

ノイマン展開法前処理によるCGアルゴリズムの ベクトル、並列処理について

石黒美佐子 鈴木信太郎
茨城大学工学部

CG系アルゴリズムは並列処理（ベクトル処理を含む）に優れ、線形方程式のスーパーコンピュータ向き解法として広く利用されている。しかしながら、その効率は、前処理をうまく並列計算できるかどうかにかかっている。ここでは、多様な係数行列に適用可能なノイマン展開法前処理についてスーパーコンピュータ HITAC S-3800 によるベクトル処理ならびに、ローカルメモリー並列計算機による並列処理の効果を検証する。差分法によるもの以外に、プラズマ流体方程式をフーリエ級数展開により離散化した行列にも適用する。

Vector and Parallel Processing of Neumann-Expansion Preconditioning for CG-like Algorithms

Misako Ishiguro and Shintaro Suzuki
Faculty of Engineering, Ibaraki University
4-12-1, Nakanarusawa-cho, Hitachi-shi, Ibaraki 316, Japan

CG-like algorithms have been widely used as an excellent solution method for linear equations on vector and/or parallel supercomputers. But the effectiveness depends on the parallelizability of the preconditioning procedure. Here, we deal with von Neumann expansion preconditioning method which is available for a wide class of matrices, and validate the vectorization effect on supercomputer HITAC S-3800 and the parallel effect on a local memory parallel processor. The method is applied to a matrix discretized by Fourier series expansion of a plasma fluid flow equation, as well as usual finite difference matrix.

1. はじめに

共役傾斜法 (Conjugate Gradients; CG) 系アルゴリズムは、正定値行列を係数とする線形方程式 $Ax = b$ (1.1)

の解法として、ベクトル、並列計算機に有効なためスーパーコンピュータで広く利用されている。実際、大型行列ではCG法に前処理を施すことによって、従来のガウスの消去法に基づくLU分解法に較べて計算量が著しく減少する。ちなみに、 $n \times n$ 密行列では、ガウスの消去法の演算数は $2/3n^3$ のオーダーになるのに対し、前処理付きCG法 (ICCG法) では、およそ $3n^2 \times$ 反復回数 となる。ここで、反復回数は初期値と行列の優対角性に依存するが n より遙かに小さい値となる。基本となる対称行列用ICCGアルゴリズムをFig. 1.1に示す。見てわかるように演算列は行列・ベクトル積、ベクトル・ベクトル演算から成り立ち、並列処理向きであることがわかる。CG法は従来よく使用されてきたRed-BlackSOR法と比べても高速なことが知られている。このほか、流体計算などで非定常計算を行うとき、前反復値が良いゲスとなって反復回数は少なくなるという利点もある。

CG法の前処理 (Preconditioning) は、元の行列Aの条件数 (=最大固有値と最小固有値の絶対値の比) を小さくして反復回数を減少させるためのものである。前処理で重要なのは、前処理行列の選び方と並列、ベクトル処理に有効な前処理方法の採用である。前処理行列Kは、 $K^{-1}A \approx I$ となる行列、つまり、元の行列に近く簡単な構造をなすものが良い。前処理方法としては、反復計算の中で $Ax = b$ を近似的に解く方法と、 $K^{-1}A$ の全ての固有値を1に近づける方法^[1]とが知られているが、ここでは多様な行列に対応できることを考慮して前者の方法を用いる。

前処理行列Kは、対称行列では不完全Cholesky分解 (IC)、非対称では不完全LU分解 (ILU) によるものがよく用いられる。Fig. 1.1では、 $K^{-1}r = r'$ の演算が前処理に当たる。これは前処理行列Kを係数とする方程式 $r = Kr'$ (1.2) を r' について解くことに置き代わる。前処理方程式を解くのは、元の方程式 $Ax = b$ を解くのに較べてラフなものでもよい。K = LUとにおいて、2回に分けて解く。

$$Ly = b, \quad Ux = y \quad (1.3)$$

(1.3) 式は回帰演算となり並列処理やベクトル処理に向かない。しかし、正方格子に差分法を適用した場合には、演算順序を斜め方向にとった超平面 (Hyperplane) 法がよく使われている。しかし、超平面法にはいくつかの欠点がある。

