要素およびピラミッド要素で計算場を離散化するハイブリッド格子の領域分割法へ拡張した．支配方程式は，3 次元 Navier-Stokes 方程式とし，時間積分には LU-SGS 陰解法，流束の計算には HLLEW 法を用いた．レイノルズ数は，5.0×10^5 とした．

3.3 計算結果

今回は分割数を 8 として計算した．領域分割プログラムを利用して空間格子を 8 分割すると，1CPU あたりの平均要素数が 119513 になる．CPU にかかる負荷は辺要素数に依存するが，分割していない空間格子の辺要素数が 896474 であることから，1CPU にかかる負荷がおよそ 8 分の 1 になっている．また，1000 ステップでの計算時間を比較すると，並列化した場合は 1CPU あたりの CPU タイムは最大で約 1 時間であり，並列化していない場合は約 8 時間であることから，8 分の 1 程度まで短縮できた．Fig.3 にそれぞれの表面圧力分布を示す．Fig.3 では，それぞれについての 44%半スパン長の Cp 分布を比較する．結果は完全に一致していることが確認できる．また粘性力を考慮した空力係数，抵抗係数もともに一致しており，粘性計算における精度が十分であることが示された．

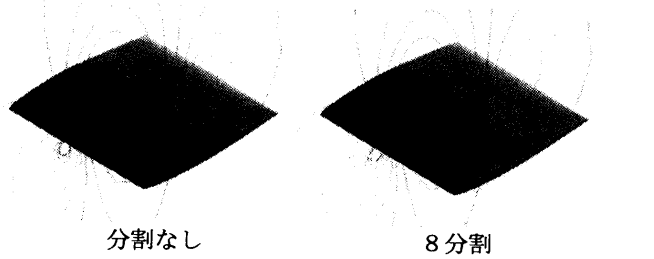

Fig.3 圧力分布図の比較

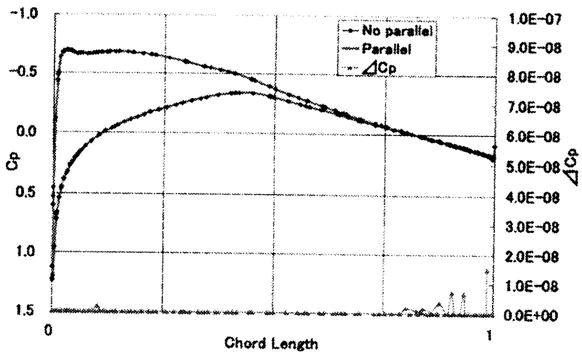

Fig.4 C p 分布の比較

4. 結論および今後の課題

非構造格子オーバーセット法の並列化に関しては，計算時間の短縮という点でその有効性が確認できた．問題点は親格子と子格子の格子点数に大きな差がある場合にそれ程大きな時間短縮はできないことである．これは，全体の計算時間は格子点数の多い格子での計算時間に依存するためで，この点を解決することが今後の課題である．

領域分割法による並列化に関しては，それぞれの CPU に対する負荷の低減という点で領域分割法が非常に有効であり，また並列計算することで計算時間も大幅に短縮できるということを確認した．また精度に関しても，並列化しても結果が変化しないことを確認でき，本手法が非常に有効であることが示された．

引用文献

1) K. Nakahashi, F. Togashi & D. Sharov: *AIAA Journal*, Vol.38, No.11 (2000) 2077-2084.

2) D. Sharov & K. Nakahashi: *AIAA Journal*, Vol.36, No.3 (1988) 484-486.

3) http://www-users.cs.umn.edu/~karupis/

4) Y. Ito & K. Nakahashi: *AIAA Paper* (2000) 2000-0924.

5) D. Sharov & K. Nakahashi: *5th International Conference on Numerical Grid Generation in Computational Field Simulation* (1996) 229-238.

6) G. Karypis & V. Kummar: *Technical Report* 95-035, University of Minnesota, 1995. A short version appears in Intl. Conf on Parallel Processing (1995).