1－18

三項対角近似因子分解による前処理共役勾配法
― スーパコン向き非対称連立一次方程式解法 ―

土肥　俊*　原田紀夫*

Tridiagonal Approximate Factorization Method: A Preconditioning Technique for Solving Nonsymmetric Linear Systems Suitable to Supercomputers

by

Shun DOI and Norio HARADA

C & C Information Technology Research Laboratories, NEC Corporation

ABSTRACT

In this paper, we propose the Tridiagonal Approximate Factorization (TF) method as a preconditioning algorithm suitable to vector and parallel computers for solving large sparce nonsymmetric systems of linear equations. In this method, the preconditioning matrix is constructed as a product of component tridiagonal matrices, each of which consists of matrix elements corresponding to a spatial axis of the original P.D.E. problem. The inversion process of this preconditioning matrix is reduced to a series of simple forward-and-backward substitutions, which have inherent parallelism and hence can be easily vectorized.

Comparison is made between the TFBCG algorithm, which applies TF preconditioning to the Bi-Conjugate Gradient algorithm, and the ILUBCG/MILUBCG algorithms, the most powerful and popular methods on conventional scaler computers. Numerical tests on the SX-2 supercomputer show that the TFBCG program is 3 to 9 times faster than the conventional (M) ILUBCG programs, and also 1 to 4 times faster than the sophisticated (M) ILUBCG programs utilizing the list-vector technique.

1. はじめに

ベクトル演算方式のスーパーコンピュータ等，並列計算機の実用化に伴って，高い並列性を持つアルゴリズムの重要性が増している。本稿では移流拡散方程式の差分近似による規則・疎な非対称連立一次方程式を取り上げ，その並列処理向き解法を検討している。

規則・疎な非対称連立一次方程式解法として，現在，前処理付き双対共役勾配法（PBCG法），前処理付き共役残差法（PCR法）等が注目されている。中でも不完全LU分解による前処理を用いた（M）ILUBCG法，（M）ILUCR法は詳細な検討がなされている[1]。しかしこれらは"自然な"プログラミングでは不完全LU分解に伴う前進後退代入が逐次処理となり並列化できない。計算格子に対して斜め方

＊日本電気㈱C＆C情報研究所

向の並列性があり（図1），ベクトル計算機ではリストベクトルを用いたベクトル化がなされるが[2]，それは等間隔メモリアクセスに比べて転送速度が低く，計算機の性能を十分引き出せない。また並列に実行できる演算量が斜め方向の格子点数に等しく，計算の進行に伴って変化するため，プロセッサアレイ型並列計算機ではこれがプロセッサの遊びとなる欠点がある。

筆者らは文献3)で規則・疎な非対称連立一次方程式のための前処理手法として，三項対角近似因子分解法（Tridiagonal Approximate Factorization; TF法）を提案した。TF法は従来の不完全LU分解法に比べてより"自然な"並列性を持ち，ベクトル計算機，プロセッサアレイに適している。収束性は，TF法を双対共役勾配法（BCG法）に適用したTFBCG法をILUBCG法と数値例で比較することにより，ほぼ同程度であることを確認している[3]。本稿では3次元の移流拡散方程式を例に取り，当社のスーパーコンピュータSX-2での実測によって，TFBCG法がILUBC法より高速であり，MILUBCG法と比べても同等以上であることを示す。

2. 前処理付き反復法

規則・疎な非対称N元連立一次方程式を

$$\boldsymbol{A}\boldsymbol{u}=\boldsymbol{f} \tag{1}$$

と表わす。係数行列Aは図2の構造を持つものとする。

式(1)に対する前処理付き反復法はこれと等価な式

$$(\Lambda_1^{-1}A\Lambda_2^{-1})(\Lambda_2\boldsymbol{u})=\Lambda_1^{-1}\boldsymbol{f} \tag{2}$$