- (1)適用できるは係数行列の形が5点階差や7点階差式に基づくものに限られる、
- (2)ベクトル長が"次元落ち"し小さくなり、十分な並列性が得られない、
- (3)計算順序の並べ変えが必要で、データ参照時にメモリー競合が起こる。

そこで、もっと広範囲な行列に適応可能で、かつ、並列処理が容易なノイマン展開法^[2]を試し前処理の効果を検討する。既に、疎な非対称ブロック3重対角行列でIBM 3090の6 CPUでの数値実験結果が報告されている^[3]。

ここでは、差分法行列に対しては、ローカルメモリー型並列計算機 FAME 上での数値実験についても示すと共に非対称行列用のCGSTAB法^[4] (Fig. 1.2) にノイマン展開法前処理を施したときのベクトル処理効果を示す。さらに、プラズマ流体方程式をフーリエ展開と差分法併用で離散化した行列にもノイマン展開法前処理を適用する。プラズマ流体方程式では高い並列性が得られ、マルチプロセッサ型スーパーコンピュータ向きであるが、今回はテスト環境が整わないので、スーパーコンピュータによるベクトル処理に留める。

Choose x_0 .
 $p_0 = r_0 = b - Ax_0$, $r'_0 = K^{-1}r_0$.
 For $i=0$ until Convergence DO
 1. Compute and Store Ap_i
 2. Compute (p_i, Ap_i)
 3. $\alpha_i = (r_i, r_i) / (p_i, Ap_i)$
 4. $x_{i+1} = x_i + \alpha_i p_i$
 5. $r_{i+1} = r_i - \alpha_i Ap_i$
 6. $r'_{i+1} = K^{-1}r_{i+1}$
 7. Compute (r_{i+1}, r'_{i+1})
 8. $\beta_i = (r_{i+1}, r'_{i+1}) / (r_i, r'_i)$
 9. $p_{i+1} = r_{i+1} + \beta_i p_i$
 EndFor

Fig. 1.1 前処理付きCG法

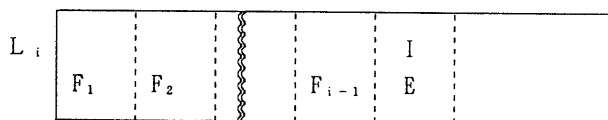
Choose x_0 , $r_0 = b - Ax_0$.
 $\rho_0 = \alpha = \omega_0 = 1$, $v_0 = p_0 = 0$.
 For $i=1$ until Convergence DO
 1. $\rho_i = (r_0, r_{i-1})$
 2. $\beta = (\rho_i / \rho_{i-1}) (\alpha / \omega_{i-1})$
 3. $y = K^{-1}p_{i-1}$
 4. $p_i = r_{i-1} + \beta (p_{i-1} - \omega_{i-1}v_{i-1})$
 5. $v_i = Ay$
 6. $\alpha = \rho_i / (r_0, v_i)$
 7. $s = r_{i-1} - \alpha v_i$
 8. $z = K^{-1}s$ (where $K = LU$)
 9. $t = Az$
 10. $\omega_i = (L^{-1}t, L^{-1}s) / (L^{-1}t, L^{-1}t)$
 11. $x_i = x_{i-1} + \alpha y + \omega_i z$
 12. $r_i = s - \omega_i t$
 EndFor

