

ブロック分割による reduced system の前処理

井上俊夫[†] 野寺隆^{††}

大型で疎な正則行列を係数行列としてもつ連立1次方程式の解法の1つに GMRES(m)法がある。一般に、反復法は前処理を用いて収束性を改善させるのだが、前処理にも様々なものがある。ここでは Y. Saad らが提案した不完全 LU 分解 (ILU) に類似した消去法とブロック分割を利用し、reduced system を構成する方法を使用する。この方法を用いると、与えられた連立1次方程式より大幅に次元の小さな方程式 (reduced system) を反復法によって解くことができ、さらに直接代入により方程式の近似解を得ることができる。その結果、行列とベクトルとの積や反復回数を減少させ、計算量、記憶容量を削減できる。本稿では、この前処理を用いた GMRES(m)法と従来の GMRES(m)法の収束性などを数値実験によって比較し、本稿で提案した前処理の有効性を示す。

Preconditioning by reduced system with block factorization

TOSHIO INOUE[†] and TAKASHI NODERA^{††}

We consider the linear systems of equations with a large, sparse, nonsingular coefficient matrix. The GMRES(m) method is one of major iterative solver for these linear systems. The computation cost and memory space of the GMRES(m) method are more expensive as the number of iteration increase. Consequently, the GMRES(m) method usually requires preconditioning technique to reduce the cost of work. In this paper, we mainly discuss the preconditioning using reduced system, although there is a lot of scheme for preconditioning. At last, numerical experiments show that the GMRES(m) method with this proposed preconditioning is more effective than the standard GMRES(m) method.

1. はじめに

理工学における様々な現象を記述する楕円型偏微分方程式の境界値問題などを有限要素法や有限差分法によって離散化して得られる大型で疎な正則行列を係数行列に持つ連立1次方程式

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

の解法について考える。方程式(1)の解法には様々な方法が存在するが、現時点では式(1)の係数行列 A が対称行列である場合には CG 法が最も有効な算法であり、行列 A が非対称行列である場合には GMRES 法が有効な算法の1つであるといえる。GMRES 法には計算の途中でブレイクダウンを起こさないことや、各反復において残差ノルムを最小にするなど好ましい性質があるが、反復の毎に記憶容量が増えていくという難点もある。そのため m 回反復したところで初期近似解と残差ベクトルを更新するリスタート版の GMRES(m)法を用いることが多い。また GMRES 法に限らず反

復法に対しては前処理を行うことも慣例となっている。前処理とは、式(1)の両辺に行列 A の近似逆行列を前処理行列として掛けることで収束性を改善する方法のことである。ここではその前処理行列を構成する方法としてシュールコンプリメント (Schur-complement) を用いて reduced system を構成する方法を用い、更にその reduced system に対して右側前処理を施す算法を考える。

まず2章で GMRES(m)法について述べる。3章および4章では実際に GMRES(m)法に対して施す前処理の構成方法について述べ、5章でその前処理を施した GMRES(m)法と従来の GMRES(m)法の収束性を比較した数値実験の結果を示す。最後に6章においてまとめと結論を述べる。

2. GMRES(m)法

GMRES 法は、1986年に Y. Saad ら¹⁾によって提案された連立1次方程式を解くクリロフ部分空間法の1つである。最初に初期近似解を x_0 とする。初期残差ベクトルを $r_0 = b - Ax_0$ とした時に第 i 回目の反復における残差ノルムは次式で決定される。

[†] 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

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

ただし、ここで、 z はクリロフ部分空間:

$$K_i(A, r_0) = \{r_0, Ar_0, A^2 r_0, \dots, A^{i-1} r_0\}$$

上の正規直交ベクトル列 $V_i = \{v_1, v_2, \dots, v_i\}$ によって $z = V_i y$ と表され、 y はその時の残差ノルムを最小にするように選ばれる。つまり、次式の最小二乗問題を解くことになる。

$$\begin{aligned} J(y) &= \|\beta v_1 - AV_i y\|_2 \\ &= \|V_{i+1}(\beta e_1 - H_i^* y)\|_2 \\ &= \|\beta e_1 - H_i^* y\|_2 \end{aligned} \quad (2)$$

