

# 左前処理行列を適応的に決定する GMRES( $m$ ) 法の有効性について

鈴木 洋夫<sup>†</sup> 野寺 隆<sup>††</sup>

大型で疎な非対称正則行列を係数とする連立 1 次方程式の反復解法の 1 つとして、GMRES ( $m$ ) 法がある。この算法はリスタートを行なうことによって、記憶容量や計算時間を軽減しているが、このことが原因で近似解を構成するために必要な固有ベクトルの情報を失い、算法の収束性に悪影響を及ぼすことがある。そこで、本稿では近似不変部分空間を用いて固有ベクトルの情報を付加するような左前処理行列を適応的に決定し、収束性を改善させる算法について述べ、並列計算機 (Origin2000) での数値実験によりその有効性を検証する。

## Effectiveness of Adaptively Left Preconditioned GMRES( $m$ ) Method

HIROO SUZUKI<sup>†</sup> and TAKASHI NODERA<sup>††</sup>

GMRES( $m$ ) method is one of the iterative algorithms for solving the linear systems of equations with a large, sparse, and nonsymmetric coefficient matrix. Since this restarted method is able to reduce both computational work and its memory requirements, the GMRES( $m$ ) method is often used. However, using the restart, GMRES( $m$ ) method loses the information of some eigenvectors corresponding to the smallest eigenvalues, and its convergence may slow down or stall. In this paper, we propose the new algorithm which determines left preconditioner adaptively and improves the convergence rate of algorithm. At last, we show the effectiveness of algorithm by using the numerical experiments on the parallel machine Origin 2000.

### 1. はじめに

計算機の高性能化により、理工学における複雑な現象を記述する非線形問題を数値的に解くことが可能になり、より良いアルゴリズムの開発が盛んに行なわれている。中でも偏微分方程式の境界値問題を有限差分法や有限要素法を用いて離散化することで得られる、大型で疎な正則行列を係数とする連立 1 次方程式

$$Ax = b, \quad A \in R^{n \times n}, \quad x, b \in R^n \quad (1)$$

を解くための反復法の研究は重要な位置を占めている。

本稿で扱う GMRES( $m$ ) 法は、係数が非対称である場合に用いられる反復法の 1 つである。この算法はリスタートを行なうことで直交ベクトルの本数を  $m$  本に制限し、記憶容量と計算時間を減少させることが可能な、実用的な解法として用いられている。しかし、リスタートを行なうことで近似解を構成するために必要な固有ベクトルの情報が失われ、算法の収束性に悪影響を及ぼす場合がある。

そこで、本稿ではこのような欠点を改善する方法と

して、近似不変部分空間を用いて固有ベクトルの情報を付加する左前処理行列を GMRES( $m$ ) 法の反復の中で適応的に構成する方法を考える。

第 2 節で GMRES( $m$ ) 法について述べ、第 3 節で行列の前処理と本稿で提案する前処理行列の構成方法について述べる。第 4 節では数値実験の結果を示し、その考察を行なう。最後に第 5 節でまとめと今後の課題について述べる。

### 2. GMRES( $m$ ) 法

GMRES( $m$ ) 法は、1986 年に Y. Saad ら<sup>1)</sup> によって開発されたもので、大型で疎な非対称正則行列を係数とする連立 1 次方程式の反復解法の 1 つである。この解法は、クリロフ部分空間

$$K_m(A, r_0) = \{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\} \quad (2)$$

上に正規直交基底

$$V_m = (v_1, v_2, \dots, v_m) \quad (3)$$

をアーノルディ過程を用いて生成し、近似解を次のように構成する。

$$x_m = x_0 + V_m y \quad (4)$$

ただし、 $x_m, r_m (= b - Ax_m)$  はそれぞれ第  $m$  回目の反復における近似解、残差ベクトルである。このとき、 $r_m$  は次の最小条件から決まる。

<sup>†</sup> 慶應義塾大学大学院理工学研究科  
 Graduate School of Science and Technology, Keio University

