

リスタート周期を動的に変える GMRES(m) 法

羽 部 充[†] 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、リスタート周期 m の選択法の 1 つとして Adaptive GMRES($m, lmax, itmax$) 法を提案する。この算法は、Sosonkina ら¹⁰⁾によって提案された Adaptive GMRES($m, itmax$) 法の改良版であり、動的にリスタート周期の値を増減させることで残差ノルムがゼロに収束するまでの計算時間を改善する。並列計算機 SGI Origin 2000 による数値実験でいくつかの算法と比較することでその有効性を検証する。

GMRES(m) Method with Changing the Restart Cycle Dynamically

MITSURU HABU[†] and TAKASHI NODERA^{††}

The GMRES(m) method, which is one of the Krylov subspace methods, must be given the restart cycle m for each problems previously. Since the value of m affects convergence of a residual norm, it is important to select m suitably. In this paper, we propose the adaptive GMRES($m, lmax, itmax$) method which has a choosing process of m . This strategy is the improved version of adaptive GMRES($m, itmax$) method proposed by Sosonkina *et al.*¹⁰⁾ and it reduces the total computation time by increasing and decreasing the value of restart cycle m . At least the effectiveness of this new strategy is shown by numerical experiments, which is implemented on SGI Origin 2000.

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),7)}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 の値を大きくす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なる。津野¹¹⁾は、適切なリスタート周期の決定法の 1 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、1998 年に Sosonkina ら¹⁰⁾によって提案された Adaptive GMRES($m, itmax$) 法もリスタート周期を自動的に選択する算法である。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。しかし、Adaptive GMRES($m, itmax$) 法は以

[†] 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

前のリスタート周期よりも小さい値ではリスタートをしないため、選択されるリスタート周期は単調増加となる。そのためリスタート周期を大きくし過ぎてしまう傾向があり、問題によっては残差ノルムが0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。そこで本稿では、Adaptive GMRES($m, itmax$)法を基本算法とし、リスタート周期の適応的な選択法について述べる。

2章では、GMRES法とGMRES(m)法について述べる。次に、3章では、Sosonkinaら¹⁰⁾が提案したAdaptive GMRES($m, itmax$)法について述べる。4章では、リスタート周期を適応的に決定するAdaptive GMRES($m, lmax, itmax$)法を提案する。最後に5章において、数値実験により各算法を比較検証し、6章で結論を述べる。

2. GMRES法とGMRES(m)法について

この章ではGMRES法とGMRES(m)法の特性に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、大型で疎な非対称行列を係数行列とする連立1次方程式の解法である。GMRES法は、各反復において残差ノルムを最小化するという性質を持っている。すなわち、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、 y は残差ノルムが最小になるように選ばれる。よって、式(2)は、以下の最小2乗問題に帰着できる。

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

ただし、 $\beta = \|r_0\|_2$ とする。また、式(3)は上ヘッセンベルグ行列 \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法は近似解 x_i を計算するために、正規直

```

choose  $x_0$ ,
 $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
start
for  $i := 1$  to  $m$  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
 $x_0 := x_k$ ;  $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
goto start

```

図1 GMRES(m)法
Fig. 1 GMRES(m) method.

交系 V_i と上ヘッセンベルグ行列 \bar{H}_i を保存しておく必要がある。そのため反復回数が増えるにつれ、記憶容量と計算量が増加していくので、そのままでは実用的ではない。そこで実際には、 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。GMRES(m)法の算法をまとめたものを図1に示す。通常、 m はリスタート周期と呼ばれる。この m の値を動的に変化させる方法を3章で述べる。

2.3 GMRES法の正規直交過程

GMRES法の正規直交ベクトル $v_j \in V_i$ の直交化には、主に修正グラムシュミット法とハウスホルダー法⁵⁾が使われている。算法の特徴として、修正グラムシュミット法は演算量が少ないが、反復が進むにつれて丸め誤差により、正規直交ベクトルどうしの直交性が失われることがある。これに対しハウスホルダー法で直交化を行った場合には、修正グラムシュミット法と比較してより多くの演算を必要とするが、数値的安定性が得られる。しかし、修正グラムシュミット法を用いた場合でも正規直交ベクトルの直交性が失われているにもかかわらず、残差ノルムが収束する様子は八