Fig. 1.2 前処理付きCGSTAB法

2. ノイマン展開前処理法

2.1 ノイマン展開法

もとの行列を不完全LU分解し、対角要素による正規化後の下三角行列をLとおく。これをm段に分ける。mはnの約数、例えばx, y 標ではy方向の格子の数とおく。これを、 $L_i = I + E + F$, $F = \sum_{j=1, i-1} F_j$ に区分する。



$$\text{方程式 } L_i x = b_i \text{ は } (I + E) x_i = b_i - \sum_{j=1, i-1} F_j x_j \quad (2.1)$$

$(I + E)^{-1}$ にノイマン展開を施せば、

$$x_i = (I - E + E^2 - E^3 + \dots) (b_i - \sum_{j=1, i-1} F_j x_j) \quad (2.2)$$

Eの要素の絶対値が1よりかなり小さいなら、2次程度で展開を打切ることができる。また、計算量を減らすためにFの要素がゼロに近いものなら適当に省略してよい。この計算法では演算に回帰演算は含まず、ブロックごとに独立に計算できるため、並列処理において
はプロセッサ間で同期の必要がない。5点有限差分行列の場合には、Eの展開を2次で打ち切っても反復回数の増加は少ない (Table 2.1) ので、

$$x_i = (I - E + E^2) (b_i - F_{i-1} x_{i-1}) \quad (2.3)$$

$i = 1, 2, \dots, m$ の順に計算すれば、右辺は既知量となり、ベクトル x_i の要素数がベクトル長となる。ベクトル長は3次元→2次元のように次元落ちするが超平面法のように極端な小ベクトルは現れない。上三角行列Uにも同様な方法を適用する。

このようにノイマン展開法は超平面法では計算できないデータ参照関係、例えば、有限要素法から得た帯行列やフーリエ変換と有限差分法の併用によって得られる行列などに適用可能である。行列が優対角性の強いものであれば必ずしもブロック行列でなくてもよい。

2次元5点有限差分行列を係数とする線形方程式をICCG法を用いて解くとき、前処理として超平面法とノイマン展開法を適用した場合の収束に達するまでの反復回数の比較をTable 2.1に示す。また、汎用大型機 HITAC M-880、スーパーコンピュータ S-820 と S-3800での計算時間の比較をFig. 2.1に示す。2次元問題のように小規模計算では前処理によって計算時間は短縮されない。反復回数は約 1/3に減少するが、前処理のコストが相対的に大きくなるためである。ノイマン展開法は超平面法に比べて同程度の計算時間となる。

2.2 非対称行列向きCGSTAB法のベクトル処理

ポアソン方程式で表現されている定常熱伝導問題を2次元形状で差分法によって離散化すると、典型的な非対称5重対角行列を係数とする線形方程式を解く問題に帰着する。CGSTAB法 (Fig. 1.2) とノイマン展開法前処理の組み合わせでベクトル処理効果を測定した。結果は Fig. 2.2に示すとおりである。前処理による計算時間短縮が少ないのは、前処理の回数が1反復あたり 2.5回必要なためである。前処理を減らすため、CGSTAB法での ω_i を求める計算を前処理なしで行う方法も試みた。この方が計算時間の改善となった。

$$\omega_i = (L^{-1}t, L^{-1}s) / (L^{-1}t, L^{-1}t) \Rightarrow (t, s) / (t, t)$$

