ハイパフォーマンス
コンピューティング 87－2
（２００１．７．２５）

複素数演算を回避するDeflated-GMRES(m)法について

鈴木洋夫† 森屋健太郎† 野寺隆††

大型で疎な非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)法があるが，リスタートを行うことによって，近似解を構成するために必要な絶対値の小さな固有値に対する固有ベクトルの情報を失い，収束性が悪くなることがある．Deflated-GMRES(m)法は，問題に応じて自動的に前処理行列を構成し、このような欠点を改善する算法であるが，係数行列の固有値と固有ベクトルを用いるため，係数行列が実行列の場合でも複素数演算が必要となる．そこで本稿では，Deflated-GMRES(m)法の計算時間をさらに改善するために，複素数演算を避けるDeflated-GMRES(m)法を考え，並列計算機 Origin2000 上での数値実験によってその性能を評価する．

Deflated-GMRES(m) method avoiding complex arithmetic

Hiroo Suzuki ,† Kentaro Moriya † and Takashi Nodera††

The GMRES(m) method is one of iterative solver for solving the linear systems of equations with a large, sparse, and nonsymmetric coefficient matrix. However, using restart, the GMRES(m) method looses the infomation of some eigenvectors corresponding to small eigenvalues, and then its convergence may slow down or stall. The Deflated-GMRES(m) method constructs a preconditioner adaptively for the GMRES(m) method and reduces the negative effects of the restart procedure. But the method must use complex arithmetic even if the coefficient matrix is real. In this paper, we consider the Deflated-GMRES(m) method avoiding complex arithmetic to improve computation time and we show the effectiveness of the algorithm by using the numerical experimetns on the parallel computer Origin 2000.

1. はじめに

理工学の分野における多くの問題では，自然現象をモデル化して偏微分方程式を構成し，解析するということが行われる．偏微分方程式を有限差分法や有限要素法を用いて離散化すると，大型で疎な行列を係数とする連立１次方程式

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

が得られることから，連立１次方程式の解法の研究が古くから行なわれて来た．

本稿で扱う GMRES(m) 法は，係数行列が非対称である場合に用いられる解法の１つである．しかし，必要な記憶領域や計算時間を軽減するために行うリスタートが原因で，近似解を構成するために必要な絶対値の小さな固有値に対する固有ベクトルの情報を失い，収束性が悪くなることがある．Deflated-GMRES(m)法は，失われる固有ベクトルの情報を付加するような前処理行列を問題に応じて自動的に構成することで，このような欠点を改善する算法であるが，係数行列の固有値と固有ベクトルを用いるため，係数行列が実行列の場合でも複素数演算が必要になる．前処理行列の構成に必要な計算時間が新たに加わる上に，複素数演算が加わることは，収束性が改善されると同時に改善されるはずの計算時間を犠牲にしている．

そこで本稿では，Burrage ら[5]によって提案された算法を基本算法とし，複素数演算を避けるための工夫を加えた算法を提案し，その性能を評価する．

第２節において，GMRES(m)法について述べる．第３節では，行列の前処理と Deflated-GMRES(m)法について述べ，第４節では複素数演算を避ける Deflated-GMRES(m)法を提案する．第５節では，数値実験の結果を示し，その考察を行う．最後に第６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[1]によって開発された算法で，大型で疎な非対称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．この算

† 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
†† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

