

適応的なリスタートを用いた疑似残差法

稲 津 隆 敏[†] 野 寺 隆[†]

疑似残差法は、大規模な非対称疎行列を係数行列とする連立一次方程式の反復解法のひとつである。このアルゴリズムには様々なバリエーションがあり、厳密版、再出発版、打切版、結合版の4種類に大別できる。しかし、厳密版はそれまでに計算されたすべての残差ベクトルを記憶する必要があり、大規模な問題では実行不可能である。また再出発版や打切版は、問題によって収束性にばらつきがある。結合版は、打切版をある時点でリスタートさせる方法であり、打切版の収束の速さと再出発版の計算誤差蓄積の少なさを性質として受け継ぐので、収束性は安定している。しかし、結合版のうち一定の周期でリスタートするものは、再出発版の悪い性質も受け継いでいる。そこで、反復の途中で打切版部分の収束性が悪いと推定される場合にのみリスタートする適応的な結合版を提案するとともに、最後に並列計算機 AP1000 を用いた数値実験結果を報告する。

An Adaptive Restarting Procedure for Pseudo-Residual Methods

TAKATOSHI INADU[†] and TAKASHI NODERA[†]

This paper is concerned with PRES (pseudo-residual) methods, one of the iterative methods for solving large and sparse nonsymmetric linear systems of equations. The main variants of PRES methods are exact, restarted, truncated, and combined methods. Unfortunately, in the exact methods, the storage requirements and the computational work increase with the iteration. The restarted and truncated methods also have the unstable convergence in some practical problems. It is generally to say that the combined methods can not breakdown, it is stable, because these procedures based on the restarting process of the truncated methods. However, the combined methods that restart periodically represent bad convergence behavior in some case. The possibility of the restarting technique may be exploited to develop efficient algorithm and to enhance robustness. In this paper, we propose the new combined algorithm, which is called the adaptive combined PRES method, to monitor the convergence behavior of residual. This adaptive procedure is attractive because it only has the restarting process. At last, we present some numerical experiments to illustrate the above fact using the parallel computer AP1000.

1. はじめに

現在、数理解物理学や工学の分野では、流体の動きや固体の変形などの解析に、計算機を使った数値モデルの定量的な解析手法を用いることが多い。その場合、与えられた方程式を有限差分法や有限要素法を用いて離散化すると、大規模で疎な係数行列を持つ連立一次方程式

$$Ax = b \quad (1.1)$$

が得られ、この方程式の近似解 x を求めることに多くの計算時間をとられることになる。通常、連立一次方程式 (1.1) の解法には、ガウスの消去法に代表される

直接法と共役勾配法のような反復法が考えられるが、係数行列の性質などを考慮すると、このような問題では反復法が用いられることが多い。しかし、係数行列 A が非対称行列になる場合には、現在においても決定的な解法と呼べるアルゴリズムは提案されていない。本稿では、方程式の離散近似によって得られる大規模で疎な非対称正則行列を係数とする連立一次方程式の反復解法のひとつである PRES 法^{2),3)} (Pseudo-Residual Method, 疑似残差法とも呼ばれる) について考える。

方程式 (1.1) の解は、次の式 (1.2) を最小化するものとも考えられる。

$$\frac{1}{2}(Ax - b)^T A^{-1}(Ax - b) \quad (1.2)$$

実際、式 (1.2) を x で微分すると、勾配は $Ax - b$ と

[†] 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

```

(1) choose an arbitrary  $x_0$ ,
    compute  $r_0 = Ax_0 - b$ ,
     $P$ : right-hand preconditioner.
(2) for  $k = 0, 1, 2, \dots$ 
    compute  $x_{k+1}$  and  $r_{k+1}$  so that
         $x_{k+1} \in x_0 + K_k(PA, Pr_0)$ ,
         $r_{k+1} = Ax_{k+1} - b$ ,
    with
         $r_{k+1}^T Z r_{k+1-i} = 0 (i=1, \dots, \sigma_k)$ , (2.1)
    where  $Z$  is an auxiliary, regular
    matrix, or with
         $\|r_{k+1}\|_Z =$ 
         $\min_{\mu_{k+1-\sigma_k}, \dots, \mu_k} \|B_{k+1}(AP, \sigma_k)\|_Z$ , (2.2)
    where
         $B_{k+1}(AP, \sigma_k) =$ 
         $r_{k+1-\sigma_k} + \sum_{i=1}^{\sigma_k} \mu_{k+1-i} AP r_{k+1-i}$ ,
    if residual norm converges, es-
    cape the loop.
endfor
    
```

図1 一般共役勾配法のアルゴリズム
Fig. 1 Algorithm of GCG Method.

なる。Hestenes と Stiefel¹⁾ によって開発された CG 法 (Conjugate Gradient Method, 共役勾配法とも呼ばれる) は、係数行列 A が正定値対称な連立一次方程式の反復解法である。ここでは、行列 A を非対称としているので、CG 法の考え方を非対称行列系へ拡張した GCG 法^{2),3),5),6),10)} (Generalized Conjugate Gradient Method, 一般共役勾配法とも呼ばれる) が有効となる。

PRES 法は現在までに提案された様々な GCG 法の中の 1 つのアルゴリズムであり、いろいろなバリエーションが考えられる。ここでは前処理を行わない 3 種類の PRES 法の残差ノルムの収束性に関する数値実験、考察を行うのだが、PRES 法の収束性は問題に強く依存する。そこで、より安定した収束性が得られる適応的なリスタート技法を用いた新しい PRES 法を提案する。

2. PRES 法

PRES 法 (疑似残差法) は GCG 法 (一般共役勾配法) の中の 1 つのアルゴリズムである。よって、PRES 法の詳細について述べる前に、GCG 法について考える。

表1 一般共役勾配法の分類
Table 1 Family of GCG Methods.

