

並列マシン Cenju2 による非圧縮性流体解析

村松 一弘, 鷲尾 巧, 土肥 俊

NEC C&C 研究所 コンピュータ・システム研究部

非圧縮 Navier-Stokes 方程式解法の一つである SMAC 法を、領域分割法を用いて並列マシン Cenju2 上に実現する。また圧力の離散 Poisson 方程式求解のために数種の並列線形ソルバをインプリメントし、それらの比較を行う。比較した線形ソルバとしては、Block(B)-MILU 前処理つき Bi-CGSTAB 法、red/black(RB)-ILU 前処理つき Bi-CGSTAB 法、Scaling 前処理つき Bi-CGSTAB 法、red/black(RB)-SOR 法の 4 種類である。例題として、2 次元正方キャビティ内流れを用いる。比較した全てのケースにおいて、B-MILU 前処理つき Bi-CGSTAB 法を用いた計算が最も実行時間が短く、RB-SOR 法と比べた場合、その $\frac{1}{4}$ の実行時間である。またプロセッサ台数の増加に対する速度向上率に関しては、格子点が 216×216 の問題に対して、16 台で 15 倍の向上率を獲得する。

NUMERICAL STUDY OF INCOMPRESSIBLE FLUID DYNAMICS WITH CENJU2 PARALLEL MACHINE

Kazuhiro Muramatsu, Takumi Washio and Shun Doi

Computer System Research Laboratory, C&C Research Laboratories, NEC Corporation

4-1-1, Miyazaki, Miyamae-ku, Kawasaki-shi, Kanagawa, 216 Japan

The SMAC method which is for solving an incompressible Navier-Stokes equations is implemented on Cenju2 parallel machine with a domain decomposition method. Several parallel linear solvers for the Poisson equations for pressure are implemented. The linear solvers we adopted are the Bi-CGSTAB method with the Block(B)-MILU, red/black(RB)-ILU and Scaling preconditioners, and red/black(RB)-SOR method. Incompressible driven square cavity flow, which is a standard two-dimensional flow model, is used for evaluation. It becomes clear that the Bi-CGSTAB method with the B-MILU preconditioner is most efficient in the four linear solvers and its execution time is 4 times as short as the RB-SOR method. It is also shown that, for 16PEs, speedup factor is about 15 when mesh size is 216×216 .

1 はじめに

近年、数値計算の様々な分野で計算機の高速化が求められている。流体解析の分野でも、乱流の直接シミュレーションのような詳細な流れ場のシミュレーションが求められており、スーパーコンを用いても莫大な時間を要する。このような要求に応えるマシンとして、並列計算機が注目されつつある。これは複数の演算素子が通信を行いつつ、並行して計算を行うものである。従来の逐次計算機に比較し、並列計算機は、演算素子の増加による計算能力の高速化が可能であるという点で、プログラムが複雑になるという欠点を補ってあまりある能力を有している。

本報告では、当社の並列マシン Cenju2 を用いて、2次元流れの標準的な検定問題である正方キャビティ内流れに対する数値解析を行う。特に、圧力の Poisson 方程式を解く際の連立一次方程式の種々の解法を比較することによって、並列マシンで有効な解法を調べる。また実行時間における通信時間の割合を求めることによって、流体解析における Cenju2 の通信性能を調べることも行う。

解析における2次元流れの解法としては、標準的な SMAC (Simplified-Marker-and-Cell) 法を用い、移流項の風上差分には3次精度の K-K(Kawamura-Kuwahara) スキームを採用する [1]。圧力の Poisson 方程式を解く際の連立一次方程式の解法には、グスタフソン加速を伴う Block(B)-Modified ILU(MILU) 前処理つき Bi-CGSTAB 法 [2],[3]、red/black(RB)-ILU 前処理つき Bi-CGSTAB 法、Scaling 前処理つき Bi-CGSTAB 法、red/black(RB)-SOR 法の4種類を採用する [4]。