ただし、 $\beta = \|r_0\|_2$ 、 $e_1 = (1, 0, \dots, 0)^T$ であり、 H_i^* はアーノルディ過程から得られる上ヘッセンベルグ行列である。式(2)は一般に H_i^* をギブンス回転行列でQR分解することによって解くことができる。こうして得られた y を用いて第 i 回目の近似解は、 $x_i = x_0 + V_i y$ で求められる。

GMRES法は反復回数を重ねると正規直交ベクトルの数が増大するため、計算量と記憶容量の削減のためにリスタートを用いることが多い。これは反復回数 i が m を越えた時に、新しく初期解 x_0 を x_m によって置き換えて、初期残差を計算し、これを収束するまで繰り返す方法である。この方法をGMRES(m)法と呼び、以降はこの方法を用いる。

3. reduced system の構成

この章では連立1次方程式(1)に対して施す前処理として消去法を用いて reduced system を構成する方法について述べる。

まず、式(1)の係数行列の要素の並びを換えることを考える。図1に示された Greedy なアルゴリズムを

```

set  $\sigma_i = \sum_{k=1}^n a_{i,k}$ 
set  $F = \emptyset$ 
for  $j := 1$  to  $n$  do
begin
  if node  $j$  is not marked and  $\sigma_i < |a_{i,j}|$ 
     $F := F \cup j$ 
    mark  $j$  and all neighbors
  else
    mark node  $j$  and add it to  $F^c$ 

```

図1 Greedy なアルゴリズム

用いて、求められた番号をインデックスに持つ行列の要素を左上の部分に集まるように行と列を並び替える²⁾。この並び換えにより、左上のブロック行列は対角行列になる。以降は、並び換えた方程式を式(3)と

次に、方程式(1)を

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (3)$$

とブロックに分割して考える。この時 A_1 は上で行った並びかえのために、対角行列になる。このブロック2元連立1次方程式(3)を展開すると次の方程式を得る。

$$A_1 x_1 + A_2 x_2 = b_1 \quad (4)$$

$$A_3 x_1 + A_4 x_2 = b_2 \quad (5)$$

この方程式を消去法によって x_1 を消去して解くと、次のような方程式が作られる。

$$(A_4 - A_3 A_1^{-1} A_2) x_2 = b_2 - A_3 A_1^{-1} b_1 \quad (6)$$

今、この式(6)の係数行列を $B = A_4 - A_3 A_1^{-1} A_2$ とおく。これを A_1 に関するシュールコンプリメントと呼ぶ¹⁾。また式(6)を reduced system と呼ぶ。この reduced system を解いて、 x_2 が求めれば、 x_1 は(1)に代入することで計算できることになる。つまり、整理すると、次のようになる。

$$B x_2 = b_2 - A_3 A_1^{-1} b_1 \quad (7)$$

$$x_1 = A_1^{-1} (b_1 - A_2 x_2) \quad (8)$$

ただし、式(7)、式(8)のなかで A_1^{-1} が必要になるので、最初に並び換えを行うものとする。

このような形にしたことにより式(7)の解が得られれば、式(8)に代入することで式(1)の解を得ることができる。結果として反復法を用いて計算する方程式の次元を小さくすることができ、計算量、記憶容量ともにコストを削減できる。これがこの前処理の最大の利点である。しかし、このような形にしたことにより、式(7)の係数行列 B の非ゼロ要素の数が多く、構造も複雑になり、条件数を悪化させる問題となってしまう。上記の理由から reduced system に対して右側前処理を施すことにする。最終的に次の方程式を解くことになる。

