(I −23)　非構造格子に基づく非圧縮粘性流れの並列計算手法の構築

中央大学　学員　○弓削　武慎
中央大学大学院　学員　玉井　典
中央大学　正員　樫山　和男

1. はじめに

一般に流体解析は構造解析と比較して圧倒的に自由度の多い大規模計算となる。このため膨大な計算時間と計算機容量を必要としている。

本報告はこれらの問題を解決するために、非構造格子に基づく非圧縮粘性流れに対する並列計算法を構築、その有効性について検討を行ったものである。

2. 基礎方程式と有限要素方程式

基礎方程式には、(1) 式のナビエ・ストークスの運動方程式と、(2) 式のオイラーの連続方程式を用いる。

$$\frac{\partial u_i}{\partial t} + u_j u_{i,j} = -p_{,i} + \frac{1}{Re} u_{i,jj} \quad (1)$$

$$u_{i,i} = 0 \quad (2)$$

ここで、$_{,i}$ は i方向の偏微分、u_i は i方向の流速、pは圧力、Reはレイノルズ数である。

解析手法として、MAC 法のアルゴリズムを用いるものとし、(1)、(2) 式に対し三角形一次要素を用いたガラーキン法により空間方向の離散化を行うと、以下の有限要素方程式が求まる。

$$A_{\alpha i \beta i} p_\beta^{n+1} = -\frac{1}{\Delta t} H_{\alpha\beta i} u_{\beta i}^n - K_{\alpha\beta\gamma j} u_{\beta j}^n u_{\gamma i}^n \quad (3)$$

$$\bar{M}_{\alpha\beta} u_{\beta i}^{n+1} = \bar{M}_{\alpha\beta} u_{\beta i}^n - \Delta t (K_{\alpha\beta\gamma j} u_{\gamma i}^n + H_{\alpha\beta i} p_\beta^{n+1} + \frac{1}{Re} S_{\alpha j \beta j} u_{\beta i}^n) \quad (4)$$

上記の有限要素式を解く際に、流速に関しては質量行列 M を集中化して陽的に、圧力に関しては EBESCG(Element by Element SCG) 法を用いて陰的に解くものとした [1]。EBESCG 法は近似条件を満たすまで繰り返し計算を行なうが、並列化が容易なアルゴリズムであるため、計算時間の短縮が期待できる。

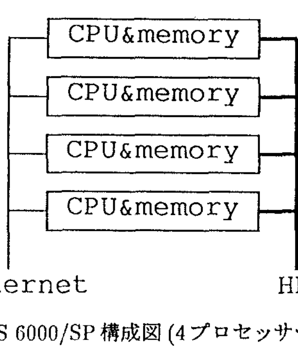

図 1 RS 6000/SP 構成図 (4プロセッサーの場合)

3. 並列計算

並列計算機として IBM RS6000/SP(SP2) を用いた。RS6000/SP は分散メモリー型並列計算機であり、単体で 300 MFLOPS の性能を持つ (図 1 参照)。各プロセッサー間は HPS（ハイパフォーマンススイッチ）と呼ばれる高速ネットワークで結ばれている。HPS は 40MB/sec と Ethernet の 32 倍のデータ転送能力を持ち、通信時間の短縮を図ることができる。この結果、並列化による計算時間の短縮分を通信時間で相殺してしまうことが少ないものとなる。

本報告における並列計算手法を示すと以下のようになる (図 2 参照)。

1) 非構造格子に対する領域分割を行ない、小領域データと小領域境界上節点データを得る。
2) 各プロセッサーはデータを割り当てられた小領域の分だけ読み込む。
3) 各プロセッサーは割り当てられた小領域について圧力、流速を求める。
4) この時、隣接する小領域を受け持つプロセッサー間のネイバリング通信、及び全プロセッサー間でのグローバル通信を行なう [1,2]。
5) 時間ステップが終了するまで、3),4) を繰り返す。
6) 各プロセッサーの結果を親プロセッサーにまとめ、出力する。

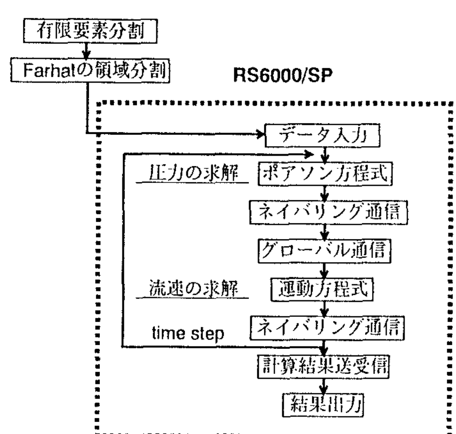

図 2 並列計算の流れ

小領域への分割は Farhat により提案された領域分割法を用いて行なう [3]。これにより任意の有限要素メッシュが小領域に分割され、かつ各小領域の要素数、領域境界の節点数がほぼ同数となる。このことから各プロセッサー