に対する反復法であると定義される。$M=\Lambda_1\Lambda_2$ は前処理行列と呼ばれる。即ち前処理付き反復法のアルゴリズムは前処理行列Mの構成法（前処理法）と双対共役勾配法等の基礎反復法の組み合せとして実現される。

さて，前処理付き反復法では毎回の反復で前処理行列Mの逆行列操作が必要である（後掲式(3)）。通常前処理以外の演算量はAの元数Nに関して1次のオーダーである。従って前処理法も同様に，M^{-1}の計算量が$O(N)$であることが要求される（逆行列操作の容易性）。

共役勾配法系統の手法は係数行列Aに重複または

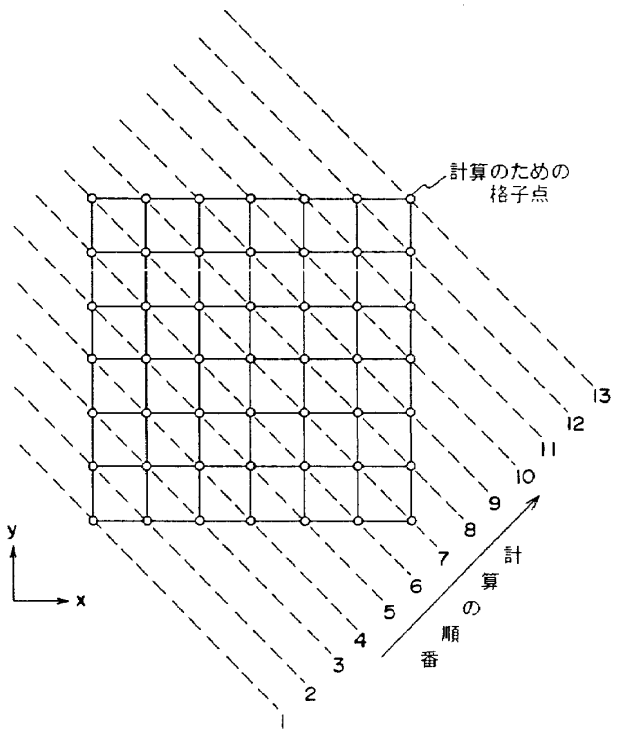

図1 ILU法における並列性

密集固有値があると収束が速いことが知られている。従って二つめの要件として，MはAに"似た"行列であることが要求される（近似性）。

さらに並列演算を特徴とする計算機では，M^{-1}の計算が想定する計算機に適した並列性を持つことが重要である（並列性）。

以上の「逆行列操作の容易性」，「近似性」，「並列性」を前処理行列の三要件と呼ぶことにする。

本稿では基礎反復法として双対共役勾配法を用い，前処理法の比較・評価を行う。ここでは式(2)において$\Lambda_1=I$（単位行列），$\Lambda_2=M$（前処理行列）とする。

前処理共役勾配法は次のように表される。

$$\begin{aligned}
&\boldsymbol{r}_0=\boldsymbol{f}-A\boldsymbol{u}_0,\\
&\boldsymbol{p}_0=M^{-1}\boldsymbol{r}_0,\quad \boldsymbol{p}_0^*=\boldsymbol{r}_0^*=\boldsymbol{r}_0\ ;\\
&k=0,\ 1,\ 2,\ \cdots\\
&\quad \alpha_{k+1}=(\boldsymbol{r}_k,\ \boldsymbol{r}_k^*)/(A\boldsymbol{p}_k,\ \boldsymbol{p}_k^*),\\
&\quad \boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\alpha_{k+1}\boldsymbol{p}_k,\\
&\quad \boldsymbol{r}_{k+1}=\boldsymbol{r}_k-\alpha_{k+1}A\boldsymbol{p}_k,\\
&\quad \boldsymbol{r}_{k+1}^*=\boldsymbol{r}_k^*-\alpha_{k+1}(M^{-1})^TA^T\boldsymbol{p}_k^*,\\
&\quad \beta_{k+1}=(\boldsymbol{r}_{k+1},\ \boldsymbol{r}_{k+1}^*)/(\boldsymbol{r}_k,\ \boldsymbol{r}_k^*),\\
&\quad \boldsymbol{p}_{k+1}=M^{-1}\boldsymbol{r}_{k+1}+\beta_{k+1}\boldsymbol{p}_k,\\
&\quad \boldsymbol{p}_{k+1}^*=\boldsymbol{r}_{k+1}^*+\beta_{k+1}\boldsymbol{p}_{k+1}^*.
\end{aligned} \tag{3}$$