評価の結果、連立一次方程式の解法としては、グスタフソン加速を伴う B-MILU 前処理つき Bi-CGSTAB 法が優れていることが明らかになる。通信性能に関しては、全計算メッシュ数が大きい場合および使用プロセッサ台数が少ない場合には十分な速度向上率が得られる。例えば、計算メッシュ数 216×216 で16台のプロセッサを用いる場合、15倍程度の速度向上率が得られる。

2 Cenju2

Cenju2 は VR3000 を要素プロセッサとし、これらを多段接続網で結合した並列マシンである。

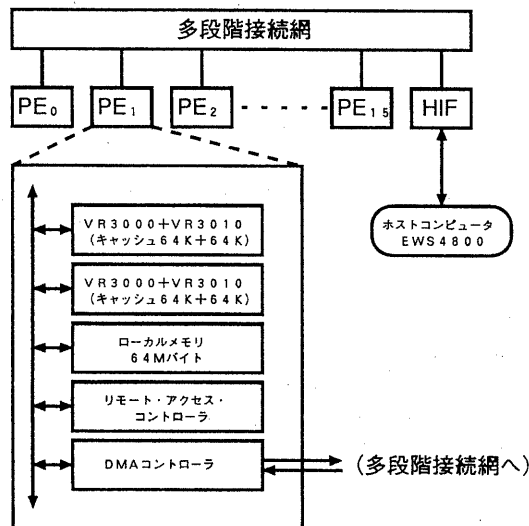


図1: Cenju2 システム構成図

要素プロセッサ (PE) には、高い実効性能が実証されている VR3000(25MHz) を搭載しており、64M バイトのローカルメモリを装備している。(図 1 参照)

ネットワークは図 2 のように 4×4 のクロスバースイッチを用いた多段パケット交換網で、通信距離の点で有利な構成になっている [5]。

プログラムは、FORTRAN 上で付加された並列ライブラリ関数を用いて並列化を行う。用いた並列ライブラリ関数は、複数の PE に同じメモリイメージをコピーし、並列実行を開始する FORK (cjsforknf)、すべての PE 間で同期をとるバリア同期 (cjsysbarrier)、PE 間で相互のメモリに書き込むデータ転送 (cjrmdawr) の 3 つである。

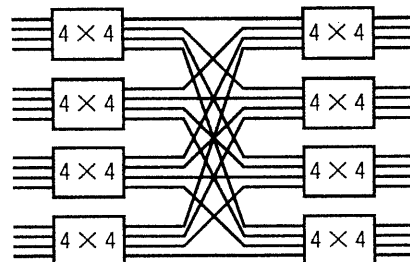


図 2: 多段階接続網図

3 基礎方程式と SMAC 法

流体の速度を $\mathbf{v} = (u, v)$ 、圧力を p 、レイノルズ数を Re とすると、2次元非圧縮性流体の基礎方程式は、連続の式

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (1)$$

および Navier-Stokes 方程式

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (2a)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (2b)$$

である。ここでは、簡単化のために (2) 式を以下のようにおく。

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{v}) - \frac{\partial p}{\partial x}, \quad \frac{\partial \mathbf{v}}{\partial t} = g(\mathbf{v}) - \frac{\partial p}{\partial y} \quad (3)$$

$$f(\mathbf{v}) = \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y} \quad (4a)$$

$$g(\mathbf{v}) = \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y} \quad (4b)$$

連続の式、Navier-Stokes 方程式の時間に関する離散化は、次のように行う。

$$\frac{\partial u^n}{\partial x} + \frac{\partial v^n}{\partial y} = 0 \quad (5)$$

$$u^n = u^{n-1} + \Delta t \left\{ f(\mathbf{v}^{n-1}) - \frac{\partial p^n}{\partial x} \right\} \quad (6a)$$

$$v^n = v^{n-1} + \Delta t \left\{ g(\mathbf{v}^{n-1}) - \frac{\partial p^n}{\partial y} \right\} \quad (6b)$$

