

対角項スケーリングを施した共役勾配法のベクトル計算機における有効性について†

速 水 謙^{††} 原 田 紀 夫^{†††}

ベクトル処理を行うスーパーコンピュータの能力を十分に発揮するためにはベクトル処理に適したアルゴリズムの開発が重要である。本論文ではベクトル処理能力を最大限に活用するという立場から、行列の対角項によるスケーリングを施した共役勾配法がベクトル計算機向き的高速な連立1次方程式の解法として有効であることを示す。この解法は偏微分方程式の離散近似等で生じる正定値対称スパースな連立1次方程式の行列を対角項でスケーリングした後共役勾配法を適用するもので、100%ベクトル処理可能なので高度のベクトル計算が実現でき、また高い収束性を保持していることによりベクトル計算機上で高速な計算ができるという特長をもつ。また、本手法は所要記憶容量も少なく済み、プログラムも簡単である。まず、拡散方程式の有限差分近似で生じる行列を例にとり、対角項スケーリングの共役勾配法に対する前処理としての効果を論じ、それが拡散係数およびメッシュ間隔の空間的な非一様性に依存することを示す。次に、3次元拡散方程式の有限差分近似に対する様々な数値実験の結果（使用機種 NEC SX-2）、異方性のない場合には拡散係数が空間的に激しく変動する場合も含めて、本手法はベクトル化 (M) ICCG 法に比してかなりの高速性 (2.5~2.9 倍) をもつこと、また有限要素法などで生じる不規則スパースな問題に対しても有効であることを示す。

1. はじめに

高度のベクトル処理能力をもつスーパーコンピュータの出現に伴い、大規模数値シミュレーションにおいて数値計算アルゴリズムがいかにそのベクトル処理能力を生かすかが重要になってきている。本論文では高度のベクトル計算が可能なアルゴリズムが重要であるという立場から連立1次方程式のベクトル計算機向き解法としての対角項スケーリングを施した共役勾配法の有効性を明らかにする。

近年、偏微分方程式の離散近似に伴う大規模で条件数の大きい対称正定値な連立1次方程式の有力な反復解法として (M) ICCG 法 ((Modified) Incomplete Cholesky-Conjugate Gradient algorithm)^{1), 2), 10)} が提案されている。この方法は従来の SOR 法 (Successive Over Relaxation method), ADI 法 (Alternate Direction Implicit method) 等の反復法およびそれらの改良版に比べ、収束性が極めて良いため、広く使用されている。しかし、ベクトル処理を行うスーパーコンピュータのアルゴリズムという観点からは、(M) ICCG 法はその反復の中にベクトル処理に向かない前進後退代入を含むため、ベクトル処理能力を十分に生かしき

れない面がある。この対策として近似逆行列の Neumann 級数展開などの近似多項式を用いる方法^{11), 12)} などがあるが、複雑である割にはベクトル性能、収束性が思わしくない。また、リストベクトルを用いて (M) ICCG 法の前進後退代入をベクトル化する手法⁹⁾ が提案されているが、間接アドレスを用いるため連続アクセスに比べてデータのアクセスが遅くなり、元数に比べてベクトル長が短いため、ベクトル計算機の性能を十分に生かしていない。

このように (M) ICCG 法は反復数が少ないという極めて良い収束性を有するがベクトル処理に向かない部分を含んでいる。本論文では、アルゴリズムの実行時間の高速性は単なる反復数だけでなく反復数と反復当たりの実行時間の積であるという視点に立ち、もとの行列にその対角項によってスケーリングを施してから共役勾配法 (Conjugate Gradient algorithm)⁴⁾ を適用する方法がベクトル計算機向きのアルゴリズムとして有効であることを明らかにする。この方法は 100%ベクトル処理可能で、所要記憶容量も少なく済み、3次元の拡散方程式の有限差分近似の様々な例題のスーパーコンピュータ上の数値実験の結果、反復数という面では (M) ICCG 法には及ばないが、高度のベクトル処理能力をもつ計算機での実行時間という面ではリストベクトルを用いたベクトル化 (M) ICCG 法より、多くの場合高速であることを示す。ただし、係数やメッシュ間隔の極端な異方性を含む問題に対しては遅くなることもある。

† On the Effectiveness of the Diagonally Scaled Conjugate Gradient Algorithm on Vector Computers by KEN HAYAMI (Information Basic Research Laboratory, C&C Information Technology Laboratories, NEC Corporation) and NORIO HARADA (Scientific Computing Systems Department, EDP Systems Engineering Division, NEC Corporation).

†† 日本電気(株) C&C 情報研究所情報基礎研究部

††† 日本電気(株)情報処理システム技術本部科学技術システム部

2. 従来の解法

従来のアルゴリズムについては広く知られているのでここでは関連する部分について簡単にふれるに止めたい。

2.1 CG 法 (Conjugate Gradient algorithm)⁴⁾

前処理付き共役勾配法の基礎となる共役勾配法 (以下 CG 法と略記する) は, 正定値対称な行列 A による連立1次方程式

$$Ax = b \tag{1}$$

の解 x を, 初期解 x_1 をもとに以下のようにして反復的に求める手法である。

$$\begin{aligned} r_1 &= b - Ax_1 \\ p_1 &= r_1 \\ i &= 1, 2, 3, \dots \\ \alpha_i &= (r_i, r_i) / (p_i, Ap_i) \\ x_{i+1} &= x_i + \alpha_i p_i \\ r_{i+1} &= r_i - \alpha_i Ap_i \\ \beta_i &= (r_{i+1}, r_{i+1}) / (r_i, r_i) \\ p_{i+1} &= r_{i+1} + \beta_i p_i \end{aligned} \tag{2}$$

CG 法そのものは大規模で条件数の大きい問題では収束しなかったり, 収束するまでに膨大な反復数を要する。また, CG 法の収束特性は行列 A の固有値分布に左右され, A の固有値が密集していると収束が速いことが知られている¹³⁾。

2.2 (M) ICCG 法 ((Modified) Incomplete Cholesky-Conjugate Gradient algorithm)

ICCG 法^{1),2)}は CG 法の収束を加速するためにまず行列 A の非零要素に対してのみコレスキー分解を行ういわゆる不完全コレスキー分解を行った後に, CG 法を適用するもので, 詳しくは文献 1)~3), 9) 等を参照されたい。問題点を明らかにするために簡単に記すと,

① A の不完全コレスキー分解

$$A \sim LDL^T \tag{3}$$

を行う。

② $r_1 = b - Ax_1$
 $p_1 = (LDL^T)^{-1}r_1$
 $i = 1, 2, 3, \dots$

$$\begin{aligned} \alpha_i &= \frac{(r_i, (LDL^T)^{-1}r_i)}{(p_i, Ap_i)} \\ x_{i+1} &= x_i + \alpha_i p_i \\ r_{i+1} &= r_i - \alpha_i Ap_i \end{aligned}$$

$$\begin{aligned} \beta_i &= \frac{(r_{i+1}, (LDL^T)^{-1}r_{i+1})}{(r_i, (LDL^T)^{-1}r_i)} \\ p_{i+1} &= (LDL^T)^{-1}r_{i+1} + \beta_i p_i \end{aligned} \tag{4}$$

