

自動リスタート過程を持つ GMRES(m) 法の性能評価

羽部 充[†] 森屋 健太郎[†] 野寺 隆^{††}

クリロフ部分空間法の1つである GMRES(m) 法は、リスタート周期 m の値を問題ごとに選択しなければならない。リスタート周期の値は残差ノルムの収束に影響を及ぼすため、適切な m を選択することが必要となる。本稿では、自動的にリスタートする機能を持つ A-GMRES($m, itmax$) 法と GMRES($\leq m_{max}$) 法を並列計算機 SGI Origin 2000 による数値実験によって比較し、それぞれの算法の性能や問題点を検証する。

Verification of the GMRES(m) method with automatic restart process

MITSURU HABU,[†] KENTARO MORIYA [†] and TAKASHI NODERA^{††}

The GMRES(m) method, which is one of the Krylov subspace method, must be selected the restart cycle m for each problems. It is important to choose the suitable value of m , because it has great impacts on numerical results. In this paper, we will verify performances of the A-GMRES($m, itmax$) method and the GMRES($\leq m_{max}$) method, which have automatic restart process, through numerical experiments.

1. はじめに

楕円型の偏微分方程式の境界値問題を数値的に解くことは、有限要素法や有限差分法で離散化された係数行列を持つ連立1次方程式

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

を解くことに帰着されることが多い。ただし、係数行列 A は大型で疎な正則行列とする。 A が対称行列の場合には、CG 法¹⁾ が有効である。しかし、 A が非対称行列である場合は GCG 法関連の算法^{2), 3), 6)} や GMRES 法⁴⁾ など様々あるが、それぞれの算法の有効性は解くべき問題に依存している。GMRES 法は、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を持っているためよく利用される算法である。一般には、正規直交ベクトル数を m 本に制限し、記憶容量と1回の反復にかかる計算を抑えた GMRES 法のリスタート版である GMRES(m) 法⁴⁾ が使われることが多い。このリスタート版である GMRES(m) 法の m の値は問題によって決定しなければならない。リスタート周期 m の値を小さくすると計算コストは低くて済むが、残差ノルムがゼロに収束しない場合もある。また、極端に m の値を大きく取ると残差ノル

ムがゼロに収束するまでに膨大な計算時間を必要とすることがある。そこで、適切なリスタート周期の決定法が必要になる。津野¹⁰⁾ は、適切なリスタート周期の決定法の一つとして GMRES($\leq m_{max}$) 法を提案した。この算法は、残差多項式のゼロ点分布の情報をもとにリスタート周期を決定する算法である。GMRES($\leq m_{max}$) 法は、GMRES(m) 法の計算に加えて残差多項式のゼロ点を計算する必要があるが、リスタートのタイミングを早めることによって不必要な正規直交ベクトルの生成を行わないため、GMRES(m) 法と比較して良い結果を示している。また、近年我々は、Sosonkina⁵⁾ の Adaptive GMRES(m) 法に基づいて Adaptive GMRES($m, itmax$) 法を提案した。この算法は、GMRES($\leq m_{max}$) 法よりも低い計算コストでリスタートのタイミングを決定できる利点を持っている。本稿では、自動リスタート型の算法である Adaptive GMRES($m, itmax$) 法と GMRES($\leq m_{max}$) 法を数値実験によって比較評価し、その性能を検証する。

第2節では、GMRES 法と GMRES(m) 法について述べる。次に、第3節では、Adaptive GMRES($m, itmax$) 法と GMRES($\leq m_{max}$) 法について述べる。第4節では、前処理行列の構成法について述べる。最後に第5節において、数値実験により各算法を比較検証し、第6節で結論を述べる。