ここで Δt は時間間隔、上添字は離散化された時間ステップ n を表している。SMAC 法では、(6) 式を速度の近似解 \mathbf{v}^* を用いて、2 段階に分割する [6],[7]。

$$u^* = u^{n-1} + \Delta t \left\{ f(\mathbf{v}^{n-1}) - \frac{\partial p^{n-1}}{\partial x} \right\} \quad (7a)$$

$$v^* = v^{n-1} + \Delta t \left\{ g(\mathbf{v}^{n-1}) - \frac{\partial p^{n-1}}{\partial y} \right\} \quad (7b)$$

$$u^n = u^* - \Delta t \cdot \frac{\partial \delta p}{\partial x} \quad (8a)$$

$$v^n = v^* - \Delta t \cdot \frac{\partial \delta p}{\partial y} \quad (8b)$$

ここで $\delta p = p^n - p^{n-1}$ である。(8) 式の発散をとり、(5) 式を用いて δp に関するポアソン方程式を得る。

$$\frac{\partial^2 \delta p}{\partial x^2} + \frac{\partial^2 \delta p}{\partial y^2} = \frac{1}{\Delta t} \left\{ \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right\} \quad (9)$$

SMAC 法では一般に速度成分と圧力が図 3 に示すように異なる点で定義される、いわゆるスタッガード格子を採用する。空間離散化に関しては、このスタッガード格子を用いて行う。

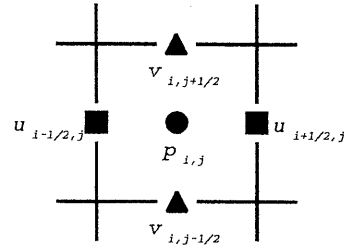


図 3: スタッガード格子

4 並列計算手法

解析対象としたモデルは、図 4 のような 2 次元正方キャビティ内流れである。並列計算では、この領域を各方向の格子数を等分に分割して各 PE に割り当てるが(図 5 参照)、スタッガード格子を用いているため、分割した領域の境界上に速度成分を含む。それゆえこの速度成分に対しては特別な配慮を必要とする。

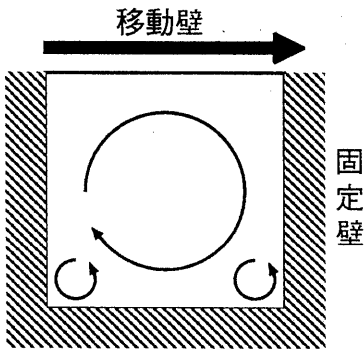


図 4: 2次元正方キャビティ内流れ

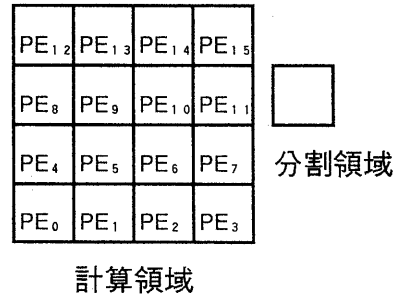


図 5: 領域分割

図6は、SMAC法における各PE間のデータの通信まで考慮したアルゴリズムである。

圧力のPoisson方程式の解法には、グスタフソン加速を伴うB-MILU前処理つきBi-CGSTAB法、RB-ILU前処理つきBi-CGSTAB法、Scaling前処理つきBi-CGSTAB法、RB-SOR法の4種類を用いる[2]～[4]。

(9)式を5点差分で離散化し、格子点を標準的な順序で並べて得られる連立一次方程式を形式的に

$$Ax = b \quad (10)$$

と書く。前処理つきBi-CGSTAB法とは、(10)式に等価な方程式

$$AM^{-1}(Mx) = b \quad (11)$$

に対してBi-CGSTAB法を適用するものである。ここで、 M は A の近似行列(前処理行列と呼ぶ)である。

B-MILU前処理では、 A をMILU分解する際に A のPE間のインタフェース部の要素を無視するものであり、前処理行列 M は

$$M_{B-MILU}(\alpha) = (L_B + D_B(\alpha))D_B(\alpha)^{-1}(D_B(\alpha) + U_B) \quad (12)$$