$$B N y = b_2 - A_3 A_1^{-1} b_1 \quad (9)$$

$$x_2 = N y \quad (10)$$

$$x_1 = A_1^{-1} (b_1 - A_2 x_2) \quad (11)$$

この前処理付き GMRES(m)法のアルゴリズムを図2に示す。

4. シュールコンプリメントの近似逆行列

この章では反復法を用いて実際に解く連立1次方程式(7)に対して施す前処理について述べる。つまり A_1 に関するシュールコンプリメント $B (= A_4 - A_3 A_1^{-1} A_2)$ の前処理行列 N を求める方法について述べる。この行列 B はもとの式(1)の係数行列 A と比較すると密行列になっているので、近似逆行列には正確なものを求めた方がよいが、非ゼロ要素の構造が複雑になっているため、それが難しい。そこで、行列 B の近似逆行列を構成する方法として、以下の3つの方法を使用した。

```

choose  $x'_0$ 
 $b'_1 := b_2 - A_3 A_1^{-1} b_1$ ;
 $r_0 := b' - BNx'_0$ ;
 $\gamma := \|r_0\|_2$ ;  $v_1 := r_0/\gamma$ ;
start
for  $n := 1$  to  $m$  do
begin
 $k_1 := Nv_n$ ;
 $k_2 := A_2 k_1$ ;
 $k_3 := A_1^{-1} k_2$ ;
 $k_4 := A_3 k_3$ ;
 $k_5 := A_4 k_1$ ;
 $w := k_5 - k_4$ ;
for  $i := 1$  to  $n$  do
begin
 $h_{i,n} := w^T v_i$ ;
 $w := w - h_{i,n} v_i$ ;
end
 $h_{n+1,n} := \|w\|_2$ ;
 $v_{n+1} := w/h_{n+1,n}$ ;
compute  $y_n = \min_j \|\gamma e_1 - H_n^* y\|$ ;
end
 $x'_m := x'_0 + V_m y_m$ ;
if  $\|b' - BNx'_m\| \leq \epsilon$  then
stop iteration
endif
 $x'_0 := x'_m$ ;  $r_0 := b' - BNx'_0$ ;
 $\gamma := \|r_0\|_2$ ;  $v_1 := r_0/\gamma$ ;
goto start
if  $\|b' - BNx'_m\| \leq \epsilon$  then
 $x_2 = Nx'$ ;
 $x_1 = A_1^{-1}(b_1 - A_2 x_2)$ ;

```

図2 前処理付き GMRES(m) 法

4.1 Jacobi 法

Jacobi 法は、行列 B に対して近似逆行列 N を次のようにして求める反復法である。すなわち、 $n_{ij} \in N$ と $b_{ij} \in B$ に対して

$$n_{ij} = \begin{cases} 1/b_{ij} & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases} \quad (12)$$

というように近似逆行列を求める。この方法は近似逆行列を求める算法として最も基本的なものである。この方法を用いる場合、厳密な逆行列を求めることはできないが、計算量と記憶容量は非常に少なくすむ。またこの方法には並列計算が行いやすいという利点もある。

4.2 Newton 法

Newton 法³⁾は、方程式 $f(x) = 1/x - a = 0$ に対して Newton 法を適用することに基づく方法であり、行列 B に対して近似逆行列 N を次のようにして求める反復法である。すなわち、初期解 $N_0 = D^{-1}$ とする。ここで D^{-1} とは次のようなものである。 $d_{ij} \in D^{-1}$ と $b_{ij} \in B$ に対して

$$d_{ij} = \begin{cases} 1/b_{ij} & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases} \quad (13)$$

つまり、上記の Jacobi 法で求められるものと等しいものである。この $N_0 = D^{-1}$ を用いて、次のような反復式で計算できることになる。

$$N_{k+1} = (2I - N_k B) N_k \quad (14)$$

Newton 法の反復回数については、行列 N の優対角性や、どの程度疎行列かということにもよるが、ここでは 2 回とした。これは反復回数を 1 回から 10 回まで変化させた時に 2 回の時が最も収束時間が短かったからである。この方法は式 (14) を見ると分かるように行列の演算が多く使われているため、生成される近似逆行列が密構造になり、計算コストがかかる。

4.3 要素の付加

上で述べたように Newton 法は密構造の近似逆行列を構成するため、計算時間の短縮が難しくなっている。そこで行列の演算にかかる時間を減少させるために次のような方法を用いた。行列 $N(7)$ として、元の行列 N の七重対角部分は元の N と等しい値を持ち、それ以外の部分はゼロを要素に持つような行列を考える。つまり $n(7)_{ij} \in N(7)$ と $n_{ij} \in N$ に対して

$$n(7)_{ij} = \begin{cases} n_{ij} & \text{if } |i-j| < 5 \\ 0 & \text{else} \end{cases} \quad (15)$$