2. GMRES 法と GMRES(m) 法について

この節では GMRES 法と GMRES(m) 法の特性に

[†] 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

ついて簡単に述べる。

2.1 GMRES 法

1986年にSaadら⁴⁾によって提案されたGMRES法は、クリロフ部分空間反復解法の1つである。初期近似解を x_0 、初期残差ベクトルを $r_0 = b - Ax_0$ とすると、 i 回目の反復における残差ノルムは次式で決定できる。

$$\min_{z \in K_i} \|b - A(x_0 + z)\|_2 = \min_{z \in K_i} \|r_0 - Az\|_2 \quad (2)$$

ただし、 z はクリロフ部分空間 K_i の正規直交系 $\{v_1, v_2, \dots, v_i\} = V_i$ の線形結合として $z = V_i y$ と表される。最終的に式(2)は、以下の最小2乗問題に帰着できる。

$$\begin{aligned} J(y) &= \|\beta v_1 - AV_i y\|_2 \\ &= \|V_{i+1}(\beta e_1 - \bar{H}_i y)\|_2 \\ &= \|\beta e_1 - \bar{H}_i y\|_2 \end{aligned} \quad (3)$$

ただし、 $\beta = \|r_0\|_2$ とする。また、式(3)は \bar{H}_i をギブンス回転行列を用いて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、 i 回目の反復におけるGMRES法の近似解 x_i は、 $x_i = x_0 + V_i y$ で求められる。

2.2 GMRES(m)法

GMRES法は反復回数が増えるにつれて記憶容量と計算量が増加していくため、実際には、 m 回の反復ごとにその時点での近似解 x_m を初期近似解 x_0 として再出発するGMRES(m)法⁴⁾が使われている。通常、 m は一定の値であるが、この m の値を動的に変化させる方法を第3節で述べる。

3. リスタート周期の増減過程を持つ GMRES(m)法

GMRES(m)法のリスタート周期 m は問題によって決定しなければならないパラメータであり、適切な選択は経験に委ねられている。 m が小さい場合には残差ノルムの縮小が停滞してしまったり、全く縮小しない事がある。また、 m の値が大きすぎると1回の反復にかかる計算時間が増加するために、問題によっては小さなリスタート周期の方が速く収束する場合がある。この節では、リスタートのタイミングを自動的に決定する算法であるA-GMRES(m)法とGMRES($\leq m_{max}$)法について述べる。

3.1 Adaptive GMRES($m, itmax$)法

Adaptive GMRES($m, itmax$)法(これ以降A-GMRES($m, itmax$)法)はAdaptive GMRES(m)法を基本算法としており、次の式によってリスタートのタイミングを決定する。

$$test = m \times \frac{\log [tol / \|r^{new}\|_2]}{\log [\|r^{new}\|_2 / (1.0 + 10u) \|r^{old}\|_2]} \quad (4)$$

ただし、 r^{new} はその時点での残差ベクトル、 r^{old} は一つ前のリスタート時の残差ベクトル、 tol は収束判定条件、 u は計算機イプシロンの値である。A-GMRES($m, itmax$)法は1反復ごとに式(4)を参照して収束までのおおよその反復回数 $test$ を見積もり、その値によってリスタートするかどうかを決定する算法である。1回の反復にかかる計算時間を少なくするために、 $test$ の値が小さい場合にはその時点でリスタートを行う。 $test$ の値が大きい場合には、より多くの正規直交ベクトルを生成して十分に残差が減少するまでリスタートを行わない。このようにA-GMRES($m, itmax$)法は、残差の縮小量によってリスタートのタイミングを決定するため、局所的な収束の変化に対応できることがある。A-GMRES($m, itmax$)法の算法を図1に示す。

