

## DQGMRES( $m, k$ ) 法とその前処理について

大澤 史典<sup>†</sup> 佐藤 拓郎<sup>†</sup> 野寺 隆<sup>†</sup>

連立1次方程式の近似解法の中で、GMRES法の打ち切り版として知られるDQGMRES( $k$ )法は、記憶容量や計算コストを軽減する算法である。しかし、この打ち切り版は、不完全直交化によってしばしば算法の収束性を悪化させる。そこで、本稿ではリスタートを利用して、残差の収束性を改善させるDQGMRES( $m, k$ )法を提案する。数値実験によれば、適切な前処理を行なうDQGMRES( $m, k$ )法は、他の算法と比較して記憶容量と計算コストを減少させることができる。

### DQGMRES( $m, k$ ) method and its preconditioning

FUMINORI OHSAWA,<sup>†</sup> TAKURO SATO<sup>†</sup> and TAKASHI NODERA<sup>†</sup>

The DQGMRES( $k$ ) method for solving nonsymmetric linear system of equations is generally used with truncated procedure to reduce storage and orthogonalization costs. However, the truncated version sometimes slow down the convergence in order to use incomplete orthogonalization. By using the restarting technique, we propose the DQGMRES( $m, k$ ) method which improves the convergence of residual. At last, the results of numerical experiments show that DQGMRES( $m, k$ ) method with appropriate preconditioning strategy gives a more robust algorithm than the other scheme, at a low cost of operations and memory.

#### 1. はじめに

科学技術計算の分野では、偏微分方程式の境界値問題を有限要素法や有限差分法を用いて離散化することにより得られる大型で疎な係数行列をもつ連立1次方程式

$$Ax = b \quad (1)$$

の近似解を必要とする場合がある。従来、この問題を解くための解法として直接法と反復法があるが、計算コストの面から非定常反復法が使われることが多い。反復法の代表的な解法としてGMRES法やQMR法、BiCGStab( $l$ )法等がある<sup>6)</sup>。

本稿で扱うDQGMRES( $m, k$ )法は、GMRES法を基にした解法であるDQGMRES法のリスタート版である。DQGMRES法とDQGMRES( $m, k$ )法は1反復あたりの計算コストがGMRES法、GMRES( $k$ )法と比べて小さいという利点があるが、その反面、不完全な直交化を用いてあるので、問題により残差ノルムの収束が不安定になるというデメリットを持つ。

第2節では、GMRES法とGMRES( $k$ )法、そして不完全なアーノルディ過程を利用するDQGMRES法について簡単に述べ、リスタートを行なうことで収束を向上させるDQGMRES( $m, k$ )法を提案する。次に、

第3節では、行列の前処理として近似逆行列の構成方法について述べる。最後に第4節において2つの数値実験について述べ、DQGMRES( $m, k$ )法の収束性についての考察を述べる。

#### 2. GMRES法、DQGMRES法に関連する解法について

##### 2.1 GMRES法

GMRES法は大型で疎な非対称の行列を係数行列として持つ連立1次方程式の近似解を求めるための反復解法の1つである。GMRES法ではアーノルディ原理を用いてクリロフ部分空間上に正規直交基底

$$V_n = (v_1, v_2, \dots, v_n) \quad (2)$$

を生成する。そして、このクリロフ部分空間上で残差ノルムを最小化する。残差ノルムは以下の式により与えられる。

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

式(2)で示された正規直交基底を用いて $z = V_n y$ と表すことができる。この $V_{n+1}$ はユニタリ行列であるため、この最小問題は次の最小2乗問題に帰着できる。

$$\begin{aligned} J(y) &= \|\gamma v_1 - AV_n y\|_2 \\ &= \|V_{n+1}(\gamma e_1 - \bar{H}_n y)\|_2 \\ &= \|\gamma e_1 - \bar{H}_n y\|_2 \end{aligned} \quad (3)$$

ただし $\gamma = \|r_0\|_2$ である。通常、式(3)はGivens法な

<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  $V_n$ ;
  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(k)法  
Fig. 1 GMRES(k) method.