ウスホルダー法で直交化を行った場合と変わらない⁸⁾。Greenbaum らの最近の研究⁸⁾は、残差ノルムの収束に重要なのは正規直交ベクトルの直交性ではなく、正規直交ベクトルの線形独立性であることを示している。よって、本稿の数値実験では、正規直交基底の生成過程には修正グラムシュミット法を用いている。

3. Adaptive GMRES($m, itmax$) 法

1998 年に Sosonkina ら¹⁰⁾によって提案された Adaptive GMRES($m, itmax$) 法 (A-GMRES($m, itmax$) 法と呼ぶ) は、GMRES(m) 法のリスタート周期を適応的に変化させる方法の 1 つである。すなわち、残差ノルムの減少の割合に応じてリスタート周期を増加させることにより、収束までの計算時間を短縮させる方法である。A-GMRES($m, itmax$) 法は、次式で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周期を増加させるかどうかを決定する算法である。

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

ただし、 r^{new} はその時点での残差ベクトル、 r^{old} は 1 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。10u は停滞によって残差ベクトルのノルムの縮小がごくわずかで、計算精度によっては $\|r^{new}\|_2 = \|r^{old}\|_2$ と見なされる場合に分母が 0 になるのを防ぐための処置である。A-GMRES($m, itmax$) 法は、リスタートをした時点で式 (4) の値を参照し、残差ノルムの減少が不十分で、最大反復回数までに収束できないと判断した場合には、残差ノルムが十分に小さくなるまでリスタート周期を増加させる。また、反復開始時の初期リスタート周期を小さくすることで、反復を繰り返しながら次第にリスタート周期が増加することになる。よって、この処理をすることにより、ある程度問題に応じたリスタート周期を決定することができる。しかし、A-GMRES($m, itmax$) 法はリスタート周期を減らす処置がないために、問題によっては有効でない場合がある。そこで本稿では、A-GMRES($m, itmax$) 法を基本算法とし、リスタート周期を適応的に増減させる方法を提案する。

4. リスタート周期の増減過程を持つ GMRES(m) 法

GMRES(m) 法はリスタート周期が大きいほど 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。A-GMRES($m, itmax$) 法は、リスタート周期を減らす過程がないため、いったんリスタート周期を増加させた場合、収束するまでそれ以上のリスタート周期で反復することになる。しかし、リスタート周期を増加させた後、式 (4) による見積りが極端に小さくなった場合には、小さなリスタート周期の方が速く収束する可能性がある。そこで本稿では Adaptive GMRES($m, l, itmax$) 法 (これ以降 A-GMRES($m, l, itmax$) 法と呼ぶ) を提案する。A-GMRES($m, l, itmax$) 法はリスタート周期を動的に変化させる過程を持つ GMRES(m) 法である。ただし、 l はある整数である。実際にはリスタート周期の増加過程と減少過程は分かれており、リスタート周期 m を増やす過程は A-GMRES($m, itmax$) 法と同じである。リスタート周期 m を減らす過程は l に依存しており、 l 回ごとのリスタート時に初期リスタート周期に戻す処置を行う。すなわち A-GMRES($m, l, itmax$) 法は、通常は A-GMRES($m, itmax$) 法とまったく同じであるが、リスタートの回数が l の倍数になったときに、A-GMRES($m, itmax$) 法のリスタート周期を初期リスタート周期に戻す方法である。増やす過程は A-GMRES($m, itmax$) 法と同じであるので、大きなリスタート周期が必要と判断した場合にはすぐにリスタート周期を増加させることができる。また、式 (4) による見積りが小さい場合には、小さなリスタート周期のまま反復を続けることができる。この l の値を 1 とした場合にはリスタートごとに初期リスタート周期に戻るため、収束の急激な振舞いの変化に最も速く対応できる可能性がある。ここで、A-GMRES($m, l, itmax$) 法の算法を図 2 に示す。

5. 数値実験

本章では、GMRES(m) 法、A-GMRES($m, itmax$) 法、および A-GMRES($m, l, itmax$) 法の数値実験による比較を行うことにする。