となる。ここで, 反復ごとに現れる Ap , 内積等は容易にベクトル処理できるが, $s = (LDL^T)^{-1}r$ の計算は

$$\begin{aligned} i &= 1, 2, 3, \dots, n \\ s_i &= \left(r_i - \sum_{j=1}^{i-1} l_{ij} s_j \right) / l_{ii} \end{aligned} \tag{5}$$

$$\begin{aligned} i &= n, n-1, \dots, 1 \\ s_i &= \left(s_i - \sum_{j=i+1}^n l_{ji} s_j \right) / l_{ii} \end{aligned} \tag{6}$$

の前進代入, 後退代入を含み, s_i のデータの依存関係のため効率良くベクトル処理を行うには特別の工夫を要す。

また, 有限差分に対する ICCG 法の収束を加速する手法として, MICCG 法 (Modified ICCG algorithm)^{9),9),10)} が提案されている。これは, 不完全コレスキー分解 $A \sim LDL^T$ で得られる $D: dd_i$ を

$$\begin{aligned} dd_i &= 1 / [d_i - c_i \times dd_{i-1} \times \{c_i + \theta \times (a_{i+m2-1} + b_{i+m1-1})\} \\ &\quad - b_i \times dd_{i-m1} \times \{b_i + \theta \times (c_{i+1-m1} + a_{i+m2-m1})\} \\ &\quad - a_i \times dd_{i-m2} \times \{a_i + \theta \times (b_{i+m1-m2} + c_{i+1-m2})\}] \end{aligned} \tag{7}$$

のように修正したものである。ただし, (7)式は3次元の有限差分で生じる図1のような行列 A の第 i 行の非零要素に対して, 対角を d_i , 第1下対角を c_i , 第 $m1$ 下対角を b_i , 第 $m2$ 下対角を a_i (ただし $m1 < m2$), 加速パラメタを θ とおいたものである。 ($\theta = 0$ のときが ICCG 法に対応する。)

MICCG 法も ICCG 法と同様に前進後退代入のベク

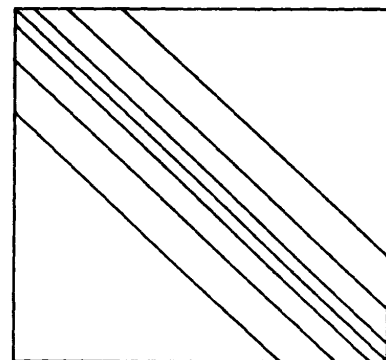


図1 3次元有限差分法による規則スパース行列のパターン

Fig. 1 Regular sparse matrix pattern due to 3 dimensional finite difference method.

トル化に特別の工夫を要す。

3. 対角項スケールリングを施した共役勾配法

3.1 理論的考察

上記の (M) ICCG 法のベクトル化の困難を克服するため、不完全コレスキー分解(3)の代りに、より単純で、ベクトル化が可能のように、行列 A をその対角項

$$D = \text{diag}[a_{11}, a_{22}, \dots, a_{nn}],$$

によりスケールリングすることを考える。すなわち、

$$A' \equiv D^{-1/2} A D^{-1/2} \quad (8)$$

とすれば A' は後に述べるように A より条件の良い正定値対称行列になると考えられる。そこで、

$$\begin{aligned} x' &= D^{1/2} x \\ b' &= D^{-1/2} b, \end{aligned} \quad (9)$$

とおき、もとの方程式(1)と同等な

$$A' x' = b' \quad (10)$$

に CG 法を適用するものである。

このように対角項スケールリングを施した共役勾配法の収束特性は行列 A' の固有値分布によって定まる。ここで、

$$A' = D^{1/2} (D^{-1} A) D^{-1/2} \quad (11)$$

より A' は $D^{-1} A$ と相似であるから、 A' の固有値分布を知るには $D^{-1} A$ の固有値分布を調べれば良い。 $D^{-1} A$ は

$$(D^{-1} A)_{ij} = a_{ij} / a_{ii} \quad (12)$$

で、 A の各行をその行の対角項で割ったものである。

行列のスケールリングの有効性については文献 15) で一般的に論じられているが、ここでは具体的に拡散方程式の有限差分近似で得られる行列に対して、対角項スケールリングによる固有値分布の変化について次の Gerschgorin の定理を用いて簡単に考察し、拡散係数と差分メッシュの及ぼす影響について論じたい。

【定理】 (Gerschgorin)¹⁴⁾

「 $A = (a_{ij})$ に対して、

$$\Delta_i = \{z \mid |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|\}$$

とおくと、 A の固有値はすべて

$$\bigcup_{i=1}^n \Delta_i \text{ 内にある。} \quad \text{」}$$

ここでは例として、3次元拡散方程式

$$\begin{aligned} -\partial/\partial x (k_x \partial u / \partial x) \\ -\partial/\partial y (k_y \partial u / \partial y) \\ -\partial/\partial z (k_z \partial u / \partial z) = f \end{aligned} \quad (13)$$

を中心差分により有限差分近似して得られる正定値対

称行列 A を考える。

行列 A の各行の非零要素は

$$\begin{aligned} -k_x / \Delta x^2, -k_y / \Delta y^2, -k_z / \Delta z^2, & \quad (\text{下対角}) \\ 2(k_x / \Delta x^2 + k_y / \Delta y^2 + k_z / \Delta z^2), & \quad (\text{対角}) \\ -k_x / \Delta x^2, -k_y / \Delta y^2, -k_z / \Delta z^2 & \quad (\text{上対角}) \end{aligned} \quad (14)$$

と考えて良い。ここでスケールリングによる固有値分布の変化を、係数 k が空間的に変化するとして考察する。もとの (実対称) 行列 $A = (a_{ij})$ の固有値 λ の値は上記定理より、

$$\lambda \in \bigcup_{i=1}^n [a_{ii} - \sum_{j \neq i} |a_{ij}|, a_{ii} + \sum_{j \neq i} |a_{ij}|] \quad (15)$$

の範囲にある。ここで係数 k が空間的に変化する場合各対角項 a_{ii} は相異なり、固有値は幅広く分散していると考えられる。

これに対して、スケールリングを施した $D^{-1} A$ の固有値 λ' は同様にして

$$\begin{aligned} \lambda' \in \bigcup_{i=1}^n [1 - \sum_{j \neq i} |a_{ij}| / |a_{ii}|, \\ 1 + \sum_{j \neq i} |a_{ij}| / |a_{ii}|] \end{aligned} \quad (16)$$

