

残差ノルムの収束停滞を適応的に回避する GMRES(m) 法

森屋 健太郎[†] 野寺 隆^{††}

GMRES(m) 法は大型で疎な連立 1 次方程式の反復法の 1 つであり, m 回の反復ごとにリスタートを行うことで直交化に必要な計算量を減少させる算法である. しかし, このリスタートが原因で残差ノルムの収束の停滞を引き起こすことがある. 本稿では, GMRES(m) 法における残差ノルムの収束の停滞を判定するパラメータ ζ の 2 つの計算式を導入し, リスタート周期を適応的に切り替える GMRES(m) 法の新しい改良版を提案する.

GMRES(m) Method Avoiding the Stagnation of Convergence of the Residual Norm in Adaptation

KENTARO MORIYA^{††} and TAKASHI NODERA^{††}

GMRES(m) method is one of the iterative methods for solving large sparse linear systems of equations, and it decreases the orthogonalization cost by restarting at every m iterations. However, the restart often causes the stagnation of the convergence of residual norm. In this paper, we study two formulas of the parameter ζ which avoids the stagnation of convergence in GMRES(m) method. We also design the new variant of GMRES(m) method by switching the restart cycle adaptively.

1. はじめに

偏微分方程式の境界値問題は, 理工学の諸現象を解析するのにしばしば用いられる. このような問題を計算機で数値的に解くためには有限差分法や有限要素法などで離散近似をして連立 1 次方程式

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

を得ることになる. ただし, 方程式 (1) の係数行列 A は大型で疎な正則行列とする. 通常, 方程式 (1) は反復法によって解かれるが, 本稿では非定常反復法の 1 つである GMRES(m) 法について取り扱う. GMRES(m) 法は GMRES 法にリスタートを適用することで直交化にかかる計算量を減少させ, 問題の規模にかかわらず計算機への実装を可能にした算法である. しかし, リスタートを行うことで残差ノルムの収束がしばしば停滞することがある.

本稿では, Moriya ら⁷⁾が提案した残差ノルムの収束の停滞を判定するパラメータの 2 つの計算式について考察する. さらに, この 2 つの計算式を必要に応じて

で使い分けることにより, 残差ノルムの収束の停滞を判定し GMRES(m) 法のリスタート周期を適応的に切り替える新しい算法を提案する. 数値結果から, このようにリスタート周期を切り替える GMRES(m) 法が, 従来の GMRES(m) 法に比べて短い計算時間で残差ノルムを収束させることを示す.

2. GMRES(m) 法

本章では GMRES(m) 法について簡単に述べることにする. 算法の詳細に関しては, Saad ら²⁾を参照してほしい.

GMRES(m) 法は, アーノルディ過程¹⁾から m 次元の正規直交基底 $V_m = (v_1, \dots, v_m)$ とヘッセンベルグ行列 \bar{H}_m を生成し, 残差ノルムの最小 2 乗問題

$$\min_{x_m \in \mathcal{K}_n} \|b - Ax_m\|_2^2 = \min_{y \in \mathcal{K}_m} \|\beta e_1 - \bar{H}_m y\|_2^2 \quad (2)$$

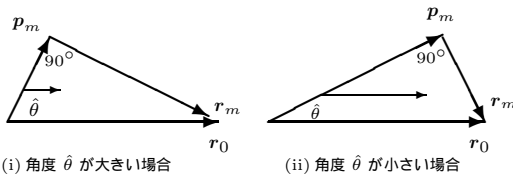
を解くことになる. ただし, $\beta = \|r_0\|_2$ とする. ここで, ヘッセンベルグ行列 \bar{H}_m は, ij 成分に $h_{i,j} = (v_i, Av_j)$ を持ち, その行列の大きさは $(m+1) \times m$ である. 近似解 x_m はこの m 本の正規直交基底によって

$$x_m = x_0 + V_m y_m \quad (3)$$

として更新される. GMRES(m) 法では式 (2) で示される m 次元の最小 2 乗問題を解いたときに, 残差ノ

[†] 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



(i) 角度 $\hat{\theta}$ が大きい場合 (ii) 角度 $\hat{\theta}$ が小さい場合

図 1 ベクトル r_m , r_0 , p_m の関係

Fig. 1 The relation among the vectors r_m , r_0 and p_m .

ルムが収束していないとその時点で求まっていた残差ベクトルと初期近似解を初期値として算法のリスタートを行うことになる。しかし、残差ノルムはあくまで m 次元の最小 2 乗問題によって、局所的な最小化が行われているにすぎないので、リスタートによってしばしば残差ノルムの収束の停滞が起こることがある。