<sup>††</sup> 慶應義塾大学理工学部  
 Faculty of Science and Technology, Keio University

```

choose  $x_0$ ,
 $r_0 := b - Ax_0$ ;
 $\gamma := \|r_0\|_2$ ;  $v_1 := r_0/\gamma$ ;
start
for  $n := 1$  to  $k$  do
begin
  Arnoldi process applied to  $A$ 
    to compute  $V_m$ ;
  compute  $y_n = \min_y \|\gamma e_1 - \bar{H}_n y\|_2$ ;
  if  $\|b - Ax_n\|_2 \leq \epsilon$  then
    stop iteration
  endif
end
 $x_0 := x_k$ ;  $r_0 := b - Ax_k$ ;
 $\gamma := \|r_0\|_2$ ;  $v_1 := r_0/\gamma$ ;
goto start

```

図 1 GMRES( $m$ )法  
Fig. 1 GMRES( $m$ ) method.

```

 $\hat{v} := Av_n$ ;
for  $i := 1$  to  $n$  do
begin
   $h_{i,n} := \hat{v}^T v_i$ ;
   $\hat{v} := \hat{v} - h_{i,n} v_i$ ;
end
 $h_{n+1,n} := \|\hat{v}\|_2$ ;
 $v_{n+1} := \hat{v}/h_{n+1,n}$ ;

```

図 2 アーノルディ過程  
Fig. 2 Arnoldi process.

$$\min_{z \in K_n} \|b - A(x_0 + z)\|_2 = \min_{z \in K_n} \|r_0 - Az\|_2 \quad (5)$$

ただし、 $z = V_m y$ である。ここで、 $V_m$ がユニタリ行列であることを用いると次の $m$ 次元関数 $J_m(y)$ の最小2乗問題に帰着される。

$$\begin{aligned} J_m(y) &= \|\beta v_1 - AV_m y\|_2 \\ &= \|V_{m+1}(\beta e_1 - \bar{H}_m y)\|_2 \\ &= \|\beta e_1 - \bar{H}_m y\|_2 \end{aligned} \quad (6)$$

ただし、 $\beta = \|r_0\|_2$ であり、 $\bar{H}_m$ はアーノルディ過程で生じる $(m+1) \times m$ 上ヘッセンベルグ行列である。式(7)は通常 $\bar{H}_m$ をQR分解することで解くことができる。図1にGMRES( $m$ )法を、図2にアーノルディ過程をまとめる。

GMRES( $m$ )法では、直交ベクトルの本数を $m$ 本に制限し、 $m$ 回の反復で求めた近似解 $x_m$ を初期近似解 $x_0$ として再び $m$ 回の反復を行なうリスタートを用いることで、記憶容量と計算時間を減少させている。ただし、 $m$ はリスタート周期である。

### 3. 行列の前処理

式(1)に直接反復法を適用しても、収束するまでに多くの計算時間を要したり、場合によっては全く収束しないことがある。これは、一般に反復法の収束性は係数行列の固有値と密接な関係があり、固有値の分布の状況によって収束性が決まるからである。また、本稿で扱うGMRES( $m$ )法は、リスタートを行なうことによって近似解を構成のために必要な固有ベクトルの情報を失い、このことが収束を遅らせる原因になる場合がある。そこで式(1)を

$$M^{-1}Ax = M^{-1}b \quad (7)$$

あるいは

$$AM^{-1}y = b, \quad x = M^{-1}y \quad (8)$$

と行列 $M$ による変換を行なってから反復法を適用することがある。式(7)や(8)は式(1)と同じ解を持つが、係数行列 $M^{-1}A$ や $AM^{-1}$ の固有値の分布が改善されていると、収束を改善することができる。

このように、固有値の分布が改善されるように元の連立1次方程式を変換することを行列の前処理といい、 $M^{-1}$ を前処理行列という。また、式(7)の変換を左側前処理、式(8)の変換を右側前処理という。

近年、固有ベクトルの情報を付加するような右側前処理を用いたGMRES( $m$ )法の改良版であるDEFLATION法<sup>6)</sup>やDEFLATED-GMRES法<sup>4)</sup>が提案されているが、本稿で提案する算法は左側前処理を用いたものである。