という要素を持つ行列を考える。

また同様に近似逆行列 $N(11)$ として、 $n(11)_{ij} \in N(11)$ と $n_{ij} \in N$ に対して

$$n(11)_{ij} = \begin{cases} n_{ij} & \text{if } |i-j| < 6 \\ 0 & \text{else} \end{cases} \quad (16)$$

を考える。

5. 数値実験

本稿で述べた前処理を施した GMRES(m) 法を計算機に実装し従来の GMRES(m) 法と比較した数値実験を行なった。

数値実験は以下の環境で行なった。

- 計算機: Compac GS160
- OS: Tru64 V5.1A
- CPU: 1 GHz Alpha 21264 × 16
- Memory: 16GB

使用するプロセッサの数は 8 つとし、数値実験は以下の条件のもとで行なった。

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

実装の際の並列化に関しては、行列とベクトルの積、ベクトルの内積、ベクトルの和、ベクトルのスカラー倍の計算部分を並列化した。

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。

$$-u_{xx} - u_{yy} + Du_x(x, y) = G(x, y)$$

$$u(x, y)|_{\partial\Omega} = 1.0$$

ただし、メッシュ幅を $h = 1/513$ として、この方程式を 5 点中心差分近似を用いて離散化する。これにより得られた行列の次元は 262144 である。また前処理を用いた時に実際に計算する方程式の行列の次元、つまり reduced system の次元の大きさは 131072 である。真の解を $u(x, y) = 1.0$ と設定して右辺を決定し、 $Dh = 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}$ と変化させる。また全ての算法のリスタート周期は $m = 10, 20, 30, 40, 50$ と変化させて数値実験を行なった。また、計算時間については clock() 関数で求めた値を秒単位で表した。その計算時間は 3 回の数値実験による平均値を用いた。反復回数については、GMRES(m) のなかのアーノルディ過程の 1 反復につき 1 回と数えた。ここで、シュールコンプリメントの近似逆行列を求める方法として Jacobi 法、Newton 法で構成されたものに対して要素の付加を施した方法を用いたものをそれぞれ RJ-GMRES(m) 法、RN(7)-GMRES(m) 法、RN(11)-GMRES(m) 法と記述することにする。

この結果を表 1 に示す。ここでの結果は、各解法に対する反復回数 (\approx 行列とベクトルとの積の回数) と計算時間を記述した。この表を見ると反復回数においては reduced system による前処理付き GMRES(m) 法が従来の GMRES(m) 法より大幅に減少している。なかでも RN(11)-GMRES(m) 法が最も反復回数が少ない。また計算時間において従来の GMRES(m) 法と RJ-GMRES(m) 法を比較すると、 $Dh = 2^{-5}$ 、リスタート周期 $m = 20$ の時に最大で 70% 短縮されている。RN(7)-GMRES(m) 法、RN(11)-GMRES(m) 法と従来の GMRES(m) 法との比較では、どちらもの方法も $Dh = 2^{-5}$ 、リスタート周期 $m = 20$ の時に最大で約 75% 短縮されている。RN(7)-GMRES(m) 法と RN(11)-GMRES(m) 法とを比較すると、反復回数は当然ながら RN(11)-GMRES(m) 法の方が少ない。計算時間に関しては、RN(7)-GMRES(m) 法の方が短縮されている場合とそうではない場合が同じくらいである。また RN(7)-GMRES(m) 法、RN(11)-GMRES(m) 法の両方法と RJ-GMRES(m) 法とを比較すると、リスタート周期 m が小さい時は RJ-GMRES(m) 法の方が計算時間が短く、リスタート周期 m が大きくなると RN(7)-GMRES(m) 法、RN(11)-GMRES(m) 法が計算時間が短くなるという傾向がある。

また $Dh = 2^{-5}$ 、リスタート周期 $m = 20$ の時の反復回数、計算時間のそれぞれに対する残差ノルムの推移をグラフにしたものを図 3、図 4 に示す。計算時間のグラフを見ると、RN(7)-GMRES(m) 法が RN(11)-