3. パラメータ ζ_m の 2 つの計算方法

本章では GMRES(m) 法における残差ノルムの収束停滞を判定するパラメータ ζ_m の 2 つの計算方法について考察する。

3.1 内積による計算方法

最初に、GMRES(m) 法が生成する残差ベクトル

$$r_m = r_0 - AV_m y_m \quad (4)$$

について考える。式 (4) で表される残差ベクトルの変化の大小は、探索ベクトル $p_m = AV_m y_m$ と 1 つ前のリスタート周期での残差ベクトル r_0 のなす角度 $\hat{\theta}$ によって決定されている。図 1 (i) のように、残差ベクトル r_0 と探索ベクトル p_m のなす角度 $\hat{\theta}$ が大きいときには、 p_m の r_0 に平行となる成分が小さいので、残差ベクトル r_m の変化は小さく残差ノルムの収束が停滞を起こす可能性は増加する。それに対して図 1 (ii) のように、残差ベクトル r_0 と探索ベクトル p_m のなす角度 $\hat{\theta}$ が小さいときには、 p_m の r_0 に平行となる成分が大きいため、それだけ残差ベクトルの変化は大きくなり、残差ノルムが大幅に減少することが予測できる。そこで、次のようなパラメータ

$$\zeta_m = \frac{(r_0, p_m)}{\|r_0\|_2 \cdot \|p_m\|_2} \quad (5)$$

を導入する。このパラメータ ζ_m は残差ベクトル r_0 と探索ベクトル p_m の内積を正規化したものであり、

$$|\zeta_m| = \cos \hat{\theta} < \cos \theta \quad (6)$$

なら残差ノルムの収束は停滞して減少の見込みがないと判定する。ここで、角度 θ は残差ノルムの収束停滞を判定する角度であり、 r_0 と p_m のなす角度 $\hat{\theta}$ が θ より大きい場合に残差ノルムの収束が停滞したと見なす。ただし、 $0 \leq \hat{\theta}$, $\theta \leq 90$ とする。式 (5) で表され

るパラメータは、さまざまな文献で扱われている。たとえば、津野ら⁶⁾では、ORTHOMIN(k) 法での残差ノルムの収束停滞の判定に使っており、これを基準にして ORTHOMIN(k) 法にリスタートをかけている。Moriya ら⁷⁾では、DEFLATED-GMRES(m , k) 法のリスタート周期を切り替える判定にパラメータ $|\zeta_m|$ を用いている。本稿では、GMRES(m) 法のリスタート周期を変化させる判定に用いているが、式 (5) による $|\zeta_m|$ の計算方法を次の 3.2 節で述べる計算方法と比較し、これら 2 つの計算方法を必要に応じて使い分ける点が前者 2 つの文献と異なる。

3.2 残差ノルムを用いた計算方法

次に、3.1 節の式 (6) で計算されるパラメータ $|\zeta_m|$ は、Saad⁵⁾ [p. 135] に基づき、次のように簡単に式変形することができる。まず、残差ノルムは次のように変形できる。

$$\begin{aligned} \|r_m\|_2^2 &= (r_0 - p_m, r_0 - p_m) \\ &= (r_0 - p_m, r_0) - (r_0 - p_m, p_m) \\ &= (r_0 - p_m, r_0) - (r_m, p_m) \end{aligned} \quad (7)$$

ここで、残差ノルム $\|r_m\|_2$ を最小化するためには、図 1 で示すように、残差ベクトル r_m と探索ベクトル p_m (ここでは $p_m = AV_m y_m$) が互いに直交するようにとられる。したがって、

$$(r_m, p_m) = (p_m, r_m) = 0 \quad (8)$$

の関係があるので、式 (8) を式 (7) に代入すると

$$\begin{aligned} \|r_m\|_2^2 &= (r_0 - p_m, r_0) \\ &= \|r_0\|_2^2 - (p_m, r_0) \\ &= \|r_0\|_2^2 - \frac{(p_m, r_0)(p_m, p_m)}{\|p_m\|_2^2} \\ &= \|r_0\|_2^2 - \frac{(p_m, r_0)(p_m, r_0 - r_m)}{\|p_m\|_2^2} \\ &= \|r_0\|_2^2 \\ &\quad - \frac{(p_m, r_0)\{(p_m, r_0) - (p_m, r_m)\}}{\|p_m\|_2^2} \end{aligned}$$

となる。さらに、この式に式 (8) を代入すると、

$$\|r_m\|_2^2 = \|r_0\|_2^2 - \frac{(p_m, r_0)^2}{\|p_m\|_2^2} \quad (9)$$

となるので、式 (9) を次のように式変形して

$$\frac{(p_m, r_0)^2}{\|p_m\|_2^2 \|r_0\|_2^2} = 1 - \frac{\|r_m\|_2^2}{\|r_0\|_2^2} \quad (10)$$

を得る。式 (10) について、両辺の平方根をとると

$$\frac{|(p_m, r_0)|}{\|p_m\|_2 \|r_0\|_2} = \sqrt{1 - \frac{\|r_m\|_2^2}{\|r_0\|_2^2}} \quad (11)$$

を導くことができる。式 (11) の左辺は式 (5) における ζ_m の絶対値にほかならないので

$$|\zeta_m| = \sqrt{1 - \frac{\|r_m\|_2^2}{\|r_0\|_2^2}} \quad (12)$$

となる．反復法では後に述べる 5 章の数値例のように，計算終了の判定に残差ノルムが用いられることが多い．通常は反復する過程で，この残差ノルムは計算されているので，式 (5) の代わりに式 (12) を使用すれば，パラメータ ζ_m を求めるのに新たに内積計算をしなくて済む利点がある．しかし実際には，リスタートを行う GMRES(m) 法の残差ノルムは単調には減少せず，式 (12) のルートの中が負の数値になり計算が破綻することがある．なぜなら，通常，浮動小数点演算では，丸め誤差の影響によって，直交基底 V_m の直交性が崩れることがあるためである．式 (4) のように，残差ベクトル r_m は直交基底 V_m に依存しており，このような場合には，残差ノルム $\|r_m\|_2$ と $\|r_0\|_2$ の間に，つねに $\|r_m\|_2 \leq \|r_0\|_2$ が成立するとは限らなくなる．