現在最もよく用いられる不完全LU分解法では，Aの非零要素に対してのみLU分解を行い，Aの近似行列

$$M_{ILU}=LU \tag{4}$$

を得る。L，UはそれぞれAの下三角，上三角と同じ非零要素パターンを持つ下三角行列，上三角行列である。M_{ILU}をILU行列と呼ぶことにする。式(3)のM_{ILU}^{-1}はこれらL，Uによる前進後退代入により計算する。このとき「並列性」に関して問題が生じる。

例えば2次元5点差分の場合，前進代入$\boldsymbol{v}=L^{-1}\boldsymbol{g}$を展開すると

$$\boldsymbol{v}_{i,j}=(\boldsymbol{g}_{i,j}-\boldsymbol{l}_{i,j-1}\cdot\boldsymbol{v}_{i,j-1}-\boldsymbol{l}_{i-1,j}\cdot\boldsymbol{v}_{i-1,j})/\boldsymbol{l}_{i,j}$$

となる。格子座標i（またはj）の増加方向に計算すると，$\boldsymbol{v}_{i-1,j}$，$\boldsymbol{v}_{i,j}$間の参照関係から並列処理できない。

$i+j=$一定となる$\boldsymbol{v}_{i,j}$，即ち斜め方向の格子点群は同時処理できる（図1）。同様に，3次元7点差分の場合も格子座標方向には逐次計算となるが，$i+j+k=$一定面上の点で並列化できる。ベクトル計算機ではリストベクトルを用いることによって強制ベクトル化することが行われる。[2] しかしベクトルを用いた間接メモリアクセスは等間隔の直接メモリアクセスに比べて転送速度が低く，ベクトル計算機の性能を十分に引き出せない。同様のことは，ILU法にGustafssonによる収束加速のための改良を施したMILU法，[4] 野寺による行列分離法[5] についてもあてはまる。

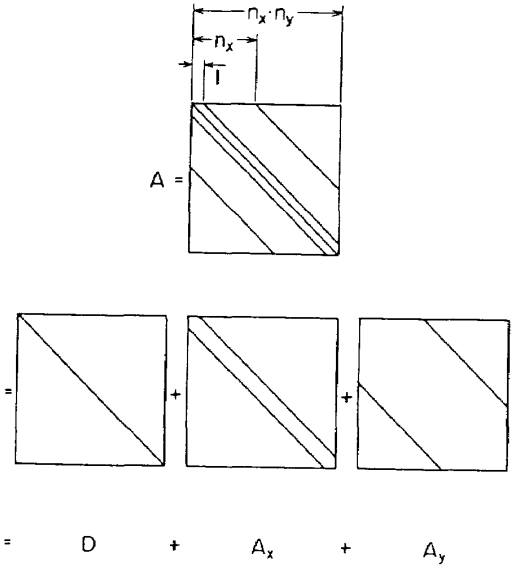

図2.1　2次元5点差分係数行列の構造

3. 三項対角近似因子分解法

はじめに2次元5点差分近似の例で説明する。係数行列Aは図2.1の構造を持つ。Aを対角行列D，x方向の微分に対応する非対角要素からなる行列A_x，y方向に対応する行列A_yに分割する。即ち，

