

## 非対称行列系に対する疑似残差法について

稲津 隆敏 野寺 隆

慶應義塾大学工学部数理科学科

移流拡散方程式の離散近似は、非対称な係数を持つ大規模な連立一次方程式を生成する。疑似残差法 (PRES) は、このような方程式を解く右側前処理付き一般共役勾配法の1つである。この方法は、各反復で疑似残差という仮の残差のノルムを最小化するのが特徴であるが、真の残差のノルムは激しく振動する性質をもつ。そこで Schönauer のスムージング [1, 2] を疑似残差法に適用し、真の残差のノルムを単調減少させることを試みる。数値実験によれば、スムージングは残差の収束を安定化するが、収束時間に対してはよい効果は認められなかった。

## Psuedo-Residual Methods for nonsymmetric linear systems

Takatoshi Inadu, Takashi Nodera

Faculty of Science and Technology, Keio University

Psuedo-residual method (PRES) is proposed for solving large nonsymmetric systems of linear equations. This method is one of the generalized conjugate gradient methods which treat the linear system with right-hand side preconditioner. Discretization of partial differential equations by finite difference method produces the system like that.

The norm of true residuals may oscillate heavily, because we minimize the norm of psuedo residuals in each iteration of the method. Therefore we will apply Schönauer's smoothing algorithm to the PRES in order to stabilize and accelerate the convergence of the norm of true residuals. Primarily experiments suggest that the PRES method with smoothing algorithm has almost monotone convergence, but the cost of its method becomes more expensive than the cost of the method without smoothing algorithm.

# 1 はじめに

移流拡散方程式を有限差分法によって離散化したときに得られる非対称行列系  $Ax = b$  ( $A \in R^{n \times n}$ ,  $x, b \in R^n$ ) に対する反復解法の1つである疑似残差法について考える。ただし,  $A$  は正則かつ非対称な大規模疎行列である。

本稿では, 以下に示す記号・定義を用いる。ただし,  $A, Z$  は行列,  $x$  はベクトルを表すものとする。

- $\|x\|_Z = \sqrt{x^T Z x}$ :  $x$  の  $Z$ -ノルム。ただし  $Z$  は正定値対称とする。特に  $Z = I$  のとき,  $\|x\|_I = \|x\|$ : ユークリッドノルム
- $K_n(A, x) = \text{span}(x, Ax, \dots, A^n x)$ : クリロフ空間
- $\prod_1^k(A) = \sum_{i=1}^k \theta_i A^i + I$ :  $A$  の行列多項式。ただし,  $\prod_1^k(0) = 1$  とする。

## 2 疑似残差法

この節では, 一般共役勾配法の一つである疑似残差法の定義・種類とその性質を説明する。

### 2.1 疑似残差法の定義

**Definition 2.1** (一般共役勾配法, Weiss [2])

$\forall x_0$  を初期近似解とし, 次の繰り返しを一般共役勾配法という。

1.  $r_0 = Ax_0 - b$ : 初期残差
2.  $P$  を右側前処理行列とする。  $k \geq 0$  に対し,

$$x_{k+1} \in x_0 + K_k(PA, Pr_0)$$

のような  $x_{k+1}$  を計算する。ただし, このとき, 次の二つの条件のうちどちらか一方を満たすものとする。

$Z$  を正定値対称行列としたとき,

$$\text{(条件 1)} \quad r_{k+1}^T Z r_{k+1-i} = 0 \quad (1 \leq i \leq \sigma_k) \tag{2.1}$$

$$\text{(条件 2)} \quad \|r_{k+1}\|_Z = \min_{\mu_{k+1-\sigma_k}, \dots, \mu_k} \left\| r_{k+1-\sigma_k} + \sum_{i=1}^{\sigma_k} \mu_{k+1-i} A P r_{k+1-i} \right\|_Z \tag{2.2}$$