#### 3.1 スペクトルの情報を用いた行列の前処理

この節では、GMRES( $m$ )法の反復の中で、アーノルディ過程により得られる、行列 $A$ のスペクトルの情報を用いた前処理を考える。

行列 $A$ の固有値を $\lambda_1, \lambda_2, \dots, \lambda_n$  ( $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|$ ,  $|\lambda_n| = 1$ )とする。また、行列 $V_k \in R^{n \times k}$ を $V_m$ の最初の $k$ 列から成るものとする。行列 $M$ を

$$M = V_k H_k V_k^T + W_k W_k^T \quad (9)$$

とおく。本稿ではこの形式の行列の逆行列を左側からの前処理行列として用いることにする。ただし、逆行列は

$$M^{-1} = V_k H_k^{-1} V_k^T + W_k W_k^T \quad (10)$$

で与えられ、 $W_k W_k^T = I_n - V_k V_k^T$ である。 $M$ は

$$M = \begin{bmatrix} V_k & W_k \end{bmatrix} \begin{bmatrix} H_k & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} V_k^T \\ W_k^T \end{bmatrix}$$

と書くことができる。ただし、 $I_{n-k}$ は $(n-k) \times (n-k)$ の単位行列である。このとき、 $M^{-1}$ は

$$M^{-1} = \begin{bmatrix} V_k & W_k \end{bmatrix} \begin{bmatrix} H_k^{-1} & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} V_k^T \\ W_k^T \end{bmatrix}$$

となる。ところで、行列 $A$ は

$$\tilde{A} = \begin{bmatrix} V_k^T \\ W_k^T \end{bmatrix} A \begin{bmatrix} V_k & W_k \end{bmatrix} = \begin{bmatrix} H_k & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix}$$

と相似であり、 $\tilde{A}_{22}$ の固有値は $\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n$ である。このことから、

$$M^{-1}A = \begin{bmatrix} V_k & W_k \end{bmatrix} \begin{bmatrix} I_k & H_k^{-1}\tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} V_k^T \\ W_k^T \end{bmatrix}$$

となり、この式から $M^{-1}A$ は固有値 $\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n, 1$ を持つことがわかる。つまり、前処理行列 $M^{-1}$ を組み込むと、絶対値が小さい方から $k$ 番目までの固有値をすべて1に変換することができる。それによって係数行列 $M^{-1}A$ の絶対値の小さい固有値が減少し、収束性を改善することができる。

### 3.2 前処理行列の構成法

この節では、IRA過程 (Implicitly Restarted Arnoldi Process) を用いて行列 $A$ の近似不変部分空間を定義し、それを用いて前処理行列をどのように構成するを考える。

#### 3.2.1 IRA過程

Sorensenら<sup>2),3)</sup>によって提案されたIRA過程は、陰的な原点シフトを行なうQR法に基づいた算法で、アーノルディ過程をリスタートする手法であり、固有値問題に用いられている。この方法を用いることで、行列 $A$ の固有値の精密な近似値とそれに対する近似不変部分空間の基底を求めることができる。

アーノルディ過程によって生成される $m \times m$ の上ヘッセンベルグ行列 $H_m$ に対して、 $H_m - z_1 I_m = Q^{(1)} R^{(1)}$ を定義する。ただし、 $Q^{(1)}, R^{(1)} \in R^{m \times m}$ 、 $Q^{(1)}$ をユニタリ行列、 $R^{(1)}$ を上三角行列、 $z_1$ を原点シフトの大きさとする。このとき、

$$\begin{aligned} (A - z_1 I_m)V_m &= V_m(H_m - z_1 I_m) + f_m e_m^T \\ (A - z_1 I_m)V_m &= V_m Q^{(1)} R^{(1)} + f_m e_m^T \\ (A - z_1 I_m)(V_m Q^{(1)}) &= (V_m Q^{(1)})(R^{(1)} Q^{(1)}) \\ &\quad + f_m e_m^T Q^{(1)} \\ A(V_m Q^{(1)}) &= (V_m Q^{(1)})(R^{(1)} Q^{(1)} + z_1 I_m) \\ &\quad + f_m e_m^T Q^{(1)} \end{aligned}$$