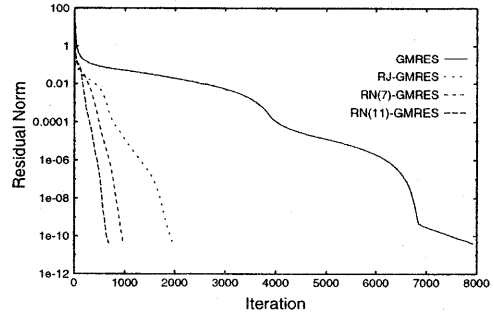


図 3 数値例 1 ($Dh = 2^{-5}, m = 20$) における反復回数と残差ノルムの推移

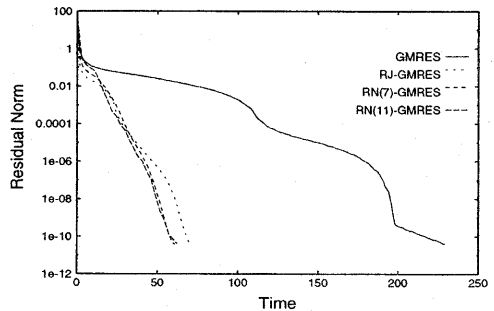


図 4 数値例 1 ($Dh = 2^{-5}, m = 20$) における計算時間と残差ノルムの推移

GMRES(m) 法より計算時間が短くなってはいるが、それほど大きな差は見られない。

5.2 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。

$$-u_{xx} - u_{yy} + D\{(y - 1/2)u_x(x, y) + (x - 1/3)(x - 2/3)u_y(x, y)\} = G(x, y)$$

$$u(x, y)|_{\partial\Omega} = 1 + xy$$

数値例 1 と同様にメッシュ幅を $h = 1/513$ として、この方程式を 5 点中心差分近似を用いて離散化する。これにより得られた行列の次元は 262144 であり、前処理を用いた時に実際に計算する reduced system の行列の次元は 131072 である。真の解を $u(x, y) = 1.0 + xy$ と設定して右辺を決定し、 $Dh = 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}$ と変化させる。また全ての算法のリスタート周期は $m = 10, 20, 30, 40, 50$ と変化させて数値実験を行なった。また計算時間についても数値例 1 同様、clock() 関数で求めた値を秒単位で表し、その計算時間は 3 回の数値実験による平均値を用いた。反復回数については、GMRES(m) 法のなかのアーノルディ過程の 1 反復につき 1 回と数えた。ここでも、シュールコンプリメントの近似逆行列を求める方法として Jacobi 法、Newton 法に要素の付加を施したものをそ

表 1 数値例 1 の結果 (I : 反復回数, T : 計算時間 (秒))

Dh	2^{-5}		2^{-4}		2^{-3}		2^{-2}	
	I	T	I	T	I	T	I	T
Method								
GMRES(10)	—	—	6758	126.81	4056	76.23	1834	33.21
GMRES(20)	7920	245.42	3876	129.68	2520	84.24	2038	48.17
GMRES(30)	5821	287.32	2996	148.12	2038	100.76	2348	123.07
GMRES(40)	4396	263.61	2638	174.36	2316	153.32	2543	156.82
GMRES(50)	4198	263.62	3092	259.72	2586	215.32	2617	160.35
RJ-GMRES(10)	3630	78.45	1602	58.00	915	33.15	741	22.12
RJ-GMRES(20)	1980	75.99	1140	73.09	869	44.65	960	34.81
RJ-GMRES(30)	1575	88.44	1034	77.48	1119	84.12	1239	85.00
RJ-GMRES(40)	1259	96.24	1196	122.21	1308	134.28	1318	136.10
RJ-GMRES(50)	1200	117.09	1118	146.08	1647	214.92	1549	149.01
RN(7)-GMRES(10)	1707	74.91	960	54.68	629	35.52	615	30.02
RN(7)-GMRES(20)	989	61.11	749	60.88	717	58.60	645	37.51
RN(7)-GMRES(30)	944	75.81	891	96.20	896	96.52	912	86.91
RN(7)-GMRES(40)	896	87.96	1076	141.10	890	117.41	892	103.84
RN(7)-GMRES(50)	900	110.49	1047	172.24	1191	196.92	1410	126.78
RN(11)-GMRES(10)	990	73.41	537	53.00	449	44.12	531	31.79
RN(11)-GMRES(20)	687	63.69	569	70.12	743	92.41	630	42.78
RN(11)-GMRES(30)	672	74.34	674	99.12	806	118.83	883	88.91
RN(11)-GMRES(40)	654	86.64	767	136.48	897	151.77	853	99.67
RN(11)-GMRES(50)	674	104.10	594	122.88	665	138.72	1208	100.76