に含まれることがわかる。ここで(14)式より

$$\sum_{j \neq i} |a_{ij}| / |a_{ii}| = 1 \quad (17)$$

に注意すれば $D^{-1} A$ の固有値は区間 $[0, 2]$ 内に限定され、1の近傍に集中するものと考えられる。したがってもとの行列 A の固有値分布の範囲が限定されないのに対してスケールリングを施したものが固有値が密集し、CG法の収束性が改善されるものと考えられる。

以上よりスケールリングは係数 k が空間的に (したがって行列の行ごとに) 変化する場合に有効に働く。同様にして、差分のメッシュ間隔が空間的に変化する不等間隔メッシュの場合も (14) 式の対角項の $\Delta x, \Delta y, \Delta z$ が行ごとに変化するのでスケールリングが有効に働くことがわかる。

3.2 アルゴリズム

上記の (8) 式のスケールリングと (9) 式の変換で得られるもとの方程式 (1) と同様な

$$A' x' = b' \quad (10)$$

に CG 法を適用し、もとの変数 A, x, b 等で表すことにより次のアルゴリズムを得る。

- ① $D^{-1} = \text{diag}[1/a_{11}, 1/a_{22}, \dots, 1/a_{nn}]$ を求める。
- ② $r_1 = b - A x_1$
 $p_1 = D^{-1} r_1$

$$\begin{aligned}
 & i=1, 2, 3, \dots \\
 & \alpha_i = \frac{(\mathbf{r}_i, D^{-1}\mathbf{r}_i)}{(\mathbf{p}_i, A\mathbf{p}_i)} \\
 & \mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i \\
 & \mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i A\mathbf{p}_i \\
 & \beta_i = \frac{(\mathbf{r}_{i+1}, D^{-1}\mathbf{r}_{i+1})}{(\mathbf{r}_i, D^{-1}\mathbf{r}_i)} \\
 & \mathbf{p}_{i+1} = D^{-1}\mathbf{r}_{i+1} + \beta_i \mathbf{p}_i
 \end{aligned} \tag{18}$$

ただし、 $A\mathbf{p}$, $D^{-1}\mathbf{r}$, 内積等は反復当たり1回計算すれば良い。

このアルゴリズムをここでは簡便のため仮に Scaled Conjugate Gradient algorithm と呼び、以下 SCG 法と略記する^{6)~8)}。SCG 法は、ICCG 法のような反復ごとの前進後退代入(5), (6)式を含まないので、行列 A の非零要素を格納する適切なデータ構造を考慮すれば容易に 100% ベクトル処理可能である。表 1 に一般の場合の SCG 法と (M) ICCG 法の反復当たりの演算量と必要な記憶領域を比較する。反復当たりの演算量、記憶領域とも SCG 法は (M) ICCG 法より少なく済む⁹⁾。また、SCG 法は ICCG 法に比べてプログラムも簡単である。

CG 法の前処理として行列の対角項によるスケーリングを用いること自体はたとえば文献 5), 13) 等で既に提案されているが、本論文では、高度のベクトル処理能力をもつ計算機におけるアルゴリズムとしての高速性に着目しその効果を種々の場合に対して実証するものである。また、前処理としてのスケーリングがどのような場合に有効に働くかを分析する。

4. 有限差分法に対する数値実験結果

ここでは、偏微分方程式の有限差分近似で生じる規則スパースな連立1次方程式に対して SCG 法、CG 法、ベクトル化 (M) ICCG 法を数値実験により比較

表 1 SCG 法と (M) ICCG 法の反復当たりの演算量、記憶領域

Table 1 Computation per iteration and required memory for SCG and (M) ICCG algorithms.

	反復当たりの演算量	記憶領域
SCG 法	$2nz + 12n$	$0.5nz + 5.5n$
(M) ICCG 法	$4nz + 12n$	$\frac{nz + 5n}{(0.5nz + 5.5n)}$

注) n : 元数, nz : 行列 A の非零要素数。

* 表 1 の (M) ICCG 法の記憶領域が $(0.5nz + 5.5n)$ とあるのは有限差分に伴う規則スパース行列の場合に、 L と A の下対角部分の記憶領域を共有させた場合である。

する。

4.1 数値例

数値例としては 3 次元拡散方程式

$$\begin{aligned}
 & -\partial/x(k_x \partial u/\partial x) \\
 & -\partial/\partial y(k_y \partial u/\partial y) \\
 & -\partial/\partial z(k_z \partial u/\partial z) = f
 \end{aligned} \tag{19}$$

を図 2 の各直方体領域 (MODEL-A~D³⁾, E) を $m \times m \times m$ に等分割し、ますめに変数を対応させた 7 点中心差分で解く。境界条件は図 2 で現れている面に対しては $\partial u/\partial n = 0$, 現れていない面に対しては $u = 0$ とする。

まず、行列の次数の及ぼす影響を調べるためにケース①では各方向のメッシュ数 m を変化させる。次にケース②では拡散係数 k の空間的な変動による行列の悪条件の影響を調べる。さらにケース③, ④では各々、拡散係数 k および離散化のメッシュ間隔の異方性の影響について調べる。

ケース① (メッシュ数 m の変化)

MODEL-A で各方向のメッシュ数を $m=10, 20,$

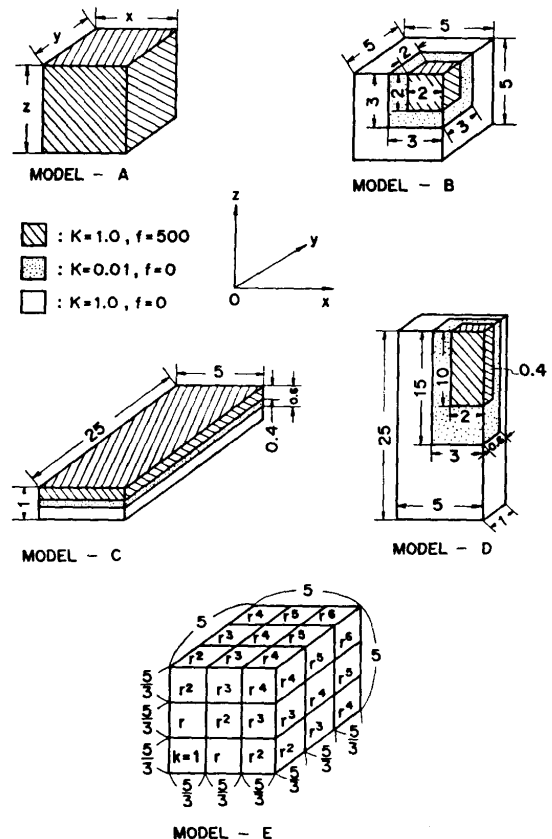


図 2 3次元拡散方程式の計算領域
Fig. 2 Computational domain for 3 dimensional diffusion equation.

30, 40, 50 と変化させる. したがって m^3 の元数の連立 1 次方程式を得る.

(ただし $x=y=z=5$, $k_x=k_y=k_z=1$, 領域で一様に $f=500$.)

