

# BiCGStab 法の前処理付きアルゴリズムに対する改善

伊藤 祥司<sup>1,a)</sup> 片桐 孝洋<sup>1</sup> 櫻井 隆雄<sup>2</sup> 猪貝 光祥<sup>3</sup> 大島 聡史<sup>1</sup> 黒田 久泰<sup>4</sup>  
直野 健<sup>2</sup>

受付日 2011年10月7日, 採録日 2012年1月13日

**概要:** 前処理付き BiCGStab (PBiCGStab) 法の改善アルゴリズムを提案する. 前処理付き BiCG 法に CGS 法の導出手順を適用すると, CGS 法の合理的な前処理付きアルゴリズムが構成される. この手法を PBiCGStab 法へと拡張するにあたり, BiCGStab 法に現れる MR 演算に対し論理面からの新たな考察を行い, 適用できることを示した. 本提案アルゴリズムが従来の PBiCGStab よりも合理的であることと, 数値実験により本提案の有効性を示す.

**キーワード:** 線形方程式, クリロフ部分空間法, 前処理系

## An Improvement in Preconditioned Algorithm of BiCGStab Method

SHOJI ITOH<sup>1,a)</sup> TAKAHIRO KATAGIRI<sup>1</sup> TAKAO SAKURAI<sup>2</sup> MITSUYOSHI IGAI<sup>3</sup>  
SATOSHI OHSHIMA<sup>1</sup> HISAYASU KURODA<sup>4</sup> KEN NAONO<sup>2</sup>

Received: October 7, 2011, Accepted: January 13, 2012

**Abstract:** An improved preconditioned BiCGStab algorithm (improved PBiCGStab) is proposed. Rational preconditioned algorithm of CGS has been constructed, by applying the derivation procedure of the CGS to the preconditioned BiCG. In order to extend this approach to the BiCGStab, minimum residual part of the BiCGStab must be considered logically. This proposed algorithm is also more rational than the conventional typical PBiCGStab mathematically. Numerical results show advantages of this improved PBiCGStab.

**Keywords:** linear systems, Krylov subspace method, preconditioned system

### 1. はじめに

自然現象の解明や工学問題の解決では, 数値シミュレーションによる解析がさかんである. ここで数値的に近似解を求めることは, 多くの場合, 線形方程式の求解に帰着される. 大規模で非対称な係数行列を持つ線形方程式

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

の求解では, 前処理付き BiCGStab 法 (PBiCGStab 法, Preconditioned BiCGStab method) [17] がしばしば用いられる. PBiCGStab 法は, アルゴリズムの演算量や所要メモリ量も少なく, 良好な収束性を示す.

多くの反復解法では, 求解状況の改善や収束性向上のために前処理付きアルゴリズムが用いられる. クリロフ部分空間法の前処理による収束性への影響が非常に大きいことは, 体系的な評価によっても示されている [8]. しかし, 前処理付きアルゴリズムの設計自体が悪いと, どの前処理演算を併用しても求解には至らないこともある [6]. このような背景からも, 前処理付きアルゴリズムを適切に構築することはきわめて重要である. したがって, BiCGStab 法の前処理付きアルゴリズムにおける問題点の改善や求解品質の向上は, 実用の観点からも非常に重要である.

<sup>1</sup> 東京大学情報基盤センター  
Information Technology Center, The University of Tokyo,  
Bunkyo, Tokyo 113-8658, Japan

<sup>2</sup> 日立製作所中央研究所  
Central Research Laboratory, Hitachi, Ltd., Yokohama,  
Kanagawa 244-0817, Japan

<sup>3</sup> 日立製作所超 LSI システムズ  
Hitachi ULSI Systems Co., Ltd., Yokohama, Kanagawa 244-  
0817, Japan

<sup>4</sup> 愛媛大学大学院理工学研究科  
Graduate School of Science and Engineering, Ehime Univer-  
sity, Matsuyama, Ehime 790-8577, Japan

a) itosho@cc.u-tokyo.ac.jp

ところが、従来から標準的に用いられてきている前処理付き BiCGStab 法のアルゴリズムでは、非合理的な前処理変換がなされている。この問題点は、CGS 法 [14] に対する前処理変換に起因しており、以後、双ランチョス系の前処理付きアルゴリズムでは、同様の問題点が継承されてきている。前処理付き CGS (PCGS) 法に対しては、前処理付き BiCG (PBiCG) 法 [2], [10] から CGS 法を導出する手順に基づく合理的なアルゴリズムが提案されており [9], PBiCGStab 法に対してもその方法を適用したい。このとき、BiCG 法と CGS 法に対して、BiCGStab 法の前処理変換の議論では、本質的に異なる点がある。BiCG 法と CGS 法は前処理変換の方向に対する“一致性”（本稿の中では、両側・左・右の異なる方向からの前処理変換に対し、最終的に得られるアルゴリズム記述が一致する特性を指す [5], [6]）があるが、BiCGStab 法では、残差ベクトルのノルムを最小化する演算（MR 部分, Minimum Residual part）の係数をともなうため、一致性がない。本稿では、CGS 法に対して提案した合理的な前処理変換を BiCGStab 法に適用するにあたり、一致性の観点から前処理付きアルゴリズムの構造を分析したうえで、右前処理の BiCGStab 法へ拡張して適用できることを示す。

本稿の構成として、2 章では、議論の背景として PBiCG 法と PCGS 法の重要な論点である係数  $\alpha_k, \beta_k$  の導出と特徴について概説し、前処理変換方向に対する一致性について説明する。3 章では、BiCGStab 法の係数の導出と前処理系について概説し、従来から用いられている PBiCGStab 法の問題点を指摘したうえで、それを改善するよう PBiCG 法に基づく合理的な改善アルゴリズムを提案する。4 章の数値実験で、従来版 PBiCGStab 法と本提案版との求解状況の違いを確認し、本提案アルゴリズムの有効性を確認する。5 章でまとめる。