となる。ただし、 $f_m = h_{m+1,m} v_{m+1}$ である。ここで、 $V_m^{(1)} = V_m Q^{(1)}$ 、 $H_m^{(1)} = R^{(1)} Q^{(1)} + z_1 I_m$ とおくと、

$$AV_m^{(1)} = V_m^{(1)} H_m^{(1)} + f_m e_m^T Q^{(1)} \quad (11)$$

を得る。このとき、 $H_m^{(1)}$ もまた上ヘッセンベルグ行列である。続いて、 $H_m^{(1)} - z_2 I_m = Q^{(2)} R^{(2)}$ について、同様のことをする。その結果、 $m-k$ 回の反復の後には

$$AV_m^{(m-k)} = V_m^{(m-k)} H_m^{(m-k)} + f_m e_m^T Q \quad (12)$$

となる。ただし、 $Q = Q^{(1)} Q^{(2)} \dots Q^{(m-k)}$ である。ここで、

$$H_m^{(m-k)} = \begin{bmatrix} H_k^{(m-k)} & G \\ \gamma_k e_1 e_k^T & \hat{H}_{m-k} \end{bmatrix} \quad (13)$$

とし、式(16)の両辺の第 $k$ 列までを取り出すと、

$$AV_k^{(m-k)} = V_k^{(m-k)} H_k^{(m-k)} + f_k^{(m-k)} e_k^T \quad (14)$$

となる。ただし、

$$f_k^{(m-k)} = V_{m-k}^{(m-k)} e_1 \gamma_k + f_m e_m^T Q e_k \quad (15)$$

である。ここまでの記述は、陽的な原点シフトを行なうQR法のアルゴリズムに基づいているが、数値的安定性を得るために実装では陰的な原点シフトを行なうQR法のアルゴリズムを用いる<sup>2)</sup>。

#### 3.2.2 原点シフトの大きさ

原点シフトの大きさ $z_1, z_2, \dots, z_{m-k}$ について考える。原点シフトの大きさは、行列 $A$ の固有値の小さい方から $k$ 個の固有値と、その固有値に関する近似不変部分空間を精密に求めることができるように選ぶ必要がある。

$m \times m$ の上ヘッセンベルグ行列 $H_m$ の固有値を $\{\theta_j^{(m)}\}_{j=1}^m$ とする。ただし、

$$|\theta_1^{(m)}| \leq |\theta_2^{(m)}| \leq \dots \leq |\theta_m^{(m)}| \quad (16)$$

である。 $H_m = V_m^T A V_m$ が行列 $A$ の直交射影であることから、 $\theta_j^{(m)}$ が $A$ の固有値の精密な近似であると考えられる。このことから、

$$z_j = \theta_{m+1-j}^{(m)}, \quad 1 \leq j \leq m-k \quad (17)$$

とおく。つまり、 $z_j$ は行列 $A$ の大きい方から $m-k$ 個の固有値を近似している。原点シフトの大きさをこのように選ぶことで、 $H_k^{(m-k)}$ の固有値は行列 $A$ の小さい方から $k$ 個の固有値を近似し、 $V_k^{(m-k)}$ がそのような固有値に対する近似不変部分空間の基底となる。

#### 3.2.3 近似不変部分空間の判定

次の条件を満足したとき、 $V_k^{(m-k)}$ が行列 $A$ の絶対値の小さい方から $k$ 個の固有値に対する近似不変部分空間の基底であるとする。

$\{\theta_j^{(k)}, y_j^{(k)}\}$ を $H_k^{(m-k)}$ の固有値と固有ベクトルの対とすると、

$$\|f_k^{(m-k)}\| \|e_k^T y_j^{(k)}\| \leq \|H_k^{(m-k)}\| \epsilon, \quad 1 \leq j \leq k \quad (18)$$