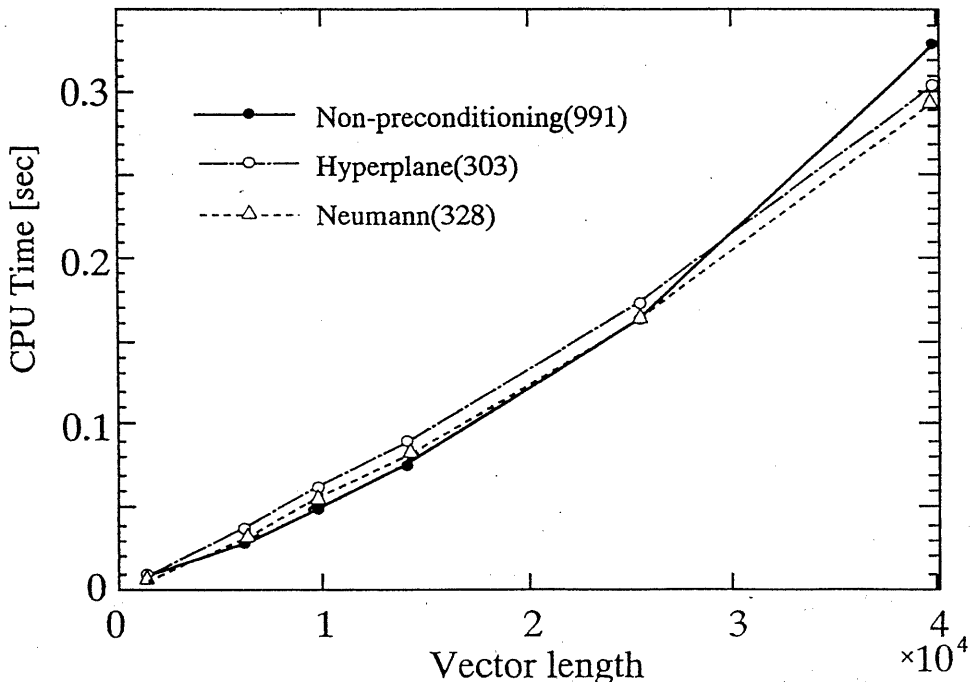


Fig. 2.1 各種前処理付きCG法の計算時間

Table 2.1 前処理付きCG法の反復回数

ベクトル長	前処理なし	超平面法	ノイマン法
1600	205	60	65
100*100	512	156	168
200*200	991	303	328

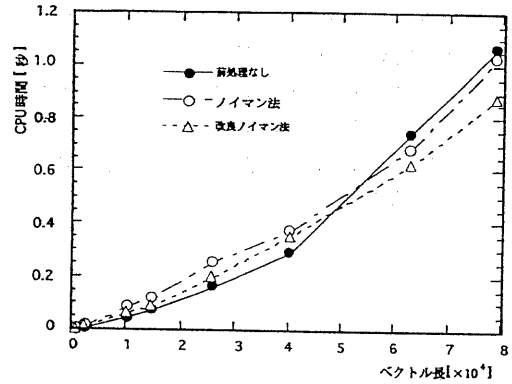


Fig. 2.2 CGSTAB法の計算時間の比較

3. 有限差分行列への並列処理の適用

3.1 並列処理向き前処理

ローカルメモリー型並列計算機によって計算を行う場合、係数行列及びベクトルを各プロセッサ (PE) に分割し割当ての方法をとる。前処理では、2つのPEにまたがる要素を計算するとき、隣のPEでの演算値を必要とするため、このまま並列計算はできない。プロセッサ間に重複部分を置くことも検討したが、前反復値を利用することになり反復回数の減少につながらないことが数値実験からわかった。ここでは、Fig. 3.1 に示すようにプロセッサ数の小行列に分割し重複のない分割方法を用い、前処理部分の演算を完全にプロセッサ間で独立に行う方法を採用した。

前処理行列の作成には、2通りの方法がある^[3]。

- (1) 全体前処理：まず、係数行列をILU分解し、 $A \approx LU$ 、その後小行列 A_i に対応する L_i, U_i を L, U から要素を抜き出して作る。
- (2) 局所前処理：分割した各々の小行列 A_i ごとにILU分解する方法である： $A_i \approx L_i U_i$ 。

反復回数の比較ではTable 3.1に示すように局所前処理の方が好結果が得られた。また、並列化による増加は約10%であった。(2.3)式の計算ではPEのつながりでの右辺の x_{i-1} に対応する計算を無視するので精度は落ちるが、前処理として許容の範囲と見なせる。