3.3 ζ_m の計算法

3.1 節と 3.2 節で述べた ζ_m の 2 つの計算方法には，次のような長所と短所がある．

- 式 (5) から求める場合

長所：直交基底 V_m の直交性が崩れてもブレイクダウンを起こすことがなく，つねに計算が可能である．

短所：内積計算が必要となるため，式 (12) に比べて計算量が多くかかる．

- 式 (12) から求める場合

長所：新たに内積計算をする必要がないため，式 (5) で必要とされる内積の計算量を抑えることができる．

短所： $\|r_m\|_2 > \|r_0\|_2$ のときには計算をすることができない．

本節では，残差ノルムの収束の停滞を判定する際に $\|r_m\|_2 > \|r_0\|_2$ ならば式 (5) で，そうでなければ式 (12) によって ζ_m を計算することを提案する．このような計算法によって， ζ_m を計算する際に内積計算のオーバヘッドを極力抑えることが可能である．

4. 適応的にリスタート周期を切り替える算法

本章では，3 章で述べた ζ_m を計算する 2 つの式を必要に応じて使い分け，さらにリスタート周期を動的に切り替えることで残差ノルムの収束停滞を回避する算法について提案する．

4.1 リスタート周期の変化法

GMRES(m) 法において，残差ノルムの収束が停滞する原因の 1 つはリスタート周期が小さすぎるため

である²⁾．このような場合にリスタート周期を一時的に大きな値に切り替えると，残差ノルムの収束の停滞が回避されやすくなる．そこで本節では次のようにして，通常はリスタート周期を m_{min} に設定しておき，残差ノルムの収束が停滞したときだけ最大値 m_{max} ($m < m_{max}$) の範囲内でリスタート周期を増加させる算法について提案する．

- (1) リスタートするたびに ζ_m を式 (5)，または式 (12) から計算する．
- (2) $|\zeta_m| < \cos \theta$ かつ $m + m_{min} < m_{max}$ なら，リスタート周期 m に m_{min} を加算する．
- (3) $|\zeta_m| \geq \cos \theta$ もしくは $m = m_{max}$ なら，リスタート周期 m を m_{min} に設定する．

ここでいう θ とは，3.1 節で述べた残差ノルムの収束停滞を判定する角度のことをさす．ステップ (1) では，リスタートをするときにはつねに残差ノルムの収束の停滞を判定している． ζ_m を計算する際には，残差ノルム $\|r_m\|_2$ と $\|r_0\|_2$ の大小関係から，式 (5) と式 (12) を 3.3 節で提案した方法によって使い分けることにする．ステップ (2) では， $|\zeta_m| < \cos \theta$ のとき残差ノルムの収束が停滞したと見なし，リスタート周期 m を m_{min} だけ増加させている．ここで，リスタート周期を一度に m_{max} まで変化させないのは，直交化の計算が必要以上に行われることを抑止するためである．それに対してステップ (3) では $|\zeta_m| \geq \cos \theta$ のとき残差ノルムの収束の停滞が解消されたとして，リスタート周期を m_{min} に戻している．また， $m = m_{max}$ のときには必ずリスタート周期を m_{min} にしている．これはリスタート周期が最大値 m_{max} であるときには，残差ノルムの収束の停滞が判定されてもこれ以上リスタート周期を加算することによって停滞を改善することができないためである．つまり，このような場合には無駄に直交化の計算が行われることを回避するために，リスタート周期を m_{min} に切り替えている．しかし，このようにリスタート周期を変化させるだけでは，角度 θ の値によって残差ノルムの収束の停滞が判定される条件が異なってしまう．そこで，次節では Moriya ら⁷⁾が提案した算法の考えに基づき，角度 θ も適応的に変化させることを考える．

4.2 角度 θ の変化法

残差ノルム $\|r_m\|_2$ の収束の停滞判定は，式 (6) における角度 θ の値が大きいと残差ノルムの収束の停滞が判定されにくくなり，収束の停滞を解消することができなくなる可能性がある．それに対して， θ が小さいとリスタート周期が必要以上に切り替わることになり直交化に必要な計算量だけが増加する．そこで，

Moriya ら⁷⁾が提案した算法に基づき、以下のようにして角度 θ を適応的に変化させることにする。

- (1) 初期値として θ を γ 度に設定する。
- (2) 残差ノルムの収束が停滞して、リスタート周期 m_{min} が増加するとき、直前の ζ_m を c_1 に登録する。
- (3) 残差ノルムの収束の停滞が解消されて、リスタート周期が $m = m_{min}$ に切り替わるなら、その直後の ζ_m を c_2 に登録する。
- (4) $m = m_{max}$ かつ $|\zeta_m| < \cos \theta$ かつ $\theta + \gamma < 90^\circ$ なら、 θ を γ 度だけ加算する。
- (5) $|c_1| > |c_2|$ かつ $\theta + \gamma < 90^\circ$ なら、 θ を γ 度だけ加算する。