ケース② (係数 k の空間変化)

(1) MODEL-B, C, D に示すように係数 k および熱源 f が分布するものとする.

(ただし, $k=k_x=k_y=k_z$, メッシュ数 $m=20$.)

(2) MODEL-E に示すように係数 $k=k_x=k_y=k_z$ が空間的に 1 から r^6 まで変化する. ただし, $r^6=10^3, 10^6, 10^9$.

(ただし $x=y=z=5$, $m=20$, $f=500$.)

ケース③ (係数 k の異方性)

MODEL-A で

(i) $k_x=1, k_y=5, k_z=25$

(ii) $k_x=1, k_y=10, k_z=100$

(ただし, $x=y=z=5$, $m=20$, $f=500$.)

ケース④ (メッシュ間隔の異方性)

MODEL-A で

(i) $x=2, y=5, z=10$

(ii) $x=1, y=5, z=25$

(ただし $k_x=k_y=k_z=1$, $m=20$, $f=500$.)

以上でメッシュの番号付けは図 2 で x, y, z 方向の順に付け, 行列 A の非零要素の値は, 図 1 のような対角方向の 1 次元配列に格納し, 行列とベクトルの積のベクトル処理を容易にした. また各算法とも解の初期ベクトルはすべて $x_1=0$ とし, 収束の判定には相対残差 L_2 ノルム:

$$\|b - Ax_i\|_2 / \|b\|_2 \quad (20)$$

を用いた.

なお, 計算はすべてスーパーコンピュータ SX-2 上で行ったが, アルゴリズムの本質的な性格は一般のベクトル計算機においても変わらない.

4.2 ICCG 法と MICCG 法の比較

まず MICCG 法の ICCG 法に対する優位性を確認するために, ケース①~④に対して各算法で相対残差 L_2 ノルムが 10^{-6} に達するのに要した反復数を表 2 に比較する.

MICCG 法は加速パラメータ $\theta=0.975, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0$ (ICCG) のうち最適であったものを示した.

行列の次数の増加につれて MICCG 法の ICCG 法に対する加速の効果が増す. また, 次数を $m=20$ の場合に固定した場合, 係数 k およびメッシュ間隔に異

表 2 ICCG 法と MICCG 法の比較
Table 2 Comparison of ICCG and MICCG algorithms.

ケース	反復回数の比	
	ICCG 法	MICCG 法の最適パラメータ θ
① 行列の次数の変化 (MODEL-A)		
$m=10$	1.14	0.9
$m=20$	1.55	0.95
$m=30$	1.80	0.975
$m=40$	1.90	0.975
$m=50$	1.97	0.975
② 係数 k の空間変化 ($m=20$)		
MODEL-A	1.55	0.95
MODEL-B	1.59	0.95
MODEL-C	2.56	0.975
MODEL-D	1.58	0.975
MODEL-E		
(i) $r^6=10^3$	1.58	0.975
(ii) $r^6=10^6$	1.45	0.9
(iii) $r^6=10^9$	1.36	0.95
③ 係数 k の異方性 ($m=20$)		
(i) $k_x=1, k_y=5, k_z=25$	1.86	0.975
(ii) $k_x=1, k_y=10, k_z=100$	1.81	0.975
④ メッシュ間隔の異方性 ($m=20$)		
(i) $x=2, y=5, z=10$	1.82	0.975
(ii) $x=1, y=5, z=25$	1.59	0.975

注) 打ち切り相対残差 L_2 ノルムは 10^{-6} .

方性がある場合は, そうでない場合に比べて MICCG 法の ICCG 法に対する加速の効果が増す傾向がある. MODEL-C では特に MICCG 法の効果が顕著である.

MICCG 法のパラメータ θ は問題により $0.9 \sim 0.975$ が最適であった.

4.3 CG 法と SCG 法の比較

次に, SCG 法におけるスケールリングの前処理としての効果を評価するために, 前処理をしないただの CG 法と SCG 法を比較する. 表 3 にケース①~④に対して CG 法で打ち切り相対残差 L_2 ノルム 10^{-6} を達成するのに要したベクトル処理の実行時間と反復数, および SCG 法との比較を示す.

CG 法と SCG 法の反復数の比較より, 拡散係数 k が空間的に変化する場合はスケールリングが有効に働いている. (特に MODEL-E (ii), (iii) では CG 法は

表 3 SCG 法と CG 法の比較 (ベクトル実行)
Table 3 Comparison of SCG and CG algorithms
(vector processed).

ケース	CG 法		CG/SCG	
	実行時間 (msec)	反復数	実行時間	反復数
① 行列の次数の変化 (MODEL-A)				
$m=10$	1.54	41	0.933	1.05
$m=20$	19.8	80	0.900	1.00
$m=30$	97.5	121	0.886	1.01
$m=40$	309	162	0.885	1.01
$m=50$	754	203	0.898	1.00
② 係数 k の空間変化 ($m=20$)				
MODEL-A	19.8	80	0.900	1.00
MODEL-B	72.5	294	3.05	3.42
MODEL-C	142	576	1.01	1.13
MODEL-D	239	970	2.83	3.18
MODEL-E				
(i) $r^s=10^s$	189	768	7.38	8.26
(ii) $r^s=10^s$	収束せず	10,000	—	—
(iii) $r^s=10^s$	収束せず	10,000	—	—
③ 係数 k の異方性 ($m=20$)				
(i) $k_x=1, k_y=5,$ $k_z=25$	38.7	157	0.872	1.02
(ii) $k_x=1, k_y=10,$ $k_z=100$	45.9	186	0.913	1.02
④ メッシュ間隔の異方性 ($m=20$)				
(i) $x=2, y=5, z=10$	39.8	161	0.900	1.03
(ii) $x=1, y=5, z=25$	44.5	177	0.908	1.02

注) 打ち切り相対残差 L_2 ノルムは 10^{-6} .

10,000 回の反復でも収束しないが SCG 法は 100 回程度で収束する。) これは 3.1 節で述べた理論的考察によって説明される。

拡散係数 k が空間的に一様な場合は (係数 k , メッシュ間隔 Δ 等が異方的な場合も含めて) 前処理としてのスケーリングがほとんど効いていない。これは係数 k が空間的に一様な場合は (14) 式からもわかるように行列 A の各行の対角項が (境界条件に対応する行の対角項を除けば) 等しいため、 $D^{-1}A$ と A の固有値分布は定数倍を除いてほぼ同等だからである。したがってこの場合は、SCG 法は (18) 式で反復ごとに $D^{-1}r$ を 1 回計算する分だけ CG 法に比べて 10% ほど実行時間が増す。

なお、反復ごとに残差を計算する必要がなければ、スケーリングを施した方程式 (10) にそのまま CG 法を適用し、残差ベクトル r を知りたいときだけ $r = D^{1/2}r'$