3.2 GMRES($\leq m_{max}$)法

津野ら¹⁰⁾によって提案されたGMRES($\leq m_{max}$)法は、残差多項式のゼロ点の分布状況からリスタートのタイミングを判断する算法である。残差ノルムは、残差多項式のゼロ点分布が係数行列 A の固有値分布に近いほどより大きく縮小する。リスタートを行うと、その時点での残差多項式のゼロ点は以降の反復における残差多項式のゼロ点の一部として固定されるため、残差多項式のゼロ点分布が A の固有値分布と一致している時にリスタートを行うのが理想的である。しかし、実際には A の固有値分布を得ることは困難であるため、 A の固有値分布に対する残差多項式のゼロ点分布の近似度を確かめることはできない。そこでGMRES($\leq m_{max}$)法では、残差多項式のゼロ点を複素平面上でなるべく集中させないことで、残差多項式のゼロ点分布が未知である A の固有値分布に重なる可能性を高めるようにリスタートを行っている。ここで、GMRES($\leq m_{max}$)法の k 回目の反復で構成される k 次の残差多項式 $P_k(t)$ は次のように表すことができる。

$$P_k(t) = P_{k'}^{GMRES}(t) \prod_{j=1}^i P_{m_j}^{(j)}(t),$$

$$k = \sum_{j=1}^i m_j + k'$$

ただし、 $P_{k'}^{GMRES}(t)$ は直前のリスタート以後に構成された k' 次多項式で、 $P_{m_j}^{(j)}(t)$ は直前のリスタートより前に構成された m_j 次多項式である。 $P_{m_j}^{(j)}(t)$ はリスタートによって固定されたゼロ点を解として持つ多項式であるため、 $P_{k'}^{GMRES}(t)$ の解であるゼロ点の分布が固定されたゼロ点の分布に重ならないときにリスタートをすることがGMRES($\leq m_{max}$)法の目標である。すなわち、新しいゼロ点 $\lambda_j^{(k)}$ すべてが、それぞれを中心とする長方形領域

```

choose  $x_0$ ,
 $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
 $m_1 := 1$ ;  $m_2 := m$ ;
start
for  $i := m_1$  to  $m_2$  do
begin
 $\hat{v} := Av_i$ ;
for  $j := 1$  to  $i$  do
begin
 $h_{j,i} := \hat{v}^T v_j$ ;
 $\hat{v} := \hat{v} - h_{j,i} v_j$ ;
end
 $h_{i+1,i} := \|\hat{v}\|$ ;
 $v_{i+1} := \hat{v}/h_{i+1,i}$ ;
compute  $y_i = \min_y \|\beta e_1 - \bar{H}_i y\|$ ;
if  $\|b - Ax_i\| \leq tol$  then
stop iteration
endif
end
test :=  $m_2 \times \frac{\log [tol/\|r^{new}\|]}{\log [\|r^{new}\|/(1.0+10U)\|r^{old}\|]}$ ;
if  $m_2 \leq mmax - p$  and
test  $\geq smv \times (itmax - itno)$  then
 $m_1 := m_2 + 1$ ;
 $m_2 := m_2 + p$ ;
goto start
endif
 $x_0 := x_{m_2}$ ;  $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
 $m_1 := 1$ ;  $m_2 := m$ 
goto start

```

図1 A-GMRES($m, itmax$)法

$$\Gamma(\lambda_j^{(k')}) := \left\{ z \in \mathbb{C} \left| \begin{array}{l} \Re(\lambda_j^{(k')} - z) < \frac{Mre}{2(k-1)}, \\ \Im(\lambda_j^{(k')} - z) < \frac{Mim}{2(k-1)} \end{array} \right. \right\}$$

に固定されたゼロ点を含まないときにリスタートを行う。ただし、Mre、Mimはすべてのゼロ点がつくる実部と虚部の分布の幅である。実装に際して、残差多項式のゼロ点は残差多項式の係数からコンパニオン行列を生成し、QR法を用いてコンパニオン行列の固有値を求めることで計算されている。また、残差多項式のゼロ点は複素平面上で実軸に関して対称に分布しているため、ゼロ点分布は実軸に関して片側のみで比較している。まとめると、GMRES($\leq m_{max}$)法は図2のようになる。

