

Origin2400 における RB オーダリングによる 不完全ブロック分解前処理の性能評価

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次方程式の解法のひとつに GMRES(m) 法がある。通常、この方法に限らず反復法では前処理を用いて収束性を改善させる。前処理にも様々なものがあるが、ここでは Y. Saad らが提案した不完全 LU 分解 (ILU) に類似した、消去に基づくブロック分解を利用した方法を使用する。この方法を用いると、与えられた連立1次方程式のほぼ半分の次元の方程式を反復法で解くことにより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、記憶容量を削減することが可能となる。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。

Evaluation of Preconditioning by using block factorization with RB-ordering on Origin2400

TOSHIO INOUE† and TAKASHI NODERA††

We consider the linear systems of equations with a large, sparse, nonsingular matrix. The GMRES(m) method is one of major iterative solvers for solving these linear systems of equations. The computation and memory cost of this method is more expensive as the number of iteration increase. Consequently, the GMRES(m) method is usually requires preconditioning to reduce the cost of work. Although there are various methods of preconditioning, in this paper, we mainly describe the preconditioning using both block elimination and block factorization. At last, numerical experiments are given for showing that the GMRES(m) method with this preconditioning is more effective than the standard GMRES(m) method.

1. はじめに

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。ただし、係数行列 A は RB-オーダリング (RB-ordering) によって生成されたものとする。方程式 (1) の解法には様々な方法が存在するが、式 (1) の係数行列 A が対称行列である場合には現時点では 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) を用いて行列 A をブロック分解して、そのブロック分解を利用して行列 A の近似逆行列を求める方法について述べる。

まず2節で GMRES(m) 法について述べる。3節および4節では実際に GMRES(m) 法に対して施す前処理の構成方法について述べ、5節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節においてまとめと結論を述べる。

† 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

2. GMRES(m) 法

GMRES 法は、1986 年に Saad 等⁴⁾ 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 \mathbf{x}_0 とする。初期残差ベクトルを $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ 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。

$$\min_{\mathbf{z} \in K_i} \|\mathbf{b} - A(\mathbf{x}_0 + \mathbf{z})\|_2 = \min_{\mathbf{z} \in K_i} \|\mathbf{r}_0 - A\mathbf{z}\|_2$$

ただし、ここで、 \mathbf{z} はクリロフ部分空間:

$$K_i(A, \mathbf{r}_0) = \{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{i-1}\mathbf{r}_0\}$$

上の正規直交ベクトル列 $V_i = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_i\}$ によって $\mathbf{z} = V_i \mathbf{y}$ と表され、 \mathbf{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。

$$\begin{aligned} J(\mathbf{y}) &= \|\beta \mathbf{v}_1 - AV_i \mathbf{y}\|_2 \\ &= \|V_{i+1}(\beta \mathbf{e}_1 - H_i^* \mathbf{y})\|_2 \\ &= \|\beta \mathbf{e}_1 - H_i^* \mathbf{y}\|_2 \end{aligned} \quad (2)$$

ただし、 $\beta = \|\mathbf{r}_0\|_2$ 、 $\mathbf{e}_1 = (1, 0, \dots, 0)^T$ 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 \mathbf{y} を用いて第 i 回目の近似解は、 $\mathbf{x}_i = \mathbf{x}_0 + V_i \mathbf{y}$ 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 \mathbf{x}_0 を \mathbf{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、以降本稿ではこの方法を用いる。

3. ブロック分解を用いた前処理

GMRES(m) 法に対して前処理を施すとは式 (1) の係数行列 A の近似逆行列 M を前処理行列とする GMRES(m) 法を考えることに等しい。つまり、ここでは左前処理を用いることにすると

$$MA\mathbf{x} = M\mathbf{b} \quad (3)$$

という連立 1 次方程式を考えることになる。このような方程式を考えることは 1 回の反復における計算量は増えることになるが、収束までの反復回数を減らすことができる。

この節では連立 1 次方程式 (1) に対して施す前処理としてブロック小行列の消去とブロック分解により前処理行列を構成する方法について述べる。

連立 1 次方程式 (1) を

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (4)$$

とブロックに分割する。この時 A_1, A_4 は対角行列とする。このブロックで考えた 2 元連立 1 次方程式 (4)