$$A=D+A_x+A_y. \tag{5}$$

このとき，三項対角近似因子分解による前処理行列を次のように定義する。

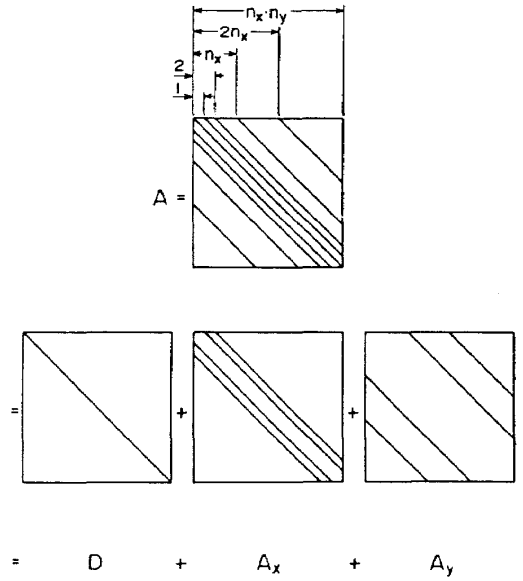

図2.2　2次元9点差分係数行列の構造

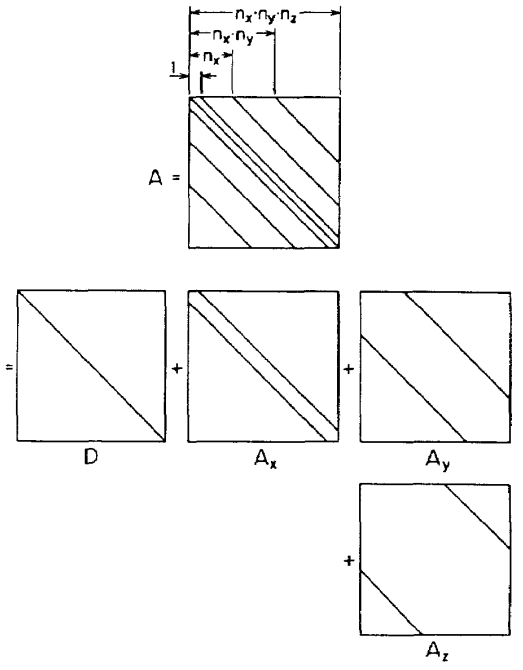

図2.3　3次元7点差分係数行列の構造

$$M_{TF}(\omega) \triangleq (D+\omega A_x)D^{-1}(D+\omega A_y). \quad (6)$$

$D+\omega A_x$, $D+\omega A_y$をそれぞれx, y方向因子と呼び，$M_{TF}(\omega)$をTF行列と呼ぶことにする。ωは収束の加速を目的とする加速パラメータである。文献3)は$\omega=1$の場合にあたる。これはそのごく自然な拡張である。以下では特に必要のない限りωを省略し，M_{TF}と書く。

逆行列操作の容易性：式(6)は反復計算の前段階で因子毎にLU分解する。即ち，

$$M_{TF} = L_x U_x D^{-1} L_y U_y.$$

このLU分解はfill-inを生じることなく完全に行える。さらに簡単な変換によりL_x, U_x, L_y, U_yの対角を1にでき，そのとき計算量はILU法のそれと等しい（表1）。

近似性：ここでは加速パラメータ$\omega=1$の場合のみを考える。近似誤差行列を定義し，近似性の簡単な評価を行う。$M-A$により近似誤差行列Rを定義すると，R_{TF}は

$$R_{TF} \triangleq M_{TF} - A = A_x D^{-1} A_y \quad (7)$$

となり，図3の構造を持つ。即ち，M_{TF}はAの非零要素部分についてこれと等しい。

R_{TF}の非零要素の値は例えばLaplace方程式$\Delta\phi=0$の場合，Aの対角要素の6.25％，異方性がある場合それ以下となる（図4）。また移流拡散方程式