ただし、 $\epsilon > 0$ は任意に設定可能とする。この条件が満たされないときは、式(18)をアーノルディ過程の初期値とし、再び $V_m, H_m$ を求めてIRA法を適用する。

#### 3.2.4 前処理行列の決定

式(22)が満たされたとき、そのときの $V_k^{(m-k)}$ 、 $H_k^{(m-k)}$ を用いて、最初の前処理行列 $M_1^{-1}$ を式(10)のように構成する。この結果得られる連立1次方程式 $M_1^{-1}Ax = M_1^{-1}b$ に対する前処理行列 $M_2^{-1}$ が同様の手順で得られる。言い換えると、連立1次方程式(1)に対する前処理行列 $M^{-1} = M_2^{-1}M_1^{-1}$ が得られる。

表 1 Origin 2000 の仕様  
Table 1 Specifacaton of Origin 2000.

OS	IRIX6.5
セルプロセッサ	MIPS R10000 195MHz
セルローカルメモリ	512MB

以下同様にして、ある整数  $\alpha_0 \geq 1$  に対して、

$$M^{-1} = M_{\alpha_0}^{-1} M_{\alpha_0-1}^{-1} \dots M_1^{-1} \quad (19)$$

という前処理行列を決定する。

#### 4. 数値実験

本稿で述べた前処理を適用した GMRES( $m$ ) 法を Pre-GMRES( $m, k, \alpha, \beta$ ) 法と呼ぶことにする。ただし、 $m$  はリスタート周期、 $k$  は近似不変部分空間の次元、 $\alpha$  は生成する前処理行列の個数、 $\beta$  は IRA 過程の最大適用回数である。

Pre-GMRES( $m, k, \alpha, \beta$ ) 法と GMRES( $m$ ) 法を、並列計算機 Origin2000 に実装し数値実験を行なった。Origin2000 の仕様を表 1 に示す。

以下に実装に際して並列化部分をまとめると、(1) ベクトルとベクトルの和、(2) ベクトルのスカラー倍、(3) ベクトルの内積、(4) 行列とベクトルの積、となる。使用するプロセッサの数を  $p$  とし、ベクトルの次元を  $n$  とすると、各プロセッサが担当する要素の数を  $n/p$  と均等に割り当てる。

また、数値実験は以下の環境で行なった。

- 収束判定条件:  $\|r_m\|_2 / \|r_0\|_2 \leq 1.0 \times 10^{-12}$
- $\epsilon = 1.0 \times 10^{-4}$
- 最大反復回数: 10000
- 初期近似解:  $x_0 = (0, 0, \dots, 0)^T$
- プログラム言語: C 言語
- 計算精度: 倍精度

##### 4.1 数値例 1

矩形領域  $\Omega = [0, 1] \times [0, 1]$  における 2 階の楕円型偏微分方程式のディリクレ境界値問題を考える。

$$\begin{aligned} -u_{xx} - u_{yy} + Du_x(x, y) &= G(x, y) \\ u(x, y)|_{\partial\Omega} &= 1 + xy \end{aligned}$$

この方程式を 5 点中心差分近似を用いて離散化し、真の解を  $u(x, y) = 1 + xy$  と設定し、右辺を決定して数値実験を行なった。このとき、メッシュの大きさは  $128 \times 128$  とした。

この実験では、リスタート周期と近似不変部分空間の次元をそれぞれ  $m = 20$ 、 $k = 10$  と固定し、 $\alpha$  を 1 から 3 まで、 $\beta$  を 1 から 9 までに設定した。 $\alpha$  の値を大きくすると、前処理を組み込むための計算量が増え、反復回数を減少させることができて計算時間を短くするには至らない場合があった。また、 $\beta$  の値が小さいと精度の悪い近似不変部分空間を前処理に用いてしまうため、逆に収束に悪影響を及ぼすことになった。