により求め、収束解を $x = D^{-1/2}x'$ より求めることにより SCG 法の計算時間は 10% 近く節約できるが、今回の数値実験では (18) 式のアルゴリズムにより反復ごとに残差計算を行わせた。

4.4 ベクトル化 (M) ICCG 法と SCG 法の比較

3.2 節で述べたように SCG 法は容易に 100% ベクトル処理可能で、ケース①の問題のサイズ $m=10\sim 50$ に応じてベクトル処理による加速率は 36~51 倍である。(ちなみに、ケース①、 $m=50$ での SCG 法の演算処理速度は 788 MFLOPS である。)

一方、2.2, 2.3 節で述べたように、(M) ICCG 法はそのままでは前進後退代入の部分が効率良くベクトル化されないため、ベクトル化率は 62% にとどまり、ケース①でのベクトル処理による加速率は 2.5 倍にとどまる。

(M) ICCG 法をベクトル化する試みも多くなされており^{11), 12)}、その中でもリストベクトルを用いる方法^{3), 9)}が有効とされている。これは前進後退代入などで、互いにデータ依存関係のない節点 (有限差分メッシュの場合 x, y, z 方向の格子座標 i, j, k が $i+j+k =$ 一定を満たす面内の節点) に関してリストベクトルを用いてベクトル化する手法である。ここでは、この手法によってベクトル化した (M) ICCG 法と SCG 法を比較する*。

まず、ケース①、MODEL-A、 $m=20$ に対する SCG 法とベクトル化 (M) ICCG 法の相対残差 L_2 ノルム (対数スケール) の推移を図 3 では反復数、図 4 ではスカラ処理したときの実行時間、図 5 ではベクトル処理したときの実行時間、を各々横軸にとったグラフで示す。

図 3 からわかるように (M) ICCG 法は SCG 法の半分以下の反復数で収束する。ところが、図 4 でスカラ処理の実行時間で比較すると SCG 法は ICCG 法より速く収束する。これは、表 1 で 3 次元有限差分法の場合 $nx \sim 7n$ とおくと反復当たりの演算量では SCG 法は (M) ICCG 法の約 2/3 であることと、ベクトル化 (M) ICCG 法でのリスト処理の負荷によるものと考えられる。

* ベクトル化 (M) ICCG 法のプログラムを作成して頂いた日本電気技術情報システム開発(株) 垣 紀子氏に感謝いたします。

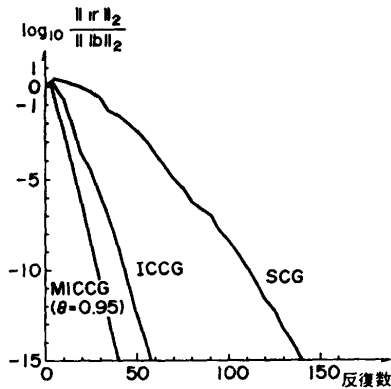


図3 SCG法と(M)ICCG法の反復数 vs. 相対残差 L_2 ノルム (ケース①, MODEL-A, $m=20$)

Fig. 3 Iterations vs. relative residual L_2 norm for SCG and (M) ICCG algorithms (Case ①, MODEL-A, $m=20$).

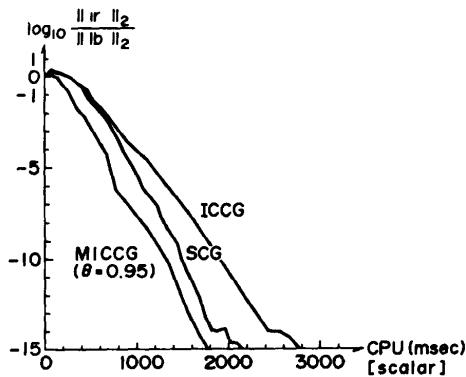


図4 SCG法と(M)ICCG法のスカラ処理実行時間 vs. 相対残差 L_2 ノルム (ケース①, MODEL-A, $m=20$)

Fig. 4 Scalar processed execution time vs. relative residual L_2 norm for SCG and (M) ICCG algorithms (Case ①, MODEL-A, $m=20$).

さらに、図5よりベクトル処理の実行時間では、SCG法がベクトル化(M)ICCG法の2倍以上速く収束する。これは、ベクトル化率ほどの手法も100%でも、後に示すようにSCG法の方がベクトル長が長く、リストによるアドレスの間接参照がないため、反復当たりの実行時間がベクトル化(M)ICCG法の1/10だからである。(なお図4、図5でスカラ処理とベクトル処理の収束状況が相対残差 10^{-15} 付近で微妙に異なるのは各処理での内積などにおける加算の順序の違いから生じるものと考えられる。)

図6にSCG法に最も不利な例として、ケース②MODEL-Cに対して図5と同様に相対残差 L_2 ノルム vs. ベクトル処理の実行時間のグラフを示す。この場合はSCG法はベクトル化ICCG法よりは速いが、

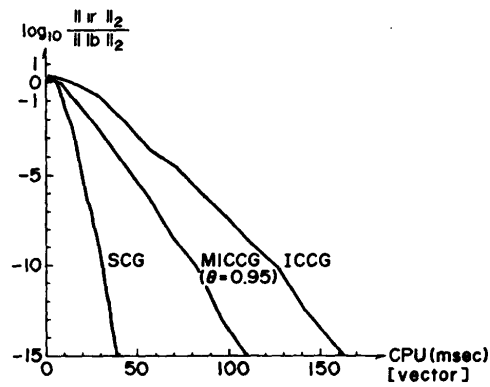


図5 SCG法と(M)ICCG法のベクトル処理実行時間 vs. 相対残差 L_2 ノルム (ケース①, MODEL-A, $m=20$)

Fig. 5 Vector processed execution time vs. relative residual L_2 norm for SCG and (M) ICCG algorithms (Case ①, MODEL-A, $m=20$).

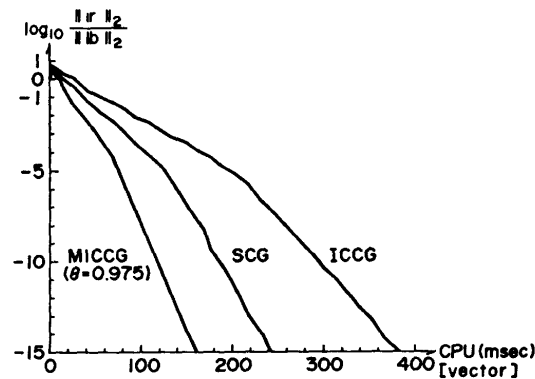


図6 SCG法と(M)ICCG法のベクトル処理実行時間 vs. 相対残差 L_2 ノルム (ケース②, MODEL-C, $m=20$)

Fig. 6 Vector processed execution time vs. relative residual L_2 norm for SCG and (M) ICCG algorithms (Case ②, MODEL-C, $m=20$).