choose $\boldsymbol{x}_0$
$\boldsymbol{r}_0 := \boldsymbol{b} - A\boldsymbol{x}_0$;
$\gamma := \|\boldsymbol{r}_0\|_2$; $\boldsymbol{v}_1 := \boldsymbol{r}_0/\gamma$;
start
for $n := 1$ to m do
begin
$\hat{\boldsymbol{v}} := A\boldsymbol{v}_n$;
for $i := 1$ to n do
begin
$h_{i,n} := \hat{\boldsymbol{v}}^T \boldsymbol{v}_i$;
$\hat{\boldsymbol{v}} := \hat{\boldsymbol{v}} - h_{i,n}\boldsymbol{v}_i$;
end
$h_{n+1,n} := \|\hat{\boldsymbol{v}}\|_2$;
$\boldsymbol{v}_{n+1} := \hat{\boldsymbol{v}}/h_{n+1,n}$;
end $\boldsymbol{y} = \min \|\gamma \boldsymbol{e}_1 - \bar{H}_n \boldsymbol{y}\|_2$;
$x_m := x_0 + V_m y$;
if $\|\boldsymbol{b} - A\boldsymbol{x}_n\|_2 \leq \epsilon$ then
stop iteration
endif
$\boldsymbol{x}_0 := \boldsymbol{x}_m$; $\boldsymbol{r}_0 := \boldsymbol{b} - A\boldsymbol{x}_m$;
$\gamma := \|\boldsymbol{r}_0\|_2$; $\boldsymbol{v}_1 := \boldsymbol{r}_0/\gamma$;
goto start

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

法は，クリロフ部分空間

$$K_m(A, \boldsymbol{r}_0) = \{\boldsymbol{r}_0, A\boldsymbol{r}_0, A^2\boldsymbol{r}_0, \cdots, A^{m-1}\boldsymbol{r}_0\} \quad (2)$$

上に正規直交基底

$$V_m = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_m) \quad (3)$$

をアーノルディ過程を用いて構成し，近似解を次のように構成する．

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + V_m \boldsymbol{y} \quad (4)$$

ただし，$\boldsymbol{x}_m$，$\boldsymbol{r}_m (= \boldsymbol{b} - A\boldsymbol{x}_m)$ は，それぞれ m 回の反復後の近似解，残差ベクトルであり，残差ノルム $\|\boldsymbol{r}_m\|_2$ が最小となるように構成されている．このとき，$\boldsymbol{y}$ は残差ノルムの最小 2 乗問題

$$\min \|\boldsymbol{b} - A\boldsymbol{x}_m\|_2 = \min_{\boldsymbol{y}} \|\gamma \boldsymbol{e}_1 - \bar{H}_m \boldsymbol{y}\|_2 \quad (5)$$

の解である．ただし，$\gamma = \|\boldsymbol{r}_0\|_2$ であり，$\bar{H}_m$ はアーノルディ過程から構成される $(m+1) \times m$ の上ヘッセンベルグ行列である．

GMRES(m) 法は，正規直交基底を m 本に制限し，m 回の反復後の近似解 $\boldsymbol{x}_m$ を初期近似解 $\boldsymbol{x}_0$ として再び m 回の反復を行なう．このようなリスタートを用いることによって，必要な記憶領域と計算時間を減少させている．ただし，m をリスタート周期と呼ぶ．図 1 に GMRES(m) 法の算法を示す．

3. 行列の前処理

GMRES(m) 法に限らず，式 (1) に反復解法を適用するときは，行列の前処理を行なうことが多い．すなわち，

$$M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{b} \quad (6)$$

あるいは，

$$\begin{cases} AM^{-1}\boldsymbol{y} = \boldsymbol{b} \\ \boldsymbol{x} = M^{-1}\boldsymbol{y} \end{cases} \quad (7)$$

というように，行列 M^{-1} による変換を行う．ただし，M^{-1} を前処理行列と呼び，式 (6) の変換を左側前処理，式 (7) の変換を右側前処理と呼ぶ．反復解法の収束性は，係数行列の固有値の分布の状況に依存しているため，適切な前処理行列を選び、固有値の分布の状況を改善することができれば，算法の収束性は改善することになる．

しかし，GMRES(m) 法の収束性が悪い場合，係数行列の固有値の分布の状況が悪いことに加えて，リスタートによる近似解を構成するために必要な絶対値の小さい固有値に対する固有ベクトルの情報が失われることがある．Deflated-GMRES(m) 法は，係数行列の固有値の分布の状況を改善し，さらに失われる固有ベクトルの情報を付加するような前処理行列を，問題に応じて自動的に構成するという優れた算法である．次の節では，Deflated-GMRES(m) 法について述べることにする．

