

リスタート周期を動的に変える GMRES(k) 法について

羽 部 充[†] 野 寺 隆[†]

非対称の大型疎行列を係数とする連立1次方程式を解く算法の一つである GMRES 法は、膨大な記憶容量や演算量を必要とするため、通常はそのリスタート版である GMRES(k) 法が用いられる。一般に、 k はリスタート周期と呼ばれ、標準の GMRES(k) 法では一定の値である。近年、Sosonkina ら⁵⁾ によってこのリスタート周期の値を動的に変化させ、収束の具合に適合させる方法が提案されている。本稿では、適応的にリスタートを繰り返す Adaptive GMRES(k) 法の有効性を検証し、さらにその収束性を改善させる改良方法を提案する。

GMRES(k) method with varying its restart frequency adaptively

MITSURU HABU[†] and TAKASHI NODERA[†]

One of iterative methods to solve large and sparse nonsymmetric linear systems of equations, GMRES algorithm needs huge amount of computation and memory. Therefore, the restarted version GMRES(k) algorithm is used in practice. The algorithm is restarted every k iterations, and its value is constant. Recently, Sosonkina *et al.*⁵⁾ proposed adaptive technique of GMRES(k) algorithm which determines the restart value k based on the some criterion estimating its convergence rate. In this paper, we discuss practical issues concerning the new adaptive restarting technique of GMRES(k) algorithm and also present the effectiveness of this algorithm. Moreover, we will propose some modified version of Adaptive GMRES(k) algorithm which improves the convergence rate.

1. はじめに

本稿で取り上げる GMRES 法は、大型で疎な係数行列をもつ連立1次方程式

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

の解法の一つである。このような連立1次方程式は、偏微分方程式の境界値問題を、有限要素法や有限差分法で離散化すると得られることが多い。GMRES 法は反復法の中では非定常反復法に分類され、CG 法¹⁾ や BiCG 法³⁾⁴⁾、BiCGStab(l) 法⁹⁾¹⁰⁾ などと比較すると、より多くの計算量や記憶容量を必要とする。しかし、GMRES 法は BiCG 法系の算法と比較すると、各反復において残差ベクトルのノルムを減少させるといふ良い性質を持っているので、現在では広く使われている。

本稿は、次のような構成になっている。まず、第2節においては、GMRES 法とそれに関連する算法について述べる。次に、第3節において、リスタート周期を動的に切り変える算法について述べる。最後に、第5節において、それらの算法を比較検討し、Adaptive

GMRES(k) 法の有効性について述べる。

2. GMRES 法関連の算法について

この節では GMRES 法に関連する算法と、その特性について簡単に説明する。

2.1 GMRES 法

GMRES 法は 1986 年に Saad ら²⁾ によって提案され、大型で疎な非対称行列を係数行列とする連立1次方程式の反復解法の1つである。GMRES 法は各反復においてクリロフ部分空間での正規直交系を生成し、その空間上で残差ベクトルのノルムを最小化する算法である。また、 i 回目の反復における残差ベクトルのノルムは、次の式で与えられる。

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

ただし、式(2)は、アーノルディ過程から得られた K_i の正規直交系 $\{v_1, v_2, \dots, v_i\} = V_i$ によって $z = V_i y$ と表すことができる。また、 V_i はユニタリ行列であるため、この問題は次の最小2乗問題に帰着できる。

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

[†] 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

```

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

```

図1 GMRES(k)法の算法
Fig. 1 GMRES(k) method.

ただし、 $\beta = \|r_0\|$ である。さらに、式(3)の上ヘッセンベルグ行列 \bar{H}_i をQR分解することによって、残差ベクトルのノルムを容易に求めることができる。さらに、その近似解 x_i は $x_i = x_0 + V_i y_k$ で得られることになる。

2.2 GMRES(k)法

GMRES法は近似解 x_i を計算するために、正規直交系 V_i と上ヘッセンベルグ行列 \bar{H}_i を保存しておかなければならない。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、 k 回の反復ごとにその時点での近似解 x_k を初期近似解 x_0 として再出発するGMRES(k)法が使われている。ただし、 k はリスタート周期と呼ばれる。GMRES(k)法の算法を図1に示す。この k の値を動的に変化させる方法を第3節で述べる。