どを用いて、上ヘッセンベルグ行列 $\bar{H}$ をQR分解することでも最小化することになる。そしてGMRES法における近似解 $x_n$ は

$$x_n = x_0 + V_n y_k$$

によって計算できることになる。

GMRES法では、正規直交系 $V_n$ と上ヘッセンベルグ行列を保存する必要があるため反復回数を増す毎に記憶容量と演算量が増加していくので実用的ではないと言われている。このため一般にはリスタート版のGMRES(k)法が使われている。このGMRES(k)法では、 $k$ 回の反復毎に求めた近似解 $x_k$ を初期近似解 $x_0$ として、残差ベクトルを再計算してリスタートをする。このGMRES(k)法はGMRES法と比べて演算量と記憶容量の面で効率の良い解法である。しかしながら、リスタートによって、近似解を構成するための固有ベクトルの情報が失われることもあり、残差ノルムのゼロへの収束が不規則になる場合もある。ここで、Saadら<sup>1)</sup>によって提案されたGMRES(k)法の算法を図1に示す。

## 2.2 DQGMRES(k)法

DQGMRES(k)法はGMRES法を基にした算法で、1996年にY. Saadら<sup>2)</sup>によって提案された。DQGMRES(k)法は、GMRES法で使用されているアーノルディ過程の代わりに不完全アーノルディ過程を用いて不完全直交化を行なう手法である。アーノルディ過程の完全直交化では、新しい正規直交基底ベクトル $v_n$ は、以前に求めた全ての $\{v_1, \dots, v_{n-1}\}$ と直交するように計算するが、この不完全直交化では、正規直交基底ベクトル $v_n$ は $\{v_{n-k}, \dots, v_{n-1}\}$ のみと直交するように計算する。これにより、正規直交基底ベクトルを保存に必要な記憶容量と、行列 $\bar{H}_n$ の保存に必

```

 $\hat{v} := Av_n$ ;
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.

```

 $\hat{v} := Av_n$ ;
for  $i := \max(1, n - k + 1)$  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3 不完全アーノルディ過程  
Fig. 3 Incomplete Arnoldi process.

要な記憶容量、そして $\bar{H}_n$ における内積演算の大半を削減でき、一般に用いられているアーノルディ過程に比べて計算コストを削減することができる。しかしながら、これは完全な正規直交化でないため、反復が進むにつれて不完全アーノルディ過程において求められる正規直交基底ベクトルの精度が悪くなる。そこで、問題によっては残差ノルムの収束が不安定になることを考慮しなければならない。不完全アーノルディ過程の算法を図3に示す。

## 2.3 DQGMRES(m,k)法

DQGMRES(m,k)法はDQGMRES(k)法のリスタート版である。ここでは $m$ はリスタート周期であり、 $k$ は正規直交基底ベクトルを保存する数である。DQGMRES(m,k)法はリスタートを行なうことで、各反復において計算する正規直交基底ベクトルの精度をある程度保つことができるが、GMRES(k)法と同様に、リスタートの際に近似解を構成する固有ベクトルの情報を欠損する。

## 3. 近似逆行列による前処理

前処理とは連立1次方程式の近似解を効率的に求めるための式(1)の行列変換である。この手法の1つに次のような右前処理がある。

$$AMy = b, \quad x = My$$

ただし、 $M$ は前処理行列とする。このような変換をすることにより残差ノルムの収束を向上させることがで

きる。

まず最初に、前処理行列  $M$  を  $M \approx A^{-1}$  の形で近似するために疎行列  $M$  の最小 2 乗問題を考える。

$$\begin{aligned} \min \|AM - E\|_F^2 \\ = \sum_{k=1}^n \min \|AM_k - e_k\|_2^2 \end{aligned} \quad (4)$$