ただし、 θ は $\gamma \leq \theta < 90^\circ$ の範囲で γ 度ずつ変化させることにする。ステップ (4) ではリスタート周期が最大値であるにもかかわらず、残差ノルムの収束の停滞が解消することができないときに、リスタート周期の増加による効果がないと見なして θ を加算している。このときには 4.1 節で述べたように、リスタート周期も $m = m_{min}$ に切り替わることになる。ステップ (5) ではリスタート周期が切り替わる直前の $|\zeta_m|$ と、リスタート周期が m_{min} に切り替わった直後の $|\zeta_m|$ を比較し、前者より後者の方が小さいなら、リスタート周期の切替えによって残差ノルムの収束性が改善されなかったとして、この場合にも θ の値を加算することにする。ただし、ステップ (4) によって θ を加算した場合には、ステップ (5) の処理は行わないこととする。この算法では、リスタート周期を増加させても残差ノルムの収束が改善されない場合に θ の値を増加させ、残差ノルムの収束の停滞が判定される条件を緩和している。

4.3 新しい GMRES(m) 法

本稿では、3 章と 4 章で述べた算法によって、残差ノルムの収束停滞を回避する機能を取り入れた GMRES(m) 法のことを GMRES(m_{min}, m_{max}) 法と表記する。GMRES(m_{min}, m_{max}) 法の 2 つのパラメータは次のようになる。

- m_{min} : 残差ノルムの収束が停滞していないときに使用する短いリスタート周期。この値は使用可能なリスタート周期の最小値でもある。
- m_{max} : 残差ノルムの収束が停滞したときに使用するリスタート周期の最大値

GMRES(m_{min}, m_{max}) 法をまとめると図 2 のようになる。従来の GMRES(m) 法との違いは、アーノルディ過程によって直交化を行う前にリスタート周期を ζ_m の値から適応的に決定していること、リスター

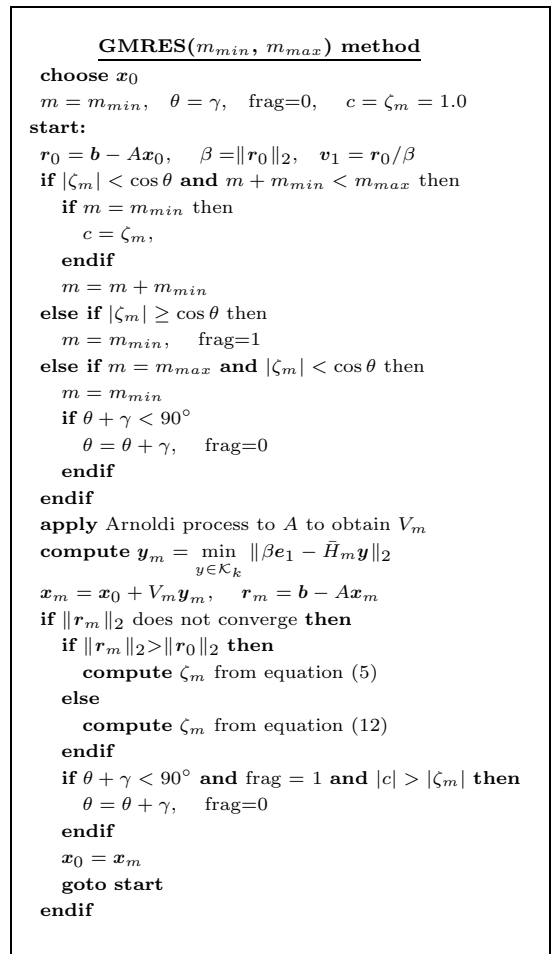


図 2 GMRES(m_{min}, m_{max}) 法
Fig. 2 GMRES(m_{min}, m_{max}) method.

トを行うたびに ζ_m を式 (5) もしくは式 (12) によって計算していることなどがあげられる。

5. 数値実験

方程式 (1) を GMRES(m) 法, GMRES(m_{min}, m_{max}) 法によって解く数値実験を行い、それぞれの算法の計算時間に関する性能を評価する。ただし、後者に関しては、 ζ_m を式 (5) のみから計算した場合と式 (12) のみから計算した場合についても示す。その際に、GMRES(m_{min}, m_{max}) 法は次のように表記する。

- GMRES(m_{min}, m_{max}) 法 + eq(5)
 ζ_m を式 (5) のみから計算した場合
- GMRES(m_{min}, m_{max}) 法 + eq(12)
 ζ_m を式 (12) のみから計算した場合
- GMRES(m_{min}, m_{max}) 法
3.3 節に基づき、 ζ_m を式 (5) と式 (12) の両方が

ら計算した場合
ここで、数値実験の環境は次のとおりである。

- CPU : PentiumIII 700 MHz
- メモリ搭載量 : 256 MB
- OS : Linux kernel 2.2.16
- 初期近似解 : $x_0 = 0$
- 残差ノルムの収束条件 : $\frac{\|r_m\|_2}{\|b\|_2} < 1.0 \times 10^{-12}$
- 計算精度 : 倍精度

評価の基準は、同じリスタート周期を持つもの
うしを比較する。たとえば、GMRES(10, 20) 法は、
GMRES(10) 法, GMRES(20) 法と比較する。なぜな

表 1 数値例 1 の GMRES(10, 40) 法における計算時間 (秒) と γ ($^\circ$) の関係 ($Dh = 2^{-2}$)
Table 1 Example 1: The relation between the computation time (sec) and γ ($^\circ$) for GMRES(10, 40) method ($Dh = 2^{-2}$).