3. Adaptive GMRES(k)法

1998年に Sosonkina ら⁵⁾によって提案された Adaptive GMRES(k)法(以下 A-GMRES(k)法と呼ぶ)は、収束の具合に応じてリスタート周期を増やしていく方法である。A-GMRES(k)法は、通常用いられるGMRES(k)法に次のような処理を行なって収束を向上させる手法である。すなわち、残差ベクトルのノルムの収束の様子が最大反復回数に対して小さい割合の

時、収束を早めるためにリスタート周期の値を増やす。収束までのおおよその反復回数を l とすると、次式によってそれを見積もることができる。

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

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収束がごくわずかで、計算精度によっては全く変化がなかったとみ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。式(4)は、反復回数に対する残差ベクトルのノルムのグラフから、1回のリスタート間における、反復回数と残差ベクトルのノルムの減少の比をとることで得られる。式(4)の値と最大反復回数までの残りの反復回数を比較し、その割合に対してリスタート周期を増やすかどうかを判断する。この値は、収束までの反復回数の見積もりを、一つ前のリスタート時とその時点の2点を通る直線で考えている。もし、その見積もりが大きく、最大反復回数までに収束できないと判断した場合、A-GMRES(k)法は残差ベクトルのノルムの減少が十分になるまでリスタート周期を増やしていく。これによりある程度問題に応じたリスタート周期をとることができる。以上をまとめると図2に示した部分をGMRES(k)法に付加すればよいことになる。

A-GMRES(k)法はリスタート周期を増やすだけである。しかし、リスタート周期が大きい程1回の反復にかかる計算時間は増加する。たとえば、一旦リスタート周期を増加させた後、式(4)による見積もりが極端に小さくなった場合は、小さいリスタート周期でも十分収束できる可能性があり、それによって計算時間も短縮できるのだが、A-GMRES(k)法はそれに対応できず、リスタート周期を増やしたままで反復し続けなければならない。そこで、もしリスタート周期を減らすことができるなら、残差ベクトルのノルムが収束するまでの計算時間を短縮できる可能性がある。次に、リスタート周期を減少させる手法について考えることにする。

3.1 リスタート周期を減らす方法

この節では、リスタート周期を減らす処置として、現在我々が実装中のいくつかの方法について述べる。

- (1) A-GMRES(k)法の増やす過程と同じようにしてリスタート周期を m づつ減らす方法
この方法は、リスタート時に式(4)の値を参照し、十分な残差ベクトルのノルムの減少があった場合に k の値を減らしてリスタートする方法である。この方法だと1回のリスタートにおいてリスタート周期の値を m しか減らすことができないため、収束の様子の急激な変化には対応しにくい。

表 2 数値例 1 の結果

Table 2 Numerical results of example 1.

η	1.5		1.7		1.9	
	T	I	T	I	T	I
GMRES(10)	16.0	106	25.1	160	42.7	277
GMRES(20)	26.5	105	37.4	156	68.3	262
GMRES(40)	45.3	104	66.0	155	118.3	257
A-GMRES(4→)	11.4	115	17.5	179	32.7	331
	4→4		4→4		4→4	

表 3 数値例 2 の結果

Table 3 Numerical results of example 2.

Dh	2^{-6}		2^{-5}		2^{-4}		2^{-3}		2^{-2}		2^{-1}	
	T	I	T	I	T	I	T	I	T	I	T	I
GMRES(10)	—	—	—	—	754.5	4612	328.4	1971	140.1	867	158.4	950
GMRESS(20)	2060.3	7613	1424.4	5422	616.0	2466	333.5	1236	274.4	1066	286.5	1062
GMRES(40)	2034.4	4164	1229.7	2724	621.4	1421	535.7	1116	629.9	1386	654.5	1445
A-GMRES(4→)	1434.1	4274	886.4	3375	488.4	1734	708.1	1535	2078.5	2027	1753.2	1964
	4→32		4→24		4→28		4→52		4→100		4→100	