と表される。ここで L_B 、 U_B はそれぞれ A のインターフェース部の要素を0とおいた狭義下三角行列、狭義上三角行列であり、 $D_B(\alpha)$ は A の対角行列を D として、次式で決定される対角行列である。

$$D = D_B(\alpha) + \text{diag}(L_B D_B(\alpha)^{-1} U_B) + \alpha \text{ rowsum}(\text{off-diag}(L_B D_B(\alpha)^{-1} U_B)) \quad (13)$$

‘rowsum’は同じ列の和を対角成分とした対角行列を意味し、 α ($0 \leq \alpha \leq 1$)は、加速係数を意味する。

RB-ILU前処理では、RBオーダリングに従って係数行列を再構成し、これを \bar{A} とする。そして前処理行列 M を

$$M_{RB-ILU} = (\bar{L} + \bar{D})\bar{D}^{-1}(\bar{D} + \bar{U}) \quad (14)$$

と表す。係数行列 \bar{A} のILU分解、および M に関する前進後退代入においては、2次元格子点 (i, j) を2つの部分集合

$$\begin{cases} S_R = \{(i, j) | \text{mod}(i+j, 2) = 0\} \\ S_B = \{(i, j) | \text{mod}(i+j, 2) = 1\} \end{cases} \quad (15)$$

に分類すると、この部分集合ごとに格子点での計算が独立にできるので、並列化することが可能になるのである。

Scaling前処理では、前処理行列 M を

$$M_{\text{Scaling}} = D \quad (16)$$

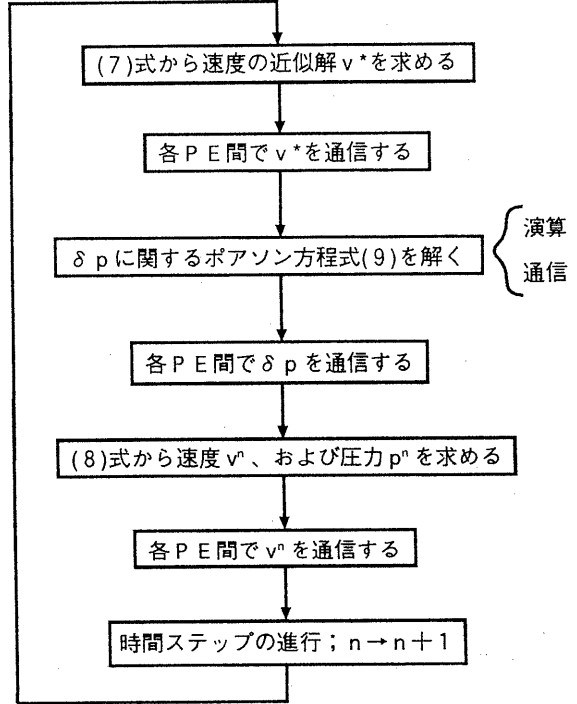


図6: 計算アルゴリズム

とおく。ILU 分解に基づく前処理に比べ反復回数が多いが、1 回当たりの演算が少なく、また並列化が容易であるという長所がある。

最後に RB-SOR 法に関して述べると、これは前述の 2 つの部分集合 $\{S_R, S_B\}$ ごとに独立に前進後退代入計算を行う方法である。この場合もやはり並列化することが可能になる。