γ	5.0	10.0	15.0	20.0	25.0	30.0
time	74.0	54.0	132.0	94.0	79.0	70.0

表 2 数値例 1 の計算結果 (time : 計算時間 (秒), iter : 反復回数)

Table 2 The numerical results for example 1 (time: computation time (sec), iter: iterations).

算法	Dh							
	2^{-3}		2^{-2}		2^{-1}		2^0	
	time	iter	time	iter	time	iter	time	iter
GMRES(10)	107.0	4220	93.0	3400	102.0	4060	80.0	3040
GMRES(20)	74.0	1920	113.0	2700	80.0	2060	86.0	2180
GMRES(30)	120.0	2280	129.0	2250	91.0	1740	89.0	1650
GMRES(40)	95.0	1440	106.0	1520	112.0	1720	123.0	1800
GMRES(50)	111.0	1350	123.0	1500	126.0	1550	151.0	1850
GMRES(60)	146.0	1560	132.0	1380	153.0	1620	179.0	1920
GMRES(10, 20) + eq(5)	48.0	1510	47.0	1460	45.0	1490	83.0	3000
GMRES(10, 20) + eq(12)	(9.0)*	(230)*	(22.0)*	(640)*	(18.0)*	(540)*	(4.0)*	(140)*
GMRES(10, 20)	45.0	1510	43.0	1460	44.0	1490	78.0	3000
GMRES(10, 30) + eq(5)	50.0	1320	57.0	1510	79.0	2660	64.0	1840
GMRES(10, 30) + eq(12)	(25.0)*	(560)*	(50.0)*	(1310)*	(32.0)*	(730)*	(6.0)*	(140)*
GMRES(10, 30)	46.0	1320	53.0	1510	74.0	2660	60.0	1840
GMRES(10, 40) + eq(5)	60.0	1460	60.0	1430	68.0	1690	78.0	1880
GMRES(10, 40) + eq(12)	(26.0)*	(490)*	(44.0)*	(940)*	(41.0)*	(860)*	(50.0)*	(1050)*
GMRES(10, 40)	54.0	1460	54.0	1430	61.0	1690	69.0	1880
GMRES(10, 50) + eq(5)	88.0	1650	85.0	1910	74.0	1740	86.0	1780
GMRES(10, 50) + eq(12)	(70.0)*	(1120)*	(60.0)*	(1040)*	65.0	1740	77.0	1780
GMRES(10, 50)	76.0	1650	79.0	1910	65.0	1740	76.0	1780
GMRES(10, 60) + eq(5)	94.0	1660	99.0	1640	96.0	1760	89.0	1960
GMRES(10, 60) + eq(12)	(64.0)*	(900)*	93.0	1640	87.0	1760	79.0	1960
GMRES(10, 60)	82.0	1660	94.0	1640	88.0	1760	79.0	1960
GMRES(20, 40) + eq(5)	76.0	1340	82.0	1440	87.0	1580	84.0	1640
GMRES(20, 40) + eq(12)	69.0	1340	75.0	1440	(23.0)*	(400)*	(19.0)*	(340)*
GMRES(20, 40)	69.0	1340	75.0	1440	79.0	1580	76.0	1640
GMRES(20, 60) + eq(5)	67.0	1180	91.0	1440	95.0	1520	101.0	1640
GMRES(20, 60) + eq(12)	62.0	1180	83.0	1440	87.0	1520	94.0	1640
GMRES(20, 60)	61.0	1180	83.0	1440	86.0	1520	94.0	1640
<< GMRES(m_{min}, m_{max}) 法で ζ_m を計算する数式の使われた回数 >>								
	式 (5)	式 (12)	式 (5)	式 (12)	式 (5)	式 (12)	式 (5)	式 (12)
GMRES(10, 20)	7	115	10	107	15	122	19	270
GMRES(10, 30)	5	83	7	95	8	235	11	126
GMRES(10, 40)	4	90	5	87	5	108	8	118
GMRES(10, 50)	4	79	4	120	0	117	0	104
GMRES(10, 60)	3	76	0	67	0	88	0	127
GMRES(20, 40)	0	48	0	51	1	59	1	65
GMRES(20, 60)	0	43	0	47	0	51	0	55

(数値)* :途中でブレイクダウンを起こしたときの計算時間および反復回数

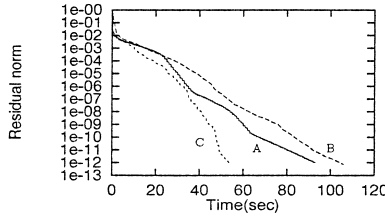
ら、リスタート周期は経験的に求まる値なので、同じ
値を持つ算法どうしを比較しないと公平な評価になら
ないためである。