4. 前処理

反復解法において、近似解を効率良く求めるために

```

choose  $x_0$ ,
 $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
start
for  $k' := 1$  to  $m_{max}$  do
begin
 $k = k + 1$ ;
 $\hat{v} := Av_{k'}$ ;
for  $i := 1$  to  $k'$  do
begin
 $h_{i,k'} := \hat{v}^T v_i$ ;
 $\hat{v} := \hat{v} - h_{i,k'} v_i$ ;
end
 $h_{k'+1,k'} := \|\hat{v}\|$ ;
 $v_{k'+1} := \hat{v}/h_{k'+1,k'}$ ;
compute  $y_{k'} = \min_y \|\beta e_1 - \bar{H}_{k'} y\|$ ;
if  $\|b - Ax_{k'}\| \leq tol$  then
stop iteration
endif
if  $k' \bmod 2 = 0$  then
残差多項式の係数を求める。
残差多項式のゼロ点を計算する。
求めたゼロ点の実部、虚部の最大、最小値を
それぞれ求める。
MreとMimを求める。
if  $k = 2$  then
goto restart
endif
領域 $\Gamma$ の中に固定されたゼロ点が存在しないとき
リスタートを行う。
endif
end
restart
求めたゼロ点を固定されたゼロ点として保存する。
 $x_0 := x_{k'}$ ;  $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
goto start

```

図2 GMRES($\leq m_{max}$)法

前処理行列が用いられることがある。係数行列に前処理を施すことにより、1回の反復における演算量は増えるが、その分収束までの反復回数を減らすことができる場合がある。本稿で述べる数値実験では行列の前処理を行なうものとし、ここではHuckle¹⁰⁾の提案した近似逆行列による右前処理行列の構成法について述べる。右前処理とは、以下のように係数行列の右側から前処理行列Mを掛けることである。

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

前処理行列Mを A^{-1} に近似するために、次のような最小2乗問題を考える。

$$\min \|AM - E\|_F^2 = \sum_{k=1}^n \min \|AM_k - e_k\|_2^2 \quad (5)$$

ただし、Eは単位行列、 M_k 、 e_k は行列M、Eのk番目の列ベクトルの成分とする。ここでベクトル M_k が疎であると仮定すると、式(5)は以下のn本の最小2

表 1 Origin 2000 の仕様

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

乗問題の式に帰着できる。

$$\min \|AM_k - e_k\|_2^2, \quad k = 1, \dots, n \quad (6)$$

ここで M_k の非ゼロ要素に関する行インデクス集合 J を定義する。また、 J と同じ行番号を持つインデクス集合 I を定義する。このインデクス集合を用いて、式 (6) は次のように最小化できる。

$$\begin{aligned} \min \|A(I, J)M_k(J) - e_k(I)\|_2^2 \\ = \min \|\hat{A}\hat{M}_k - \hat{e}_k\|_2^2, \quad k = 1, \dots, n \quad (7) \end{aligned}$$

式 (7) を解く算法は様々あるが、本稿の数値実験では Givens 法における QR 分解を用いた。また、 M_k の非ゼロ要素のインデクスは、係数行列 A の非ゼロ要素と同じインデクス集合を持つものとして M を決定している。

5. 数値実験

この節では、GMRES(m) 法、A-GMRES($m, itmax$) 法、GMRES($\leq m_{max}$) 法の数値実験による比較を行う。

5.1 Origin 2000 での実装

本稿で述べた算法を並列計算機 Origin 2000 上で実装し実行した。並列化した部分は、(1)ベクトルどうしの和、(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、各プロセッサに割り当てられるベクトルの要素数は $\tilde{n} = n/p$ となり、 i 番目のプロセッサはベクトルの $\tilde{n}(i-1)+1$ 行目から $\tilde{n}i$ 行目まで割り当てられる。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は通信することなく各プロセッサが、割り当てられたベクトルの要素に対して計算を進めていくことができる。ベクトルの内積については、まず各プロセッサが局所的な \tilde{n} 個の内積を計算し、その後でそれら p 個の値の和をとって計算する。