ベクトル化MICCG法($\theta=0.975$)の実行時間はSCG法の約2/3になっている。

次に表4にケース①~④の3次元問題に対してSCG法と、リストベクトルを用いてベクトル化した(M)ICCG法のベクトル処理の実行時間を比較する。MICCG法は表2と同じ最適な θ の値を用いた。打ち切り相対残差 L_2 ノルムは 10^{-6} である。

表4よりもまずSCG法とベクトル化ICCG法のベクトル処理実行時間を比較する。①の行列の次数の変化に対しては、SCG法はベクトル化ICCG法の3.3~4.7倍速く、行列の次数の増加につれてその比は3.3倍まで漸的に減少する。

一方向のメッシュ数を $m=20$ に固定すると、係数

k とメッシュ間隔が等方的な場合は、

① (MODEL-A) や、係数 k が空間的に激しく変化
する②の MODEL-B, E でも、SCG 法はベクトル
化 ICCG 法の 3.8~4.0 倍速い。係数 k が異方的な③
では SCG は ICCG の 2.0~2.5 倍速く、メッシュ間
隔が異方的な②の C, D や④では SCG は ICCG の
1.5~2.4 倍速い。

次に SCG 法とベクトル化 MICCG 法のベクトル処
理実行時間を比較する。①の行列の次数の変化に対し
ては、SCG 法はベクトル化 MICCG 法の 1.7~4.2 倍
速く、行列の次数の増加につれてその比は 1.7 倍まで
漸近的に減少する。

一方向のメッシュ数を $m=20$ に固定すると、係数
 k とメッシュ間隔が等方的な場合は、① (MODEL-
A) や、係数 k が空間的に激しく変化する②の MOD-
EL-B, E でも、SCG 法はベクトル化 MICCG 法の

2.5~2.9 倍速い。係数 k が異方的な③では SCG は
MICCG の 1.2~1.4 倍速く、メッシュ間隔が異方
的な②の D や、④では SCG は MICCG の 1.0~1.4
倍速い。②の MODEL-C のメッシュ間隔の異方性
と係数 k の空間変化が重なった場合のみ SCG 法はベ
クトル化 MICCG 法の 0.62 倍の速さにとどまってい
る。

以上で SCG 法はベクトル化 MICCG 法に比べて異
方性を含む問題に対しては同等かやや不利になる傾向
がある。特にメッシュ間隔の異方性は、(14)式からも
わかるように 2 乗 (Δx^2 等) できいてくるのでその
影響が大きい。

これは 4.3 節でも述べたように SCG 法のスケーリ
ングの前処理は係数 k およびメッシュ間隔 (Δx 等)
の空間的な変動に対して有効に作用するが、係数 k お
よびメッシュ間隔が空間的に一樣な場合にはほとんど

表 4 SCG 法とベクトル化 (M) ICCG 法の比較 (ベクトル実行)
Table 4 Comparison of SCG and vectorized (M) ICCG algorithms (vector processed).

ケース	SCG 法			ベクトル化 ICCG 法		ICCG / SCG (CPU)	ベクトル化 MICCG 法 (θ 最適)		MICCG / SCG (CPU)
	CPU 時間 (msec)	反復数	ベクトル 加速率	実行時間 (msec)	反復数		実行時間 (msec)	反復数	
① 行列の次数の変化 (MODEL-A)									
$m=10$	1.65	39	36	7.79	16	4.7	6.87	14	4.2
$m=20$	22.0	80	48	84.0	31	3.8	55.0	20	2.5
$m=30$	110	120	50	381	45	3.5	215	25	2.0
$m=40$	349	161	50	1,160	59	3.3	617	31	1.8
$m=50$	840	202	51	2,760	73	3.3	1,410	37	1.7
② 係数 k の空間変化 ($m=20$)									
MODEL-A	22.0	80	48	84.0	31	3.8	55.0	20	2.5
MODEL-B	23.8	86	48	94.6	35	4.0	60.3	22	2.5
MODEL-C	141	509	48	219	82	1.6	86.7	32	0.62
MODEL-D	84.4	305	47	132	49	1.6	84.0	31	1.0
MODEL-E									
(i) $r^*=10^*$	25.6	93	48	103	38	4.0	65.6	24	2.6
(ii) $r^*=10^*$	28.7	103	47	113	42	4.0	78.9	29	2.8
(iii) $r^*=10^*$	31.0	113	48	122	45	3.9	89.4	33	2.9
③ 係数 k の異方性 ($m=20$)									
(i) $k_x=1, k_y=5, k_z=25$	44.4	154	46	111	41	2.5	60.5	22	1.4
(ii) $k_x=1, k_y=10, k_z=100$	50.3	182	47	103	38	2.0	58.0	21	1.2
④ メッシュ間隔の異方性 ($m=20$)									
(i) $x=2, y=5, z=10$	44.2	157	47	108	40	2.4	60.5	22	1.4
(ii) $x=1, y=5, z=25$	49.0	174	47	73.5	27	1.5	47.1	17	0.96

注) 打ち切り相対残差 L_2 ノルムは 10^{-6} .

作用しないため、異方性による行列の条件の悪化に対して SCG 法の反復数は増加するのに対して、(M) ICCG 法の前処理は空間的に一樣な場合にも作用し、特に異方性に対して顕著に働くためである。

SCG 法を用いる場合の異方性の対策としては(14)式で

$$k_x/\Delta x^2 \sim k_y/\Delta y^2 \sim k_z/\Delta z^2,$$

となるようなメッシュ間隔を用いることが考えられる。これは物理的な意味、あるいは離散化誤差を減らす意味でも有用である。それは、 $k_x/\Delta x^2$, $k_y/\Delta y^2$, $k_z/\Delta z^2$ の値が極端に異なるような異方性のメッシュを用いた場合、シミュレーションの精度が落ちる恐れがあるからである。

次に表5に SCG 法とベクトル化 MICCG 法の反復回数、反復当たりのベクトル実行時間、全ベクトル実行時間を比較する。

表1より、反復当たりの演算量では MICCG 法は SCG 法の約 1.5 倍だが、表5より反復当たりのベクトル実行時間の比は約 10 倍である。これは、SCG 法ではベクトル長が行列の次数 $m^3=1,000\sim 125,000$ であるのに対し、ベクトル化 (M) ICCG 法ではベクトル長はデータの依存関係がない面上の節点数 (m^2 のオーダー) しかとれないこと、SCG 法では連続アドレス参照が生かせるのに対し、ベクトル化 (M) ICCG 法ではリストベクトルを用いるため、より負荷の大きい間接アドレス参照を必要とするためである。ただし、ベクトル計算機のハードウェアによってベクトル長の効果、アドレスの連続参照と間接参照の相対速度は異なるので、上記の反復当たりのベクトル実行時間の比も異なる可能性がある。

以上をまとめると、スーパーコンピュータ上の数値実験の結果、3次元拡散方程式の有限差分近似で、拡散係数 k およびメッシュ間隔が等方的な場合は、 k が空間的に激しく変化する条件の悪い問題も含めて、SCG 法はベクトル化 MICCG 法よりベクトル処理の実行時間の点でより高速である。係数 k や差分のメッシュ間隔に異方性がある場合は SCG 法の効果は相対的に半減するが、それでもほとんどの場合ベクトル化 MICCG 法とほぼ同等かより高速であることが示された。ここで得られた結果は、ベクトル計算機のハードウェアによって定量的には異なる可能性があるが、高度のベクトル処理能力をもつスーパーコンピュータにおける、行列の対角項によるスケールリングを施した共役勾配法の有効性を示している。

表5 SCG 法とベクトル化 MICCG 法の比較 (MICCG/SCG)

Table 5 Comparison of SCG and vectorized MICCG algorithms.