—は反復回数が 10000 回以上の時

それぞれ RJ-GMRES(m) 法, RN(7)-GMRES(m) 法, RN(11)-GMRES(m) 法と記述することにする。

この結果を表 2 に示す。ここでの結果は、各解法に対する反復回数 (≈ 行列とベクトルとの積の回数) と計算時間を表した。この表を見ると、反復回数においては RN(11)-GMRES(m) 法が最も少なく、従来の GMRES(m) 法と比べて、大きく減少している。また計算時間においては RJ-GMRES(m) 法と従来の GMRES(m) 法と比較して、最大で 43% ($Dh = 2^{-3}$, リスタート周期 $m = 30$ の時) 短縮されている。RN(7)-GMRES(m) 法, RN(11)-GMRES(m) 法と従来の GMRES(m) 法との比較では RN(7)-GMRES(m) 法は最大で 58% ($Dh = 2^{-4}$, リスタート周期 $m = 20$ の時), RN(11)-GMRES(m) 法は 60% ($Dh = 2^{-4}$, リスタート周期 $m = 20$ の時) 短縮されている。また RN(7)-GMRES(m) 法と RN(11)-GMRES(m) 法との比較では、多くの場合において RN(11)-GMRES(m) 法が RN(7)-GMRES(m) 法よりも計算時間が短縮されている。RJ-GMRES(m) 法と RN(7)-GMRES(m) 法, RN(11)-GMRES(m) 法の両方法とを比較すると計算時間において RN(7)-GMRES(m) 法, RN(11)-GMRES(m) 法は RJ-GMRES(m) 法よりも短くなっていることが多い。

また $Dh = 2^{-5}$, リスタート周期 $m = 40$ の時の反復回数, 計算時間のそれぞれに対する残差ノルムの推移をグラフにしたものを図 5, 図 6 に示す。これらのグラフから, RN(11)-GMRES(m) 法が計算時間, 反復回数において他の方法よりも大きな効果が見られる。また reduced system による前処理を施した 3 つの方

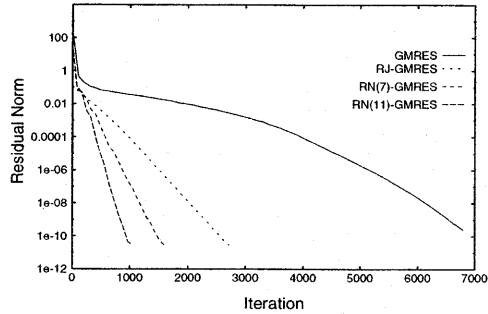


図 5 数値例 2 ($Dh = 2^{-5}$, $m = 40$) における反復回数と残差ノルムの推移

法が従来の GMRES(m) 法に対して計算時間, 反復回数のどちらにおいても大きく減少していることがわかる。

5.3 台数効果

CPU 台数	GMRES(50)	RN(11)-GMRES(50)
1	1.00	1.00
2	1.79	1.75
4	3.58	3.51
8	6.80	6.67

表 3 台数効果

ここでは、一例として数値例 1 で, $Dh = 2^{-3}$ における GMRES(50) 法と RN(11)-GMRES(50) 法の並列処理の台数効果を挙げる。CPU 台数を 1, 2, 4, 8 台と変化させ, 表 5.3 に台数効果を示す。2 つの方

表2 数値例2の結果 (I: 反復回数, T: 計算時間(秒))