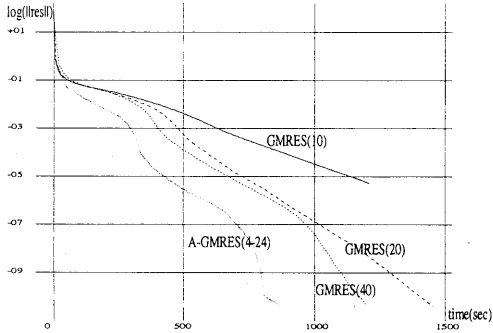


図 3 数値例 2 ($Dh = 2^{-5}$): 計算時間に対する残差ノルム
Fig. 3 Example 2 ($Dh = 2^{-5}$): The behaviour of residual norm vs. computational time (sec).

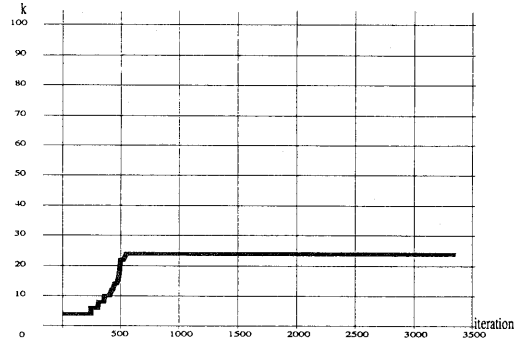


図 5 数値例 2 ($Dh = 2^{-5}$): 反復回数に対する A-GMRES(k) 法のリスタート周期
Fig. 5 Example 2 ($Dh = 2^{-5}$): The behaviour of restart frequency of A-GMRES(k) method vs. number of iterations.

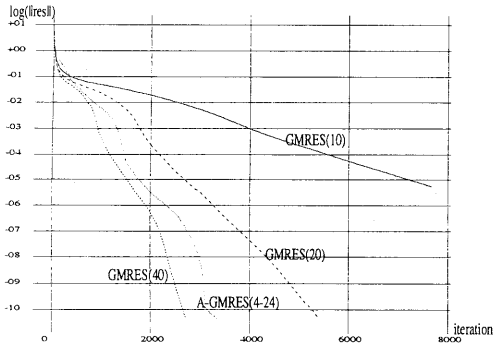


図 4 数値例 2 ($Dh = 2^{-5}$): 反復回数に対する残差ノルム
Fig. 4 Example 2 ($Dh = 2^{-5}$): The behaviour of residual norm vs. number of iterations.

時間, 反復回数ともに良い結果が得られている. 計算時間では最も速く, 2 番目に速かったものと比較しても平均で 75%程度短くなっている. $Dh = 2^{-5}$ のときの残差ベクトルのノルムの収束の様子を図 3, 図 4 に, その時の A-GMRES(k) 法のリスタート周期の増加の様子を図 5 に示す. A-GMRES(k) 法のリスタート周期は 548 回の反復までに 24 になっており, その後は増加していない. $Dh = 2^{-2}, 2^{-1}$ においては A-GMRES(k) 法のリスタート周期は上限の 100 まで増加している. $Dh = 2^{-2}$ のときの残差ベクトルのノルムの収束の様子を図 6, 図 7 に, その時の A-GMRES(k) 法のリスタート周期の増加の様子を図 8 に示す. $Dh = 2^{-2}, 2^{-1}$ での A-GMRES(k) 法の計算時間が大きくなっている

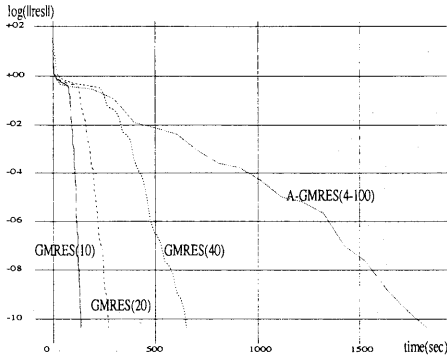


図6 数値例2 ($Dh = 2^{-2}$): 計算時間に対する残差ノルム
Fig. 6 Example 2 ($Dh = 2^{-2}$): The behaviour of residual norm vs. computational time (sec).