5 評価結果

我々が用いたモデルは、前節で述べたように 2 次元正方キャビティ内流れである。境界条件は、上壁を移動壁とし $u = 1$ 、 $v = 0$ を与え、他の壁面ではすべりの無い壁とする。計算メッシュは図 7 のような不等間隔直交格子を採用し、計算メッシュ数は 72×72 (5184)、 144×144 (20736)、 216×216 (46656) で行う。また計算条件はレイノルズ数 $Re = 1000$ 、時間間隔はそれぞれ $\Delta t = 2 \times 10^{-3}$ 、 $\Delta t = 5 \times 10^{-4}$ 、 $\Delta t = 2 \times 10^{-4}$ で、Poisson 方程式の求解における打ち切り相対残差は 10^{-3} とする。用いた PE 数は 1×1 (1)、 2×2 (4)、 3×3 (9)、 4×4 (16) である。

図 8 は、連立一次方程式に Scaling 前処理つき Bi-CGSTAB 法を採用し、50 時間ステップまで実行した場合の、PE 数と速度向上率との関係図を表している。図から計算メッシュ数が 72×72 の場合には、通信のオーバーヘッドが大きいために、PE 数が 9 台、16 台で PE 数に見合った速度向上

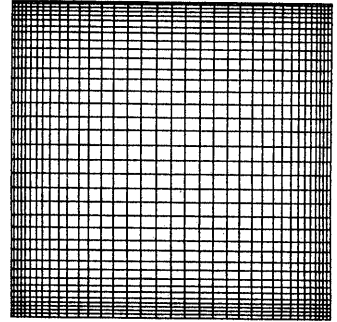


図 7: 不等間隔直交格子

率が得られないことがわかる。しかしメッシュ数が 144×144 、 216×216 と大きくなるにつれて、ほぼ台数に等しい速度向上率が得られる。特に、 216×216 で台数が 4 台あるいは 16 台の場合、台数以上の向上率が得られる。このような台数以上の速度向上率が得られる現象はスーパーリニアと呼ばれる。これは、1 台ではキャッシュメモリのミスヒットが多く、実効演算速度が 1.6MFLOPS とピーク性能の約 $\frac{1}{8}$ である一方、台数が増加するにつれてキャッシュメモリのヒット率が向上し、1 台当たりの実効演算速度が速くなるためである。実際には、台数が増えるにつれて通信のオーバーヘッドも増大するが、 216×216 の場合には演算速度の向上の効果の方が大きいと言える。以上のことは、連立一次方程式の他の解法についてほぼ当てはまる。

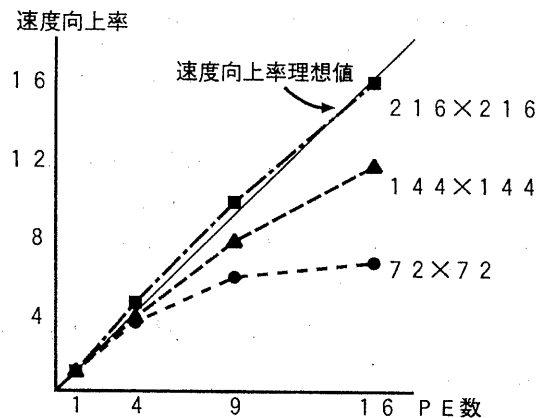


図 8: PE 数と速度向上率との関係

このような速度向上率よりも重要なことは、どのアルゴリズムが高速に解を求めるかという点である。このことは、いかに速度向上率が良くても計算量が多ければ、向上率というものは余り意味をなさないことを意味する。このような観点から、PE 数 16 台で 50 時間ステップまで実行した場合の、SMAC 法のアルゴリズム全体での演算時間および通信時間との関係を、連立一次方程式の各解法ごとに比較した図が、図 9 である。我々が用いたモデルでは、全実行時間の内、8 割以上が連立一次方程式の求解に費やされるので、そのソルバの効率が解析時間全体にかなり影響する。