ただし、 $E$  は単位行列で、列ベクトル  $M_k$ ,  $e_k$  は、行列  $M, E$  における  $k$  番目の列ベクトル成分とする。 $M_k$  の非ゼロ要素の数が少なければ、式 (4) は以下の  $n$  個の最小 2 乗問題に帰着できる。

$$\min \|AM_k - e_k\|, \quad k = 1, \dots, n \quad (5)$$

この最小 2 乗問題を効率良く、そしてある程度の精度を保って解く必要がある。そこで  $A$  の疎構造を最大限に利用するためにインデックス集合を導入する。まず、 $M_k$  の非ゼロ要素に対応した行インデックス集合  $J$  を定義する。また  $J$  と同様の行番号をもつインデックス集合  $I$  を定義する。これにより式 (5) を

$$\begin{aligned} \min \|A(I, J)M_k(J) - e_k(I)\| \\ = \min \|\hat{A}\hat{M}_k - \hat{e}_k\|, \quad k = 1, \dots, n \end{aligned} \quad (6)$$

と最小化する。この最小化問題は

- (i) 正規方程式  $\hat{A}^T \hat{A} \hat{M}_k = \hat{A}^T \hat{e}_k$  の形に変形し CG 法を利用する
- (ii) Givens 法を用いて QR 分解する
- (iii) Householder 法を用いて QR 分解する
- (iv) Gram-Schmidt 法を用いて QR 分解する

というどれか 1 つの方法で解くことができる。以上のような手順で、精度を保った近似逆行列をある程度低いコストで計算することができる。本稿の数値実験では Gram-Schmidt 法を用いた QR 分解を使用し、 $M_k$  の非ゼロ要素のインデックスの選び方は、行列  $A$  の各列に対する非ゼロ要素のインデックスに対応するように選んだ。なお、この方法で近似逆行列を求める場合は独立に列ごとに求めているので、並列計算機上に実装するときは、近似逆行列の各列を求める計算を並列に各プロセッサに割り当てるだけでよく、並列化は容易になる。

## 4. 数値実験

### 4.1 プログラムの並列化

本稿で述べた GMRES( $k$ ) 法, DQGMRES( $m, k$ ) 法, そして近似逆行列を前処理行列として用いた GMRES( $k$ ) 法, DQGMRES( $m, k$ ) 法を並列計算機 Origin 2000 上で実装し数値実験を行なった。並列化した部分は、ベクトルどうしの和、ベクトルのスカラー倍、ベクトルの内積、行列とベクトルの積、前処理行列を求める計算全体とした。

問題となる行列の次元を  $d$ 、使用するプロセッサの数を  $p$  とする。 $d$  はロードアンバランスを少なくする

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

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

ために  $p$  の倍数になるようにした。これにより並列化によって各プロセッサに割り当てられる要素数は  $n/p$  個となり、各プロセッサにかかる処理はほぼ均等になる。ベクトルどうしの和、ベクトルのスカラー倍、行列とベクトルの積、前処理行列を求める計算全体はそれぞれのプロセッサが通信をすることなくそれぞれ計算を進めることができる。ベクトルの内積においては各プロセッサが局所的な内積を計算し、全てのプロセッサにおいて処理が終了した段階でそれらの和をとることで計算をした。

### 4.2 Origin 2000 への実装

並列計算機 Origin 2000 上でセルの台数を 8 個として数値実験を行なった。DQGMRES( $m, k$ ) 法のリスタート周期を 10, 20, 40 と設定し、保存する正規直交基底ベクトルの数を 5 と固定して、リスタート周期と保存する正規直交基底ベクトルの数の比を変えて、近似逆行列による前処理を用いて実験を行なった。また、比較対象として GMRES( $k$ ) 法のリスタート周期を 10, 20, 40 と設定して、近似逆行列による前処理を使用した。近似逆行列においては、式 (6) を求めるのに Gram-Schmidt 直交化法を用いた QR 分解を使用し、 $M_k$  のインデックス集合  $J$  は行列  $A$  の各列の非ゼロ要素に対応するように選んだ。計算時間の計測は、各算法とも 3 回行ない、その平均値を取った。また数値実験は以下の環境において行なった。

- 収束条件:  $\|r_i\|/\|r_0\| \leq 1.0 \times 10^{-12}$
- 初期近似解:  $x_0 = (0, 0, \dots, 0)^T$
- 計算精度: 倍精度
- 最大反復回数: 10000