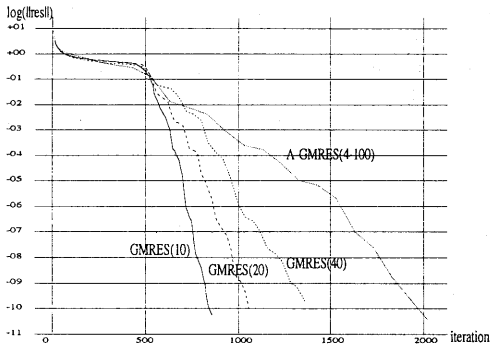


図7 数値例2 ($Dh = 2^{-2}$): 反復回数に対する残差ノルム
Fig. 7 Example 2 ($Dh = 2^{-2}$): The behaviour of residual norm vs. number of iterations.

のは、リスタート周期が増えるにつれ、1回の反復にかかる計算時間が増えてしまうからである。図7を見るとどの算法でも約500回の反復の辺りで停滞が起きている。その後は大きく収束へ向かっているが、A-GMRES(k)法はその停滞のときにリスタート周期を大きく増やしてしまうため、停滞を過ぎても収束までに時間がかかってしまっている。A-GMRES(k)法はリスタート周期を減らす処置がないためにこういった収束をする問題に対しては有効ではないことが分かった。

5.3 数値例3

矩形領域 $\Omega = [0, 1] \times [0, 1]$ における2階の偏微分方程式のディリクレ境界値問題を考える (Joubert⁹) を参照)。

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

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

メッシュ幅を $h = 1/129$ とし数値例2と同様に5点中心差分で離散化する。なお、真の解を $u(x, y) = 1 + xy$ と設定して右辺を決定し数値実験を行なった。この間

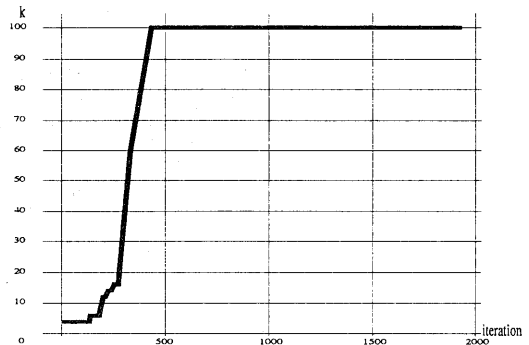


図8 数値例2 ($Dh = 2^{-2}$): 反復回数に対する A-GMRES(k) 法のリスタート周期

Fig. 8 Example 2 ($Dh = 2^{-2}$): The behaviour of restart frequency of A-GMRES(k) method vs. number of iterations.

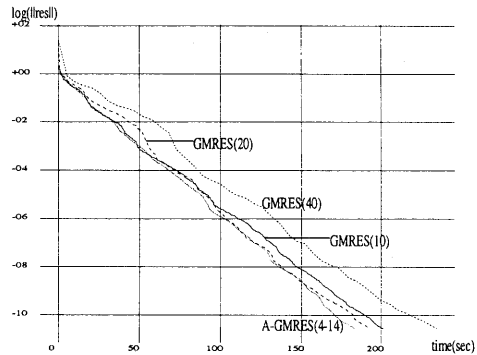


図9 数値例3 ($Dh = 2^0$): 計算時間に対する残差ノルム
Fig. 9 Example 3 ($Dh = 2^0$): The behaviour of residual norm vs. computational time (sec).

題では、行列の次元は16384とし、最大反復回数は3840回とした。 Dh の値は $2^{-1}, 2^0, 2^1$ の3通りで数値実験し、得られた結果を表4に示した。

A-GMRES(k)法は $Dh = 2^{-1}, 2^0$ ではどちらもリスタート周期が14まで増え、まずまずの結果が得られている。 $Dh = 2^0$ のときの残差ベクトルのノルムの収束の様子を図9、図10に、その時のA-GMRES(k)法のリスタート周期の増加の様子を図11に示す。 $Dh = 2^1$ においては、数値例2の $Dh = 2^{-2}, 2^{-1}$ の場合と同様にリスタート周期が100まで増えてしまい、計算時間はGMRES(20)法の約2.7倍もかかってしまっている。

6. おわりに

SGIの並列計算機Origin 2000を用いてのいくつかの数値実験によってGMRES(k)法とA-GMRES(k)

表 4 数値例 3 の結果

Table 4 Numerical results of example 3.