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。

$$(A_4 - A_3 A_1^{-1} A_2) x_2 = b_2 - A_3 A_1^{-1} b_1 \quad (5)$$

今、この方程式 (5) 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。この B を用いて A をブロック分解すると

$$\begin{aligned} A &= \begin{bmatrix} I & 0 \\ A_3 Y & I \end{bmatrix} \begin{bmatrix} A_1 & 0 \\ 0 & B \end{bmatrix} \\ &\times \begin{bmatrix} I & Y A_2 \\ 0 & I \end{bmatrix} \end{aligned} \quad (6)$$

の様に ILU 分解に類似した形で表される。ただし $Y = A_1^{-1}$ とする。この行列 A の近似逆行列 M は、それぞれのブロックについて逆行列を計算することで容易に求めることができる。

$$\begin{aligned} M &= \begin{bmatrix} I & -Y A_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} Y & 0 \\ 0 & N \end{bmatrix} \\ &\times \begin{bmatrix} I & 0 \\ -A_3 Y & I \end{bmatrix} \end{aligned} \quad (7)$$

ただし、ここで N は B の近似逆行列を表す。この M を式 (1) の前処理行列として利用する。つまり、式 (6)、式 (7) を式 (3) に代入して展開すると、次のような形になる¹⁾。

$$NBx_2 = N(b_2 - A_3 Y b_1) \quad (8)$$

$$x_1 = Y(b_1 - A_2 x_2) \quad (9)$$

ここで、 A_1 を対角行列とする様な係数行列 A を持つ連立 1 次方程式を考えている理由について述べる。実際の計算において、 A_1 の逆行列 Y も必要となるのだが、式 (8) と式 (9) を見るとわかるように A_1 の逆行列 Y はどちらの式にも存在しており、この Y を近似逆行列で置き換えると、式 (9) での代入においても誤差が伝わることになる。しかし、 A_1 に対角行列を用いることで正確な逆行列を求めることができ、その結果、式 (9) の部分における誤差を押えることができる。このため式 (1) の係数行列 A の条件に RB-オーダリングによって生成したものをを用いるという条件を付け加えている。ただし、 A_1 を対角行列にしなければならないため、 A_1 の次元はあまり大きくとることができない。つまり、実際に反復法を用いて計算する方程式 (8) の係数行列の次元を小さくすることができにくくなる。

このような形にしたことにより式 (8) の解が得られれば、式 (9) 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 (8) 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。また、同様の理由から B の近似逆行列 N を求め

```

choose  $x'_0$ 
 $b' := N(b_2 - A_3 A_1^{-1} b_1)$ ;
 $r_0 := b' - N B x'_0$ ;
 $\gamma := \|r_0\|_2$ ;  $v_1 := r_0/\gamma$ ;
start
for  $n := 1$  to  $m$  do
begin
 $k_1 := A_2 v_n$ ;
 $k_2 := A_1^{-1} k_1$ ;
 $k_3 := A_3 k_2$ ;
 $k_4 := A_4 v_n$ ;
 $k_5 := k_4 - k_3$ ;
 $w := N k_5$ ;
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  $y_n = \min_y \|\gamma e_1 - H_n^* y\|$ ;
end
 $x'_m := x'_0 + V_m y_m$ ;
if  $\|b' - N B x'_m\| \leq \epsilon$  then
stop iteration
endif
 $x'_0 := x'_m$ ;  $r_0 := b' - N B x'_0$ ;
 $\gamma := \|r_0\|_2$ ;  $v_1 := r_0/\gamma$ ;
goto start
if  $\|b' - N B x'_m\| \leq \epsilon$  then
 $x_2 = x'$ ;
 $x_1 = A_1^{-1}(b_1 - A_2 x_2)$ ;

```

図1 前処理付き GMRES(m) 法

るのが難しくなることも欠点である。

この前処理付き GMRES(m) 法のアルゴリズムを図1に示す。

4. シュールコンプリメントの近似逆行列

この節では反復法を用いて実際に解く連立1次方程式(8)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。

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

$$n_{ij} = \begin{cases} 1/b_{ij} & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases} \quad (10)$$

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済む。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いている方法であり、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

$$d_{ij} = \begin{cases} 1/b_{ij} & \text{if } i=j \\ 0 & \text{if } i \neq j \end{cases} \quad (11)$$