5.2 数値実験の詳細

数値実験は SGI の並列計算機 Origin 2000 によって行った。その際セルの台数は 8 個とした。

GMRES(m) 法のリスタート周期と GMRES($\leq m_{max}$) 法の最大リスタート周期 m_{max} は 10, 20, 40 の 3 通り、A-GMRES($m, itmax$) 法は初期のリスタート周期を 4 とし、最大のリスタート周期を 100 とした。また、最大反復回数については、A-GMRES($m, itmax$) 法は、前処理なしの場合には $30\sqrt{N}$ 回、前処理付の場合には $10\sqrt{N}$ 回と $30\sqrt{N}$ 回とした。ただし、 N は行列の次元である。GMRES(m) 法と GMRES($\leq m_{max}$) 法の最大反復回数は 10000 回とした。

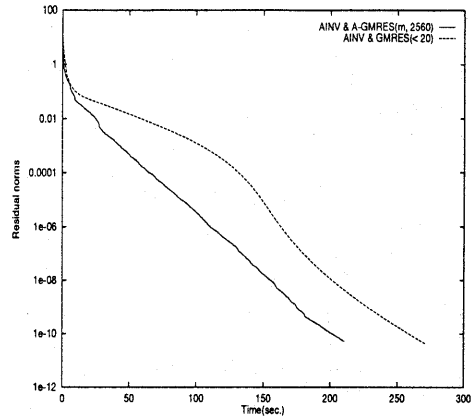


図 3 $Dh = 2^{-6}$: 計算時間に対する残差ノルム

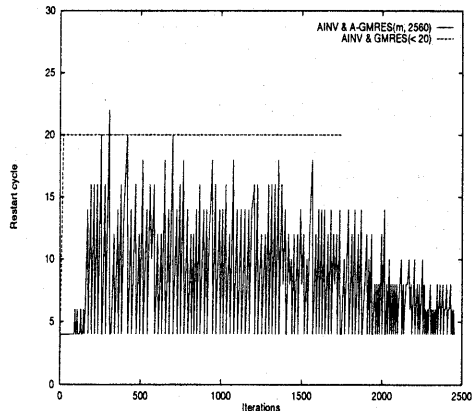


図 4 $Dh = 2^{-6}$: 反復回数に対するリスタート周期

反復回数の数え方は、1 クリロフ部分空間進む毎に 1 回とした。なお、計算時間の計測は、各算法とも 3 回行なって、その平均値を取った。

数値実験は以下の条件のもとで行った。

- 収束判定条件: $\|r_i\|/\|b\| \leq 1.0 \times 10^{-12}$
- 初期近似解: $x_0 = (0, 0, \dots, 0)^T$
- 計算精度: 倍精度
- 計算機イプシロン: $u = 1.0 \times 10^{-16}$

5.3 数値例

矩形領域 $\Omega = [0, 1] \times [0, 1]$ における 2 階の偏微分方程式の Dirichlet 境界値問題を考える (Joubert⁷) を参照)。

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

ただし、メッシュ幅を $h = 1/257$ として 5 点中心差分近似で離散化し、真の解を $u(x, y) = 1 + xy$ と設定して右辺を決定する。なお、行列の次元は 65536 となる。

表 2 数値例の結果 (その 1)

Algorithm	Dh					
	2 ⁻⁶		2 ⁻⁵		2 ⁻⁴	
	sec.	iter.	sec.	iter.	sec.	iter.
GMRES(10)	—	—	—	—	292.0	4199
GMRES(20)	—	—	457.9	4158	229.3	2029
GMRES(40)	597.6	2973	410.8	2163	283.4	1309
A-GMRES(m, 7680)	609.1	7564	472.0	7321	392.4	6739
GMRES(≤ 10)	—	—	—	—	454.7	5760
GMRES(≤ 20)	—	—	591.5	5023	301.3	2312
GMRES(≤ 40)	680.6	2762	539.2	2497	382.1	1719
AINV & GMRES(10)	382.1	3988	218.7	2426	102.7	1078
AINV & GMRES(20)	206.0	1527	159.0	1151	105.2	722
AINV & GMRES(40)	330.2	1351	237.1	925	184.5	771
AINV & A-GMRES(m, 2560)	201.0	2466	169.0	2360	136.1	1647
AINV & A-GMRES(m, 7680)	336.4	5678	299.8	4922	161.0	2735
AINV & GMRES(≤ 10)	408.4	4279	232.1	2538	110.4	1127
AINV & GMRES(≤ 20)	271.6	1742	153.1	988	127.3	823
AINV & GMRES(≤ 40)	360.5	1247	255.6	901	235.7	802