アルゴリズム (algorithm)	σ_k
厳密版 (exact method)	$k + 1$
再出発版 (restarted method)	$(k \bmod \sigma_{res}) + 1$
打切版 (truncated method)	$\min(k + 1, \sigma_{max})$
結合版 (combined method)	— (注)

(注) 打切版がある時点で再出発する方法なので、ここでは σ_k は特定できない。

2.1 GCG 法の定義と分類

図1 で定義されるアルゴリズムを、GCG 法という。通常、 σ_k は GCG 法のオーダーと呼ばれ、 σ_k の選び方により GCG 法は表1 のように分類できる。

右側前処理行列を P として、 $\tilde{A} = AP$ とすると、

$$r_{k+1} \in K_{k+1}(\tilde{A}, r_0) \tag{2.3}$$

となる。ただし、 $K_{k+1}(\tilde{A}, r_0) = \text{Span}(r_0, \tilde{A}r_0, \tilde{A}^2r_0, \dots, \tilde{A}^k r_0)$ をクリロフ部分空間とする。よって、GCG 法はクリロフ部分空間法⁴⁾ のひとつであることが分かるので、残差 r_{k+1} と誤差 e_{k+1} は次式のように表せる。

$$r_{k+1} = \prod_1^{k+1} (\tilde{A}) r_0 = \sum_{i=1}^{k+1} \theta_{i,k+1} \tilde{A}^i r_0 + r_0 \tag{2.4}$$

$$e_{k+1} = \prod_1^{k+1} (PA) e_0 = \sum_{i=1}^{k+1} \theta_{i,k+1} (PA)^i e_0 + e_0 \tag{2.5}$$

ただし、

$$\prod_1^{k+1} (\tilde{A}) = \sum_{i=1}^{k+1} \theta_{i,k+1} \tilde{A}^i + I \tag{2.6}$$

であり、係数 $\theta_{i,k+1}$ ($i = 1, \dots, k + 1$) は任意の値をとるものとする。

2.2 PRES 法の定義と分類

図2 で定義されるアルゴリズムを PRES 法という。通常、 σ_k は PRES 法のオーダーと呼ばれる。式 (2.7) は疑似残差の Z -ノルム $\|\tilde{r}_{k+1}\|_Z$ が各反復で最小になるよう係数 $\alpha_{i,k}$ を定めているので、この方法は疑似残差法と呼ばれる。ただし、 Z は適当な正定値対称行列であり、 $\|\tilde{r}_{k+1}\|_Z^2 = \tilde{r}_{k+1}^T Z \tilde{r}_{k+1}$ とする。この $\|\tilde{r}_{k+1}\|_Z$ の最小化は、GCG 法のアルゴリズム中の式 (2.1) と同値である。よって、PRES 法は GCG 法のバリエーションのひとつである。

(1) choose an arbitrary x_0 ,
compute $r_0 = Ax_0 - b$,
 Z : an auxiliary symmetric,
positive definite matrix.

(2) for $k = 0, 1, 2, \dots$
 P_k : right-hand preconditioner

$$d_k = P_k r_k,$$

$$\alpha_{i,k} = -\frac{r_{k+1-i}^T Z A d_k}{r_{k+1-i}^T Z r_{k+1-i}} \quad (2.7)$$

for $i = 1, \dots, \sigma_k$,

$$\bar{r}_{k+1} = A d_k + \sum_{i=1}^{\sigma_k} \alpha_{i,k} r_{k+1-i},$$

(\bar{r}_{k+1} : pseudo-residual)

$$\phi_k = \frac{1}{\sigma_k \sum_{i=1}^{\sigma_k} \alpha_{i,k}},$$

$$r_{k+1} = \phi_k \bar{r}_{k+1},$$

$$x_{k+1} = \phi_k (d_k + \sum_{i=1}^{\sigma_k} \alpha_{i,k} x_{k+1-i}),$$

if $\|r_{k+1}\|_Z$ converges, escape the loop.
endfor

図2 疑似残差法のアルゴリズム

Fig. 2 Algorithm of PRES Method.

PRES 法には右側前処理行列 P_k , 適当な正定値対称行列 Z , オーダ σ_k の選び方により, 表2のようなバリエーションが考えられる. ただし, 表2にあるアルゴリズムがすべてではない. 以下, $P = I$ として前処理をしない4種類のPRES法のいくつかの性質について考察する.

2.2.1 ORTHORES-E

これは右側前処理をしない厳密版である. 各反復で得られる新たな残差ベクトル r_{k+1} が r_i ($i = 0, \dots, k$) に対して Z -直交している ($r_{k+1}^T Z r_i = 0$). よって, 理論的には係数行列 A の次元の回数 (n 回) で必ず収束する. しかし, そのためには残差ベクトルを n 本記憶しておかなければならず, 相当なメモリ量が必要となる. 特に大規模な問題の場合には, このアルゴリズムの実行は不可能である. また, 計算誤差の蓄積が多い. ただし, 厳密版の場合, 残差ノルムの収束性に関して次のような解析が可能である.

右側前処理を含めたすべての厳密版において, 右側前処理行列 P_k をある行列 P に固定し, $\tilde{A} = AP$ とする. PRES 法では, 各反復において $\|\bar{r}_{k+1}\|_Z$ を最

表2 様々な疑似残差法

Table 2 Family of PRES Methods.

アルゴリズム (algorithm)	σ_k	Z	P_k
ORTHORES-E	$k+1$	I	I
ORTHORES-R (σ_{res})	$(k \bmod \sigma_{\text{res}}) + 1$	I	I
ORTHORES-T (σ_{max})	$\min(k+1, \sigma_{\text{max}})$	I	I
ORTHORES-C ($\sigma_{\text{max}}, \sigma_{\text{res}}$)	—	I	I
ATPRES (CGNE)	2	I	A^T

小化している, 真の残差ベクトル r_{k+1} は Z -ノルムについて最小化されることはない. では, 各反復で r_{k+1} はどのような性質を持つのであろうか.