図から計算メッシュ数が 72×72 の場合、通信のオーバーヘッドが大きく、実行時間の半分以上が通信にとられていることがわかる。このことから、大きな速度向上率が得られないのは明らかである。一方計算メッシュ数が 216×216 の場合には、通信のオーバーヘッドが小さく、実行時間の $\frac{1}{5}$ 以下である。これは、1PE当たりの計算メッシュ数を $m \times m$ とすると、一般に演算量は m の二乗に比例し、通信量は m に比例するので、メッシュ数を増やす程、相対的な通信量が減少するためである。従って計算メッシュ数が 216×216 の場合には、台数に見合った速度向上率が得られる。

また全ての計算メッシュ数の場合で、グスタフソン加速を伴うB-MILU前処理つきBi-CGSTAB法が、最も実行時間が短いことがわかる。具体的には、RB-SOR法の $\frac{1}{4}$ ぐらいである。これは、この方法では前処理でPE間の通信を行う必要がないので、通信のオーバーヘッドが少なく、かつ収束性が良いからである。実際に、その1時間ステップあたりの平均反復回数は 72×72 で15回、 144×144 で24回、 216×216 で33回で、これはRB-ILU前処理つきBi-CGSTAB法の $\frac{1}{3}$ 、Scaling前処理つきBi-CGSTAB法の $\frac{1}{5}$ 、RB-SOR法の $\frac{1}{10}$ 程度である。このことは、B-MILU前処理つきBi-CGSTAB法はRB-SOR法などに比較して、メッシュ形状が複雑であるような条件の悪い問題に対しても収束性が保証されるロバストな手法であると考えられる。結局、非圧縮性流体を取り扱う際の圧力のPoisson方程式を並列マシンで解く場合には、B-MILU前処理つきBi-CGSTAB法が優れていることがわかる。

最後に、数値計算の結果を示す。計算条件は計算メッシュ数 72×72 、レイノルズ数 $Re = 1000$ 、時間間隔 $\Delta t = 2 \times 10^{-3}$ で、無次元時間140(70,000時間ステップ)までの計算を行った。これに要した時間は約4.5時間である。計算メッシュ数が小さいために、2次渦は見えないが、主渦の中心の位置は信頼度が高いと言われるU.Ghiaらの計算結果と一致する[8]。ここでU.Ghiaらの計算では、主渦の中心の位置は $x = 0.5315$ 、 $y = 0.5625$ である。

6 まとめ

2次元SMAC法を領域分割法を用いて並列マシンCenju2上を実現した。Poisson方程式の求解として4種類の並列線形ソルバをインプリメントし、それらの比較を行った。

その結果、並列化のゲインについては、計算メッシュ数が 72×72 の場合、通信のオーバーヘッドによって大きな速度向上率は得られないが、 144×144 、 216×216 の場合には、台数に見