$$\mathrm{div}(-\mathrm{grad}\phi + [\,v_x,\ 0\,]^T\phi) = 0$$

では，セルペクレ数0で6.25％，2のとき8.3％と最大，以後減少し，∞では0になる（図5）。即ち異方性，移流性が大きい程M_{TF}はAに"近づく"。

並列性：逆行列操作は各因子毎に行う。例えばx方向因子$D+\omega A_x$の場合，逆行列計算は方程式

$$(D+\omega A_x)\boldsymbol{v} = \boldsymbol{g} \quad (8)$$

を解くことと等価である。行列$D+\omega A_x$は原係数行列からy方向のつながりに相当する係数A_yを取り去ったものであるから，式(8)はx方向のみに連立したn_y（y方向のメッシュ数）本の独立なn_x（x方向のメッシュ数）元連立一次方程式に他ならない。従ってその前進後退代入計算はn_y個の独立な計算である。y方向についても同様である。この並列性はADI法[6]や野木の不完全AD分解法[7]の並列性と等しい。

表1　TFBCG，ILUBCG法の演算量
（1反復1自由度当たりの演算量）

	加算	乗算
2次元5点差分	24	28
3次元7点差分	32	36

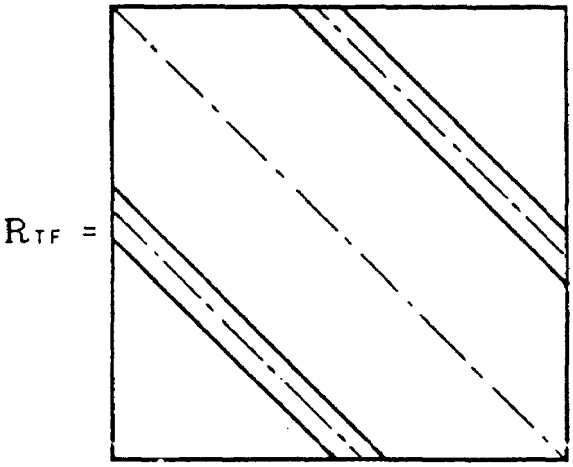

図3.　近似誤差行列の構造

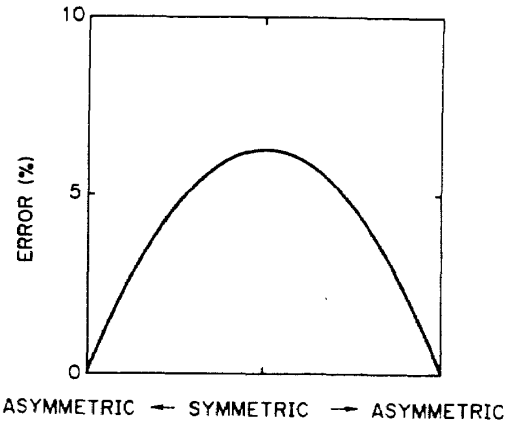

図4　TF行列の近似誤差と異方性の関係

以上の議論は，高次差分や3次元の場合に容易に拡張できる。例えば点(i, j)が$(i+2, j)$, $(i, j+2)$ともつながりを持つ2次元9点差分の場合，各因子行列は5重対角行列となるが（図2.2），それ以外は上記と大差ない。

さて，3次元7点差分の場合TF行列M_{TF}は

$$M_{TF}(\omega) \triangleq (D+\omega A_x)D^{-1}(D+\omega A_y)$$
$$D^{-1}(D+\omega A_z) \quad (9)$$