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。

$$N_{k+1} = (2I - N_k B) N_k \quad (12)$$

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。またこの方法は式(12)を見ると分かるように行列の演算が多く使われている。このために計算量が多くなる。また行列の演算が多いため並列化が難しい。

4.3 Newton 法の改良版

上で述べた様に Newton 法は行列の演算が多く使われていることにより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様な方法を用いた。行列 B' として元の行列 B の五重対角部分のみを要素として持つ行列を考える。つまり $b'_{ij} \in B'$ と $b_{ij} \in B$ に対して

$$b'_{ij} = \begin{cases} b_{ij} & \text{if } |i-j| < 3 \\ 0 & \text{else} \end{cases} \quad (13)$$

という要素を持つ行列を考える。そしてこの B' に対して上の Newton 法を行う。これにより行列の演算にかかる時間は短縮されるが、求められる近似逆行列は上記の Newton 法よりも厳密ではなくなる。また、上記の Newton 法との比較のために反復回数は2回とした。

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: Origin2400
- OS: IRIX6.5.12
- CPU: MIPS R12000 300MHz × 16
- Memory: 8GB

また使用するプロセッサの数は4つとし、数値実験は以下の条件のもとで行った。

- 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

また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 $h = 1/257$ とし、この方程式を5点中心差分近似を用いて離散化する。また上で述べた条件により領域のオーダリングにはRB-オーダリングを用いた。これにより得られた行列の次元は65536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は32768である。真の解を $u(x, y) = 1.0$ と設定して右辺を決定し、 $Dh = 2^{-2}, 2^{-1}, 2^0, 2^1$ 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、Newton法の改良版を用いたものをそれぞれSchur-J-GMRES(m)法、Schur-N-GMRES(m)法、Schur-MN-GMRES(m)法と表すことにする。

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(≈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全ての前処理付き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Schur-N-GMRES(m)法が最も反復回数が少ない。また計算時間においては従来のGMRES(m)法とSchur-J-GMRES(m)法とを比較すると、 $Dh = 2^{-2}$ 、リスタート周期 $m = 10$ の時に最大で、Schur-J-GMRES(m)法が80%ほど計算時間が短縮されている。Schur-N-GMRES(m)法と従来のGMRES(m)法を比較すると、 $Dh = 2^{-2}$ 、リスタート周期 $m = 10$ の時が最大で、計算時間が60%ほど短縮されているが、あま

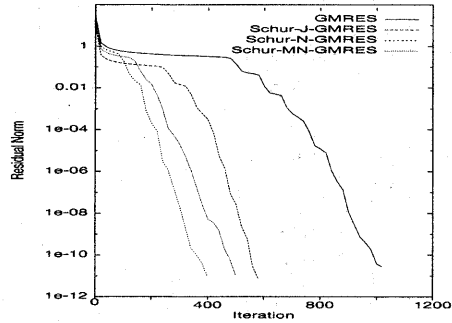


図2 数値例1($Dh = 2^{-1}, m = 20$)における反復回数と残差ノルムの推移

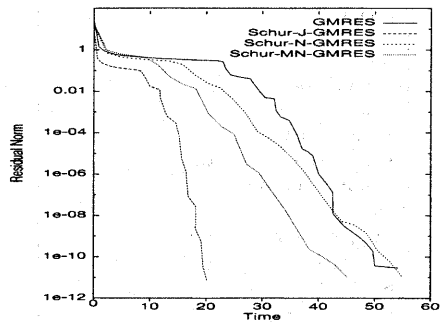


図3 数値例1($Dh = 2^{-1}, m = 20$)における計算時間と残差ノルムの推移

り短縮されていない場合も多い。これはNewton法はその計算のなかで行列とベクトルの積が多く行われることとそのため並列計算に適していないことが原因としてあげられる。またSchur-N-GMRES(m)法とSchur-MN-GMRES(m)法とを比較すると、反復回数は当然Schur-N-GMRES(m)法の方が少ないが、計算時間はSchur-MN-GMRES(m)法の方がSchur-N-GMRES(m)法よりも僅かではあるが短縮されている。

また $Dh = 2^{-1}$ 、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2、図3に示す。これらのグラフからは、反復回数においてはSchur-N-GMRES(m)法が従来のGMRES(m)法の50%以下になっているのに対して、計算時間においては両方法の間に大きな差が見られないというSchur-N-GMRES(m)法の特徴がよく見て取れる。