具体的な性能についてであるが、前処理なしの

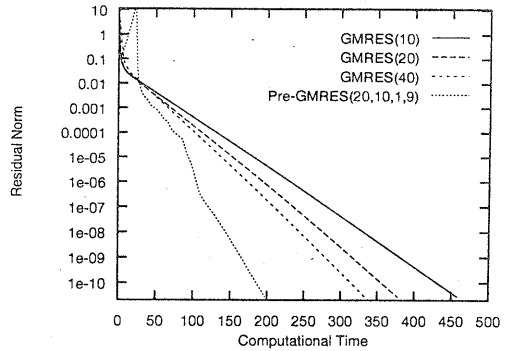


図 3 計算時間と残差ノルムの比較

Fig. 3 Computational time vs. residual norm

GMRES( $m$ ) 法では収束までに多くの反復回数と計算時間を要してしまう  $D = 1.0$  の場合に最も効果が高かった。反復回数については最も多かった GMRES(10) 法に対して、Pre-GMRES(20,10,3,2) 法が 80% 減少させることができた。計算時間については、最も長かった GMRES(10) 法に対して、Pre-GMRES(20,10,1,9) 法が 60% 短縮することができた。しかしながら、 $D = 10.0$ 、 $D = 100.0$  の場合には前処理なしの GMRES( $m$ ) 法でもリスタート周期を短く設定することで十分な性能を発揮するようになり、前処理を組み込む計算のオーバーヘッドの高さが目立つ結果となったが、それでもリスタート周期を長く設定したものと比較すると、それを上回る性能を発揮している。例えば、 $D = 100$  における Pre-GMRES(20,10,2,8) 法がその例である。図 3 に計算時間に対して最も効果が高かった  $D = 1.0$  における Pre-GMRES(20,10,1,9) 法の残差ノルムの収束の様子を示し、図 4 に反復回数に対して最も効果が高かった  $D = 1.0$  における Pre-GMRES(20,10,3,2) 法の残差ノルムの収束の様子を示す。

##### 4.2 数値例 2

矩形領域  $\Omega = [0, 1] \times [0, 1]$  における 2 階の楕円型偏微分方程式のディリクレ境界値問題を考える。

$$\begin{aligned} -u_{xx} - u_{yy} + D\{(y-1/2)u_x(x, y) \\ + (x-1/3)(x-2/3)u_y(x, y)\} &= G(x, y) \\ u(x, y)|_{\partial\Omega} &= 1 + xy \end{aligned}$$

この方程式を 5 点中心差分近似を用いて離散化し、真の解を  $u(x, y) = 1 + xy$  と設定し、右辺を決定して数値実験を行なった。このとき、メッシュの大きさは  $128 \times 128$  とした。

この実験では、 $\alpha = 2$ 、 $\beta = 9$  に固定し、 $m$  と  $k$  の値を変化させて前処理に対する影響を調べた。

数値例 2 でも数値例 1 と同様の傾向が見られた。同じリスタート周期を持つものどうしで比較した場合、 $D = 1.0$  の場合には前処理の効果が高く、反復回数に

表 2 数値例 1 の結果 (T : 計算時間(秒), I : 反復回数)

Table 2 Numerical results of example 1. (T : computational time(sec), I : Number of iteration)