間の負荷の均等化とプロセッサー間の通信量を最小とすることが可能となり、効率的な並列計算を行なうことができる。

4. 数値解析結果

数値解析例として、円柱周りの流れ解析を行うものとした。節点数の異なる3種類のメッシュ、mesh-S(節点数nx=11904),M(nx=23684),L(nx=55540)に対して、それぞれ1、2、4、8の小領域に領域分割して計算を行なった。mesh-Lを8分割した時の領域分割図を図3に示す。なお、レイノルズ数は100とした。

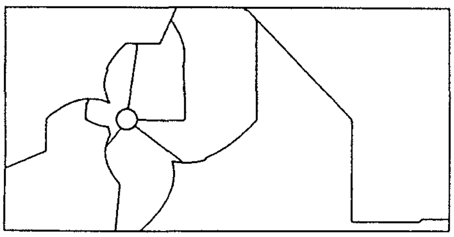

図3 領域分割図 (mesh-L, 分割数=8)

性能評価として、次式で定義される演算速度倍率と並列化効率を用いる。比較は、I/O部分を除いた時間刻み1ステップにかかる計算時間で行なった。

$$\text{演算速度倍率} = \frac{\text{プロセッサー数1の時の計算時間}}{\text{プロセッサー数Nの時の計算時間}} \quad (5)$$

$$\text{並列化効率} = \frac{\text{演算速度倍率}}{\text{プロセッサー数N}} \times 100(\%) \quad (6)$$

図4、図5にそれぞれのメッシュに対して、プロセッサー数を変化させた時の演算速度倍率、並列化効率を示す。mesh-M,Lについてはプロセッサー数の増加に伴い計算時間の短縮が確認できる。このことから、大規模計算になるほど、本手法は有効であるといえる。特に、mesh-Lについては理想倍率に近い演算速度倍率が得られており、より節点数の多いメッシュを用いた場合には、より理想倍率に近いものとなることが期待できる。また、速度向上率が低下し始めるmesh-Lの4プロセッサー時、mesh-Mの2プロセッサー時の節点数が12000〜14000であることから、RS6000/SPでは1プロセッサーあたり、約15000節点以上の場合、十分な並列化効率を得られるものと考えられる。

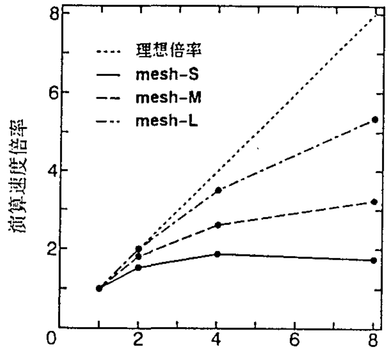

図4 演算速度倍率

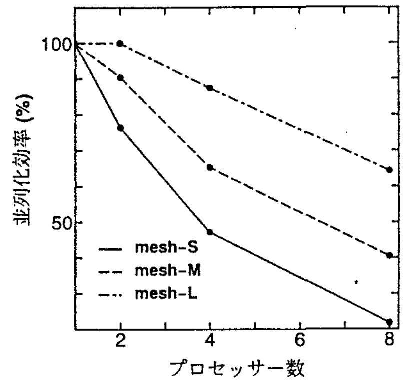

図5 並列化効率

表 1: 通信時間の割合

	PE=2	PE=4	PE=8
mesh-S	36.4%	53.9%	77.5%
mesh-M	14.5%	37.5%	62.4%
mesh-L	7.8%	21.7%	32.9%

次に表1として計算時間に占めるCPU時間と通信時間に内訳を示す。これより並列化効率の低い計算では、計算時間の多くを通信時間が占めていることがわかる。並列化効率を高めるには通信時間の割合を抑えることが重要である。

5. おわりに

本報告では、非構造格子に基づく非圧縮粘性流れの並列計算手法を構築しその有効性を検討した結果、以下のことが明らかとなった。

1) 節点数が多いほど高い計算時間短縮効果が得られ、大規模計算における並列計算の有効性が確認できた。

2) 陰的解法のEBESCG法は計算機容量を節約できる上、並列化プログラミングが容易であり、大規模並列計算に適した解法であるといえる。

今後は、三次元解析での並列計算の有効性を検討する所存である。

参考文献

[1] 玉井　典, 樫山和男:第 23 回関東支部技術研究発表会講演概要集, pp26-27(1996)[2]Kazuo Kashiyama et.al.Int,J.Num.Meth.Fluids,vol.21,pp885-900(1995)[3] Farhat,C.,Computers&Structures,28,pp576-602 (1988) [4] 青山幸也 (日本ＩＢＭ):RS 6000/SP 並列プログラミング虎の巻 (1996)