3.1 Deflated-GMRES(m) 法

GMRES(m) 法の反復の中で、デフレーションを用いて前処理行列を構成する算法を Deflated-GMRES(m) 法と呼ぶ．デフレーションとは，元の行列をその固有値と同じ固有値を持つ次元の小さい行列に変換し，固有値を求める方法である．GMRES(m) 法の反復の中で，アーノルディ過程から構成される V_m，H_m と係数行列 A の間には次のような関係が成立する．

$$H_m = V_m^T A V_m \quad (8)$$

このことから，H_m の固有値は A の固有値の 1 部であることが分かり，実際には，A の絶対値が大きい固有値と小さい固有値である．Deflated-GMRES(m) 法において前処理行列の構成に必要な固有値と固有ベクトルは，絶対が小さい固有値と，その固有値に対する固有ベクトルであるから，H_m の固有値を求めることができれば十分であり，また，H_m の大きさはリスタート周期によって決まり，一般にリスタート周期は，元の問題の大きさに比べてかなり小さいので，A の固有値を直接求めるより比較的短時間で計算することができる．

このように，デフレーションを用いることによって，前処理行列の構成に必要な固有値と固有ベクトルを比較的短時間で求めることが可能である．また，H_m は GMRES(m) 法の反復の中で必ず構成されるので，そ

の固有値と固有ベクトルを用いる前処理行列は，問題に応じて自動的に構成されることになる．Erhel ら[4]によって提案された算法 (RD-GMRES(m,k) 法，m はリスタート周期，kは付加する固有ベクトルの本数) は，最も代表的な算法であり，右側前処理を用いる．さらにこの算法は，森屋ら[7]によって，リスタート周期を適応的に変化させる改良などを加えることによって，その性能を大きく改善することに成功している．

しかし，一般に非対称行列の固有値は複素数になるので，固有値と固有ベクトルを用いて前処理行列を構成する Deflated-GMRES(m) 法は，係数行列が実行列の場合でも複素数演算が必要になる．前処理行列の構成に必要な計算時間が新たに加わる上に，複素数演算が加わることは，収束性が改善されると同時に改善されるはずの計算時間を犠牲にしている．

Baglama ら[5]によって提案された算法 (LD-GMRES(m,k) 法，m はリスタート周期，kは付加する固有ベクトルの数) は，左側前処理を用いる．本稿では，この算法の前処理行列の構成の方法を複素数演算を避けるように改良する算法を提案する．次の節では，LD-GMRES(m,k) 法について述べることにする．

3.2 LD-GMRES(m,k) 法

LD-GMRES(m,k) 法では，

$$M^{-1} = V_k H_k^{-1} V_k^T + I - V_k V_k^T \tag{9}$$

という前処理行列を構成する．ただし，kは付加する固有ベクトルの数である．このとき，V_k，H_kはアーノルディ過程から構成される V_m，H_mに次のような操作を行うことで得られる行列である．

アーノルディ過程を行列で表すと，

$$AV_m = V_m H_m + \boldsymbol{f}_m \boldsymbol{e}_m^T \tag{10}$$

となる．この等式に次のような QR 法に基づく変形をする．

$$\begin{aligned}
(A - z_1 I)V_m &= V_m(H_m - z_1 I) + \boldsymbol{f}_m \boldsymbol{e}_m^T \\
(A - z_1 I)V_m &= V_m Q^{(1)} R^{(1)} + \boldsymbol{f}_m \boldsymbol{e}_m^T \\
(A - z_1 I_m)(V_m Q^{(1)}) &= (V_m Q^{(1)})(R^{(1)} Q^{(1)}) \\
&\quad + \boldsymbol{f}_m \boldsymbol{e}_m^T Q^{(1)} \\
A(V_m Q^{(1)}) &= (V_m Q^{(1)})(R^{(1)} Q^{(1)} + z_1 I_m) \\
&\quad + \boldsymbol{f}_m \boldsymbol{e}_m^T Q^{(1)} \\
AV_m^{(1)} &= V_m^{(1)} H_m^{(1)} + \boldsymbol{f}_m \boldsymbol{e}_m^T Q^{(1)}
\end{aligned}$$