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。

表1 数値例1の結果(N: 反復回数, T: 計算時間(秒))

Dh	2 ⁻²		2 ⁻¹		2 ⁰		2 ¹	
	I	T	I	T	I	T	I	T
GMRES(10)	2420	93.30	1129	44.22	909	35.26	969	37.22
GMRES(20)	1339	73.60	1020	54.41	1099	60.00	1221	65.10
GMRES(30)	1409	99.03	1109	78.04	1309	93.82	1281	87.92
GMRES(40)	1319	112.59	1399	118.96	1280	108.99	1323	115.93
GMRES(50)	1394	142.03	1448	145.38	1598	161.51	1483	154.51
Schur-J-GMRES(10)	619	17.56	526	14.92	604	16.67	568	15.72
Schur-J-GMRES(20)	599	21.20	580	20.46	720	25.47	720	25.16
Schur-J-GMRES(30)	749	32.15	808	34.69	868	36.75	746	31.66
Schur-J-GMRES(40)	839	41.55	838	41.93	998	49.93	918	45.71
Schur-J-GMRES(50)	1000	56.96	1048	60.23	1099	62.46	1049	59.35
Schur-N-GMRES(10)	259	33.29	327	34.32	310	36.32	302	34.95
Schur-N-GMRES(20)	377	51.65	398	53.62	399	51.75	449	54.19
Schur-N-GMRES(30)	480	83.75	447	75.41	468	64.48	437	79.44
Schur-N-GMRES(40)	395	68.88	479	89.15	559	79.94	483	73.82
Schur-N-GMRES(50)	397	99.56	447	93.16	498	108.83	408	86.54
Schur-MN-GMRES(10)	349	26.72	360	33.08	379	31.25	355	31.04
Schur-MN-GMRES(20)	479	43.41	499	44.61	479	45.06	500	51.53
Schur-MN-GMRES(30)	686	60.26	629	55.93	531	60.00	652	56.62
Schur-MN-GMRES(40)	511	55.21	679	64.90	600	77.05	560	70.88
Schur-MN-GMRES(50)	697	59.93	648	67.75	744	74.56	593	65.99

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

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

数値例1と同様にメッシュ幅を $h = 1/257$ とし、この方程式を5点中心差分近似を用いて離散化する。また領域のオーダリングにはRB-オーダリングを用いた。これにより得られた行列の次元は65536であり、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は32768である。真の解を $u(x, y) = 1.0 + xy$ と設定して右辺を決定し、 $Dh = 2^{-2}, 2^{-1}, 2^0, 2^1$ 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、Newton法の改良版を用いたものをそれぞれSchur-J-GMRES(m)法、Schur-N-GMRES(m)法、Schur-MN-GMRES(m)法と表すことにする。

この結果を表2に示す。この結果は、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Schur-N-GMRES(m)法が最も少なく、しかも従来のGMRES(m)法と比べて大幅に減少している。また計算時間においてはSchur-J-GMRES(m)法と従来のGMRES(m)法とを比較すると50% ($Dh = 2^1$, リスタート周期 $m = 10$ の時) から80% ($Dh = 2^{-2}$, リスタート周期 $m = 20$ の時) 短縮されている。またSchur-N-GMRES(m)法を見ると、従来の

GMRES(m)法よりも収束するまでの計算時間が遅いことが多くある。しかし、Schur-MN-GMRES(m)法と従来のGMRES(m)法を比較すると計算時間が遅いことがなくなっていて、 $Dh = 2^{-2}$, リスタート周期 $m = 10$ の時には80%ほど短縮されている。しかし、リスタート周期 m が大きくなると計算時間は従来のGMRES(m)法とあまり変わらなくなる。Schur-MN-GMRES(m)法とSchur-N-GMRES(m)法を比較すると、それぞれの方法の特徴により反復回数はSchur-N-GMRES(m)法の方が少なくなっているが、計算時間はSchur-MN-GMRES(m)法が1度の演算にかかる計算時間を減少させるためにSchur-N-GMRES(m)法よりも短縮している。

また、 $Dh = 2^0$, 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4、図5に示す。これらのグラフからわかることは、計算時間においてSchur-J-GMRES(m)法は他の方法よりも大きな効果が見られる。またSchur-N-GMRES(m)法とSchur-MN-GMRES(m)法を見比べるとそれほど大きくはないが、Schur-MN-GMRES(m)法はSchur-N-GMRES(m)法よりも良い結果を示している。