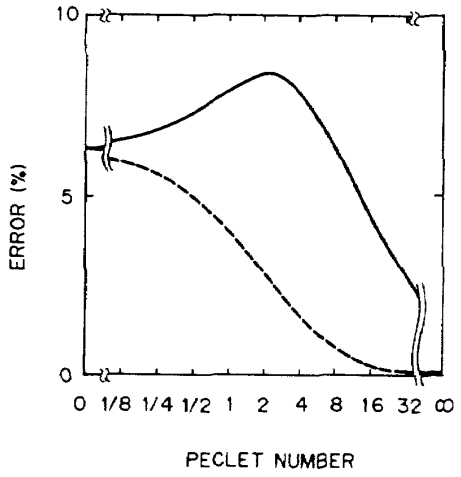

図5　TF行列の近似誤差と移流性の関係

となる。2次元と同様に各因子をそれぞれLU分解する。

$$M_{TF}=L_xU_xL_yU_yL_zU_z$$

ここでは簡単な変換によりL_x以外の対角要素が1になるようにしている。計算量はILU法と等しい（表1）。

表2はx, y, zの各方向因子における並列性とベクトル計算実行時のベクトル長，メモリアクセスパターンを示したものである。各因子とも残りの軸方向に関して独立な並列性を持つ。現在のベクトル計算機では直接アクセスのアドレスが等間隔であることから，y方向因子の計算のみ1次元相当のベクトル長となる。x方向因子の計算はn_x飛びの等間隔アクセスである。

表2　3次元TF行列のベクトル処理

因子	並列方向	ベクトル長	アクセスパターン
x	y, z	$n_y \cdot n_z$	n_x飛び
y	x, z	n_x	連続
z	x, y	$n_x \cdot n_y$	連続

4. 実験結果

3次元7点差分の例題により，収束性，計算速度についてTFBCG法，(M) ILUBCG法の比較を行う。例題は3次元流れ場内の熱拡散方程式(例題1)と，半導体デバイス内のキャリアの拡散方程式（例題2）である。(注)

実験1：例題1を用い，メッシュサイズを59×30×30に固定する。流速の変化に対する収束性，計算時間の比較を行う。

実験2：例題1を用い，流速一定（セルペクレ数＝1）とする。メッシュサイズの変化に対する収束性，計算時間の比較を行う。

実験3：例題2における2種のデバイス（TYPE1, TYPE2）を用いる。バイアス電圧による移流性の変化に対する収束性，計算時間の比較を行う。

実験はいずれもTFBCG法（$\omega=1.0$と1.4），ILUBCG法，MILUBCG法について行った。$\omega=1.4$は例題1の実測による最適値である。(M)ILUBCG法は通常のコーディングによるものと，リス

脚注

例題1　3次元流れ場内の熱拡散方程式

$$\mathrm{div}(-\mathrm{grad}\phi+\boldsymbol{v}\phi)=0,\ (0<x<10,\ 0<y,\ z<1),$$

$$\boldsymbol{v}=[v_x,\ 0,\ 0]^T,\ v_x=v_{max}\cdot\{1-(1-y)^2\}\cdot\{1-(1-z)^2\}.$$

境界条件

$$\phi=\phi_{IN}\ \text{at}\ x=0,\ \{\mathrm{grad}\phi\}_x=0\ \text{at}\ x=10,\ \{\mathrm{grad}\phi\}_n=\phi_{AIR}-\phi\ \text{at}\ y=0,\ 1,\ z=0,\ 1.$$

$\{\mathrm{grad}\phi\}_n$：$\mathrm{grad}\phi$の外向法線方向成分。

例題2　半導体デバイス内のキャリアの拡散方程式

$$\{(1/\tau)-D\,\mathrm{div}\,\mathrm{grad}+\mu E\,\mathrm{grad}\}P=\eta\phi_s/d,\ (0<x<\ell,\ 0<y<w,\ 0<z<d),$$

P：キャリア密度，τ：キャリアの寿命，D：拡散係数，μ：移動度，$E=[E_x,\ 0,\ 0]^T$：バイアス電界，η：量子効率，ϕ_s：フォトンフラックス，d：デバイスの厚さ。

境界条件

$$D\{\mathrm{grad}P\}_x=(\mu E_x+S'_x)P\ \text{at}\ x=0,\ D\{\mathrm{grad}P\}_x=(\mu E_x-S_x)P\ \text{at}\ x=\ell,\ \text{など}.$$