### 4.3 数値例 1

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

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

メッシュ幅を  $h = 1/257$  とし、この式を 5 点中心差分近似で離散化した。これにより得られた行列の次元は 65536 である。真の解を  $u(x, y) = 1 + xy$  と設定して右辺を決定して数値実験を行なった。この計算結果を表 2 に示した。この表の算法の部分は T が時間 (秒) を表し、I は反復回数を表している。

まず GMRES(10) 法に対して前処理行列として近似逆行列を適用し、収束状況を観測した。前処理をしたものは全ての場合において計算時間、反復回数とともに減少している。特に  $Dh = 2^4$  の場合では、反復回数は約 1/5、計算時間は約 1/4 に減少している。 $Dh = 2^3$  の場合には反復回数は約 1/4、計算時間は約

表 2 数値例 1 (T:計算時間(秒), I:反復回数)

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

Dh	2 <sup>-4</sup>		2 <sup>-3</sup>		2 <sup>-2</sup>		2 <sup>-1</sup>		2 <sup>0</sup>	
	T	I	T	I	T	I	T	I	T	I
GMRES(10)	529.1	5207	249.5	2304	98.2	918	88.9	881	90.3	919
GMRES(20)	245.6	2275	128.2	1044	110.9	976	127.5	1027	129.6	1090
GMRES(40)	234.1	1423	182.6	1201	206.8	1286	224.3	1385	220.6	1360
AINV-GMRES(10)	135.4	1035	74.1	553	54.3	459	66.0	531	74.1	528
AINV-GMRES(20)	136.7	770	92.0	559	104.6	659	105.5	679	111.6	692
AINV-GMRES(40)	172.1	805	151.7	769	188.5	905	185.1	851	186.4	846
DQGMRES(10,5)	226.0	2673	113.0	1300	80.5	883	102.4	1158	137.9	1488
DQGMRES(20,5)	138.1	1815	156.0	2019	110.8	1658	194.6	2341	149.6	1934
DQGMRES(40,5)	175.3	2063	138.2	1739	287.8	4066	286.2	3975	154.5	2517
AINV-DQGMRES(10,5)	101.5	862	67.6	593	65.1	618	97.7	980	82.5	806
AINV-DQGMRES(20,5)	96.7	915	96.1	899	105.06	1040	185.6	1818	78.7	899
AINV-DQGMRES(40,5)	135.6	1501	189.2	1832	172.0	1680	166.7	1760	152.4	1560

Dh	2 <sup>1</sup>		2 <sup>2</sup>		2 <sup>3</sup>		2 <sup>4</sup>	
	T	I	T	I	T	I	T	I
GMRES(10)	95.3	910	93.0	908	101.1	864	97.8	803
GMRES(20)	125.2	1052	125.8	1057	125.7	1080	114.1	987
GMRES(40)	230.1	1408	221.2	1333	190.7	1238	207.8	1262
AINV-GMRES(10)	66.3	571	63.7	533	53.3	518	52.3	480
AINV-GMRES(20)	120.5	719	118.7	687	107.9	638	97.3	621
AINV-GMRES(40)	184.0	887	190.2	920	167.9	839	189.8	943
DQGMRES(10,5)	207.5	1793	111.9	1321	87.6	1022	65.8	817
DQGMRES(20,5)	198.3	2379	139.1	1651	97.7	1107	79.0	899
DQGMRES(40,5)	192.8	3029	126.7	1970	89.1	1299	86.1	1154
AINV-DQGMRES(10,5)	84.7	747	67.7	665	79.1	701	56.6	517
AINV-DQGMRES(20,5)	75.7	738	91.7	946	81.7	899	73.7	720
AINV-DQGMRES(40,5)	90.7	1071	72.0	946	96.6	1172	163.7	1870

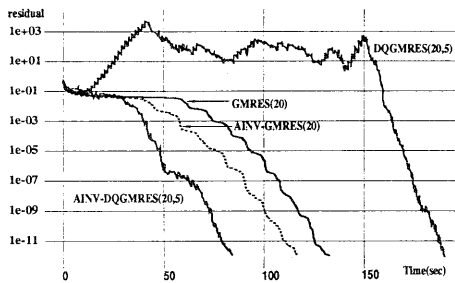


図 4 数値例 1 ( $Dh = 2^1$ ): 計算時間に対する残差ノルム  
Fig. 4 Example 1 ( $Dh = 2^1$ ): The behaviour of residual norm vs. computational time (sec).