表 3 数値例の結果 (その 2)

Algorithm	Dh					
	2 ⁻³		2 ⁻²		2 ⁻¹	
	sec.	iter.	sec.	iter.	sec.	iter.
GMRES(10)	163.3	2148	70.3	912	59.8	863
GMRES(20)	140.5	1260	119.0	1020	121.1	1023
GMRES(40)	221.2	1149	291.8	1320	263.5	1280
A-GMRES(m, 7680)	220.8	3925	125.5	1597	165.8	1191
GMRES(≤ 10)	188.9	2321	66.4	925	63.4	888
GMRES(≤ 20)	138.6	1066	119.6	1020	120.3	1060
GMRES(≤ 40)	243.4	1028	278.0	1198	324.6	1350
AINV & GMRES(10)	51.4	546	46.5	500	48.9	528
AINV & GMRES(20)	79.7	576	85.8	580	99.0	716
AINV & GMRES(40)	161.7	755	196.2	955	250.9	993
AINV & A-GMRES(m, 2560)	200.3	1003	138.1	730	137.8	639
AINV & A-GMRES(m, 7680)	49.5	833	36.3	596	31.1	467
AINV & GMRES(≤ 10)	53.2	557	49.0	522	51.7	539
AINV & GMRES(≤ 20)	93.0	601	100.9	654	99.2	639
AINV & GMRES(≤ 40)	208.7	718	285.7	952	235.8	872

Dh の値を 2⁻⁶ から 2⁻¹ までの 6 通りで行ない、その結果を表 2 と表 3 に示した。これらの表で AINV と表記されているものは、係数行列 A の近似逆行列による前処理を行なったものである。Dh が 2⁻⁶、2⁻² における残差ノルムの収束の様子を図 3 と図 5 に示した。また、Dh が 2⁻⁶、2⁻² のときの A-GMRES(m, itmax) 法と GMRES(≤ m_{max}) 法のリスタート周期の値の変化を図 4 と図 6 に示した。

A-GMRES(m, itmax) 法は前処理なしの場合には

すべての Dh で収束しているが、残差ノルムの収束までに反復回数が多くかかってしまい、計算時間において最速で収束したものはなかった。しかし、前処理付きの A-GMRES(m, itmax) 法である AINV & A-GMRES(m, itmax) 法は、Dh = 2⁻⁶, 2⁻³, 2⁻², 2⁻¹ において最も速く収束している。図 5 から分かるように、小さなリスタート周期で十分収束に達するような問題でも反復の局面によっては残差ノルムの収束が停滞に近い状況があるが、AINV & A-GMRES(m, itmax)