Dh	2 ⁻¹		2 ⁰		2 ¹	
	T	I	T	I	T	I
GMRES(10)	—	—	201.5	3105	—	—
GMRES(20)	176.7	2051	193.2	2248	255.7	2965
GMRES(40)	231.3	1686	235.8	1771	354.2	2687
A-GMRES(4→)	195.3	2705	186.9	2618	690.0	2610
	4→14		4→14		4→100	

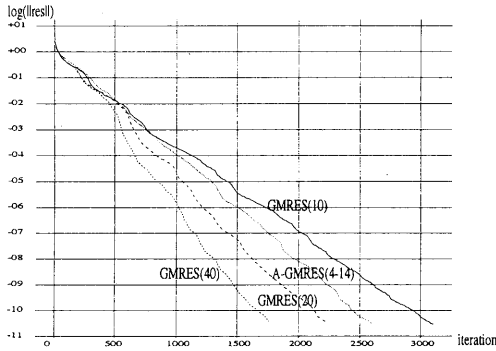


図 10 数値例 3 ($Dh = 2^0$): 反復回数に対する残差ノルム
Fig. 10 Example 3 ($Dh = 2^0$): The behaviour of residual norm vs. number of iterations.

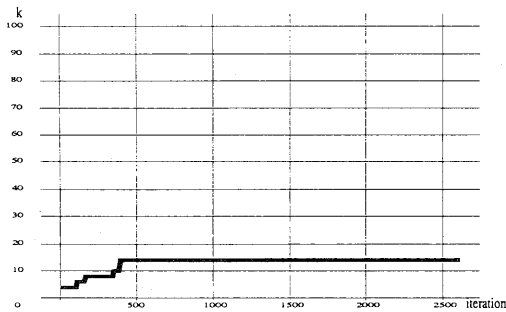


図 11 数値例 3 ($Dh = 2^0$): 反復回数に対する A-GMRES(k) 法のリスタート周期の増加
Fig. 11 Example 3 ($Dh = 2^0$): The behaviour of restart frequency of A-GMRES(k) method vs. number of iterations.

法との比較検討を行なった。A-GMRES(k)法は式(4)によって収束までのおおよその反復回数を見積もり、それによってリスタート周期を変化させる手法である。式(4)は局所的な残差ベクトルのノルムの収束と反復回数の比をとっているため、図4や図10のような直線的な収束をする問題では、A-GMRES(k)法は有効であることが分かった。しかしそうでない場合、残差ベクトルのノルムの収束が停滞した後に大きく収束へ向かう図7のような問題に対しては、リスタート周期

を増やし過ぎてしまい、収束までかなりの計算時間をとることになった。これは収束の状況に応じてリスタート周期を減らす過程がないことが原因であると思われる。

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) 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).
- 3) Lanczos, C.: Solution of systems of linear equations by minimize iteration, *J. Res. Nat. Bur. Standard*, Vol. 49, pp. 33-53 (1952).
- 4) Fletcher, R.: Conjugate gradient methods for indefinite systems, *Lecture Notes in Math.*, Vol. 506, pp. 73-89 (1976).
- 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) 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) Reichel, L. and Trefethen, L. N.: Eigenvalues and Pseudo-eigenvalues of Toeplitz Matrices, *Lin. Alg. Appl.*, Vol. 162, pp. 153-185 (1992).
- 8) Gutknecht, M. H.: Variants of BiCGSTAB for matrices with complex spectrum, *SIAM J. Sci. Comput.*, Vol. 14, pp. 1020-1033 (1993).
- 9) 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).
- 10) 森屋, 野寺: 適応的に l を変化させるBiCGStab(l)法, *情報処理学会論文誌*, Vol. 40, No. 6, pp. 2669-2678 (1999).