3.2 並列計算機FAMEによる数値実験

NEC Cenju の特殊なバージョンであるローカルメモリー型並列計算機 FAME を用いてノイマン展開法の並列化効果を検証した。FAMEは、20台のPEを装備し、クロスバー方式のプロセッサ間ネットワークを持つ。数値実験の結果、Fig. 3.2 に示すように、PE20台による処理で実時間で約10倍の高速化が得られた。

CG系解法のベクトル計算では、ベクトル長は前処理部分を除き行列の次数となり好都合であるが、ローカルメモリー型の並列計算機では問題点もある。スカラー量の α と β の計算が内積に基づくものなので (Fig. 1.1参照)、部分和を各PEに分担させるならば、部分和収集のためのデータ転送が必要となる。共有メモリーを持たないデータ転送速度の低い並列計算機ではこのことがCG法の致命的欠点となる。ローカルメモリー型並列処理では、データ転送や同期処オーバーヘッドを考えると、並列性のある計算をPEに分担させるこ

とが必ずしも最善ではない。データ転送時間との兼ね合いで使い分ける必要がある：

- (1)初期設定や逐次処理では主PE (PE0) でのみ実行、
- (2)前後のデータ転送オーバーヘッドを考慮して全てのPEで同じ計算を重複して実行する、
- (3)各PEで並列実行する。

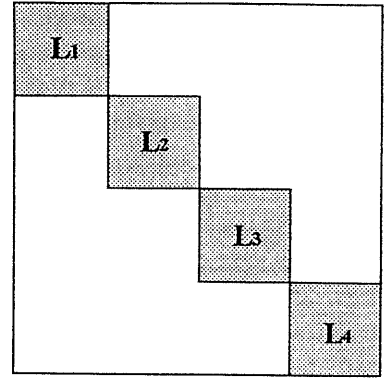


Table 3.1 並列化による反復回数 (4 CPU 仮定)

ベクトル長	並列せず	部分前処理	全体前処理
40*40	65	73	75
100*100	168	181	187
200*200	328	355	369

Fig. 3.1 前処理行列のPEへの割当

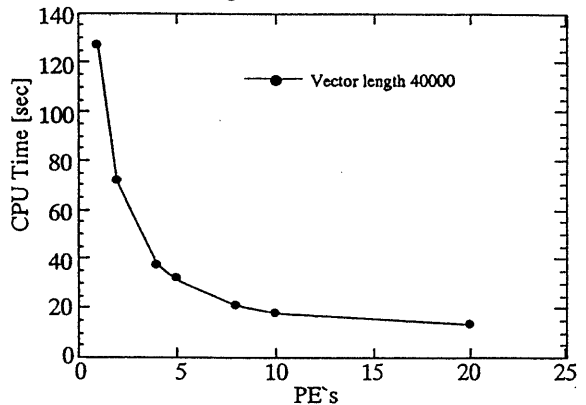


Fig. 3.2 差分方程式の並列処理

4. プラズマ流体方程式への適用

4.1 プラズマ流体方程式の離散化

プラズマ流体方程式は^[5]、プラズマ質量密度に関する連続の式、プラズマの運動方程式、状態方程式、オームの法則、磁場の電磁誘導の式など7つの方程式から成っている。方程式群をトカマク形状 (Fig. 4.1) で解くにあたり、大アスペクト比近似を用いて簡易磁気流体方程式化し、流れ関数 ψ 、磁束関数 ψ 、プラズマ圧力 P の3関数に関する方程式に帰す方法が従来から採られている。離散化は、 ζ, θ, r 座標で、ポロイダル半径方向 r は中心差分により、トロイダル、ポロイダル角度方向 ζ, θ はそれぞれフーリエ変換によりなされる^[5]。 r は R 個のメッシュに、 ζ は N 個のモードに、 θ は M 個のモードに刻まれる。ここで、 $N, M \ll R$ 。この結果、得られる行列の形は $N \times N$ ブロック対角行列となり (Fig. 4.2a)、その小行列は $M \times R \times M \times R$ で、ベクトル処理を効率的に行うために r 方向を最内側に持ってくれば、 $R \times R$ の3重対角行列 A_{mk} を要素とする正方行列となる (Fig. 4.2b)。最外側のブロック行列は、 N 個 独立に計算可能なので並列処理に最適である。