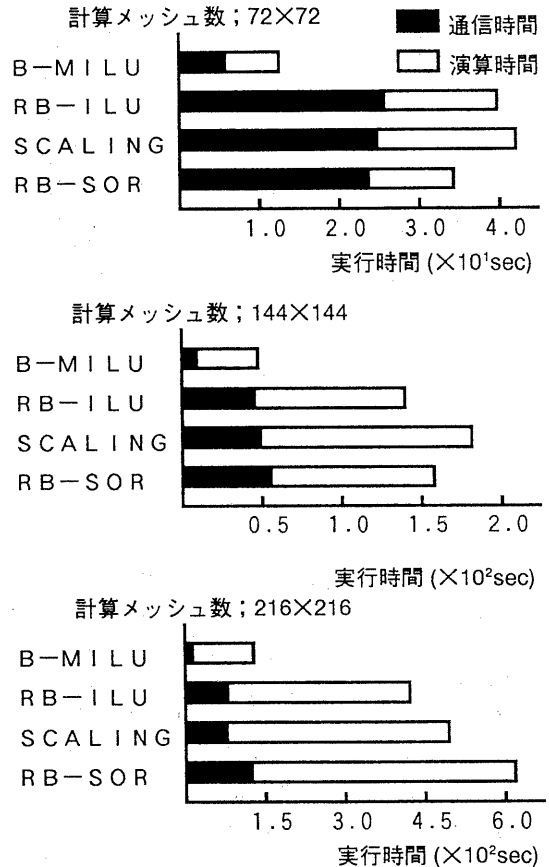


図9: PE数16台における演算時間と通信時間

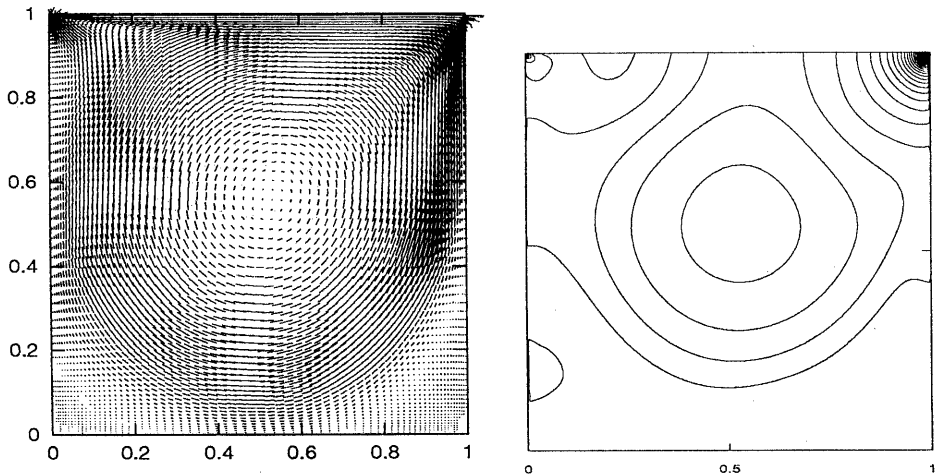


図 10: (a) 速度ベクトル図

(b) 等圧力線図

合う速度向上が得られた。例えば PE16 台を使用した場合、速度向上率は 15 程度である。これは問題規模が大きくなるにつれ、計算に対する相対的な通信コストが低下すること、また PE 数を増やすと 1 台当たりの問題規模が小さくなり、キャッシュのヒット率が向上することによる。

また比較した 4 種類の線形ソルバの中で、各 PE 内でローカルに MILU 分解を行う B-MILU 前処理つき Bi-CGSTAB 法が、他の 3 手法 (RB-ILU 前処理つき Bi-CGSTAB 法、Scaling 前処理つき Bi-CGSTAB 法、RB-SOR 法) よりも高速であり、RB-SOR 法と比較すると全実行時間は $\frac{1}{4}$ であった。この手法では、前処理で PE 間の通信がないため通信のオーバーヘッドが少なく、一方収束性は良いからである。

参考文献

- [1] T. Kawamura and K. Kuwahara, *Computation of High Reynolds Number Flow around a Circular Cylinder with Surface Roughness*, AIAA paper, 84-0340, pp.9-12, (1984).
- [2] J. M. Ortega, *Introduction to parallel and vector solution of linear systems*, Frontiers of computer science, Plenum Press, pp.145-149, (1988).
- [3] Takumi Washio and Ken Hayami, *Parallel Block Preconditioning Based on SSOR and MILU*, J. Num. Linear Algebra with Appl., World Scientific, to appear.
- [4] 小国力編著：行列計算ソフトウェア, 丸善,(1991).
- [5] 松下, 山内, 中田, 小池：並列マシン Cenju2 のアーキテクチャ, 情報処理学会計算機アーキテクチャ研究会資料 Vol.92, No.64, pp.17-23, (1992).
- [6] 保原・大宮編：数値流体力学, 東京大学出版会, (1992).
- [7] A. A. Amsden and F. H. Harlow, J. Comput. Phys., Vol.24, p.398, (1970).
- [8] U. Ghia, K. N. Ghia and C. T. Shin, J. Comput. Phys., Vol.48, p.387, (1982).