5.1 数値例 1

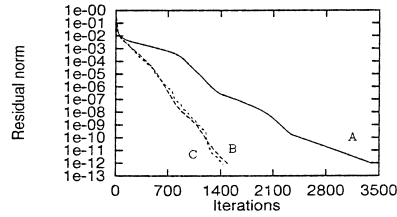
正方領域 $\Omega = [0, 1]^2$ における偏微分方程式の境界
値問題を考える⁴⁾。

$$\begin{aligned}
 & -u_{xx} - u_{yy} + D\{(y-1/2)u_x \\
 & + (x-2/3)(x-1/3)u_y\} = f(x, y) \text{ on } \Omega \\
 & u(x, y)|_{\partial\Omega} = 1 + xy
 \end{aligned}$$

ただし、右辺 $f(x, y)$ は厳密解が $u(x, y) = 1 + xy$
となるように設定する。領域 Ω をグリッド数 128^2 で
区切り 5 点中心差分によって離散化すると、次元が
16384 である連立 1 次方程式を得ることになる。最初
に、4.2 節で提案した θ を変化させる算法について、 γ
を適切な値に設定するために次のような実験を行った。
係数 $Dh = 2^{-2}$ のときの GMRES(10, 40) 法につい
て、 γ を 5.0° から 30.0° まで 5.0° ずつ変化させて
残差ノルムの収束条件を満たすまでの計算時間を計測



残差ノルム vs. 計算時間



残差ノルム vs. 反復回数

図3 数値例1: 残差ノルムの収束履歴 ($Dh = 2^{-2}$), A: GMRES(10), B: GMRES(40), C: GMRES(10, 40)

Fig. 3 Example 1: The convergence behavior of the residual norm ($Dh = 2^{-2}$), A: GMRES(10), B: GMRES(40), C: GMRES(10, 40).

表3 数値例1: GMRES(10, 40) 法のリスタート周期 m の出現回数 ($Dh = 2^{-2}$)

Table 3 Example 1: The number of the restart cycle m in GMRES(10, 40) method ($Dh = 2^{-2}$).

m	10	20	30	40
出現回数	61	19	7	5

した. ただし, h はグリッド間の幅を表し $h = 1/129$ である. 表1で示す実験結果によると, $\gamma = 10.0$ のときに計算時間が最も短縮されている. したがって, この事実をもとにして数値例1では $\gamma = 10.0$ と設定した. 次に係数 Dh を変化させたときに, 残差ノルムが収束条件を満たすまでに要した計算時間と反復回数を表2に示す. GMRES(m_{min}, m_{max}) 法では, 同じリスタート周期を持つ GMRES(m) 法に比べて計算時間が短縮されるケースが多かった. たとえば, 係数 $Dh = 2^{-2}$ のとき GMRES(10, 40) 法の計算時間は 54.0 秒であり, これは GMRES(10) 法と比べて4割程度, GMRES(10) 法と比べて5割程度短縮されている. このときの残差ノルムの収束の振舞を比較したグラフを図3に示す. また, このときの GMRES(10, 40) 法について, 出現したリスタート周期の回数を表3に示す. 表3より, $m = 10$ のリスタート周期が一番多く出現しており, 最大値の $m = 40$ のリスタート周期は出現回数が一番少なかった. ζ_m の計算法に関しては, 式(12) だけから ζ_m を計算した場合よりも式(5) と式(12) によって ζ_m を計算したほうが計算時間は全体的にさらに1割程度短縮されている. 式(5) だけから計算した場合にはルートの中が負になることがあって反復の途中でブレイクダウンを起こす場合がみられた. このときの計算時間と反復回数をカッコでくくっている. 一例として係数 $Dh = 2^{-2}$ の場合について, 式(12) のみで計算してブレイクダウンを起こしたときの ζ_m^2 の値を表4に示す. 表4のように, ζ_m^2 が負の値をとると平方根を計算することができなくなるので, ζ_m の計算はブレイクダウンを起こすこ

表4 数値例1: GMRES(m_{min}, m_{max}) 法 + eq(12) で, ブレイクダウンを起こしたときの ζ_m^2 の値 ($Dh = 2^{-2}$)

Table 4 Example 1: The value of ζ_m^2 for GMRES(m_{min}, m_{max}) method + eq(12) when breakdown occurs ($Dh = 2^{-2}$).

算法	ζ_m^2
GMRES(10, 20) + eq(12)	-0.1156
GMRES(10, 30) + eq(12)	-0.1116
GMRES(10, 40) + eq(12)	-0.0986
GMRES(10, 50) + eq(12)	-0.1082

表5 数値例2の GMRES(10, 20) 法における計算時間 (秒) と γ ($^\circ$) の関係 ($Dh = 2^{-2}$)

Table 5 Example 2: The relation between the computation time (sec) and γ ($^\circ$) for GMRES(10, 20) method ($Dh = 2^{-2}$).

γ	5.0	10.0	15.0	20.0	25.0	30.0
time	186.0	161.0	233.0	269.0	181.0	271.0

とになる.