ただし,  $\sigma_k$  は一般共役勾配法のオーダーと呼ばれるもので, 詳細は次小節で述べる。

$r_{k+1} \in K_{k+1}(AP, r_0)$  より, 一般共役勾配法はクリロフ部分空間法の一つである。よって,

$$r_{k+1} = \prod_1^{k+1}(AP) r_0 = \sum_{i=1}^{k+1} \theta_{i,k} (AP)^i r_0 + r_0 \tag{2.3}$$

$$e_{k+1} = \prod_1^{k+1}(PA) e_0 = \sum_{i=1}^{k+1} \theta_{i,k} (PA)^i e_0 + e_0 \tag{2.4}$$

と表せる。 $e_{k+1}$  は誤差を表している。

**Definition 2.2** (疑似残差法, Schönauer [3])

$\forall x_0$  を初期近似解とし, 次の繰り返しを疑似残差法 (pseudo-residual method, PRES) という。

1.  $r_0 = Ax_0 - b$ : 初期残差

2.  $P_k$  を右側前処理行列,  $Z$  を適当な正定値対称行列とする。  $k \geq 0$  に対し,

$$d_k = P_k r_k, \quad (2.5)$$

$$\alpha_{i,k} = -\frac{r_{k+1-i}^T Z A d_k}{r_{k+1-i}^T Z r_{k+1-i}} \quad \text{for } 1 \leq i \leq \sigma_k, \quad (2.6)$$

$$\bar{r}_{k+1} = A d_k + \sum_{i=1}^{\sigma_k} \alpha_{i,k} r_{k+1-i}, \quad (\bar{r}_{k+1} : \text{疑似残差}) \quad (2.7)$$

$$\phi_k = \frac{1}{\sum_{i=1}^{\sigma_k} \alpha_{i,k}}, \quad (2.8)$$

$$r_{k+1} = \phi_k \bar{r}_{k+1}, \quad (2.9)$$

$$x_{k+1} = \phi_k (d_k + \sum_{i=1}^{\sigma_k} \alpha_{i,k} x_{k+1-i}). \quad (2.10)$$

ただし,  $\sigma_k$  を疑似残差法のオーダーと呼ぶ。

**Remark 2.1** 式 (2.6) は次式を満足する。

$$\|\bar{r}_{k+1}\|_Z = \min \quad (2.11)$$

この場合, 疑似残差が最小化されるので, この方法を疑似残差法と呼んでいる。

**Remark 2.2** 式 (2.11) と式 (2.1) は同値である。

## 2.2 疑似残差法の分類

疑似残差法には, 右側前処理行列  $P_k$ , 適当な正定値対称行列  $Z$ , オーダー  $\sigma_k$  などの選び方により, 表 1 のようなバリエーションがある<sup>1</sup>。以下, 各方法の特徴を簡単に述べる。

表 1 様々な疑似残差法

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

<sup>1</sup>ただしこれらが全てのバリエーションを表しているのではない。

## 1. ORTHORES

この方法は厳密版 (exact method) と呼ばれている。それは、新たな残差がそれまでに求められた全ての残差に対して  $Z$ -直交<sup>2</sup> しているからである。厳密版は解の次元回 ( $n$  回) で必ず収束するが、そのためには残差を  $n$  個記憶しておかなければならず、相当なメモリが必要となる。また、計算誤差の蓄積が大きい。

## 2. ORTHORES-T( $\sigma_{\max}$ )

この方法は打ち切り版 (truncated method) と呼ばれている。 $\sigma_k$  は最大  $\sigma_{\max}$  であり、 $k \geq \sigma_{\max}$  において最新の  $\sigma_{\max}$  本の残差が新たな残差の計算に使用される。この方法は、厳密版に比べ必要なメモリは少なくすむが、その反面  $n$  回で収束することは保証されない。また計算誤差の蓄積も大きい。

## 3. ORTHORES-R( $\sigma_{res}$ )

この方法は再出発版 (restarted method) と呼ばれている。 $\sigma_k$  は最大  $\sigma_{res}$  であり、 $\sigma_{res} + 1$  回ごとに残差  $r_{k+1}$  を

$$r_{k+1} = Ax_{k+1} - b$$

と計算し、これを新たな初期残差として繰り返しを進める。よって、打ち切り版同様メモリは少なくすむが、 $\sigma_{res} + 1$  回ごとに再出発しているため、残差が収束することは保証されない。しかし、一定の周期ごとに再出発するので、計算誤差の蓄積は厳密版や打ち切り版に比べて小さく、確実な収束が期待できる。