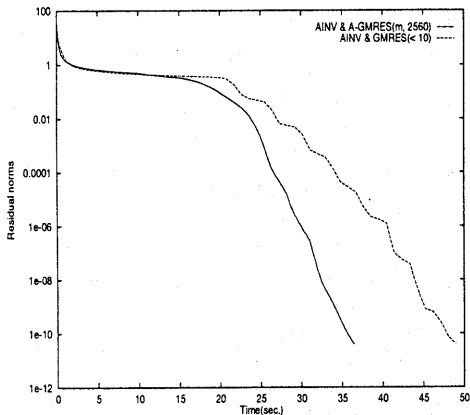


図5 $Dh = 2^{-2}$: 計算時間に対する残差ノルム

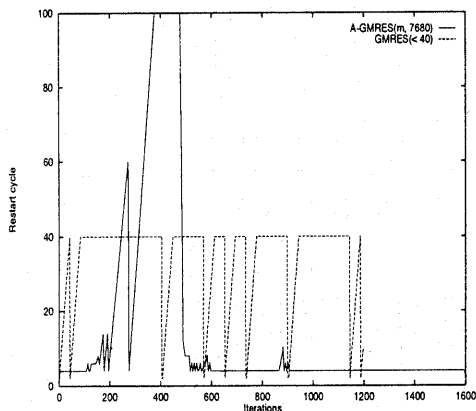


図6 $Dh = 2^{-2}$: 反復回数に対するリスタート周期

法は、それぞれの局面において適切なリスタート周期を選べたために良い結果が得られたと考えられる。一方、 $GMRES(\leq m_{max})$ 法は、全体として $GMRES(m)$ 法とほぼ同じか、もしくはそれ以上に収束までの計算時間がかかっている。図4と図6からどのようなタイミングでリスタートを行ったか分かる。 $Dh = 2^{-2}$ の場合には、反復の途中から残差ノルムが急激に収束しており、 $GMRES(10)$ 法などのリスタート周期が小さな算法が良い結果を示しているが、 $GMRES(\leq 40)$ 法は、ほぼ毎回40回の周期でリスタートしているため計算時間の短縮に至らなかった。

6. まとめ

本稿では、自動的にリスタート周期を決定するA-GMRES($m, itmax$)法と $GMRES(\leq m_{max})$ 法を、並列計算機SGI Origin 2000で実装して得られ

た数値実験の結果によって比較検証を行なった。A-GMRES($m, itmax$)法は、小さなリスタート周期で反復する傾向があるため、問題によっては収束までの反復回数が増加して計算時間がより多くかかってしまっていた。これは、パラメータである $itmax$ の値が関係しているためであると考えられる。また、 $GMRES(\leq m_{max})$ 法のリスタートのタイミングは、小さなリスタート周期が良い結果を示すような問題でも m_{max} 回の反復で行うことが多かった。これは、固定されたゼロ点と新しいゼロ点の分布比較の条件を満たすことが少なかったためである。我々の今後の課題として、数値実験を通して得られた上記の問題点を解決することが挙げられる。

参考文献

- 1) Hestenes, M. R. and Stiefel, E.: Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards*, Vol. 49, pp. 409-435 (1952).
- 2) Lanczos, C.: Solution of systems of linear equations by minimize iteration, *J. Res. Nat. Bur. Standard*, Vol. 49, pp. 33-53 (1952).
- 3) Fletcher, R.: Conjugate gradient methods for indefinite systems, *Lecture Notes in Math.*, Vol. 506, pp. 73-89 (1976).
- 4) Saad, Y. and Schultz, M. H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, Vol. 7, pp. 856-869, (1986).
- 5) Sosonkina, M., Watson, L. N., Kapania, R. K. and Walker, H. F.: A new adaptive GMRES algorithm for high accuracy, *Numer. Linear Algebra Appl.*, Vol. 5, pp. 275-297 (1998).
- 6) Gutknecht, M. H.: Variants of BiCGSTAB for matrices with complex spectrum, *SIAM J. Sci. Comput.*, Vol. 14, pp. 1020-1033 (1993).
- 7) Joubert, W.: Lanczos methods for the solution of nonsymmetric systems of linear equations, *SIAM J. Matrix. Anal. Appl.*, Vol. 13, pp. 928-943 (1992).
- 8) 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).
- 9) 津野, 野寺: 早期リスタートによるGMRES(m)法の高速化, *情報処理学会論文誌*, Vol. 40, No. 4, pp. 1760-1773 (1999).
- 10) T. K. Hucl: Efficient Computation of Sparse Approximate Inverses, *Numer. Linear Algebra Appl.*, Vol. 5, pp. 57-71 (1998).
- 11) Homer. F. W.: Implementation of the GMRES method using Householder transformations, *SIAM J. Sci. Comput.*, Vol. 9, pp. 152-163 (1988).