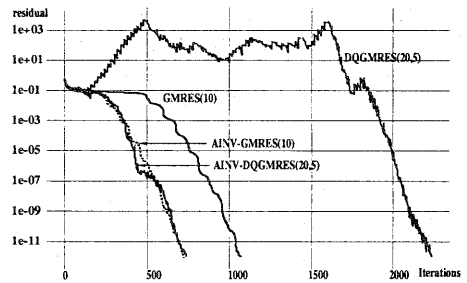


図 5 数値例 1 ( $Dh = 2^1$ ): 反復回数に対する残差ノルム  
Fig. 5 Example 1 ( $Dh = 2^1$ ): The behaviour of residual norm vs. number of iterations.

1/3 に減少しており、残差ノルムの早い収束が得られている。GMRES(20)法、GMRES(40)法に対しても、全ての場合において反復回数、計算時間はともに減少し、前処理の安定した効果が確認できる。

次に DQGMRES(10,5)法に対して同様の前処理を行なって収束状況を観測した。特に  $Dh = 2^0, 2^1$  の場合は、前処理をしない残差ノルムの収束は不規則であり、GMRES(10)法のそれと比べて反復回数、計算時間はともに多くかかっている。しかし、前処理をするこ

とによって収束が安定し、前処理をした GMRES(10)法に近い結果が出ている。それ以外の場合においても前処理をしなかった場合に比べて、反復回数、計算時間はともに減少している。

次に DQGMRES(20,5)法に対して同様の前処理を行なって収束状況を観測した。前処理をしなかった場合は収束状況は全般的に不安定であるが、前処理をすることによって前処理をした GMRES(20)法と同程度、あるいはそれ以下の計算時間で近似解が求まって

表3 数値例2 (T: 計算時間(秒), I: 反復回数)

Table 3 Numerical results of example 2. T: Computation time(sec), I: Number of iteration

Dh	2 <sup>-4</sup>		2 <sup>-3</sup>		2 <sup>-2</sup>		2 <sup>-1</sup>		2 <sup>0</sup>	
	T	I	T	I	T	I	T	I	T	I
GMRES(10)	466.7	1502	271.3	987	171.1	572	101.2	320	94.4	339
GMRES(20)	294.1	895	218.5	678	148.7	439	154.7	470	195.2	575
GMRES(40)	261.4	630	182.8	478	202.0	516	279.8	678	233.8	560
AINV-GMRES(10)	140.2	380	113.7	329	61.8	199	81.3	223	84.9	284
AINV-GMRES(20)	124.8	295	108.9	257	118.2	265	144.3	347	136.7	323
AINV-GMRES(40)	117.9	274	134.1	319	134.4	322	205.0	481	163.1	400
DQGMRES(10,5)	295.7	1226	221.6	808	125.0	457	70.8	320	90.3	353
DQGMRES(20,5)	189.8	855	132.1	599	119.8	467	120.9	504	190.1	799
DQGMRES(40,5)	98.6	617	126.3	717	138.2	748	140.2	840	183.9	998
AINV-DQGMRES(10,5)	140.7	382	72.7	310	61.9	252	101.9	324	86.0	266
AINV-DQGMRES(20,5)	92.6	325	99.5	366	85.0	320	112.7	457	88.5	339
AINV-DQGMRES(40,5)	62.9	398	123.7	666	129.1	671	207.6	1103	90.9	474

Dh	2 <sup>1</sup>		2 <sup>2</sup>		2 <sup>3</sup>		2 <sup>4</sup>	
	T	I	T	I	T	I	T	I
GMRES(10)	97.6	361	97.8	332	99.1	324	127.6	383
GMRES(20)	177.0	524	152.8	496	154.4	451	158.7	502
GMRES(40)	263.9	631	236.6	561	207.1	529	269.5	637
AINV-GMRES(10)	82.2	261	75.6	252	81.5	218	77.58	215
AINV-GMRES(20)	129.3	306	138.2	337	118.2	284	119.2	288
AINV-GMRES(40)	156.2	399	152.9	362	143.4	358	136.5	320
DQGMRES(10,5)	71.1	322	97.6	340	76.8	312	102.6	381
DQGMRES(20,5)	103.1	468	101.2	466	91.1	434	95.2	431
DQGMRES(40,5)	190.8	1037	126.2	752	110.9	677	118.7	691
AINV-DQGMRES(10,5)	65.7	233	69.9	241	63.7	217	72.2	230
AINV-DQGMRES(20,5)	99.0	427	78.2	284	90.6	371	94.4	399
AINV-DQGMRES(40,5)	131.2	708	242.6	1236	95.1	516	148.9	797