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。表 1 は、本数値実験で用いた Origin 2000 の仕様を記述したものである。並列化した部分は、(1) ベクトルどうしの和、(2) ベクトルのスカラ倍、(3) ベクトルの内積、(4) 行列とベクトルの積、の 4 つである。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

```

choose  $x_0$ ,
 $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
 $m_1 := 1$ ;  $m_2 := m$ ;
 $k := 0$ ;
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
 $h_{i+1,i} := \|\hat{v}\|_2$ ;
 $v_{i+1} := \hat{v}/h_{i+1,i}$ ;
compute  $y_i = \min_y \|\beta e_1 - \bar{H}_i y\|_2$ ;
if  $\|b - Ax_i\|_2 \leq tol$  then
stop iteration
endif
end
 $test := m_2 \times \frac{\log[tol/\|r^{new}\|_2]}{\log[\|r^{new}\|_2/(1.0+10u)\|r^{old}\|_2]}$ ;
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
 $x_0 := x_{m_2}$ ;  $r_0 := b - Ax_0$ ;
 $\beta := \|r_0\|$ ;  $v_1 := r_0/\beta$ ;
 $m_1 := 1$ ;  $k := k + 1$ ;
if  $k = l$  then
 $m_2 := m$ ;  $k := 0$ ;
goto start

```

図2 A-GMRES($m, lmax, itmax$) 法
Fig. 2 A-GMRES($m, lmax, itmax$) method.

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 個の値の和をとつ

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

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

て計算する。

5.2 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うものとし、ここでは Huckle⁹⁾ 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 をかけることである。

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

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。

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

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ックス集合 J を定義する。また、 J と同じ行番号を持つインデックス集合 I を定義する。このインデックス集合を用いて、式(precond2)は次のように最小化できる。

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

ただし、 $A(I, J)M_k(J) = \hat{A}\hat{M}_k$ とする。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ックスは、係数行列 A の非ゼロ要素と同じインデックス集合を持つものとして M を決定している。

5.3 数値例

数値実験は SGI の並列計算機 Origin 2000 を用いたが、セルの数を 8 個とした。

GMRES(m) 法のリスタート周期は 10, 20, 40 の3通り、A-GMRES($m, itmax$) 法、A-GMRES($m, l, itmax$) 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、A-

表 2 数値例 1 の結果 (1), 計算時間: time (sec.), 反復回数: iter.
 Table 2 Numerical results of example 1 (Part 1), computation time.: time (sec.), number of iterations.: iter.

算法	Dh					
	2^{-6}		2^{-5}		2^{-4}	
	time	iter.	time	iter.	time	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)	591.2	4573	373.2	3540	207.3	1778
	$m : 4 \rightarrow 26$		$m : 4 \rightarrow 24$		$m : 4 \rightarrow 28$	
A-GMRES(m , 1, 7680)	609.1	7564	472.0	7321	392.4	6739
A-GMRES(m , 2, 7680)	635.2	7644	526.6	7531	337.0	6784
A-GMRES(m , 3, 7680)	676.2	7555	486.8	7376	340.3	6808
A-GMRES(m , 5, 7680)	575.6	7035	455.7	7282	314.5	6442
A-GMRES(m , 10, 7680)	560.8	7008	533.5	7004	388.6	7286
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)	212.1	1836	195.3	1198	216.6	853
	$m : 4 \rightarrow 20$		$m : 4 \rightarrow 24$		$m : 4 \rightarrow 60$	
AINV&A-GMRES(m , 7680)	412.6	5454	336.2	4679	161.8	2080
	$m : 4 \rightarrow 8$		$m : 4 \rightarrow 8$		$m : 4 \rightarrow 8$	
AINV&A-GMRES(m , 1, 2560)	201.0	2466	169.0	2360	136.1	1647
AINV&A-GMRES(m , 2, 2560)	224.1	2538	199.6	2515	167.9	2148
AINV&A-GMRES(m , 3, 2560)	215.2	2453	179.5	2398	173.2	2311
AINV&A-GMRES(m , 5, 2560)	228.3	2464	191.3	2382	251.8	1556
AINV&A-GMRES(m , 10, 2560)	255.6	2440	231.2	2531	261.3	2277
AINV&A-GMRES(m , 1, 7680)	336.4	5678	299.8	4922	161.0	2735
AINV&A-GMRES(m , 2, 7680)	471.4	7009	269.2	4533	137.3	2248
AINV&A-GMRES(m , 3, 7680)	396.8	6632	326.3	5202	170.8	2416
AINV&A-GMRES(m , 5, 7680)	332.7	5365	239.4	3858	184.1	2934
AINV&A-GMRES(m , 10, 7680)	369.3	6159	274.4	4420	171.8	2875

GMRES($m, itmax$) 法, A-GMRES($m, l, itmax$) 法の最大反復回数は, 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) 法の最大反復回数は 10,000 回とした.