## 4. ORTHORES-C

打ち切り版がある時点で再出発するとき、その方法を結合版 (combined method) という。

## 5. ATPRES

右側前処理行列  $P_k$  を  $A^T$  とする。 $ZAA^T$  が対称になるので、 $i > 2$  において  $\alpha_{i,k} = 0$  となる。よって、 $\sigma_k = 2$  となる (weiss [1])。この場合、 $k = 0$  において  $\alpha_{2,0} = 0$  としなければならない。

この方法は正規方程式  $A^T Ax = A^T b$  を解くのに似ているが、 $A^T$  を右側前処理に利用しているため、 $A(A^T x') = b$  を解くことになる。よって、 $x = A^T x'$  により  $x$  を直接計算することができる。

## 2.3 疑似残差法の特質

この小節では、疑似残差法により求められる残差の収束性について説明する。一般に残差の評価はそのノルム<sup>3</sup>について行なわれるので、残差ノルムのことをただ単に残差と述べたりする。また、今後は右側前処理行列  $P_k$  をある行列  $P$  に固定し、 $AP = \tilde{A}$  と表現する。

一般共役勾配法の定義において、条件 2 を満たすアルゴリズムを考えると、それは真の残差  $\|r_{k+1}\|_Z$  を最小化しているため

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

となる。しかし、疑似残差法は疑似残差を最小化しているため、真の残差は  $Z$ -ノルムについて最小にはならない。では、真の残差はどのようになっているのだろうか。

<sup>2</sup> $r_{k+1}^T Z r_{k+1-i} = 0$  for  $1 \leq i \leq k+1$ .

<sup>3</sup>本稿では  $Z$  ノルム  $\|r_k\|_Z$  を用いる。

Lemma 2.1 全ての厳密版に対し、次式が成り立つ。

$$\|r_{k+1}\|_{Z\tilde{A}^{-1}}^2 = r_{k+1}^T Z\tilde{A}^{-1} r_{k+1} = r_{k+1}^T Z\tilde{A}^{-1} \prod_1^{k+1} (\tilde{A}) r_0 \quad (2.12)$$

ただし、 $\prod_1^{k+1} (\tilde{A})$  において、係数  $\theta_1, \dots, \theta_{k+1}$  は任意にとれるものとする。

Proof. 省略 (文献 [1] 参照)

Lemma 2.2  $A$  を正定値行列とすると、次の不等式が成立する。

$$|x^T A y| \leq \|x\|_A \|y\|_A \quad (2.13)$$

Proof. 省略 (文献 [1] 参照)

この二つの補助定理により、次の定理が成り立つ。

Theorem 2.1  $Z\tilde{A}^{-1}$  を正定値対称とする。このとき

$$\|r_{k+1}\|_{Z\tilde{A}^{-1}} = \min_{\theta_1, \dots, \theta_{k+1}} \left\| \prod_1^{k+1} (\tilde{A}) r_0 \right\|_{Z\tilde{A}^{-1}} \quad (2.14)$$

が厳密版において成り立つ。

Proof. Lemma 2.1 および Lemma 2.2 より、

$$\|r_{k+1}\|_{Z\tilde{A}^{-1}}^2 \leq \|r_{k+1}\|_{Z\tilde{A}^{-1}} \left\| \prod_1^{k+1} (\tilde{A}) r_0 \right\|_{Z\tilde{A}^{-1}} \quad (2.15)$$

$$\|r_{k+1}\|_{Z\tilde{A}^{-1}} \leq \left\| \prod_1^{k+1} (\tilde{A}) r_0 \right\|_{Z\tilde{A}^{-1}} \quad (2.16)$$

となる。式 (2.16) の等号は、右辺の最小性により成立する。

よって厳密版では、真の残差は  $Z\tilde{A}^{-1}$ -ノルムについて最小化される。しかし、疑似残差法では残差を  $Z$ -ノルムで評価するので、真の残差は最小化されない。再出発版や打ち切り版についても似たようなことが言える (weiss [1])。実際実験的に、疑似残差法の真の残差は繰り返しの度に激しく振動することが確認されている。