すべての厳密版において, 次の式 (2.8) が成り立つ³⁾.

$$\|r_{k+1}\|_{Z\tilde{A}^{-1}}^2 = r_{k+1}^T Z \tilde{A}^{-1} \prod_1^{k+1} (\tilde{A}) r_0 \quad (2.8)$$

ただし,

$$\prod_1^{k+1} (\tilde{A}) = \sum_{i=1}^{k+1} \theta_i \tilde{A}^i + I \quad (2.9)$$

であり, 係数 θ_i ($i = 1, \dots, k+1$) は任意の値をとるものとする. さらに, A を正定値対称行列とすると, 次の不等式 (2.10) が成り立つ²⁾.

$$|x^T A y| \leq \|x\|_A \|y\|_A \quad (2.10)$$

前述の式 (2.8) と式 (2.10) より, AZ^{-1} が正定値対称行列であるとすると, 次式が厳密版において成り立つ.

$$\|r_{k+1}\|_{Z\tilde{A}^{-1}} = \min_{\theta_1, \dots, \theta_{k+1}} \left\| \prod_1^{k+1} (\tilde{A}) r_0 \right\|_{Z\tilde{A}^{-1}} \quad (2.11)$$

よって, 厳密版では r_{k+1} は $Z\tilde{A}^{-1}$ -ノルムについて最小化され, Z -ノルムについては最小化されない. 後述の数値実験で $\|r_{k+1}\|_Z$ は各反復で激しく振動することが分かる.

前述の式 (2.11) において $Z = P = I$ (単位行列) とすると,

$$\begin{aligned} \|r_{k+1}\|_{A^{-1}} &= r_{k+1}^T A^{-1} r_{k+1} \\ &= \min_{\theta_1, \dots, \theta_{k+1}} \left\| \prod_1^{k+1} (A) r_0 \right\|_{A^{-1}} \end{aligned} \quad (2.12)$$

となり, 式 (2.12) は式 (1.2) の最小化と同値になる. よって, PRES 法は基本的に各反復で誤差ベクトル e_{k+1} を Z -ノルムについて最小化する方法である. “基本的に” と述べたのは, $\tilde{A}Z^{-1}$ が正定値対称行列である場合の厳密版においてのみ, これが証明可能であるからである²⁾.

2.2.2 ORTHORES-R (σ_{res})

このアルゴリズムは、右側前処理をしない再出発版である。 σ_k は最大 σ_{res} であり、 σ_{res} 回ごとに残差ベクトル r_{k+1} を

$$r_{k+1} = Ax_{k+1} - b \quad (2.13)$$

によって計算し、 x_{k+1} を新たな初期近似解ベクトル、さらに r_{k+1} を新たな初期残差ベクトルとして反復を進める。この操作を“リスタート (restart, 再出発)”と呼ぶ。よって、計算誤差の蓄積は少なく、計算に必要なメモリ量も少ない。ただし厳密版と異なり、 n 回の反復で必ず収束する保証はない。

2.2.3 ORTHORES-T (σ_{max})

このアルゴリズムは、右側前処理をしない打切版である。 σ_k は最大 σ_{max} であり、 $k \geq \sigma_{max} - 1$ において最新の σ_{max} 本の残差ベクトルが新たな残差ベクトル r_{k+1} の計算に用いられる。計算に必要なメモリ量は少ないが、その反面、厳密版のように n 回の反復で収束する保証はない。また、計算誤差の蓄積が多くなる可能性も高い。打切版は再出発版と異なり、 $k \geq \sigma_{max} - 1$ においてつねに σ_{max} 本の残差ベクトルを用いて $\|r_{k+1}\|_Z$ を最小化しているので、発散や振動現象を起こさなければ、再出発版より反復回数が少ない場合が多い。

2.2.4 ORTHORES-C ($\sigma_{max}, \sigma_{res}$)

このアルゴリズムは、ORTHORES-T (σ_{max}) を σ_{res} 回ごとにリスタートする結合版のひとつであり、Weiss 2) には例として PRES20 があげられている。PRES20 は、 $\sigma_{max} = 5$, $\sigma_{res} = 20$ として、PRES 法にスムージング^{3),7)} を適用して得られる残差ベクトル列を利用し、反復を進める方法である。

野寺・稲津¹¹⁾ は、結合版が打切版の速い収束性に加え、再出発版の計算誤差の蓄積の少なさを性質として受け継ぐ良い方法であることを示した。そこで、後述の数値実験において、稲津・野寺⁸⁾ で提案された残差ノルムが安定した収束性を示す ORTHORES-C (5,50) の性質について報告する。しかし、ORTHORES-C ($\sigma_{max}, \sigma_{res}$) には次の章で述べるような問題点も存在する。

2.3 ORTHORES-C ($\sigma_{max}, \sigma_{res}$) の問題点

ORTHORES-C ($\sigma_{max}, \sigma_{res}$) は σ_{res} 回ごとにリスタートするので、 σ_{res} の選び方により悪い収束性を示す場合がある (実験 3.1, 3.2 参照)。これは、ORTHORES-C ($\sigma_{max}, \sigma_{res}$) の打切版部分においてよい収束性を示している場合でも、 σ_{res} 回の周期で強制的にリスタートが発生することが原因とみられる。よって、打切版部分の収束性が悪くなったときに限りリスタートする、

適応的な結合版が必要となる。

本稿では、この適応的なリスタートを行う結合版を ORTHORES-AC (σ_{max}) (Adaptive Combined Method, 適応的な結合版) と呼び、次の章で新たなリスタートのタイミングについて提案する。