D	1.0		10.0		100.0	
	sec	iter	sec	iter	sec	iter
GMRES(10)	458.12	7060	132.63	2110	32.39	500
GMRES(20)	377.30	3580	101.43	1060	69.80	660
GMRES(30)	322.66	2550	128.75	1020	87.34	690
GMRES(40)	337.93	1920	128.46	800	146.72	880
GMRES(50)	397.24	1550	212.86	800	244.48	950
Pre-GMRES(20,10,1,1)	425.69	3010	246.06	1610	227.50	1570
Pre-GMRES(20,10,1,2)	322.41	2290	241.34	1370	234.08	1510
Pre-GMRES(20,10,1,3)	374.97	2170	225.63	1370	238.13	1610
Pre-GMRES(20,10,1,4)	298.95	2110	242.98	1470	237.47	1650
Pre-GMRES(20,10,1,5)	250.91	1770	228.79	1570	289.72	1790
Pre-GMRES(20,10,1,6)	239.07	1690	282.89	1590	270.05	1890
Pre-GMRES(20,10,1,7)	201.14	1430	271.30	1570	277.87	1950
Pre-GMRES(20,10,1,8)	200.17	1430	246.94	1490	266.46	1870
Pre-GMRES(20,10,1,9)	187.58	1410	193.78	1390	265.80	1870
Pre-GMRES(20,10,2,1)	604.24	2340	451.60	1680	391.65	1760
Pre-GMRES(20,10,2,2)	354.66	1340	270.62	1240	393.84	1540
Pre-GMRES(20,10,2,3)	218.87	1020	326.00	1240	365.74	1660
Pre-GMRES(20,10,2,4)	220.99	1040	287.47	1180	316.76	1460
Pre-GMRES(20,10,2,5)	268.36	1260	222.70	1060	386.70	1780
Pre-GMRES(20,10,2,6)	331.84	1300	215.20	1040	249.50	1180
Pre-GMRES(20,10,2,7)	286.68	1200	255.33	1020	148.96	720
Pre-GMRES(20,10,2,8)	247.70	1200	229.23	1120	105.16	560
Pre-GMRES(20,10,2,9)	278.18	1140	240.56	1180	111.70	600
Pre-GMRES(20,10,3,1)	648.01	2130	550.16	1810	664.28	1770
Pre-GMRES(20,10,3,2)	292.48	990	319.04	1090	510.42	1390
Pre-GMRES(20,10,3,3)	418.58	1170	290.10	950	404.20	1370
Pre-GMRES(20,10,3,4)	380.49	1330	263.62	950	218.94	690
Pre-GMRES(20,10,3,5)	378.91	1110	287.62	1050	164.35	550
Pre-GMRES(20,10,3,6)	478.26	1390	256.04	970	127.18	530
Pre-GMRES(20,10,3,7)	358.99	1290	293.49	1110	144.80	610
Pre-GMRES(20,10,3,8)	342.87	1270	366.52	1150	160.61	690
Pre-GMRES(20,10,3,9)	424.43	1350	417.00	1550	193.05	770

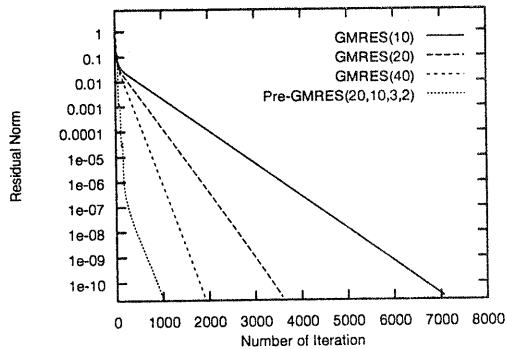


図 4 反復回数と残差ノルムの比較

Fig. 4 Number of iteration vs. residual norm

については最大で 50%。計算時間についてはリスタート周期が 10 の場合を除いて、30%から 50%程度前処理なしの GMRES( $m$ ) 法に対して上回る性能を示し

た。しかし、 $D = 10.0$  の場合には反復回数を改善できたものの、計算時間についてはリスタート周期が長いものに対してわずかな改善が見られただけであった。また、 $D = 100.0$  の場合には前処理を組み込むための計算のオーバーヘッドだけが目立つ結果となった。図 5、図 6 にそれぞれ  $D = 1.0$  における Pre-GMRES(50,10,2,9) 法の計算時間、反復回数に対する残差ノルムの収束の様子を示す。

## 5. おわりに

最後にまとめと今後の課題について述べる。数値実験の結果から、前処理なしの GMRES( $m$ ) 法では収束までに非常に多くの反復回数と計算時間を要してしまうような問題に対しては、Pre-GMRES( $m,k,\alpha,\beta$ ) 法は非常に効果的な算法であることが示された。本稿の数値例 1 と数値例 2 の  $D = 1.0$  の場合のように、計算時間を約 30%から 60%短縮できている。しかし、そのためには 4 つのパラメーターを適切に設定しなけ

表3 数値例2の結果 (T: 計算時間(秒), I: 反復回数)

Table 3 Numerical results of example 1. (T: computational time(sec), I: Number of iteration)