小行列を対角要素で正規化し ILU分解し、 $K = (I + L)(I + R)$ とおく。 L 行列は Fig. 4.2bの左下部分と同様の単位3角行列となる。 L 行列を M 段に分け、 $L_m = [F, E, I]$ とおく。フーリエ展開では基本波の寄与が大きいため、 A_{mk} 行列要素の絶対値は $m = k$ のとき大きく、 $m \neq k$ では極めて小さくなる。つまり、行列 F の要素は E, I に比べて無視

できる。そこで(2.1)式でEを3次まで展開しFを省略した形にできる。 x_m をR次のベクトルとして

$$x_m = (I - E + E^2 - E^3) b_m \quad (4.1)$$

(4-1)式は回帰式ではないので Fig. 4.2bの行列の全要素 (=M*R次) で並列計算可能となりベクトル処理や並列処理に最適である。

4.2 ベクトル処理数値実験

CGSTAB法を使って、前処理なし、正規化のみ、ノイマン展開法前処理の場合の3通りで反復回数の比較を行った($\epsilon = 10^{-6}$)。結果をTable 4.1の左2列に示す。M*R = 300, 6000の2つの場合について試したが、前処理で反復回数が約1/20と大幅に減少した(次数6000の場合)。右3列には次数6000の場合におけるHITAC M-880、S-3800での計算時間を示す。M-880のスカラ性能は、S-3800の約0.7であるから、ベクトル処理による速度向上は約80倍となりS-3800では約2.5GFLOPSの速度(最高性能の1/3~1/4)となる。

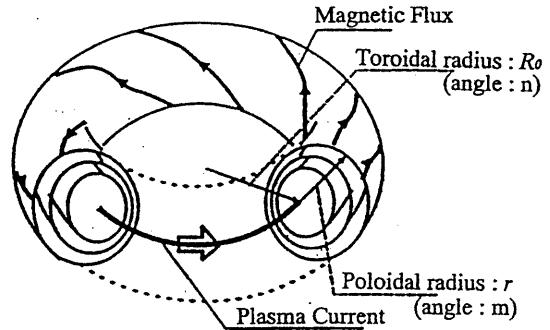


Fig. 4.1 トカマク形状

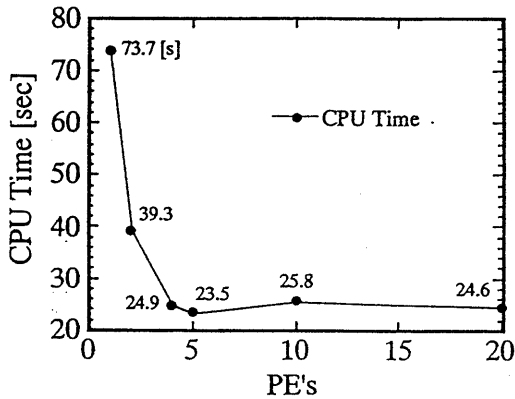


Fig. 4.3 FAME による並列処理

Table 4.1 ノイマン展開法による反復回数と計算時間

方法	M=6, R=50	M=20, R=300	M-880	S-3800
前処理なし	137回	5298回	(秒)	
正規化のみ	97	650	54.3	0.513
ノイマン法	43	230	22.8	0.209