3. ORTHORES-AC (σ_{max})

ORTHORES-T (σ_{max}) を繰り返し、一定の条件が満たされた場合にリスタートする前処理なしの PRES 法を、ORTHORES-AC (σ_{max}) と呼ぶことにする。新しい適応的なリスタート技法を提案する前に、様々な PRES 法の性質を調べるため、次のような数値実験を行った。ただし、すべての数値実験では $Z = P = I$ とした。

問題 1 (2次元楕円型偏微分方程式の境界値問題)

矩形領域 $\Omega = [0, 1] \times [0, 1]$ における 2 次元楕円型偏微分方程式のディリクレ境界条件問題

$$\begin{aligned} u_{xx} + u_{yy} + au_x + bu_y &= f(x, y) \\ u(x, y)|_{\partial\Omega} &= 0 \end{aligned}$$

に対して真の解を

$$u(x, y) = x(1-x)y(1-y)$$

と設定し、右辺を決定する。ただし、 a, b は実数係数であり、これらの値を変化させることで係数行列 A も変化することになる。この方程式を 5 点中心差分を用いて離散化し、数値実験を行った。メッシュは 256×256 の等間隔格子を用い、格子点への番号付けは整合順序を用いるものとする。

問題 2 (ブロック対角行列)

2×2 の小行列 A_j を

$$A_j = \begin{pmatrix} \lambda_{re,j} & \lambda_{im,j} \\ -\lambda_{im,j} & \lambda_{re,j} \end{pmatrix}$$

とし、係数行列 A を

$$A = \begin{pmatrix} A_1 & & & & \\ & A_2 & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & A_m \end{pmatrix} \in \mathbf{R}^{2m \times 2m}$$

とする。この行列 A の固有値は、 $j = 1, \dots, m$ に対して

$$\lambda_{j1} = \lambda_{re,j} + i \lambda_{im,j}, \quad \lambda_{j2} = \lambda_{re,j} - i \lambda_{im,j}$$

となる。これに対し、解ベクトルを $x = (1, \dots, 1)^T$ として、右辺を決定する。ただし、 $\lambda_{re,j}$ および $\lambda_{im,j}$ はある範囲 Λ_{re} および $\Lambda_{im} (\subset \mathbf{R})$ 内でランダムに決定するものとする。

各数値実験では、PRES 法の初期近似解を零とし、

表3 AP1000によるPRES法の残差ノルム $\|r_{k+1}\|/\|r_0\|$ の収束特性 (問題1)
 Table 3 Convergence property of residual norms $\|r_{k+1}\|/\|r_0\|$ for the PRES Methods using AP1000 (Problem 1).

係数		実験 3.1						実験 3.3				
		T (5)		R (5)			C (5,50)			AC (5)		
a	b	秒	反復回数	秒	反復回数	再出発回数	秒	反復回数	再出発回数	秒	反復回数	再出発回数
3	50	—	発散	—	1e-11	600	124.5	1140	22	119.2	1085	11
30	5	—	7e+01	—	1e-03	600	175.8	1611	32	157.6	1433	9
3	500	59.6	537	47.7	491	98	53.4	487	9	56.0	511	7
300	5	64.7	584	48.5	499	99	57.0	520	10	60.0	547	6
3	5000	54.0	487	—	発散	—	64.8	591	11	62.1	570	13
3000	5	50.2	453	89.5	919	183	54.8	499	9	52.3	482	13
3	5	—	3e-01	—	7e-01	600	—	2e-10	60	276.8	2500	4
30	50	158.0	1423	—	9e-12	600	129.0	1177	23	117.0	1071	20
300	500	61.0	550	53.3	548	109	60.5	552	11	59.0	557	33
3000	5000	78.8	711	—	発散	—	79.8	726	14	88.8	814	16

(注) 3000回以内で収束しなかったものに関しては、反復回数の欄に最終残差ノルムの値を記述した。

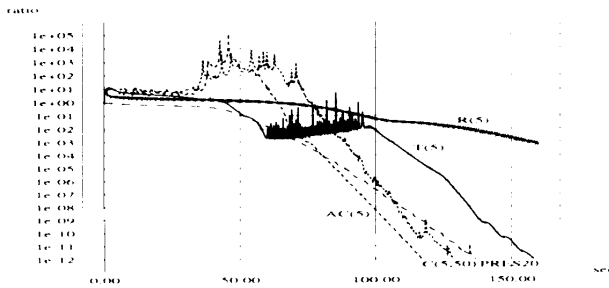


図3 PRES法の残差ノルム $\|r_{k+1}\|/\|r_0\|$ の収束の様子 (問題1: $a = 30, b = 50$)

Fig. 3 Convergence behavior of residual norms $\|r_{k+1}\|/\|r_0\|$ v.s. computational time for various PRES Methods (Problem 1: $a = 30, b = 50$).

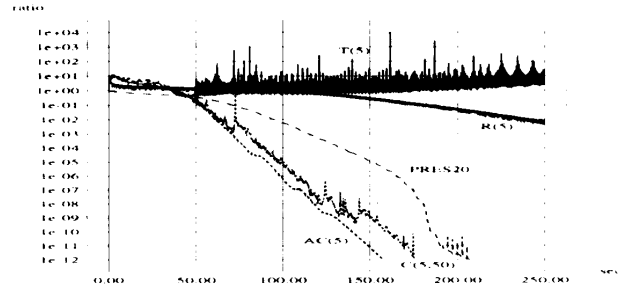


図4 PRES法の残差ノルム $\|r_{k+1}\|/\|r_0\|$ の収束の様子 (問題1: $a = 30, b = 5$)

Fig. 4 Convergence behavior of residual norms $\|r_{k+1}\|/\|r_0\|$ v.s. computational time for various PRES Methods (Problem 1: $a = 30, b = 5$).