6. おわりに

上記の数値実験では消去によるブロック分解を用いた前処理付きGMRES(m)法の有効性を示したが、シュールコンプリメントに対する前処理にJacobi法を用いた場合とNewton法を用いた場合で効果に大きな差が認められた。これはNewton法では行列とベクトルの積を数多く計算しなければならないことが原因と

表 2 数値例 2 の結果 (I : 反復回数, T : 計算時間 (秒))

Dh	2^{-2}		2^{-1}		2^0		2^1	
	I	T	I	T	I	T	I	T
Method								
GMRES(10)	6870	257.49	4000	148.88	2300	87.58	1840	69.77
GMRES(20)	3319	179.18	2039	109.08	1840	99.63	1960	105.17
GMRES(30)	2670	182.74	1829	128.53	1648	112.27	2009	136.51
GMRES(40)	2039	172.85	1879	159.63	1838	156.01	2078	176.03
GMRES(50)	2050	203.66	1795	180.26	1849	189.59	2247	224.09
Schur-J-GMRES(10)	1729	57.90	1169	39.32	720	24.15	939	31.61
Schur-J-GMRES(20)	939	38.68	878	36.21	897	36.62	940	39.79
Schur-J-GMRES(30)	960	46.21	898	42.75	959	45.63	1018	48.93
Schur-J-GMRES(40)	798	44.52	839	46.07	919	51.01	1117	62.60
Schur-J-GMRES(50)	797	49.76	999	61.74	1044	66.36	1098	71.05
Schur-N-GMRES(10)	414	65.27	558	90.52	558	100.20	792	101.69
Schur-N-GMRES(20)	596	98.40	636	116.11	592	100.12	838	110.82
Schur-N-GMRES(30)	840	134.66	710	139.42	702	115.80	954	148.73
Schur-N-GMRES(40)	1036	194.50	872	155.38	720	143.60	1118	215.76
Schur-N-GMRES(50)	1046	183.66	900	194.14	698	178.10	1200	225.59
Schur-MN-GMRES(10)	558	50.94	800	65.52	878	66.34	898	94.35
Schur-MN-GMRES(20)	756	78.12	918	82.49	798	78.94	878	108.28
Schur-MN-GMRES(30)	958	120.72	1014	102.42	838	102.36	1076	135.74
Schur-MN-GMRES(40)	1268	162.68	1028	135.68	958	111.80	1430	171.24
Schur-MN-GMRES(50)	1098	182.58	1198	151.10	1098	115.73	1398	197.78

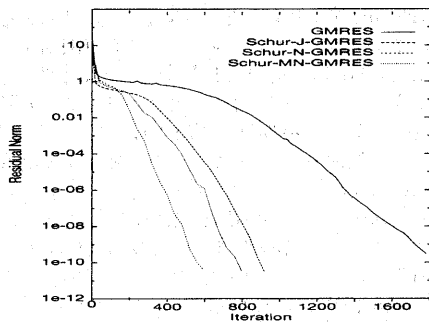


図 4 数値例 2 ($Dh = 2^0, m = 20$) における反復回数と残差ノルムの推移

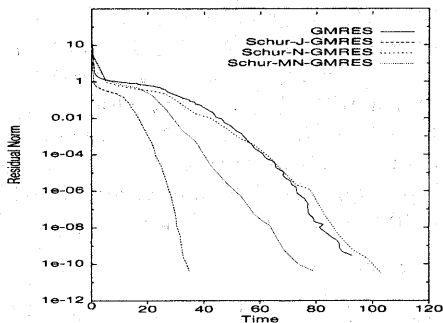


図 5 数値例 2 ($Dh = 2^0, m = 20$) における計算時間と残差ノルムの推移

して考えられる。これを解決するために Newton 法を改良したが、Jacobi 法を用いたものほどの効果は得られなかった。従って今後の課題としては (i) Newton 法をさらに改良する, (ii) LU 分解などの直接法をシュールコンプリメントに対する前処理として利用することがあげられる。また (iii) シュールコンプリメント $B = A_4 - A_3 A_1^{-1} A_2$ の形を利用して近似逆行列を求めるという方法もあげられる。

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(1): 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(5): 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.