ケース	反復回数の比	1反復当たりのベクトル実行時間の比	全ベクトル実行時間の比
① 行列の次数の変化 (MODEL-A)			
$m=10$	0.359	11.6	4.16
$m=20$	0.250	10.0	2.50
$m=30$	0.208	9.38	1.95
$m=40$	0.193	9.17	1.77
$m=50$	0.183	9.16	1.68
② 係数 k の空間変化 ($m=20$)			
MODEL-A	0.250	10.0	2.50
MODEL-B	0.256	9.89	2.53
MODEL-C	0.0629	9.78	0.615
MODEL-D	0.102	9.78	0.995
MODEL-E			
(i) $r^s=10^3$	0.258	9.93	2.56
(ii) $r^s=10^4$	0.282	9.75	2.74
(iii) $r^s=10^5$	0.292	9.89	2.88
③ 係数 k の異方性 ($m=20$)			
(i) $k_x=1, k_y=5, k_z=25$	0.143	9.55	1.36
(ii) $k_x=1, k_y=10, k_z=100$	0.115	10.0	1.15
④ メッシュ間隔の異方性 ($m=20$)			
(i) $x=2, y=5, z=10$	0.140	9.75	1.37
(ii) $x=1, y=5, z=25$	0.0977	9.82	0.961

注) 打ち切り相対残差 L_2 ノルムは 10^{-4} .

5. 不規則スパース行列への SCG 法の適用

5.1 データ格納方式

次に、不規則スパースで正定値対称な大規模連立1次方程式への SCG 法の適用を検討する。

有限要素法などで生じる不規則スパース行列に対して SCG 法を適用する際、ベクトル計算機上で効率良く行列とベクトルの積を計算するための行列のデータ構造が重要となる。

従来、対称な不規則スパース行列を格納するのに次のような行方向リストがよく用いられてきた。

$A(k)$: 行列 A の対角, 上対角の非零要素

$JA(k)$: $A(k)$ の列番号

$IA(i)$: i 行目の対角要素の要素番号

N : 元数

この行方向リストで行列とベクトルの積 $s=Ap$ を計

算するには、

$$\begin{aligned}
 & i=1, 2, \dots, N \\
 & \quad S(i) = A(IA(i)) * P(i) \\
 & i=1, 2, \dots, N \\
 & \quad k = IA(i)+1, \dots, IA(i+1)-1 \\
 & \quad \quad S(i) = S(i) + A(k) * P(JA(k)) \\
 & \quad \quad S(JA(k)) = S(JA(k)) + A(k) * P(i)
 \end{aligned} \tag{21}$$

とすれば良い。

しかし、有限要素法などでは1節点につながる節点数は限られているので、(21)式の k による最深 do loop 長は短いため、高いベクトル性能を引き出しにくい。そこで、上記の問題を克服する、ベクトル計算機向きのデータ構造として、図7のような対角方向リストを提案する。図7で、

- $AD(j)$: 行列 A の対角要素,
- $AU(m)$: 行列 A の上対角非零要素,
- $JC(m)$: $AU(m)$ の列番号,
- $KD(i)$: i 本目の対角方向リストは第 $KD(i)$ 上対角上にある,
- $ID(i)$: i 本目の対角方向リストの先頭要素の要素番号,
- NT : 対角方向リストの本数,
- N : 元数,

とすると、対角方向リストで行列とベクトルの積 $s = Ap$ を計算するには、

$$\begin{aligned}
 & j=1, 2, \dots, N \\
 & \quad S(j) = AD(j) * P(j) \\
 & i=1, 2, \dots, NT \\
 & \quad k = KD(i) \\
 & \quad m = ID(i), \dots, ID(i+1)-1 \\
 & \quad \quad j = JC(m) \\
 & \quad \quad S(j-k) = S(j-k) + AU(m) * P(j) \\
 & \quad \quad S(j) = S(j) + AU(m) * P(j-k)
 \end{aligned} \tag{22}$$

とすれば良い。

有限要素メッシュが規則的な場合は、図1の有限差分の場合と似て行列の非零要素は対角方向に並ぶ。また複雑なメッシュでも空間的に規則的に(たとえば x, y, z 方向の順に)番号付けを行えば、対角方向に並ぶ傾向をもつ。したがって一般に(22)式の対角方向リストに沿った、 m による最深 do loop 長(ベクトル長)は行方向リストの場合に比べて長いと考えられる。また、do loop 内では $S(j), P(j)$ 等の引数は単調

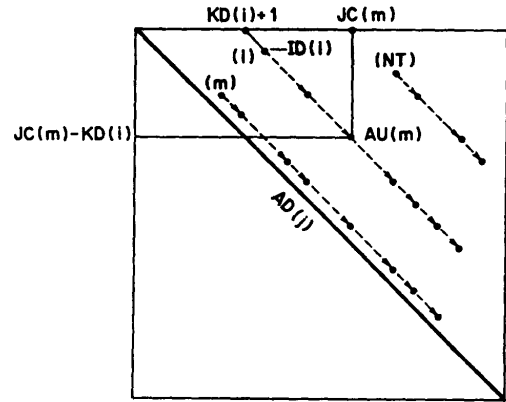


図7 ランダム・スパース行列の対角方向リストデータ格納方式

Fig. 7 Diagonally oriented list data structure for random sparse matrices.

増加であるから、ベクトル処理に際するデータ参照関係の問題も生じない。

5.2 有限要素法に対する数値実験結果

3次元の複雑な形状での静電界の有限要素解析に対して、SCG法を適用した数値実験結果を示す。計算機はスーパーコンピュータ SX-2 を用いた。表6に行方向リスト、対角方向リストを用いた SCG 法について、打ち切り相対残差 L^2 ノルムを 10^{-15} としてベクトル処理の実行時間と反復数を比較した。

対角方向リストを用いた SCG 法は、行方向リストを用いた SCG 法に比べてベクトル処理の実行時間で2~3倍速い。これは表7に示すように、行方向リストの平均ベクトル長が6であるのに対して、対角方向リストの平均ベクトル長が25~64と長くなっている

表6 3次元有限要素法で生じるランダム・スパース行列に対する SCG 法(対角方向、行方向リスト)の比較

Table 6 Comparison of SCG algorithms using row oriented and diagonally oriented lists for random sparse matrices arising in 3 dimensional finite element analysis.