### 3 スムージング

前節の説明通り、疑似残差法の真の残差は激しく振動する。よって、残差を単調減少させ、収束を加速化するために、Schönauer のスムージング [1, 2] を疑似残差法に適用する。

Definition 3.1 (スムージング, Schönauer) 前節の疑似残差法に以下の計算を加え、単調減少する新たな残差  $s_{k+1}$  を得る。  $s_{k+1}$  のことをスムージングした残差と呼び、この計算をスムージングと呼ぶ。

$$\begin{aligned} s_0 &= r_0, \\ x_0^s &= x_0, \\ \gamma_k &= -\frac{s_k^T Z(r_{k+1} - s_k)}{(r_{k+1} - s_k)^T Z(r_{k+1} - s_k)}, \\ s_{k+1} &= s_k + \gamma_k(r_{k+1} - s_k), \\ x_{k+1}^s &= x_k^s + \gamma_k(x_{k+1} - x_k^s). \end{aligned}$$

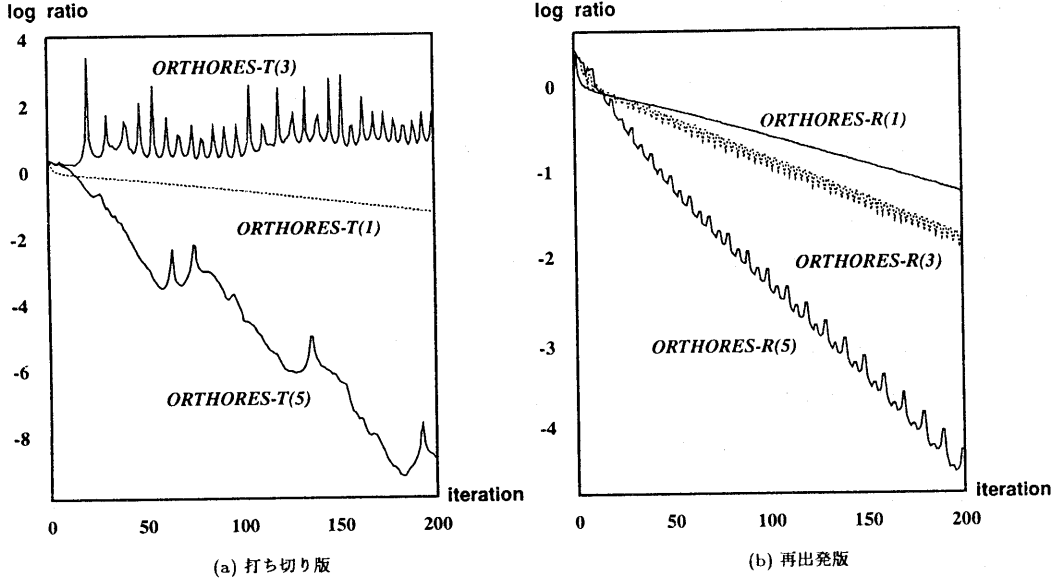


図1 通常の残差の初期残差に対する比  $\log \frac{\|r_{k+1}\|}{\|r_0\|}$

この残差の性質として、以下のようなことが言える。

$$\begin{aligned} s_{k+1} &= s_k + \gamma(r_{k+1} - s_k) \\ &= \gamma r_{k+1} + (1 - \gamma)s_k \end{aligned}$$

より、

$$\|s_{k+1}\|_Z \leq \|s_k\|_Z, \quad (3.1)$$

$$\|s_{k+1}\|_Z \leq \|r_{k+1}\|_Z. \quad (3.2)$$

式(3.1)より、 $\|s_{k+1}\|_Z$ は単調減少する。さらに式(3.2)より、元の残差より収束が速くなる。

## 4 数値実験

矩形領域  $\Omega = [0, 1] \times [0, 1]$  における移流拡散方程式のディリクレ境界条件問題

$$\begin{aligned} u_{xx} + u_{yy} + 3u_x + 5u_y &= f(x, y) \\ u(x, y)|_{\partial\Omega} &= 0 \end{aligned}$$