反復回数の数え方は, 1 クリロフ部分空間進むごとに 1 回とした. なお, 計算時間の計測は, 各算法とも 3 回行って, その平均値をとった.

各例題の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: $u=1.0 \times 10^{-16}$

5.3.1 数値例 1

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 (Joubert⁶) を参照).

$$\begin{aligned}
 -u_{xx} - u_{yy} + Du_x(x, y) &= G(x, y), \\
 u(x, y)|_{\partial\Omega} &= 1 + xy.
 \end{aligned}
 \tag{8}$$

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. Dh の値を 2^{-6} から 2^{-1} までの 6 通りで行い, 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ったものである. Dh が 2^{-6} と 2^{-1} における残差ノルムの収束の様子を図 3 と図 5 に, そのとき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($m, l, itmax$) 法が最も速く収束しているのは, $Dh = 2^{-6}, 2^{-3}, 2^{-2}, 2^{-1}$ の場合である. 図 4 から, $Dh = 2^{-6}$ での A-GMRES($m, l, itmax$) 法のリスタート周期の値が小刻みに振動している様子が分かる. 特に AINV&A-GMRES($m, 1, 2560$) 法のリスタート周期は, 大きな値と小さな値を交互に繰り返しており, 計算時間は最短となっている. 式 (4) の見積りから決定されたリスタート周期は, 各々の局面で適切な値だったといえる. $Dh = 2^{-1}$ では, A-GMRES($m, l, itmax$) 法はいったん大きなリスタート周期で反復するが, その後, 反復回数 250 回前後からは最小のリスタート周期で収束に至っている. 図 5

表 3 数値例 1 の結果 2, 計算時間: time (sec.), 反復回数: iter.

Table 3 Numerical results of example 1 (Part 2), computation time.: time (sec.), number of iterations.: iter.

算法	Dh					
	2^{-3}		2^{-2}		2^{-1}	
	time	iter.	time	iter.	time	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)	263.7	1443	699.6	1988	668.5	1920
	$m : 4 \rightarrow 52$		$m : 4 \rightarrow 100$		$m : 4 \rightarrow 100$	
A-GMRES(m , 1, 7680)	220.8	3925	125.5	1597	165.8	1191
A-GMRES(m , 2, 7680)	161.5	2855	147.2	1653	145.2	1076
A-GMRES(m , 3, 7680)	223.4	4337	134.9	1726	134.0	1125
A-GMRES(m , 5, 7680)	215.8	3500	185.5	1769	169.5	1261
A-GMRES(m , 10, 7680)	198.8	2739	144.8	1629	126.2	1106
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)	510.3	839	519.9	1069	388.6	771
	$m : 4 \rightarrow 94$		$m : 4 \rightarrow 100$		$m : 4 \rightarrow 100$	
AINV&A-GMRES(m , 7680)	48.8	754	45.7	541	54.4	625
	$m : 4 \rightarrow 6$		$m : 4 \rightarrow 10$		$m : 4 \rightarrow 14$	
AINV&A-GMRES(m , 1, 2560)	200.3	1003	138.1	730	137.8	639
AINV&A-GMRES(m , 2, 2560)	126.9	936	150.8	730	139.5	662
AINV&A-GMRES(m , 3, 2560)	94.3	911	102.6	649	235.5	772
AINV&A-GMRES(m , 5, 2560)	293.6	1246	243.2	851	357.7	912
AINV&A-GMRES(m , 10, 2560)	668.0	839	214.8	879	357.6	916
AINV&A-GMRES(m , 1, 7680)	49.5	833	36.3	596	31.1	467
AINV&A-GMRES(m , 2, 7680)	52.7	879	31.4	499	32.1	467
AINV&A-GMRES(m , 3, 7680)	47.2	785	33.0	534	38.4	473
AINV&A-GMRES(m , 5, 7680)	55.0	905	34.2	528	33.4	489
AINV&A-GMRES(m , 10, 7680)	55.8	883	34.3	525	46.0	568