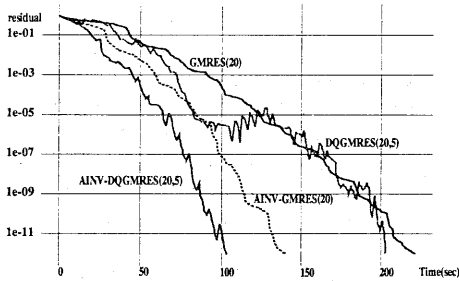


図6 数値例2 ( $Dh = 2^0$ ): 計算時間に対する残差ノルム  
Fig. 6 Example 2 ( $Dh = 2^0$ ): The behaviour of residual norm vs. computational time (sec).

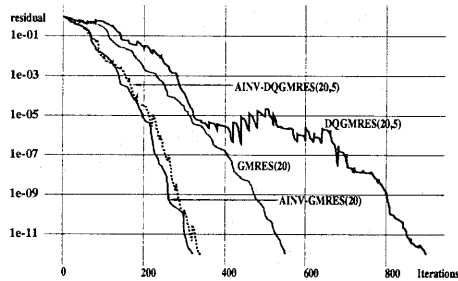


図7 数値例2 ( $Dh = 2^0$ ): 反復回数に対する残差ノルム  
Fig. 7 Example 2 ( $Dh = 2^0$ ): The behaviour of residual norm vs. number of iterations.

いる。特に、 $Dh = 2^1$ の場合は計算時間、反復回数とともに大きく減少し、安定した収束を見せている。この残差ノルムの収束履歴を図4と図5に示した。前処理をすることにより反復回数、計算時間の減少とともに、残差ノルムの収束も安定していることがわかる。

最後にDQGMRES(40,5)法に対して同様の前処理を行なって収束状況を観測した。多くの場合において前処理を行なうことで計算時間、反復回数はともに減少したが、 $Dh = 2^{-3}, 2^4$ の場合においては、収束は全

く改善されておらず、計算時間、反復回数ともに前処理をしない場合よりも余計にかかってしまった。また $Dh = 2^3$ の場合は反復回数は減少したものの計算時間は増加している。このケースでは、 $m$ に対して $k$ は小さくとってあり、不完全アーノルディ過程で求められる正規直交ベクトルの精度が非常に悪くなっていくため、残差ノルムの収束が不安定になっていると思われる。このように、残差ノルムの収束が不安定であるような状況では、前処理が全く効果を発揮していない

場合があることがわかる。

#### 4.4 数値例 2

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

$$-u_{xx} - u_{yy} - u_{zz} + Du_x = G(x, y, z), \\ u(x, y)|_{\partial\Omega} = 0.0$$

メッシュ幅を  $h = 1/65$  として、この式を 7 点中心差分近似で離散化した。得られた行列の次元は 262144 である。真の解を

$u(x, y, z) = \exp(xyz) \sin(\pi x) \sin(\pi y) \sin(\pi z)$  と設定して右辺を決定して数値実験を行なった。この結果を表 3 に示した。

まず GMRES(10) 法に対して、前処理行列として近似逆行列を適用し、収束状況を観測した。数値例 1 の時とはほぼ同様に、近似逆行列による前処理をすることにより、全ての場合において計算時間、反復回数はともに減少し、特に  $Dh = 2^{-4}$  の場合には、前処理をすることで計算時間は半分以下に減少し、反復回数は  $1/4$  以下に減少した。GMRES(20) 法、GMRES(40) 法の場合も全ての場合においても前処理をすることで、計算時間、反復回数はともに減少し、近似逆行列により残差ノルムの収束向上していることがわかる。