トベクトルを用いて強制ベクトル化したものについて測定した。収束の打ち切り条件は相対2乗残差で 10^{-15} とした。

個々の比較結果を図6～9に示した。例題1と2では収束性の傾向が異なる。TF法の加速パラメータは，例1では反復数を2割程度減少させるが，例

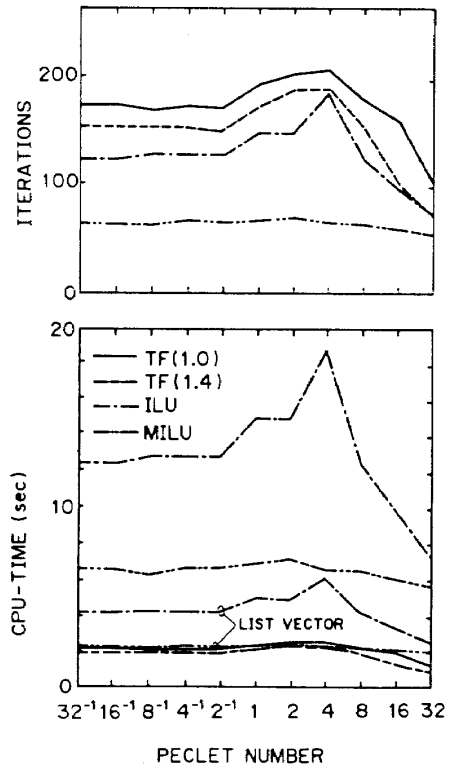

図6 数値実験の結果（実験1）

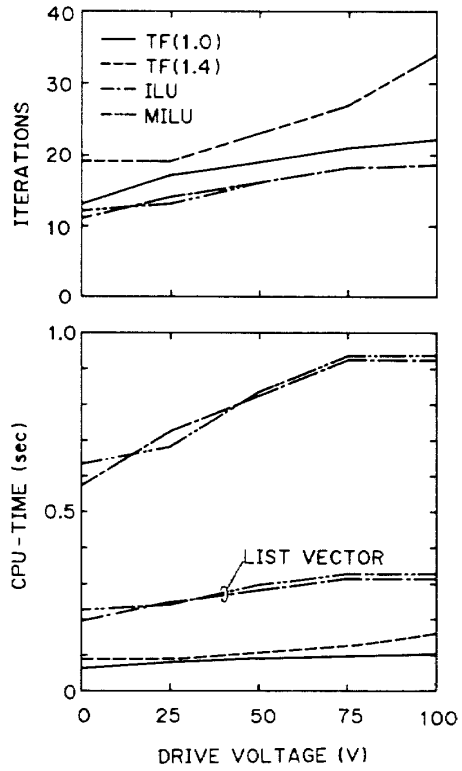

図8 数値実験の結果（実験3, TYPE1, $n_x \times n_y \times n_z = 201 \times 21 \times 6$）

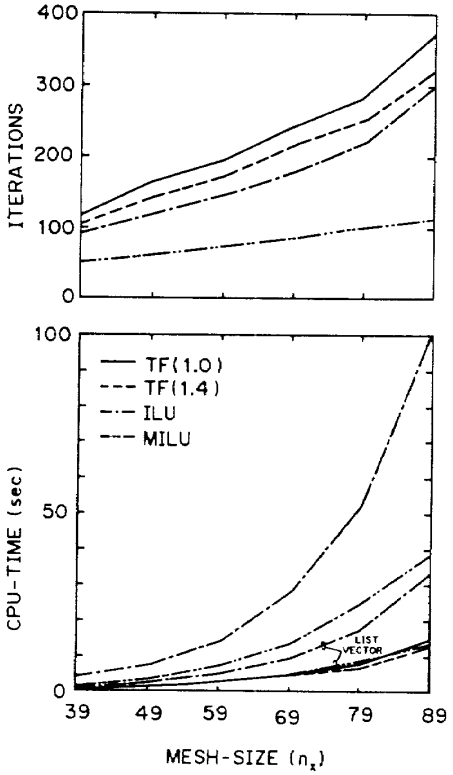

図7 数値実験の結果（実験2, $n_y = n_z = (n_x + 1)/2$）

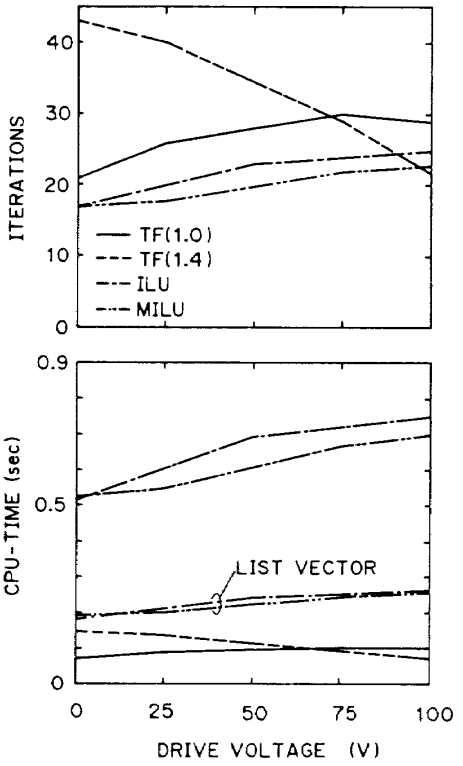

図9 数値実験の結果（実験3, TYPE2, $n_x \times n_y \times n_z = 51 \times 51 \times 6$）

2では効果がない。ILU法に対するGustafssonの改良も同様に例2に対しては殆ど効果がない。TF法の反復数はILU法の1～6割増し程度である。MILU法に対しては例2ではILU法と変わらないが，例1では1.8～3.3倍とやや差が大きい。

一方，計算時間ではTF法は通常のコーディングによる(M)ILU法の1/3～1/9，リストベクトルを用いて強制ベクトル化を行ったILU法の2/3～1/3となった。強制ベクトル版MILU法と比べても例2では1/2～1/4，反復数の差の大きい例1でも同程度となった。なお，例2 TYPE1では370 MFLOPSの演算速度が得られた。

5. むすび