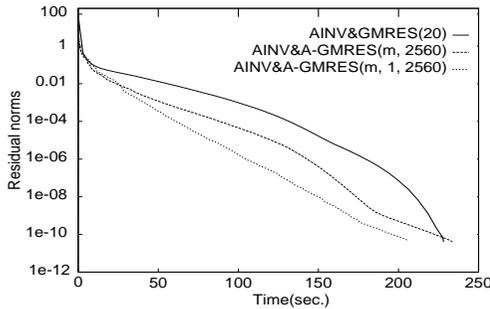


図 3 数値例 1 ($Dh = 2^{-6}$): 計算時間に対する残差ノルム
Fig.3 Example 1 ($Dh = 2^{-6}$): The behaviour of residual norm vs. computation time (sec.).

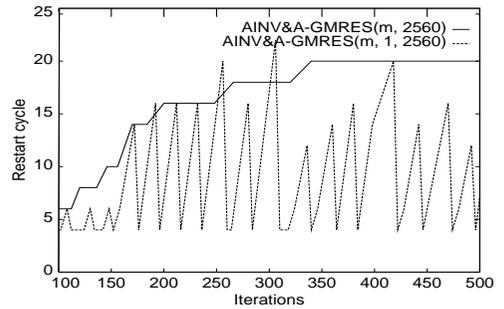


図 4 数値例 1 ($Dh = 2^{-6}$): 反復回数に対するリスタート周期
Fig.4 Example 1 ($Dh = 2^{-6}$): The behaviour of restart cycle vs. number of iterations.

において, 計算時間 20 秒以前では停滞が起きていて, それ以後急激に収束している. A-GMRES(m , $itmax$) 法はその停滞のときにリスタート周期を大きくしてしまうため, 計算時間をより多く必要とする. 特に, A-GMRES(m , 7680) 法と AINV&A-GMRES(m , 2560) 法は, m の値を最大リスタート周期まで上げてしまい, その結果膨大な計算時間を必要とした. しか

し, AINV&A-GMRES(m , 5, 2560) 法や AINV&A-GMRES(m , 10, 2560) 法も残差ノルムがゼロに収束するまでに多くの計算時間を必要とする. これは l が大きいため, 収束の様子が急激に変化した際に, リスタート周期を減らす処置が遅れたためである.

5.3.2 数値例 2

矩形領域 $\Omega = [0, 1] \times [0, 1]$ における 2 階の偏微分

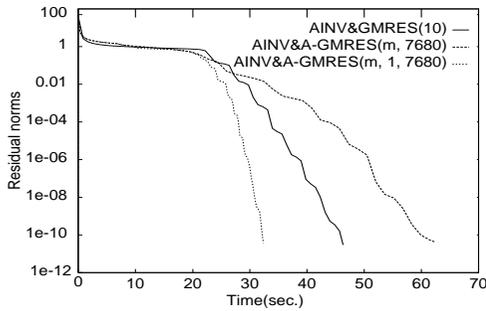


図5 数値例1 ($Dh = 2^{-1}$): 計算時間に対する残差ノルム
Fig. 5 Example 1 ($Dh = 2^{-1}$): The behaviour of residual norm vs. computation time (sec.).

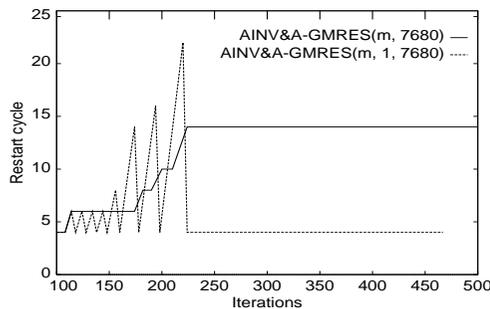


図6 数値例1 ($Dh = 2^{-1}$): 反復回数に対するリスタート周期
Fig. 6 Example 1 ($Dh = 2^{-1}$): The behaviour of restart cycle vs. number of iterations.