この変形を $m-k$回繰り返すと，

$$AV_m^{(m-k)} = V_m^{(m-k)} H_m^{(m-k)} + \boldsymbol{f}_m \boldsymbol{e}_m^T Q \tag{11}$$

が得られる．ただし，$Q = Q^{(1)} Q^{(2)} \cdots Q^{(m-k)}$である．このとき，$H_m$の固有値 $\{\theta_i\}_{i=1}^m$が，

$$|\theta_1| \le |\theta_2| \le \cdots \le |\theta_m| \tag{12}$$

であるとし，原点シフトの大きさ z_iを，

$$z_i = \theta_{m+1-i}, \quad 1 \le i \le m-k \tag{13}$$

とする．このとき，$H_m^{(m-k)}$の第 k行第 k列 $H_k^{(m-k)}$の対角成分に，係数行列 A の絶対値が小さい方から k個の固有値が並び，$V_m^{(m-k)}$の第 k列 $V_k^{(m-k)}$の各列には，対応する固有ベクトルが並ぶ．このようにして $H_k = H_k^{(m-k)}$，$V_k = V_k^{(m-k)}$が構成される．これらの行列を用いて式 (9) のように前処理行列を構成する．

このようにして構成される前処理行列を M_1^{-1}とすると，

$$M_1^{-1} A\boldsymbol{x} = M_1^{-1}\boldsymbol{b} \tag{14}$$

に対する前処理行列 M_2^{-1}を全く同じ方法で構成することができる．このように，任意の整数αに対して，

$$M^{-1} = M_\alpha^{-1} M_{\alpha-1} \cdots M_1^{-1} \tag{15}$$

という前処理行列を構成することができるが，αの値を必要以上に大きくすると，前処理行列を構成するために必要な計算と，それを適用するために必要な計算のオーバーヘッドが大きくなり逆効果になる．

一般に非対称行列の固有値は複素数であり，LD-GMRES(m) 法では，それを原点移動に用いて QR 法を適用するため，V_k，H_kは複素行列になる．それによって，前処理行列を構成した後の GMRES(m) 法の反復は複素数演算となる．そこで次の節では，LD-GMRES(m,k) を改良し，複素数演算を回避することができる Deflated-GMRES(m) 法について述べる．

4. 複素数演算を回避する方法

前節で述べた LD-GMRES(m,k) 法を OLD-GMRES(m,k) 法，この節で述べる LD-GMRES(m,k) 法の改良版を NLD-GMRES(m,k) 法と呼ぶ．

実行列の複素固有値は，必ず共役複素数の対として現れることから，OLD-GMRES(m,k) 法で用いられる QR 法の原点移動を共役複素数の対として，QR 法の反復 2 回を 1 回で計算するダブル QR 法を用いる．すなわち，次のような変形を行う．

$$\begin{aligned}
&AV_m = V_m H_m + \boldsymbol{f}_m \boldsymbol{e}_m^T \\
&W = H_m^2 - (z_1 + \bar{z}_1) H_m + z\bar{z}_1 I \\
&W = Q^{(1)} R^{(1)} \\
&A(V_m Q^{(1)}) = (V_m Q^{(1)})(Q^* H_m Q) + \boldsymbol{f}_m \boldsymbol{e}_m^T Q^{(1)} \\
&AV_m^{(1)} = V_m^{(1)} H_m^{(1)} + \boldsymbol{f}_m \boldsymbol{e}_m^T Q^{(1)}
\end{aligned}$$

このようにして得られる $V_m^{(1)}$，$H_m^{(1)}$は実行列になる．従って，この変形を繰り返して得られる V_k，H_kも実行列になり，前処理行列を構成した後の GMRES(m) 法の反復は実数演算となる．図 3 に NLD-GMRES(m,k) 法の算法を示す．