に対して解を

$$u(x, y) = (1 - e^x)(1 - x)y(1 - e^{1-y})$$

と設定し、右辺を決定する<sup>4</sup>。これを  $20 \times 20$  のメッシュで5点中央差分近似を用いて離散化し、数値実験を行った(行列  $A$  は  $400 \times 400$  となる)。疑似残差法の初期近似解は零ベクトルとした。今回は前処理は行っていない。また計算には Sun のワークステーション SPARCstation10 model 30 を使用した。

<sup>4</sup>真の  $u$  に対する誤差も調べるため。

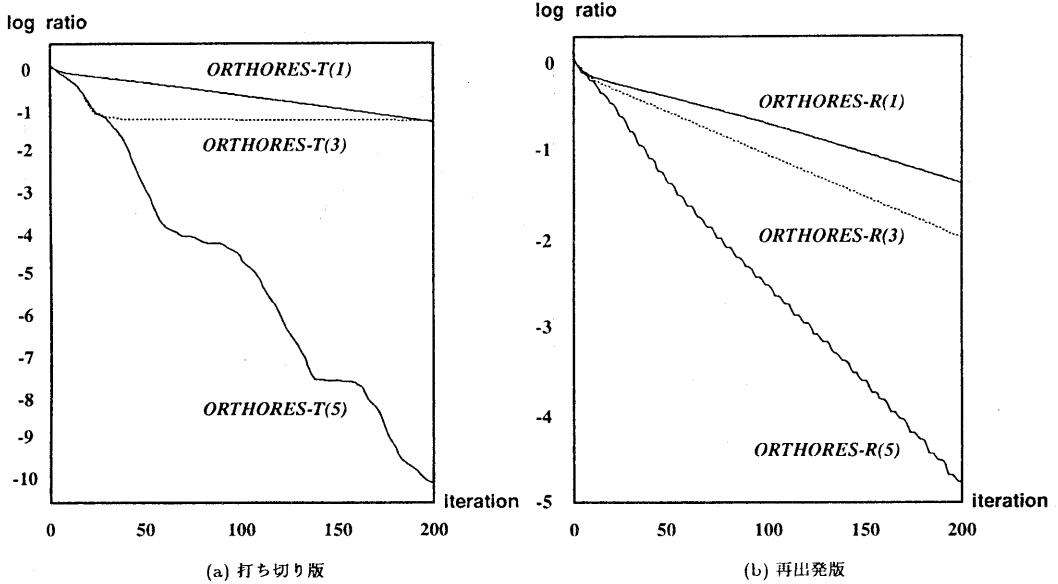


図 2 スムージングされた残差の初期残差に対する比  $\log \frac{\|s_{k+1}\|}{\|s_0\|}$

図 1 と 2 は、初期残差に対する残差の比が繰り返しごとにどのように変化するかを示す。実験した疑似残差法の種類は打ち切り版と再出発版で、それぞれオーダ  $\sigma_k$  を 1, 3, 5 と変化させた。また各方法について、通常の残差とスムージングされた残差の両方を調べた。

オーダの違いによる残差の収束性を見ると、オーダ 5 以外はよい収束性は得られていない。原因は、前処理を行っていないので、行列の条件数が大きいためと考えられる。また移流項の係数 3, 5 をそれぞれ 100, 150 とすると、オーダ 3 でもかなりよい収束性が得られる。よって、行列の条件をよくすることによって、より低いオーダでの速い収束が期待できる。

次に打ち切り版と再出発版との差を見る。オーダ 5 では、打ち切り版の方がよりよい収束性を示す。その反面オーダ 3 では、再出発版の方がゆっくりながら確実な収束を示している。まとめると、再出発版はオーダに比例するように収束性がよくなり、各オーダでゆっくり確実に収束する。打ち切り版はそのときの条件に左右され、よい条件下ではより速く収束するが、条件によっては収束しない場合もある。