方程式の Dirichlet 境界値問題を考える (Joubert⁶⁾ を参照)。

$$-u_{xx} - u_{yy} + D\{(y - 1/2)u_x(x, y) + (x - 1/3)(x - 2/3)u_y(x, y)\} = G(x, y), \quad (9)$$

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

ただし、メッシュ幅を $h = 1/257$ とし数値例1と同様に5点中心差分で離散化する。厳密解を $u(x, y) = 1 + xy$ と設定して右辺を決定し、数値実験を行った。なお、この問題では行列の次元は65536となる。 Dh の値は、 2^{-6} から 2^{-1} の6通りで数値実験し、得られた結果を表4と表5に示した。

A-GMRES($m, 1, 7680$)法は前処理を行わない算法の中では全体的に最も良い結果となっている。前処理付きの算法では、AINV&A-GMRES($m, 2560$)法と AINV&A-GMRES($m, 7680$)法、AINV&A-GMRES($m, 1, 2560$)法と AINV&A-GMRES($m, 1, 7680$)法をそれぞれ比較してみると、同じ算法でも計算時間に大きな差が出ている。これは最大反復回数がパラメータになっているためである。最大反復回数を大きく

とり過ぎると小さなリスタート周期のまま計算し続けてしまう。それでも残差ノルムは最大反復回数までに収束しているが、AINV&GMRES(m)法の結果と比較すると最適なリスタート周期を選んではないことが分かる。AINV&A-GMRES($m, 2560$)法は Dh の値が 2^{-1} のときにリスタート周期を上げ過ぎてしまい、大幅に計算時間がかかることになる。AINV&GMRES(m)法や AINV&A-GMRES($m, itmax$)法は、 Dh の値によっては多くの計算時間が必要な場合があるが、AINV&A-GMRES($m, 1, 2560$)法は、 Dh がどの値でも極端に計算時間がかかることもなくバランス良く収束していることが分かる。

6. おわりに

本稿では、適応的にリスタート周期を決定するA-GMRES($m, l, itmax$)法を提案し、数値実験でGMRES(m)法、A-GMRES($m, itmax$)法との比較を行った。A-GMRES($m, itmax$)法の問題点として、リスタート周期を大きくし過ぎてしまうために発生する計算時間のオーバーヘッドがあるが、A-GMRES($m, l, itmax$)法はそのオーバーヘッドを十分に減少できた。その結果 AINV&A-GMRES($m, 1, 2560$)法は、全数値実験のトータル計算時間において最も速く収束し、AINV&A-GMRES($m, 2560$)法に対して18%程度計算時間の短縮を達成した。しかし、数値例1の $Dh = 2^{-1}$ において、AINV&A-GMRES($m, 5, 2560$)法や AINV&A-GMRES($m, 10, 2560$)法は収束までに多くの計算時間を必要とする。このことから、 l の値は計算時間の短縮に影響を与えていることが分かる。また、最大反復回数によっても計算時間に大きな違いがある。数値例2において、 $Dh = 2^{-6}, 2^{-5}$ では、AINV&A-GMRES($m, l, 7680$)法よりも AINV&A-GMRES($m, l, 2560$)法の方が、短い計算時間で残差ノルムは収束判定条件まで収束したが、 $Dh = 2^{-2}, 2^{-1}$ においては、逆に AINV&A-GMRES($m, l, 7680$)法の方が AINV&A-GMRES($m, l, 2560$)法よりも速く収束している。本稿の数値実験では最大反復回数を $10\sqrt{N}$ 回または $30\sqrt{N}$ 回としているが、最大反復回数がパラメータとなっているため、そのとり方によって結果が大きく異なっている。今後の課題として、様々な問題や前処理の有無に対して適切な最大反復回数の決定があげられる。

参考文献

- 1) Hestenes, M.R. and Stiefel, E.: Methods of

表4 数値例2の結果1, 計算時間: time (sec.), 反復回数: iter.

Table 4 Numerical results of example 2 (Part 1), computation time.: time (sec.), number of iterations.: iter.