Dh	2^{-5}		2^{-4}		2^{-3}		2^{-2}	
	I	T	I	T	I	T	I	T
Method								
GMRES(10)	—	—	—	—	—	—	6550	102.57
GMRES(20)	—	—	9648	231.08	5900	143.96	4520	97.01
GMRES(30)	—	—	5545	192.52	4573	160.20	3872	132.67
GMRES(40)	6798	312.44	4997	229.72	5398	250.56	3979	201.82
GMRES(50)	6497	376.36	4748	275.04	4623	270.52	3870	248.71
RJ-GMRES(10)	6640	158.80	3996	95.43	2641	63.16	2200	52.18
RJ-GMRES(20)	2676	105.96	2119	82.88	2200	86.08	1598	64.35
RJ-GMRES(30)	2155	120.04	1980	110.36	1618	90.61	1666	96.89
RJ-GMRES(40)	2716	203.72	1753	131.67	1675	126.2	1753	161.07
RJ-GMRES(50)	2497	236.44	1792	169.61	1597	151.52	1686	168.06
RN(7)-GMRES(10)	3178	158.68	1818	89.96	1339	64.41	1365	80.01
RN(7)-GMRES(20)	2198	145.92	1480	96.40	1358	88.72	1148	72.61
RN(7)-GMRES(30)	1497	123.12	1258	104.32	1706	88.35	1342	100.75
RN(7)-GMRES(40)	1594	157.45	1355	135.08	1193	119.52	1380	120.82
RN(7)-GMRES(50)	1497	183.36	1297	159.42	1199	146.39	1399	149.81
RN(11)-GMRES(10)	1640	136.64	1138	94.65	1020	84.52	780	70.17
RN(11)-GMRES(20)	1353	136.92	912	92.28	836	84.21	876	79.98
RN(11)-GMRES(30)	1077	128.48	779	91.12	891	106.72	947	112.75
RN(11)-GMRES(40)	1034	142.48	947	131.92	878	119.96	1028	110.92
RN(11)-GMRES(50)	1001	158.72	896	143.08	1185	189.53	1153	120.74

—は反復回数が10000回以上の時

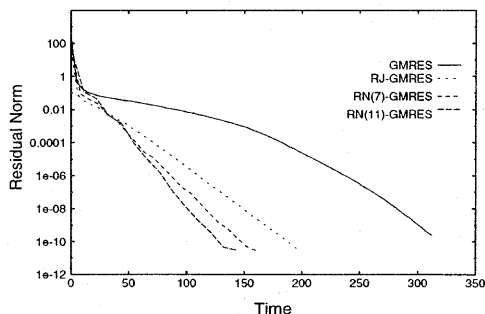


図6 数値例2($Dh = 2^{-5}$, $m = 40$)における計算時間と残差ノルムの推移

法における台数効果の違いは10%程であった。しかし、表1より実際の計算時間は10%以上の差があった。また同様に表1より、RN(11)-GMRES(50)法の反復回数がGMRES(50)法よりも減少していることから、RN(11)-GMRES(50)法の性能がGMRES(m)法に比べて、向上した原因は残差ノルムの収束性が向上したためである。

6. おわりに

上記の数値実験ではブロック分割による reduced system の前処理付き GMRES(m)法の有効性を示した。シュールコンプリメントに対する前処理として Newton 法に要素の付加を施した方法と Jacobi 法を用いた。反復回数においては Newton 法に要素の付加を施した方法は Jacobi 法より大きく減少したが、計

算時間においてはそれほど大きく減少はしなかった。これは Newton 法では Jacobi 法よりも行列とベクトルの積を多く計算しなければならないことが原因として考えられる。

従って今後の課題としては Newton 法をさらに改良する、特に要素の付加を行う部分を改良すること、またシュールコンプリメントに対する前処理として LU 分解などの別の方法を用いるということが挙げられる。

参考文献

- 1) Grosz, L.: *Preconditioning by incomplete block elimination*, Numerical Linear Algebra with Applications, 2000; 7: 527-541.
- 2) Saad, Y. and Zhang, Z.: *Diagonal threshold techniques in robust multi-level ILU preconditioners for general sparse linear systems*, Numerical Linear Algebra with Applications, 1999; 21: 257-280.
- 3) Pan, V. and Schreiber, R.: *An improved Newton iteration for the generalized inverse of a matrix with applications*, SIAM Journal on Scientific and Statistical Computing, 1991; 12: 1109-1130.
- 4) Saad Y. and Schultz M. H.: *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 1986; 7: 856-869.