ベクトル計算機上でのTF法の有効性を示した。当社のスーパーコンピュータSX-2での実測では，TFBCG法はリストベクトルにより高速化されたILUBCG法より高速であり，同様のMILUBCG法と比べても同等以上である。プログラムの使い勝手の点でも，リストベクトルによる(M)ILU法のような，"はみ出し部分"を持つ配列[2]の使用やリストの設定等の必要がない。

ここではTF法の前処理法としての評価に重点を置き，基礎反復法としてはBCG法のみを用いた。同様に，CR法やCGS法[8]にも適用できる。それらを含めた，より多くの例題による評価が重要である。

最後に，半導体デバイスの例題を提供下さった当社材料開発試作センターの小田氏に感謝致します。

参考文献

1) 村田健郎：前処理付き共役勾配法・共役残差法，情報処理，Vol.27, No.5, pp.498-507 (1986).

2) 後 保範：ベクトル計算機向きICCG法，京都大学数理解析研究所講究録，No.514, pp. 110-134 (1984).

3) 土肥 俊，原田紀夫：三項対角行列分離による前処理共役勾配法，情報処理学会第32回全国大会予稿集，5E-5 (1986).

4) I. Gustafsson: A Class of First Order Factorization Methods, BIT, No. 18, pp. 142-156, 1978.

5) 野寺 隆：大型疎行列に対するPCG法，Proc. of the SEMINAR ON MATHEMATICAL SCIENCE, No.7, 慶応大 (1983).

6) R.S. バーガ著，渋谷政昭訳：計算機による大型行列の反復解法，サイエンス社.

7) 野木達夫：不完全AD(HV)分解，京都大学数理解析研究所講究録，No.585, pp.240-258 (1986).

8) C. den Heijer: Iterative Methods for Non-symmetric Linear Systems, Proc. of Int. Conf. on Simulation of Semiconductor Devices and Processes, pp. 267-285, SWANSEA, 1984.