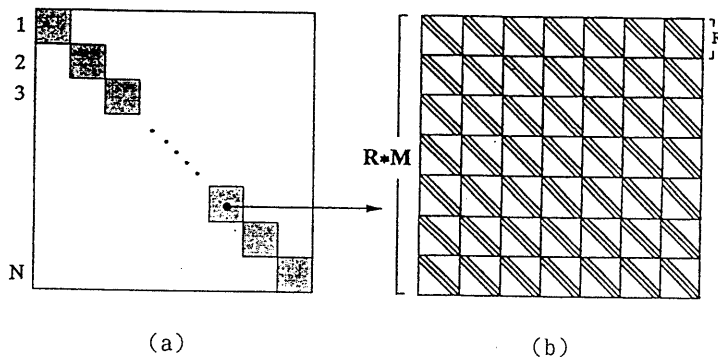


Fig. 4.2 プラズマ流体方程式における離散化行列

プラズマ流体方程式行列の計算では、 $N \times M \times R$ ($\geq 20 \times 20 \times 300$) の並列性を有し、これを、 N 方向に並列、 M と R 方向にベクトル計算すれば、CG法の並列処理のオーバーヘッドもないので、マルチプロセッサ型スーパーコンピュータに最適なアルゴリズムと言える。

並列計算機 FAMEによる数値実験結果についてはFig. 4.3 に示す。PE 5 台までは台数効果が出るが、PE数が増えると、CGSTAB法 (Fig. 1.2) で行列・ベクトル積の計算に先だてて y や z ベクトルの全要素を送っているので転送のオーバーヘッドが大きい。データ転送速度の遅いシステムでは、むしろ、行列を縦割りして各PEに分担させ前処理なしで実行する方がよい。

5. まとめ

差分方程式に適用した場合には、以下のことがわかった。

- (1) ノイマン展開法は超平面法に比べて反復回数の増加は少なく (Table 2.1)、計算時間はノイマン展開法の方がやや少ない (Fig. 2.1)。
- (2) ノイマン法のCGSTAB法への適用では、反復回数は1/3に減少するが、前処理の演算量が多いため前処理による計算時間の減少は約 80%に留まった (Fig. 2.2)。
- (3) ローカルメモリー型並列計算機 FAMEでの実験では、20 台のPEで約10倍の速度向上であった (Fig. 3.3)。これは、 α 、 β の計算が内積に基づくものな部分積の転送が必要となり通信量が大きいためである。FAMEは、PE間のデータ転送速度が実効で4 Mバイトと遅いので、演算時間と同程度のデータ転送時間を必要とする。

プラズマ流体方程式に適用した場合には以下の知見を得た。

- (4) 優対角行列のためノイマン展開法は回帰性のない演算として適用でき、反復回数を1/20に減らすなど前処理法として適している。ベクトル処理では、スカラー計算に比べて約80倍の速度向上が得られた。3次元トカマク形状では、全格子を対象とした並列性を有するので、並列・ベクトル処理に最適である。
- (5) FAMEでの並列処理では、5 台のPEで約3.5倍の速度向上を得るが、台数が増加するとデータ転送オーバーヘッドのため効果が上がらなかった (Fig. 4.3)。
- (6) マルチプロセッサ型スーパーコンピュータでは良い効果が期待できる。

参考文献

- [1] 田中伸厚、寺坂晴夫：共役勾配法の前処理のためのブロック化行列の多項式に関する研究、情報処理学会（6年前期）全国大会、7L-5（1994）。
- [2] H.A. Van der Vorst: A Vectorizable Variant of Some ICCG Method, SIAM J. Stat. Comput., Vol. 3-3, pp. 45-54 (1982).
- [3] G. Radicati di Brozoro et al.: Parallel Conjugate Gradients like Algorithm for Solving Sparse Nonsymmetric Linear Systems on a Vector Processor, Parallel Computing 11, pp. 223-239 (1989).
- [4] H.A. van der Vorst: Bi-CGSTAB: A Fast Smoothly Converging Variants of Bi-CG for Nonsymmetric Linear System, SIAM J. Sci. Stat. Comput., Vol. 13-2, pp. 631-644 (1992).
- [5] T. Takeda, K. Tani, et al.: Plasma Simulator METIS for Tokamak Confinement and Heating Studies, Parallel Computing, Vol. 18, pp. 743-765 (1992).