次に DQGMRES(10,5) 法に対して、同様の前処理を行なって収束状況を観測した。 $Dh = 2^{-1}$  の場合においては前処理をしても収束性は改善されておらず、より計算時間がかかってしまっているが、それ以外の場合においては計算時間、反復回数はともに減少しており、前処理をした GMRES(10) 法と比べて同程度、あるいはそれよりも短い計算時間で近似解が求まっている。

次に DQGMRES(20,5) 法に対して、同様の前処理を行なって収束状況を観測した。前処理をすることで、全ての場合において反復回数、計算時間はともに減少した。特に  $Dh = 2^{-4}, 2^0$  の場合は、前処理をすることで計算時間、反復回数はともに半分以下に減少している。前処理をした GMRES(20) 法と比較すると、全ての場合において計算時間はより短くなっていて、残差ノルムの収束は安定している。この残差ノルムの収束履歴を図 6 と図 7 に示した。前処理をすることにより反復回数、計算時間の減少とともに、残差ノルムの収束も安定していることがわかる。

最後に DQGMRES(40,5) 法に対して、同様の前処理を行なって収束状況を観測した。多くの場合は、前処理をすることによって反復回数、計算時間はともに減少したが、 $Dh = 2^{-1}, 2^2, 2^4$  の場合は計算時間、反復回数はともに増加しており、収束の改善は見られなかった。これは数値例 1 の DQGMRES(40,5) 法の場合と同様に、正規直交基底ベクトルの精度が悪いために残差ノルムの収束が極めて不安定になり、前処理を行なっても収束は全く改善されていないものと思われる。

## 5. おわりに

並列計算機 Origin 2000 を用いて、GMRES( $k$ ) 法と DQGMRES( $m, k$ ) 法に対して近似逆行列による前処理を適用し、数値実験により比較検討を行なった。まず GMRES( $k$ ) 法に前処理を適用した場合、全ての場合において収束が改善され、計算時間と反復回数は大きく減少した。DQGMRES( $m, k$ ) 法に近似逆行列による前処理を適用した場合、多くの場合に有効であり、前処理をした GMRES( $k$ ) 法と同程度あるいはそれ以下の計算時間で近似解を求めることができた。図 4 と図 5 に示した場合では GMRES( $k$ ) 法を大きく下回る計算時間で近似解を求めることができた。しかしながら、DQGMRES(40,5) 法のような  $m$  に対して  $k$  が極度に小さい場合には、残差ノルムの収束が極めて不安定になり、前処理を適用しても残差ノルムの収束は改善されず、前処理をしない場合よりも計算時間が余計にかかるような場合もあった。これは不完全アーノルディ過程で求められる正規直交基底ベクトルの精度が上がらないことが原因であると思われる。このようなケースでは  $k$  の値を大きくして、正規直交基底ベクトルの計算精度をあげる必要がある。

リスタート周期  $m$  に対して、それに適当な  $k$  を設定した DQGMRES( $m, k$ ) 法においては、前処理を適用することで残差ノルムの収束が安定し、計算時間を大幅に減少することができた。

## 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) Y. Saad and K. Wu: DQGMRES: A Direct Quasi-minimal Residual Algorithm Based on Incomplete Orthogonalization. *Numerical Linear Algebra with Applications*, Vol.3, No. 4, pp. 329-343 (1996).
- 3) T. F. Chan, E. Chow, Y. Saad, and M. C. Yeung: Preserving Symmetry in Preconditioned Krylov Subspace Methods. *SIAM J. Sci. Comput.*, Vol. 20, No. 2, pp. 568-581 (1998).
- 4) T. K. Huckle: Efficient Computation of Sparse Approximate Inverses. *Numerical Linear Algebra with Applications*, Vol.5, pp. 57-71 (1998).
- 5) 野寺, 野口: AP1000 における BiCGStab( $l$ ) 法の有効性について. 情報処理学会論文誌, Vol.38, No.11, pp. 2089-2101 (1997).
- 6) R. Weiss: Parameter-Free Iterative Linear Solvers, Akademie Verlag, 1996.