5.2 数値例2

領域 $\Omega = [0, 1]^3$ 上での次の偏微分方程式の境界値問題を考える³⁾.

$$a_1 u_{xx} + a_2 u_{yy} + a_3 u_{zz} + R(a_4 u_x + a_5 u_y + a_6 u_z) + a_7 u = g(x, y, z) \text{ on } \Omega$$

ただし,

$$\begin{aligned} a_1 &= 2 + \sin(2\pi x) \cos(2\pi y) \cos(2\pi z) \\ a_2 &= 2 + \cos(2\pi x) \sin(2\pi y) \cos(2\pi z) \\ a_3 &= 2 + \cos(2\pi x) \cos(2\pi y) \sin(2\pi z) \\ a_4 &= \sin(4\pi x), \quad a_5 = \sin(4\pi y), \quad a_6 = \sin(4\pi z) \\ a_7 &= \sin(2\pi x) \sin(2\pi y) \sin(2\pi z) \end{aligned}$$

であり, 右辺 $g(x, y, z)$ と境界条件は厳密解が

$$u(x, y, z) = \sin(2\pi x) \cos(2\pi y) \sin(2\pi z)$$

となるように設定する. 領域 Ω をグリッド数 50^3 で区切り7点中心差分によって離散化すると, 次元が125000である連立1次方程式を得ることになる. 数値例1と同様にして, 最初に係数 $Dh = 2^{-2}$ の GMRES(10, 20) 法について γ を変化させて残差ノルム

表 6 数値例 2 の計算結果 (time : 計算時間 (秒), iter : 反復回数)

Table 6 The numerical results for example 2 (time: computation time (sec), iter: iterations).

算法	Dh							
	2^{-3}		2^{-2}		2^{-1}		2^0	
	time	iter	time	iter	time	iter	time	iter
GMRES(10)	261.0	880	258.0	880	306.0	1000	(...) [†]	(...) [†]
GMRES(20)	198.0	460	190.0	440	238.0	540	560.0	1300
GMRES(30)	249.0	450	268.0	480	336.0	570	379.0	660
GMRES(40)	298.0	440	269.0	400	301.0	440	395.0	560
GMRES(50)	320.0	400	378.0	450	375.0	450	374.0	450
GMRES(60)	389.0	420	390.0	420	394.0	420	460.0	480
GMRES(10, 20) + eq(5)	175.0	420	169.0	400	215.0	510	280.0	700
GMRES(10, 20) + eq(12)	166.0	420	161.0	400	204.0	510	(69.0)*	(170)*
GMRES(10, 20)	165.0	420	161.0	400	204.0	510	279.0	700
GMRES(10, 30) + eq(5)	176.0	420	169.0	400	225.0	490	260.0	600
GMRES(10, 30) + eq(12)	167.0	420	159.0	400	217.0	490	(161.0)*	(360)*
GMRES(10, 30)	168.0	420	159.0	400	217.0	490	256.0	600
GMRES(10, 40) + eq(5)	176.0	420	170.0	400	225.0	490	260.0	600
GMRES(10, 40) + eq(12)	167.0	420	159.0	400	215.0	490	(162.0)*	(360)*
GMRES(10, 40)	167.0	420	161.0	400	214.0	490	256.0	600
GMRES(10, 50) + eq(5)	177.0	420	170.0	400	225.0	490	259.0	600
GMRES(10, 50) + eq(12)	168.0	420	160.0	400	216.0	490	(162.0)*	(360)*
GMRES(10, 50)	167.0	420	161.0	400	216.0	490	256.0	600
GMRES(10, 60) + eq(5)	178.0	420	169.0	400	224.0	490	260.0	600
GMRES(10, 60) + eq(12)	167.0	420	159.0	400	215.0	490	(161.0)*	(360)*
GMRES(10, 60)	167.0	420	161.0	400	215.0	490	255.0	600
GMRES(20, 40) + eq(5)	205.0	320	205.0	320	251.0	380	330.0	520
GMRES(20, 40) + eq(12)	199.0	320	198.0	320	239.0	380	325.0	520
GMRES(20, 40)	199.0	320	199.0	320	240.0	380	324.0	520
GMRES(20, 60) + eq(5)	205.0	300	207.0	300	248.0	380	331.0	520
GMRES(20, 60) + eq(12)	198.0	300	200.0	300	237.0	380	325.0	520
GMRES(20, 60)	198.0	300	199.0	300	236.0	380	325.0	520
((GMRES(m_{min} , m_{max}) 法で ζ_m を計算する数式の使われた回数))								
	式 (5)	式 (12)	式 (5)	式 (12)	式 (5)	式 (12)	式 (5)	式 (12)
GMRES(10, 20)	0	31	0	29	0	37	2	50
GMRES(10, 30)	0	31	0	29	0	32	3	38
GMRES(10, 40)	0	31	0	29	0	32	3	38
GMRES(10, 50)	0	31	0	29	0	32	3	38
GMRES(10, 60)	0	31	0	29	0	32	3	38
GMRES(20, 40)	0	12	0	12	0	14	0	19
GMRES(20, 60)	0	12	0	12	0	14	0	19

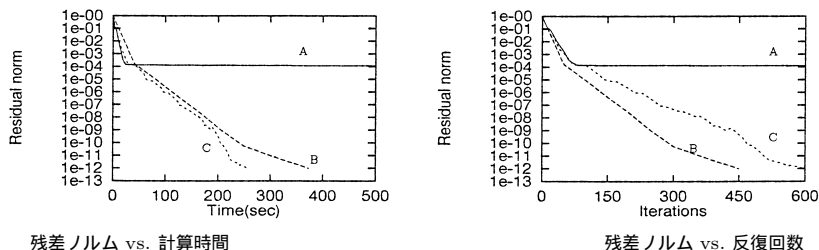
(...)[†] : 30 分の計算時間で残差ノルムが収束しなかった場合
 (数値)* : 途中でブレイクダウンを起こしたときの計算時間および反復回数