残差ノルムの収束条件を

$$\frac{\|r_{k+1}\|}{\|r_0\|} \leq 1.0 \times 10^{-12}$$

として、64個のセルを持つ並列計算機 AP1000 (富士通) を用いて倍精度計算を行った。最大反復回数は3000回として、3000回の反復で収束しきらなかったアルゴリズムについては、数値実験結果を示す表の反復回数の欄に $\|r_{3000}\|/\|r_0\|$ の値を記入した。この場合、たとえば“6e-03”は 6×10^{-3} を示す。

また、適切なリスタート基準を考えるため、各反復における ϕ_k および $\alpha_{i,k}$ の値を観測した。

実験 3.1 いろいろな PRES 法 (問題 1)

問題1について、 a, b の値を変化させ、再出発版、打切版、結合版の3つのPRES法を利用して解を求めた。この数値実験の結果を表3に示す。また、 $a = 30, b = 50$ と $a = 30, b = 5$ の場合の残差ノルムの収束の様子を図3と図4に示す。これらの結果から、再出発版と打切版は収束性が大変不安定であることが分かる。それに対し結合版は、安定した収束性を示している。

実験 3.2 いろいろな PRES 法 (問題 2)

問題2について、 $m = 2048, \Lambda_{re} = \{1.0\}, \Lambda_{im} =$

表4 PRES法の残差ノルム $\|r_{k+1}\|/\|r_0\|$ の収束特性 (問題2)
 Table 4 Convergence property of residual norms $\|r_{k+1}\|/\|r_0\|$ for the PRES Methods using AP1000 (Problem 2).

ORTHORES-	実験 3.2			実験 3.4
	T (5)	R (5)	C (5,50)	AC (5)
時間 (秒)	40.0	—	—	39.9
反復回数	2438	発散	6×10^{-6}	2429
再出発	0	—	—	1

(注) 3000回以内で収束しなかったものに関しては、反復回数の欄に最終残差ノルムの値を記述した。

$[-100, 100]$ として、再出発版、打切版、結合版の3つのPRES法でを用いて解を求め、数値実験の結果を表4に、残差ノルムの収束の様子を図5に示す。この数値実験の場合、打切版が最も良い収束性を示している。結合版は、リスタートがうまく機能せず、打切版より悪い収束性を示している。このような例が存在することから、最適なリスタートのタイミングを決定することが重要なポイントとなる。

図3, 図4, 図5より、ORTHORES-T (σ_{max}) の収束性に関して次の予想をたてることができる。

予想 3.1 ORTHORES-T (σ_{max}) は、 σ_{max} 回

表5 ϕ_k と $\alpha_{i,k}$ の観測値の例 (問題1: $a = 30, b = 5, \text{ORTHORES-T (5)}$)
 Table 5 Observation of ϕ_k and $\alpha_{i,k}$ (Problem 1: $a = 30, b = 5, \text{ORTHORES-T (5)}$).

k	ϕ_k	$\alpha_{1,k}$	$\alpha_{2,k}$	$\alpha_{3,k}$	$\alpha_{4,k}$	$\alpha_{5,k}$	$\ r_{k+1}\ /\ r_0\ $
0	-1.7610e+02	-5.7e-03	+0.0e+00	+0.0e+00	+0.0e+00	+0.0e+00	6.63e+00
1	-3.9042e+00	-5.0e-01	+2.4e-01	+0.0e+00	+0.0e+00	+0.0e+00	6.57e+00
2	-3.8950e+00	-5.0e-01	+2.4e-01	-2.0e-04	+0.0e+00	+0.0e+00	6.54e+00
3	-3.8867e+00	-5.0e-01	+2.4e-01	-1.3e-05	-2.3e-04	+0.0e+00	6.53e+00
4	-3.8792e+00	-5.0e-01	+2.4e-01	-6.5e-06	-1.1e-05	-2.5e-04	6.53e+00
5	-3.8764e+00	-5.0e-01	+2.4e-01	+3.2e-07	-3.3e-06	-1.1e-05	6.56e+00
6	-3.8737e+00	-5.0e-01	+2.4e-01	+6.5e-06	+4.2e-06	-3.3e-06	6.62e+00
7	-3.8593e+00	-5.0e-01	+2.4e-01	+8.1e-06	+1.2e-05	+3.7e-06	6.72e+00
8	-3.6211e+00	-5.2e-01	+2.5e-01	-6.8e-05	+3.3e-05	+7.1e-06	7.47e+00
9	-1.9648e+00	-8.3e-01	+3.2e-01	-1.7e-03	+3.6e-04	-5.2e-05	8.77e+00
10	-1.8235e+00	-1.2e+00	+6.9e-01	-1.4e-02	+2.8e-03	-5.3e-04	7.33e+00
...
180	-1.9207e+00	-1.0e+00	+4.8e-01	+1.8e-05	-3.6e-03	-1.5e-03	2.66e+00
181	-1.9199e+00	-1.0e+00	+4.8e-01	-1.0e-04	-3.5e-03	-1.3e-03	2.66e+00
182	-1.9214e+00	-1.0e+00	+4.8e-01	+6.0e-06	-3.6e-03	-1.5e-03	2.64e+00
183	-1.9192e+00	-1.0e+00	+4.8e-01	-8.8e-05	-3.5e-03	-1.3e-03	2.64e+00
184	-1.9221e+00 (ϕ_k の安定)	-1.0e+00	+4.8e-01	-5.9e-06	-3.6e-03	-1.5e-03	2.62e+00
...
448	-2.2416e+00	-1.0e+00	+5.5e-01	+9.1e-03	+2.5e-02	+7.5e-03	1.05e+00
449	-2.9277e+00	-9.5e-01	+5.7e-01	+9.0e-03	+2.4e-02	+2.4e-03	1.62e+00
450	-5.6842e+00	-1.0e+00	+7.8e-01	+1.9e-02	+3.8e-02	+3.1e-03	4.40e+00
451	+3.1549e+00 (正の ϕ_k)	-1.1e+00	+1.2e+00	+2.8e-02	+8.6e-02	+2.2e-02	7.06e+00
452	-5.3886e-01	-9.5e-01	-7.8e-01	-2.4e-02	-9.5e-02	-7.5e-03	2.03e+00
453	-1.1319e+00	-1.0e+00	+1.5e-01	-4.5e-03	-2.1e-02	-5.0e-05	1.11e+00
454	-1.2606e+00	-1.0e+00	+2.5e-01	+1.0e-03	-4.3e-03	-1.5e-03	7.18e-01
...