スムージングの効果を調べるため、図 3 に初期残差に対する残差の比が実行時間に対してどのように変化するかを示す。図の通り、スムージングは収束を安定化するものの、時間的に加速化するとは言えない。これは、スムージングにおいて余分なベクトル演算を行っているためと考えられる。ただし、これはシングルプロセッサ環境での結果なので、ベクトルプロセッサを利用すると、スムージングの威力が発揮されるであろう。また前処理を加えることにより、よりよい効果が期待できる。その辺りについては、当日より詳細な実験結果を示す。

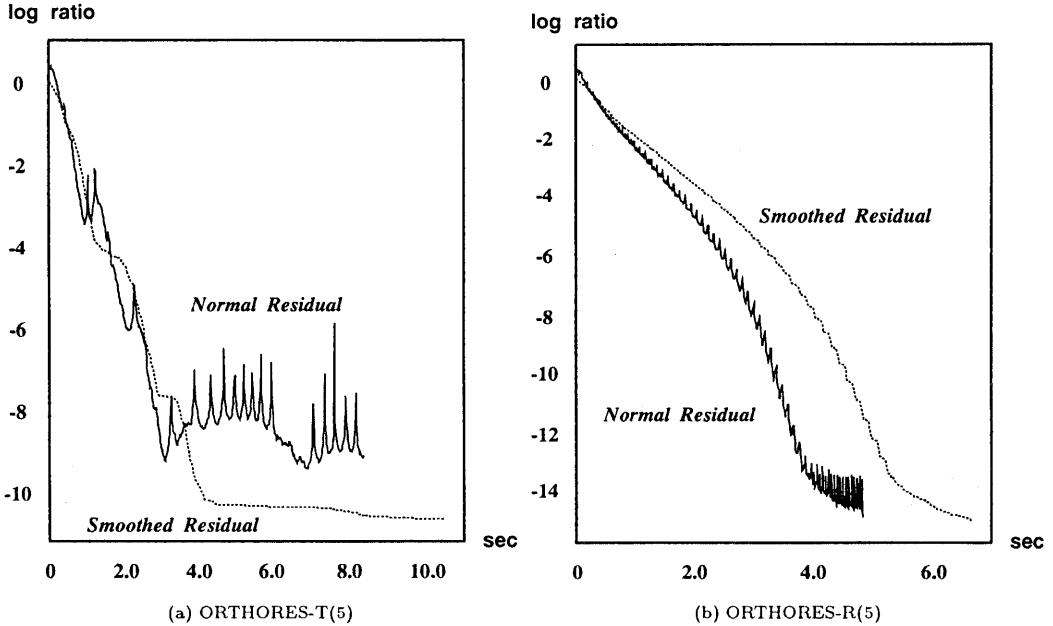


図 3 実行時間で見る残差の収束性

## 5 まとめ

疑似残差法は  $P_k, Z, \sigma_k$  の選択により様々なバリエーションがある。今回は主に  $\sigma_k$  を変えて実験を行った。その結果、各  $\sigma_k$  について、問題によって収束する場合としない場合があることがわかった。しかしおむねオーダーを 5 にすれば収束するようである。 $Z$  や  $P_k$  に関しては、 $Z = AP$  として真の残差を最小にする方法や、 $P_k = A^T Z$  として誤差を最小にする方法などがある (weiss [4])。また今回の実験では、スムージングによって残差の収束は安定するものの、時間的な加速化は確認できなかった。

今後は、主に右側前処理とその並列化について研究していきたい。

## 参考文献

- [1] R. Weiss. *Convergence Behavior of Generalized Conjugate Gradient Methods*. Doctoral Thesis, University of Karlsruhe. Interner Bericht des Rechenzentrums, 43/90, 1990.
- [2] R. Weiss. *Properties of Generalized Conjugate Gradient Methods*. Numerical Linear Algebra with Applications, Vol.1(1), 45-63, 1994.
- [3] W. Schönauer, H. Müller, E. Schnepf. *Pseudo-Residual Type Methods for the Iterative Solution of Large Linear Systems on Vector Computers*. Parallel Computing 85, Elsevier Science Publishers B. V., North Holland, 1986.
- [4] R. Weiss. *Error-Minimizing Krylov Subspace Methods*. SIAM J. Sci. Comput. Vol.15, No.3, pp.511-527, May 1994.