算法	Dh					
	2 ⁻⁶		2 ⁻⁵		2 ⁻⁴	
	time	iter.	time	iter.	time	iter.
GMRES(10)	—	—	—	—	—	—
GMRES(20)	—	—	—	—	—	—
GMRES(40)	1766.1	7276	1500.7	6419	910.2	3632
A-GMRES(m, 7680)	1154.6	6921	1315.6	6912	890.8	4718
	4→24		4→28		4→28	
A-GMRES(m, 1, 7680)	1090.2	7613	976.1	7530	1082.2	7584
AINV&GMRES(10)	1687.7	7075	1540.6	6432	1375.1	5641
AINV&GMRES(20)	1008.0	3643	946.4	3237	662.4	1996
AINV&GMRES(40)	710.2	1980	721.2	1871	481.5	1388
AINV&A-GMRES(m, 2560)	614.4	2090	581.2	2042	545.8	1973
	4→22		4→22		4→24	
AINV&A-GMRES(m, 7680)	1348.9	5889	1467.4	6199	1523.1	6668
	4→8		4→8		4→8	
AINV&A-GMRES(m, 1, 2560)	628.6	2503	613.1	2441	664.9	2527
AINV&A-GMRES(m, 1, 7680)	1556.6	6699	1476.6	6422	1458.9	6310

表5 数値例2の結果2, 計算時間: time (sec.), 反復回数: iter.

Table 5 Numerical results of example 2 (Part 2), computation time.: time (sec.), number of iterations.: iter.

算法	Dh					
	2 ⁻³		2 ⁻²		2 ⁻¹	
	time	iter.	time	iter.	time	iter.
GMRES(10)	—	—	—	—	—	—
GMRES(20)	1307.5	7504	1019.7	6338	944.0	5605
GMRES(40)	955.1	3862	904.4	3637	823.7	3691
A-GMRES(m, 7680)	1252.8	7270	1052.4	6677	794.6	5487
	4→24		4→24		4→20	
A-GMRES(m, 1, 7680)	953.7	7507	914.1	7608	897.2	7615
AINV&GMRES(10)	757.8	3188	933.2	3752	735.3	3040
AINV&GMRES(20)	474.9	1720	638.8	2163	539.3	1987
AINV&GMRES(40)	532.0	1383	556.5	1596	589.2	1715
AINV&A-GMRES(m, 2560)	601.8	2264	549.0	1908	1076.4	1732
	4→20		4→26		4→100	
AINV&A-GMRES(m, 7680)	1051.4	4615	1208.3	5152	1164.6	5053
	4→8		4→6		4→6	
AINV&A-GMRES(m, 1, 2560)	676.4	2522	675.3	2561	700.0	2558
AINV&A-GMRES(m, 1, 7680)	1490.2	6469	1099.0	4765	1263.1	5505

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.*, No.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) Homer, F.W.: Implementation of the GM-

RES method using Householder transformations, *SIAM J. Sci. Comput.*, Vol.9, pp.152–163 (1988).

- 6) Joubert, W.: Lanczos methods for the solution of nonsymmetric systems of linear equations, *SIAM J. Matrix. Anal. Appl.*, Vol.13, pp.928–943 (1992).
- 7) Gutknecht, M.H.: Variants of BiCGSTAB for matrices with complex spectrum, *SIAM J. Sci. Comput.*, Vol.14, pp.1020–1033 (1993).
- 8) Greenbaum, A., Rozložník, M. and Strakoš, Z.: Numerical Behaviour of the Modified Gram-Schmidt GMRES Implementation, *BIT*, Vol.37, No.3, pp.706–719 (1997).

- 9) Huckle, T.K.: Efficient Computation of Sparse Approximate Inverses, *Numer. Linear Algebra Appl.*, Vol.5, pp.57–71 (1998).
- 10) 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).
- 11) 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).
(平成 13 年 3 月 19 日受付)
(平成 14 年 3 月 14 日採録)
- 12) Habu, M. and Nodera, T.: GMRES(m) Algorithm with Changing the Restart Cycle Adaptively, *Proc. ALGORITHMY 2000, Conference on Scientific Computing*, pp.254–263 (2000).



羽部 充 (正会員)

1977 年生まれ。2000 年 3 月慶應義塾大学理工学部数理科学科卒業。現在、慶應義塾大学大学院理工学研究科 (基礎理工学専攻) 在学中。ハイパフォーマンス・コンピューティングや数値計算分野に興味を持つ。



野寺 隆 (正会員)

1982 年慶應義塾大学大学院工学研究科博士課程 (数理工学専攻) 修了。現在、同大学助教授。その間、1986 年より 1 年間米国スタンフォード大学客員教授。大規模な行列計算の算法の研究開発に従事。ハイパフォーマンス・コンピューティングや文書処理に興味を持つ。著書に『楽々 L^AT_EX』(共立出版)等がある。工学博士。エッセイスト。SIAM, 日本応用数理学会会員。