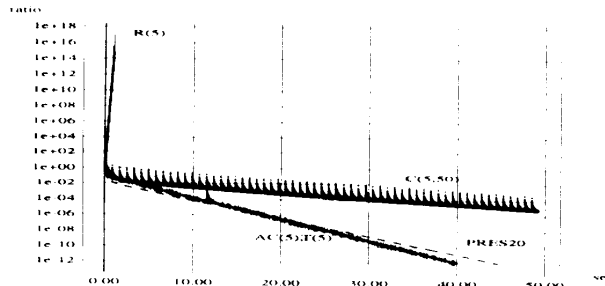


図5 PRES法の残差ノルム $\|r_{k+1}\|/\|r_0\|$ の収束する様子 (問題2)
 Fig. 5 Convergence behavior of residual norms $\|r_{k+1}\|/\|r_0\|$ v.s. computational time for various PRES Methods (Problem 2).

ごとに最小の残差ノルムが更新されていけば、残差ノルムが収束する方向に進んでいる。

ここで我々は、PRES法のアルゴリズム中で得られる実数 ϕ_k と $\alpha_{i,k}$ の値に注目する。各々の数値実験データを観測し (例として表5, 図6を示す), データを分析した結果からさらに次の2つの予想をたてることができる。

予想3.2 PRES法のアルゴリズムにおける ϕ_k の値が正のとき、残差ノルムは飛躍的に増大する。

予想3.3 ORTHORES-T (σ_{\max}) は、 ϕ_k および $\alpha_{i,k}$ がある値に近づいて安定すると、残差ノルムの収束性はその付近で一定となる。逆に、 ϕ_k および $\alpha_{i,k}$ が各反復で不安定な値を示す場合、残差ノルムの収束性が変化する可能性がある。

予想3.3における“残差ノルムの収束性が一定になる”とは、“発散していれば発散、振動していれば振動、収束していれば収束するペースが一定になる”という意味である。

これらの予想をもとに、ORTHORES-AC (σ_{\max}) のアルゴリズムを図7のように提案する。図7に示したアルゴリズムの(3)では、最小の残差ノルムの更新を行っている。もし更新されたならば、そのまま打切版部分を繰り返す。更新されなければ、(5)において ϕ_{k+1-i} ($i = 1, \dots, \sigma_{\max}$) の符号を調べ、正であればリスタートする。正でなければ、今度は ϕ_k と $\alpha_{i,k}$ の安定性を調べる。ただし、 ϕ_k は $\alpha_{i,k}$ から生成されるものなので、このアルゴリズムでは ϕ_k のみを使用する。安定基準は、

$$\frac{V[\phi_{k+1-i}]}{E[\phi_{k+1-i}]^2} < \epsilon_{\text{stab}} \tag{3.1}$$

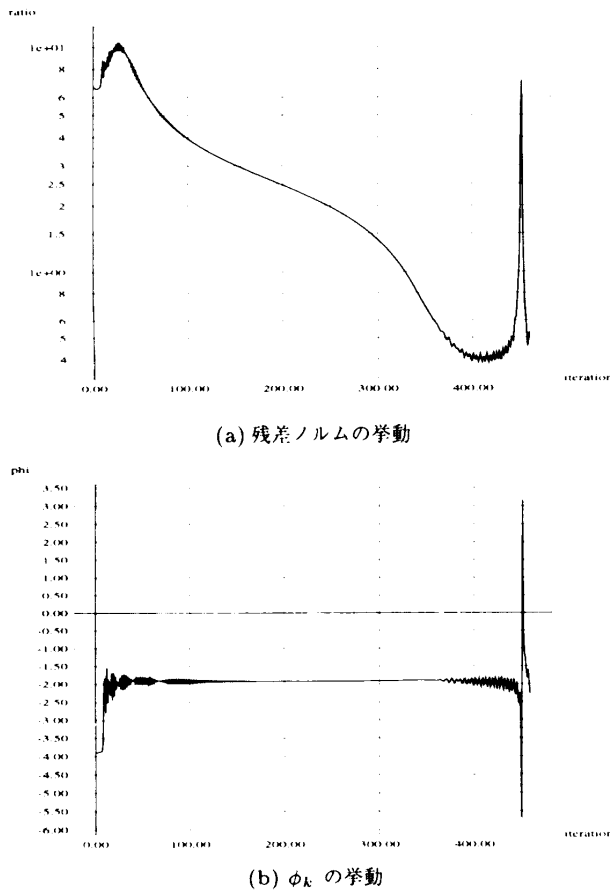


図6 残差ノルム $\|r_{k+1}\| / \|r_0\|$ と ϕ_k の関係
(問題1: $a = 30$, $b = 5$, ORTHORES-T (5))

Fig. 6 Relation between residual norms $\|r_{k+1}\| / \|r_0\|$ and ϕ_k (Problem 1: $a = 30$, $b = 5$, ORTHORES-T (5)).