	ケース1	ケース2	ケース3	ケース4
元数	1,167	2,039	3,575	6,309
ベクトル実行時間 (msec)				
行方向 SCG 法	50.4	93.9	157	278
対角方向 SCG 法	24.1	36.9	55.9	94.3
行方向 / 対角方向	2.1	2.5	2.8	2.9
両 SCG 法の反復数	19	20	20	21

注) 打ち切り相対残差 L_2 ノルムは 10^{-15} .

表7 ランダム・スパース行列に関するデータ
Table 7 Data concerning the random sparse matrices.

	ケース1	ケース2	ケース3	ケース4
元数 (N)	1,167	2,039	3,575	6,309
上対角要素数 (MX)	6,584	12,702	20,887	36,221
半バンド幅	327	327	420	691
対角方向リストの本数 (NT)	267	282	359	568
対角方向リストの平均ベクトル長 (MX/NT)	25	45	58	64
行方向リストの平均ベクトル長 (MX/N)	6	6	6	6

のが主な原因と考える。なお表7のデータは、節点は空間的に複雑に分布しており要素の大きさもまちまちであるが、節点番号は空間的に規則的に付けられており、半バンド幅が小さいので、対角方向リストに向いている。逆にそのように空間的に規則的な番号付けを行うことが対角方向リストを生かす上で重要である。より大規模な問題や、より差分メッシュに近い規則性をもつ有限要素メッシュに対しては対角方向リストはさらに威力を発揮すると考えられる*。

6. おわりに

本論文では、偏微分方程式の離散近似などで生じる正定値対称で大規模スパースな連立1次方程式のベクトル計算機向きの反復解法として、行列の対角項によるスケールを施した共役勾配法が有効であることを明らかにした。

本手法は、反復数が少し増す傾向にあるが、反復当たりの処理が少ない上100%ベクトル処理可能で容易に高速ベクトル計算が実現できるのでベクトル計算機上の実行時間の面で有効である。また、本手法は所要記憶量も少なく済み、プログラムが簡単である。

3次元の拡散方程式の有限差分近似で生じる様々な問題に対してスーパーコンピュータ (NEC SX-2) 上の数値実験より、差分メッシュに極端な異方性のある場合を除いて、本手法はベクトル化 (M) ICCG 法より高速であることを示した。

また、有限要素法などで生じる対称不規則スパース行列に対して、本手法のベクトル処理に適した対角方

向リストによるデータ格納方式を提案し、その有効性を示した。

今後ベクトル処理を行うスーパーコンピュータによる大規模な数値シミュレーションがますます重要となってくる。本手法はこれに対する有力なアルゴリズムの一つと考える。

謝辞 貴重な助言を頂いた査読者の方々に感謝いたします。

参考文献

- 1) Meijerink, J. A. and Van der Vorst, H. A.: Iterative Solution Method for Linear Systems of Which the Coefficient Matrix Is a Symmetric M-matrix, *Math. Comp.*, Vol. 31, pp. 148-162 (1977).
- 2) Kershaw, D. S.: The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations, *J. Comp. Phys.*, Vol. 26, pp. 43-65 (1978).
- 3) 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, 京都 (1984).
- 4) Hestenes, M. R. and Stiefel, E.: Method of Conjugate Gradients for Solving Linear Systems, *J. Res. Nat. Bur. Stand.*, No. 49, pp. 409-436 (1952).
- 5) Fox, R. L. and Stanton, E. L.: Developments in Structural Analysis by the Direct Energy Minimization, *AIAA*, Vol. 6, No. 6, pp. 1036-1042 (1968).
- 6) 速水 謙, 原田紀夫: Scaled CG 法再考, 第30回情報処理学会全国大会論文集, pp. 1731-1732 (1985).
- 7) Hayami, K. and Harada, N.: The Scaled Conjugate Gradient Method and Vector Processors, *Proceedings of the First International Conference on Supercomputing Systems*, pp. 213-221, Florida (1985).
- 8) 速水 謙, 原田紀夫: ベクトル計算機における Scaled CG 法の有効性について, 情報処理学会数値解析研究会資料, 17-4 (1986).
- 9) 村田健郎, 小国 力, 唐木幸比古: スーパーコンピュータ (科学技術計算への応用), 丸善, 東京 (1985).
- 10) Gustafsson, I.: A Class of First Order Factorization Methods, *BIT*, Vol. 18, pp. 142-156 (1978).
- 11) Johnson, O. G. and Paul, G.: Optimal Parameterized Inverse Preconditioning for Conjugate Gradient Calculations, *Res. Rep.*, RC 8644, IBM Research Center, Yorktown Heights, NY (1981).

* 3次元静電界有限要素法の解析データの使用を快諾して頂いた日本電気(株) C&C 情報研究所土肥 俊氏に感謝いたします。

- 12) van der Vorst, H. A.: A Vectorizable Variant of Some ICCG Methods, *SIAM J. Sci. Stat. Comput.*, Vol. 3, No. 3, pp. 350-356 (1982).
- 13) 野寺 隆: 大型疎行列に対する PCG 法, Seminar on Mathematical Sciences, No. 7, Keio University (1983).
- 14) 一松 信: 数値解析, 朝倉書店, 東京 (1982).
- 15) Bauer, F.L.: Optimally Scaled Matrices, *Numer. Math.*, Vol. 5, pp. 73-87 (1963).

(平成元年 3 月 30 日受付)

(平成元年 7 月 18 日採録)



速水 謙 (正会員)

1956年生. 1979年東京大学工学部計数工学科(数理工学専攻)卒業. 1981年同大学院修士課程修了. 同年, 日本電気(株)入社. 以来, LSIの配線収容性予測, 数値シミュレーション; 特に連立一次方程式の反復解法, 数値積分法, 境界要素法による電磁界解析等の研究に従事. 1986年-88年英国 Computational Mechanics Institute 客員研究員. 現在, 日本電気(株) C & C 情報研究所情報基礎研究部主任. シミュレーション学会, 境界要素法研究会, International Society for Boundary Elements 各会員.



原田 紀夫 (正会員)

昭和 42 年名古屋大学理学部数学卒業. 昭和 44 年同大学院修士課程修了. 同年 4 月日本電気(株)入社. 同社研究所にて計算機システムの性能評価, 乱数発生法, 組み合わせ最適化法, LSI の統計力学的配置理論, 並列処理の平均膨張率理論およびスーパーコンピュータ利用技術等の研究に従事. 昭和 53 年工学博士. 現在同社情報処理システム技術本部科学技術システム部部長代理. 電子通信学会(現電子情報通信学会)昭和 59 年度論文賞および米澤ファウンダーズメダル受賞記念特別賞受賞, 昭和 62 年度論文賞受賞. 日本数学会, 日本シミュレーション学会各会員.