ダブル QR 法を用いる場合，Wは，非ゼロ要素が対角の下 2 段までの行列であり，この行列を QR 分解するためには，ハウスホルダー変換を用いることになる．NLD-GMRES(m,k) 法において最も計算時間を必要とする部分である．複素数演算を回避することによって生じる新たな計算が，どの程度の影響を持つかが問題である．

```
choose x_0
r_0 := b − Ax_0;
γ := ‖r_0‖_2;  v_1 := r_0/γ;
start
for n := 1 to m do
begin
  v̂ := Av_n;
  for i := 1 to n do
  begin
    h_{i,n} := v̂^T v_i;
    v̂ := v̂ − h_{i,n} v_i;
  end
  h_{n+1,n} := ‖v̂‖_2;
  v_{n+1} := v̂/h_{n+1,n};
end
y = min ‖γe_1 − H̄_n y‖_2;
x_m := x_0 + V_m y; if ‖b − Ax_n‖_2 ≤ ε then
  stop iteration
endif
if α_0 ≤ α then
  compute eigenvalue {θ_i}_{i=1}^m of H_m
  and order them;
  z_i = θ_{m+1−i};
  for i := 1 to m − k do
  begin
    W := H_m^2 − (z_i + z̄_i)H_m + z_i z̄_i I;
    W := QR;
    H_m^(i) := Q* H_m Q;  V_m^(i) := V_m Q;
  end
  H_k := H_k^(i);  V_k := V_k^(i);
  M_α^{-1} := V_k H_k^{-1} V_k^T + I − V_k V_k^T;
  M^{-1} := M_α^{-1} M^{-1};
endif
x_0 := x_m;  r_0 := b − Ax_m;
γ := ‖r_0‖_2;  v_1 := r_0/γ;
goto start
```

図 2 NLD-GMRES(m) 法
Fig. 2 NLD-GMRES(m) method.

5. 数値実験

本稿で述べた NLD-GMRES(m,k) 法を，OLD-GMRES(m,k) 法，RD-GMRES(m,k) 法，および行列の前処理を行わない GMRES(m) 法と比較するために，各算法を並列計算機 Origin2000 に実装し，数値実験を行った．ただし，Deflated-GMRES(m) 法の各算法の付加する固有ベクトルの本数は 4 本とした．また，OLD-GMRES(m,k) 法，NLD-GMRES(m,k) 法の反復の中で構成する前処理行列は 1 個とした．Origin2000 の仕様を表 1 に示す．

表 1 Origin 2000 の仕様
Table 1 Specificaton 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

実装に際して並列化した部分をまとめると，(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，である．使用する CPU の数を p とし，ベクトルの次元を n とすると，各 CPU が担当する要素の数を n/p と均等に割り当てる．また，固有値の計算には CLAPACK を用いた．

また，数値実験は以下の環境で行った．

- 収束判定条件：$\epsilon = \|r_m\|_2/\|r_0\|_2 \leq 1.0 \times 10^{-12}$
- 最大反復回数：10000
- 初期近似解：$x_0 = (0, 0, \ldots, 0)^T$
- プログラム言語：C 言語
- 計算精度：倍精度
- CPU の数：8 個

5.1 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．

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

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 128×128 とした．結果を表 2 に示す．

5.2 数値例 1 の結果と考察

デフレーションを用いて前処理行列を適応的に構成する Deflated-GMRES(m) 法の全てが $Dh = 2^{-6}$ の場合に良い結果を示している．前処理行なうの算法の中で最も速く収束した NLD-GMRES(50,4) 法は，前処理を行なわない算法の中で最も速く収束した GMRES(40) 法と比較して，計算時間を約 60%程度短縮することができた．また，Dh の値が大きくなると，前処理を行なわない算法が良い性能を示し，前処理行列を構成するために必要な計算のオーバヘッドの高さが算法に悪影響を与えることが分かる．

次に，NLD-GMRES(m,k) 法と OLD-GMRES(m,k) 法を比較する．ほとんど全ての場合に対して，NLD-GMRES(m,k) 法が OLD-GMRES(m,k) 法より良い結果を示し，計算時間を短縮することができた．特に $Dh = 2^{-6}$，$m = 50$ の場合，約 38%程度短縮することができた．このことから，複素数演算を回避することによって LD-GMRES(m,k) 法の性能を改善することができることが分かる．