とする。ただし、 V は分散、 E は算術平均を表し、 $\varepsilon_{\text{stab}}$ は任意に決定できる。この安定基準に達していれば、PRES法のリスタートを行うものとする。

以下に、ORTHORES-AC (σ_{max}) の数値実験結果を報告する。

実験 3.3 ORTHORES-AC (σ_{max}) (問題 1)

実験 3.1 と同じ条件で、ORTHORES-AC (5) を用いて問題 1 の解を求めた。数値実験結果を表 3 に示す。また、 $a = 30$, $b = 50$ と $a = 30$, $b = 5$ の場合の残差ノルムの収束の様子を図 3 と図 4 に示す。ただし、この実験では ORTHORES-AC (5) の ϕ_k の安定基準を

$$\varepsilon_{\text{stab}} = 1.0 \times 10^{-3}$$

とした。この値は、様々な数値実験を行った中で、最も良い結果が得られた値である。これらの結果より、適応的なリスタート技法を用いる ORTHORES-AC (5) は問題 1 に対して安定した収束性を示し、さらに最も収束が速い場合も見受けられる。そこで同じく残差ノ

```

(1) set initial  $\|r_{\min}\| = \|r_0\|$ 
(2) do ORTHORES-T ( $\sigma_{\text{max}}$ )
    for  $\sigma_{\text{max}}$  iterations
      (Truncated Method Part)
(3) minres_updated = FALSE
    for  $i = 0, \dots, \sigma_{\text{max}} - 1$ 
      if ( $\|r_{\min}\| > \|r_{k+1-i}\|$ )
         $\|r_{\min}\| = \|r_{k+1-i}\|$ 
        minres_updated = TRUE
      endif
    endfor
(4) if (minres_updated)
    go to (2)
  else
    go to (5)
  endif
(5) for  $i = 1, \dots, \sigma_{\text{max}}$ 
    if ( $\phi_{k+1-i} > 0$ )
      restart and go to (2)
    endif
  endfor
(6) compute average and variance
    of  $\phi_{k+1-i}$  ( $i = 1, \dots, \sigma_{\text{max}}$ )
(7) if (variance/average2 <  $\varepsilon_{\text{stab}}$ )
    restart and go to (2)
  else
    go to (2)
  endif
endif

```

図7 ORTHORES-AC (σ_{max}) のアルゴリズム
Fig. 7 Algorithm of ORTHORES-AC (σ_{max}).

ルムの収束性が安定している ORTHORES-C (5,50) と比較すると、次のような性質が分かる。

ORTHORES-R (5) の残差ノルムの収束が速い場合、ORTHORES-C (5,50) の収束も速いことが多い。それとは対照的に、ORTHORES-T (5) の残差ノルムが速く収束する場合は、ORTHORES-AC (5) も速く収束することが多い。よって、一定周期ごとに必ずリスタートする結合版は、再出発版の性質を強く受け継ぐ。適応的な結合版は、各反復における打切版部分の残差ノルムの収束状況を観測しながらリスタートの実行を決定するので、打切版の性質を強く受け継ぐ。

実験 3.4 ORTHORES-AC (σ_{max}) (問題 2)

実験 3.2 と同じ条件で、ORTHORES-AC (5) を用いて問題 2 の解を求めた。 ϕ_k の安定基準は実験 3.3 と同じとした。数値実験結果を表 4 と図 5 に示す。この場

合, ORTHORES-AC (5) は1度だけリスタートして, ORTHORES-T (5) とほぼ同じ計算時間で収束した. これは, ORTHORES-AC (σ_{\max}) は ORTHORES-T (σ_{\max}) が収束する方向に進んでいけばリスタートしない方がよい, という考え方の正当性を示す1つの例である.

また, ORTHORES-AC (5) はリスタートの実行を決定するために余分な計算を必要とするが, 反復回数がそれほど変わらない ORTHORES-T (5) とほぼ同じ時間で収束したことを考慮すると, この計算のオーバーヘッドは無視できるくらい小さいことが分かる. これは, ORTHORES-AC (σ_{\max}) で追加されたアルゴリズムにベクトルの加減算, スカラ倍, 内積計算などのベクトル演算がいっさい含まれないことが原因となっている.

4. その他の話題

4.1 演算量の比較

前処理なしの PRES 法でリスタートを行う場合に必要となる演算は, 行列とベクトルの掛算が1回, ベクトルの減算が1回である. 一般的に考えれば, この計算のオーバーヘッドは計算に相当な時間のかかる行列とベクトルの掛算が含まれているので反復法の約1回の反復に相当する.

5点中心差分によって離散化された問題の場合, 係数行列 A の1行あたりの非零要素数は, 境界に接する部分を除き5個である. よって, このような問題の場合, 行列とベクトルの掛算は5回のベクトル演算 (ベクトルの加減算, スカラ倍, 内積計算) に相当する.

ここで, PRES 法においてリスタートする場合としない場合の演算量を比較する. 簡潔にするため, ここでは $\sigma_{\max} = \sigma_{\text{res}} = 5$ とする.

リスタートする場合, 上記の理由から, ベクトル演算6回分が余計にかかる. ただし, リスタートにより次の反復でのオーダ σ_k は1となる.

PRES 法では, オーダ σ_k が1増加すると, $\alpha_{i,k}$ の計算に内積計算1回, \bar{r}_{k+1} と x_{k+1} のそれぞれの計算に, スカラ倍1回, 加算1回, 合計5回のベクトル演算が余計にかかる.

ここで再出発版と打切版の演算量を比較する. 5回の反復における両者の演算量の差は, ベクトル演算を単位に考えると, 再出発版の計算コストが6に対し, $k \geq 4$ においてつねにオーダ5で反復する打切版の計算コストは,

$$5 \times (4 + 3 + 2 + 1) = 50$$

である. よって, 打切版の方が5回の反復につき44

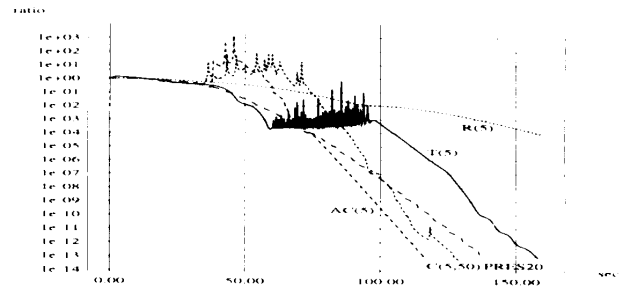


図8 PRES法の誤差ノルム $\|e_{k+1}\| / \|e_0\|$ の収束の様子 (問題1: $a = 30, b = 50$)

Fig. 8 Convergence behavior of error norms $\|e_{k+1}\| / \|e_0\|$ for the PRES Methods (Problem 1: $a = 30, b = 50$).