の収束条件を満たすまでの計算時間を計測した。ただし、 h はグリッド間の幅で $h = 1/51$ である。表 5 の計測結果によると、この問題でも数値例 1 と同様、 $\gamma = 10.0$ のときに計算時間が最も短縮されている。したがって、この事実をふまえて数値例 2 でも $\gamma = 10.0$ と設定した。次に、係数 Dh を変化させて残差ノルムが収束条件を満たすまでに要した計算時間と反復回数を表 6 に示す。係数 $Dh = 2^0$ のときの一例として、図 4 に GMRES(10, 50) 法を GMRES(10) 法、GMRES(50) 法と比較したときの残差ノルムの収束履歴を示す。GMRES(10, 50) 法は GMRES(50) 法と比べて反復回数は余分にかかっているものの計算時間は 4 割程度減少している。したがって、 γ の値は係数 $Dh = 2^{-2}$ のときを基準にして最適化されているにもかかわらず、係数 $Dh = 2^0$ のときでも効果的に作用していることが推測される。 ζ_m の計算に関しては、数値例 2 でも式 (5) のみを使った場合に比べて、式 (5) と式 (12) の両方を使用した場合は最大で 1 割程度計算

時間が短縮されている。それに対して、式 (12) のみを使った場合には、係数 $Dh = 2^0$ のときにブレイクダウンが発生した。この問題に関しては、GMRES(10, m_{max}) 法ではリスタート周期は最大で 30 までしか変化しなかった。係数 Dh が 2^{-3} から 2^{-1} のとき、GMRES(10, 20) 法は計算時間は GMRES(20) 法と比べて 2 割程度減少しており、GMRES(20, 40) 法と GMRES(20, 60) 法での計算時間は GMRES(20) 法とほぼ同じ時間であった。一方、係数 Dh が 2^0 のとき、GMRES(10, m_{max}) 法では 256 秒から 280 秒程度で残差ノルムはゼロに収束している。このとき GMRES(10) 法では 30 分計算しても残差ノルムはゼロに収束していない。

6. ま と め

本稿では GMRES(m) 法における残差ノルムの収束の停滞を判定するパラメータの 2 つの計算式について考察し、さらにこれらの計算式を用いてリスタート周



残差ノルム vs. 計算時間

残差ノルム vs. 反復回数

図4 数値例2: 残差ノルムの収束履歴 ($Dh = 2^0$), A: GMRES(10), B: GMRES(50), C: GMRES(10, 50)

Fig. 4 Example 2: The convergence behavior of the residual norm ($Dh = 2^0$), A: GMRES(10), B: GMRES(50), C: GMRES(10, 50).

期を適応的に切り替える GMRES(m_{min}, m_{max}) 法を提案した。残差ノルムの収束停滞を判定する式 (5) と式 (12) では、後者のほうが計算量は少ないが、数値例 1 と 2 で示すように反復の途中でブレイクダウンを起こすケースがみられた。それに対して、前者の計算式ではブレイクダウンを起こすことはありえないが、内積計算を新たに行う必要があるので式 (12) よりも計算量を多く必要とする。しかし、この 2 つの計算式を必要に応じて使い分けるとブレイクダウンを起こすことがなく、さらに式 (5) のみを使用した場合に比べて計算時間を 1 割程度短縮することが数値例 1 と 2 の数値実験から確認された。したがって、本稿で提案した GMRES(m_{min}, m_{max}) 法を使用することで、GMRES(m) 法における残差ノルムの収束を加速させ、さらに残差ノルムの収束判定を行う数式の内積計算に要するオーバーヘッドを削減することが可能である。

今後の課題は、パラメータ γ を理論的に設定できる算法を提案することである。

参考文献

- 1) Arnoldi, W.: The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem, *Quart. Appl. Math.*, Vol.9, pp.17–29 (1951).
- 2) Saad, Y. and Schultz, M.K.: GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, No.7, pp.856–869 (1986).
- 3) Schönauer, W.: *Scientific Computing on Vector Computers*, North Holland (1987).
- 4) Joubert, W.: Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations, *SIAM J. Matrix. Anal. Appl.*, Vol.13, No.3, pp.928–943 (1992).

- 5) Saad, Y.: *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company (1996).
- 6) 津野, 野寺: 適応的なリスタートを用いた ORTHOMIN(k) 法, 情報処理学会論文誌, Vol.40, No.3, pp.1350–1353 (1999).
- 7) Moriya, K. and Nodera, T.: The DEFLATED-GMRES(m, k) Method with Switching the Restart Frequency Dynamically, *Numer. Linear. Algebra. Appl.*, Vol.7, pp.569–584 (2000).

(平成 13 年 8 月 13 日受付)

(平成 14 年 4 月 16 日採録)



森屋健太郎 (正会員)

1999 年 3 月慶應義塾大学大学院理工学研究科数理科学専攻修士課程修了。同年富士通株式会社に入社し現在に至る。2000 年 9 月慶應義塾大学大学院理工学研究科後期博士課程 (基礎理工学専攻) に入学。2002 年 3 月同研究科後期博士課程修了。工学博士。計算機の性能評価および並列計算機による大規模計算に興味を持つ。



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。