しかし，NLD-GMRES(m,k) 法と RD-GMRES(m,k) 法を比較すると，反復回数は NLD-GMRES(m,k)

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 : computationl time(sec), I : Number of iteration)

D	2^{-6}		2^{-5}		2^{-4}	
算法	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	137.06	810
GMRES(40)	309.95	1880	250.93	1040	179.59	800
GMRES(50)	370.95	1450	342.73	1050	305.06	950
RD-GMRES(10,4)	333.15	4340	201.58	3120	155.56	2390
RD-GMRES(20,4)	145.90	1800	148.42	1520	185.31	1960
RD-GMRES(30,4)	136.00	1380	168.00	1410	137.33	1380
RD-GMRES(40,4)	138.84	1040	154.90	1080	180.81	1240
RD-GMRES(50,4)	125.44	800	112.68	800	141.15	1000
OLD-GMRES(10,4)	443.62	3600	744.18	5000	437.50	3520
OLD-GMRES(20,4)	299.33	1800	189.31	1360	282.46	1560
OLD-GMRES(30,4)	184.31	840	192.53	840	173.97	1050
OLD-GMRES(40,4)	193.61	900	160.56	760	144.91	660
OLD-GMRES(50,4)	200.76	750	252.25	750	209.11	650
NLD-GMRES(10,4)	332.04	3300	200.47	5000	154.45	3260
NLD-GMRES(20,4)	144.79	1400	147.31	1320	184.20	1180
NLD-GMRES(30,4)	143.89	840	166.89	810	136.22	1060
NLD-GMRES(40,4)	137.73	1000	143.79	720	189.70	640
NLD-GMRES(50,4)	124.33	800	111.57	700	140.04	600

法の方が少ないが，計算時間はほぼ等しい．反復回数が少ないが計算時間が等しいということから，NLD-GMRES(m,k) 法の前処理行列の構成に必要な計算のオーバヘッドは RD-GMRES(m,k) 法と比較してかなり大きいことが分かる．このことが，複素数演算を必要とする RD-GMRES(m,k) 法を上回る性能に至らない原因である．

5.3 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．

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

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 128×128 とした．結果を表 3 に示す．

5.4 数値例 2 の結果と考察

数値例 2 の結果も，数値例 1 と全く同じ傾向を示した．

数値例 1 と同様に，Deflated-GMRES(m) 法の性能は $Dh = 2^{-6}$ の場合に最も良く，NLD-GMRES(40,4) 法は GMRES(30) 法と比較して，計算時間を約 56%程度短縮することができた．

また，NLD-GMRES(m) 法と OLD-GMRES(m) 法を比較すると，数値例 1 と同様にほとんどの場合において計算時間を短縮することができ，特に $Dh = 2^{-6}$，$m = 10$ の場合，約 77% 程度短縮することができた．

しかし，RD-GMRES(m) 法と比較するとほぼ同じ性能を示し，上回る性能を示すには至らなかった．

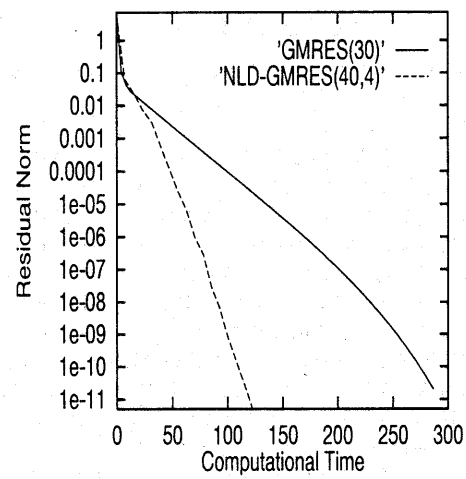

図 3　計算時間に残差ノルムの収束の様子
Fig. 3　Computational time vs. residual norm

図 3 に，$Dh = 2^{-6}$ の場合の，NLD-GMRES(m,k) 法の中で最も計算時間が短かった NLD-GMRES(40,4) 法と，GMRES(m) 法の中で最も計算時間が短かった，GMRES(30) の計算時間に対する残差ノルムの収束の様子を示す．