回ものベクトル演算を再出発版より多く行っていることになる.

実際に表3の中の $a = 300, b = 500$ における両者の計算時間と反復回数を比較すると, 反復回数はほぼ等しいにもかかわらず, 計算時間は打切版の方が約1.2倍かかっている.

よって, 5点中心差分によって離散化された問題において反復回数が等しい場合, 前処理なしの PRES 法はリスタートした方が全体の演算量は少なくなるといえる.

4.2 誤差について

前述の厳密版の解析で述べたように, PRES 法は基本的に各反復で誤差ノルムを最小化するアルゴリズムである. よって数値実験では, 真の解を構成した後に方程式の右辺を計算し, 誤差ノルムの挙動もあわせて観測した. その結果を, 図8に示す.

同じ問題で残差ノルムと誤差ノルムの挙動を見比べると, 残差が激しく振動する部分を除き, 誤差の方がなめらかに収束している. また最終的な値も, 誤差の方がより精度が高く求まることが多い.

4.3 PRES20 との比較

PRES20は, スムージング⁷⁾を用いた ORTHORES-C (5,20) である. Weiss 2) によれば, この PRES20 は結合版の中で最適なアルゴリズムである. そこで, PRES20 を用いて実験3.1と3.2に示した問題の解を求め, その残差ノルムの収束性を図3, 図4のグラフに示す.

これらのグラフからも理解できるように, PRES20 はスムージングの処理に計算コストがかかるため, 時間的観点から見た残差ノルムの収束性は, 他の結合版に比べ劣る.

5. まとめ

本稿では様々な PRES 法の残差ノルムの収束性を

分析し、大規模な連立一次方程式の反復解法として、より安定かつ高速に収束する新しい PRES 法を提案した。

PRES 法の打切版と再出発版は残差ノルムの収束性が大変不安定であり、問題により収束結果が大きく異なる。そこで考えられるのが、打切版をある時点でリスタートさせる結合版である。結合版は、残差ノルムが反復の途中で発散や振動しない場合の打切版の収束の速さと、再出発版の計算誤差蓄積の少なさを性質として受け継ぐ、比較的安定なアルゴリズムといえる。しかし、ORTHORES-C ($\sigma_{\max}, \sigma_{\text{res}}$) や PRES20 などの結合版は周期的にリスタートするので、残差ノルムの収束性が多少悪くなる場合がある。そのような性質は、それらのアルゴリズムの打切版部分において良い収束性を示している場合でも、あるタイミングで強制的にリスタートすることに起因する。

そこで、反復の途中で打切版部分の残差ノルムの収束性を調べることにより、リスタートのタイミングを柔軟に決定する適応的な結合版を提案し、数値実験によりその有効性を示した。ここに示した数値実験から、適応的な結合版は安定かつ高速な収束性を持つことが確認された。

今後の課題としては、

- (1) 3つの予想の証明
 - (2) ϕ_k の安定基準の最適化
 - (3) $\sigma_{\max}, \sigma_{\text{res}}$ の最適化
- などがあげられる。

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) Weiss R.: Convergence Behavior of Generalized Conjugate Gradient Methods, PhD Thesis, University of Karlsruhe (1990).
- 3) Weiss R.: Properties of Generalized Conjugate Gradient Methods, *Numerical Linear Algebra with Applications*, Vol.1, No.1, pp.45-63 (1994).
- 4) Weiss R.: A Theoretical Overview of Krylov Subspace Methods, *Applied Numerical Mathematics*, Vol.19, No.3, pp.207-234 (1995).
- 5) Weiss R.: Minimization Properties and Short

Recurrences for Krylov Subspace Methods, *Electronic Transactions on Numerical Analysis*, pp.57-75 (1994). (ETNA: Kent State University, etna@mcs.kent.edu)

- 6) Schönauer W. and Weiss R.: An Engineering Approach to Generalized Conjugate Gradient Methods and Beyond, *Applied Numerical Mathematics*, Vol.19, No.3, pp.175-206 (1995).
- 7) 稲津, 野寺: 非対称行列系に対する疑似残差法について, *情報研報*, Vol.95, No.56, pp.27-34 (1995).
- 8) 稲津, 野寺: 非対称行列系に対する疑似残差法と前処理について, *情報研報*, Vol.95, No.59, pp.19-24 (1995).
- 9) 稲津, 野寺: 適応的なリスタート技法を用いた疑似残差法, PCW '95 Japan, p.P2-H-1 (1996).
- 10) 野寺: 共役勾配法の未解決問題について, 京都大学数理解析研究所講究録掲載予定 (1996).
- 11) 野寺, 稲津: リスタートによる疑似残差法の収束性の加速, *情報処理学会論文誌*, Vol.37, No.6, pp.1237-1240 (1996).

(平成 8 年 1 月 16 日受付)

(平成 8 年 6 月 6 日採録)

稲津 隆敏 (正会員)



1970 年 10 月 25 日生。1995 年 3 月慶應義塾大学理工学研究科 (数理学専攻) 修了。同年、東芝入社。行列計算の算法の開発や、ハイパフォーマンス・コンピューティングに興味を持つ。修士 (工学)。

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。同年、同大学数理科学科助手、1989 年講師となり現在に至る。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 \LaTeX 』などがある。工学博士。エッセイスト。SIAM, 日本応用数理学会会員。