D	1.0		10.0		100.0	
	sec	iter	sec	iter	sec	iter
GMRES(10)	448.53	7000	244.22	3070	86.35	890
GMRES(20)	348.95	3600	170.78	1240	116.24	780
GMRES(30)	380.61	2490	195.93	1110	1137.06	810
GMRES(40)	309.95	1880	250.93	1040	179.59	800
GMRES(50)	370.95	1450	342.73	1050	305.06	950
Pre-GMRES(10,5,2,9)	899.30	6180	448.49	3730	462.71	2820
Pre-GMRES(20,5,2,9)	294.08	1630	216.54	1430	284.52	1670
Pre-GMRES(30,5,2,9)	202.28	1000	178.14	1090	225.19	1120
Pre-GMRES(40,5,2,9)	289.10	930	297.40	1130	317.60	1050
Pre-GMRES(50,5,2,9)	325.31	960	324.71	910	390.97	1010
Pre-GMRES(10,10,2,9)	1208.83	5760	751.26	3470	312.71	1540
Pre-GMRES(20,10,2,9)	344.08	1320	196.55	1080	273.04	1240
Pre-GMRES(30,10,2,9)	232.28	710	173.87	680	216.42	830
Pre-GMRES(40,10,2,9)	257.74	700	209.12	780	306.83	820
Pre-GMRES(50,10,2,9)	275.18	670	353.09	870	356.73	850

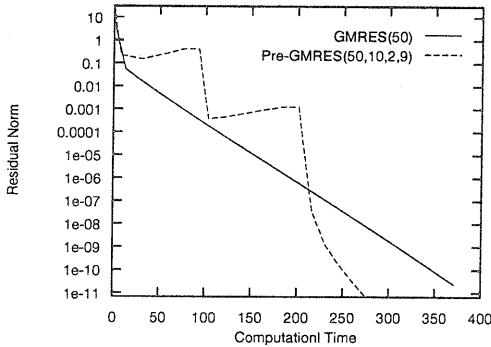


図5 計算時間と残差ノルムの比較

Fig. 5 Computational time vs. residual norm

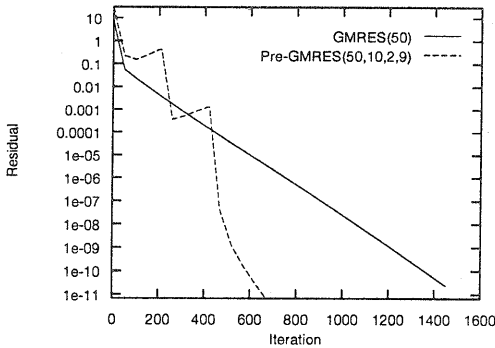


図6 反復回数と残差ノルムの比較

Fig. 6 Number of iteration vs. residual norm

ればならない。そこで今後の課題としては、前処理を組み込むことによって増えてしまう演算量の計算時間に対する影響とのバランスを考慮に入れながら、最適なパラメーターの値を自動的に決定することがあげられる。

### 参考文献

- 1) Y. Saad and M. H. Schultz: GMRES : A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comp.*, Vol. 7, No. 3, pp. 856-869 (1986).
- 2) D. C. Sorencen : Implicit Application of Polynomial Filters in a K-Step Arnoldi Method, *SIAM J. Matrix Anal. Appl.*, Vol. 13, No. 1, pp. 357-385 (1992).
- 3) R. Lehoucq: Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration: Ph.D. Thesis, Rice University, Houston (1995).
- 4) J. Erhel, K. Burrage and B. Pohl : Restarted GMRES Preconditioned by Deflation, *Journal of Computational and Applied Mathematics*, No. 69, pp. 303-318 (1996).
- 5) J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel : Adaptively Preconditioned GMRES Algorithms, *SIAM J. Sci. Comput.*, Vol.20, No. 1, pp. 243-269 (1998).
- 6) Burrage, K. and Erhel, J. : On the Performance of Various Adaptive Preconditioned GMRES Strategies, *Numer. Linear Algebra Appl.*, No. 5, pp. 101-121 (1998).