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 2. (T : computationl time(sec), I : Number of iteration)

D	2^{-6}		2^{-5}		2^{-4}	
算法	sec	iter	sec	iter	sec	iter
GMRES(10)	422.87	6330	318.25	4440	150.95	2180
GMRES(20)	292.45	2920	163.80	1420	112.33	1080
GMRES(30)	285.18	2100	202.11	1320	131.89	960
GMRES(40)	309.17	1800	284.46	1360	169.94	960
GMRES(50)	380.37	1600	250.23	1200	209.16	1000
RD-GMRES(10,4)	400.61	5040	367.19	4050	280.90	3260
RD-GMRES(20,4)	180.16	1860	202.87	1800	205.85	2060
RD-GMRES(30,4)	173.27	1260	130.32	1140	108.62	960
RD-GMRES(40,4)	128.96	840	121.14	760	119.01	760
RD-GMRES(50,4)	141.98	750	131.27	750	136.82	750
OLD-GMRES(10,4)	899.30	6180	448.49	3470	312.71	1540
OLD-GMRES(20,4)	294.08	1320	196.55	1080	273.04	1240
OLD-GMRES(30,4)	202.28	710	173.87	680	216.42	830
OLD-GMRES(40,4)	257.74	700	209.12	780	306.83	820
OLD-GMRES(50,4)	275.18	670	353.09	870	356.73	850
NLD-GMRES(10,4)	389.50	6090	366.08	3230	289.79	1270
NLD-GMRES(20,4)	179.05	1160	201.76	1020	204.74	1080
NLD-GMRES(30,4)	172.16	720	129.21	640	107.51	810
NLD-GMRES(40,4)	127.85	720	120.03	720	117.90	800
NLD-GMRES(50,4)	140.87	650	130.16	850	135.71	850

6. まとめと今後の課題

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，NLD-GMRES(m,k) 法は，OLD-GMRES(m,k) 法の性能を大きく上回ることが分かり，左側前処理を用いる Deflated-GMRES(m) 法に対しては，複素数演算を回避することが，算法の性能の向上につながることが分かった．数値例 1，数値例 2 の全ての場合において，計算時間を平均約 30%程度短縮することができた．しかし，右側前処理を用いる Deflated-GMRES(m) 法である RD-GMRES(m,k) 法の性能と比較すると，ほぼ同じ性能に留まることが分かった．複素数演算を必要としない NLD-GMRES(m,k) 法が複素数演算を必要とする RD-GMRES(m,k) 法ほぼ同じ性能を示したということは，NLD-GMRES(m,k) 法の前処理行列の構成に必要な計算のオーバヘッドがかなり大きいことが分かる．この主な原因は，ダブル QR 法の中で，非ゼロ要素が対角の下 2 段まで現れる行列の QR 分解にハウスホルダー変換を用いることと，行列の 2 乗を計算する必要があることである．

今後の課題は，NLD-GMRES(m,k) 法において，前処理行列の構成に用いるダブル QR 法について考察し，計算のオーバーヘッドを小さくすることである．また，森屋ら[7]のように，リスタート周期の適応的な決定方法などを考え，パラメータを自動的に決定することである．

参 考 文 献

1) Saad, Y. and Schults, M. H. : 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) Sorencen, D. C. : 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) Lehoucq, R. : Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration: Ph.D. Thesis, Rice University, Houston (1995).

4) Erhel, J., Burrage, K and Pohl, B. : Restarted GMRES Preconditioned by Deflation, *Journal of Computational and Applied Mathmatics*, No. 69, pp. 303–318 (1996).

5) Baglama, J., Calvetti, D., Golub, G. H., and Reichel, L. : 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.*, Vol. 5, pp. 101–121 (1998).

7) Moriya, K. and Nodera, T. : The DEFLATED-GMRES(m,k) method with switching the restart frequency dynamically , *Numer. Linear Algebra Appl.*, Vol. 7, pp. 569–584 (2000).