## 2. BiCG 法と CGS 法の導出におけるポイントと前処理付きアルゴリズム

双ランチョスシステムのクリロフ部分空間法は、式 (1) の双対系（シャドウ）として

$$A^T x^\sharp = b^\sharp \quad (2)$$

を構成する BiCG 法 [2], [10] から発展した解法であり、BiCG 法から CGS 法 [14]、さらに、MR 部分を追加した BiCGSTAB 法 [17] をはじめとする様々な解法へと発展してきている (図 1)。これらのアルゴリズムでは、残差ベクトルの定義と、アルゴリズム中で用いる  $\alpha_k, \beta_k$  などの係数の定義が非常に重要である。

本章では、次章における議論の準備として、前処理なしの BiCG 法から CGS 法の導出のポイント (図 1 の「BiCG」から「CGS」への矢印) を概説し、BiCG と CGS とは等価な関係であることを示す。そのうえで、各々の前処理系に

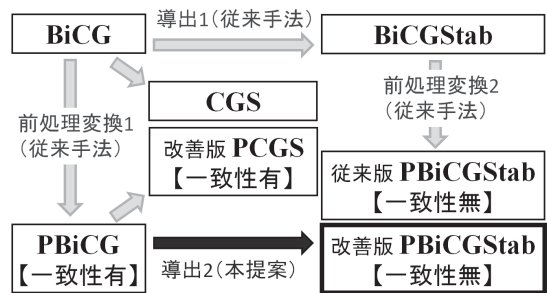


図 1 本稿で議論する双ランチョス系アルゴリズムと各々の前処理付きアルゴリズムの関係

Fig. 1 Relations between bi-Lanczos type algorithm and pre-conditioned algorithms.

おける一致性と  $\alpha_k, \beta_k$  に対する特徴について述べる (図 1 の「前処理変換 2」および「PBiCG」から「改善版 PCGS」への矢印)。

BiCGStab 法に関する従来手法である、「導出 1」と「前処理変換 1」部分、および、「導出 2」については、次章で議論する。

### 2.1 線形方程式求解アルゴリズムの前処理系

本稿において、「前処理付きアルゴリズム」、および、「前処理系」とは、前処理演算子（前処理行列） $K$  をともない記述された求解アルゴリズム、および、 $K$  に基づく演算子により変換された系を指すこととし、“不完全コレスキー分解”や“不完全 LU”などに基づく、いわゆる、“前処理演算”とは異なる。一例として、前処理系への変換は、元の線形方程式 (1) に対し、

$$A \approx K = K_L K_R$$

の前処理行列を用い、

$$\begin{aligned} \tilde{A}\tilde{x} &= \tilde{b}, & \tilde{A} &\Rightarrow K_L^{-1}AK_R^{-1}, \\ \tilde{x} &\Rightarrow K_Rx, & \tilde{b} &\Rightarrow K_L^{-1}b \end{aligned} \quad (3)$$

と変換することに相当するが、実際には式 (3) の変形自体は行わず、式 (1) を求解するアルゴリズムに対して、式 (3) の求解と等価となる変換を施した前処理付きアルゴリズムを構築する。本稿では、前処理系の行列やベクトルや係数に対しては  $\tilde{A}$  のように“~ (チルダ)”を付して表す。特に、式 (3) は式 (1) の両側から前処理変換したものであり“両側変換”と呼ばれ、

$$K_L = K, K_R = I \quad (4)$$

としたものは“左変換” ( $I$  は単位行列)、

$$K_L = I, K_R = K \quad (5)$$

としたものは“右変換”と呼ばれ、各々は、

$$\tilde{A} \Rightarrow K^{-1}A, \tilde{x} \Rightarrow x, \tilde{b} \Rightarrow K^{-1}b,$$

$$\tilde{A} \Rightarrow AK^{-1}, \tilde{\mathbf{x}} \Rightarrow K\mathbf{x}, \tilde{\mathbf{b}} \Rightarrow \mathbf{b}$$

と、式 (1) を変換することに相当する [1], [3], [18].

## 2.2 BiCG 法における着眼点

BiCG 法 [2], [10] では、係数行列  $A$  が非対称である線形方程式 (1) を解くために、その双対系 (2) を導入し、線形系と双対系における残差ベクトル

$$\mathbf{r}_k = R_k(A)\mathbf{r}_0, \quad (6)$$

$$\mathbf{r}_k^\sharp = R_k(A^T)\mathbf{r}_0^\sharp \quad (7)$$

と探索方向ベクトル

$$\mathbf{p}_k = P_k(A)\mathbf{r}_0, \quad (8)$$

$$\mathbf{p}_k^\sharp = P_k(A^T)\mathbf{r}_0^\sharp \quad (9)$$

について、

$$\langle \mathbf{r}_i^\sharp, \mathbf{r}_j \rangle = 0 \quad (i \neq j) \quad (\text{双直交性}), \quad (10)$$

$$\langle \mathbf{p}_i^\sharp, A\mathbf{p}_j \rangle = 0 \quad (i \neq j) \quad (\text{双共役性}) \quad (11)$$

が成り立つ。ここで、 $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$  (初期残差ベクトル),  $\mathbf{r}_0^\sharp = \mathbf{b}^\sharp - A^T\mathbf{x}_0^\sharp$  (初期シャドウ残差ベクトル) である。また、式 (6)~(9) で示した  $k$  次多項式は、

$$R_k(z) = R_{k-1}(z) - \alpha_{k-1}zP_{k-1}(z), \quad R_0(z) = 1,$$

$$P_k(z) = R_k(z) + \beta_{k-1}P_{k-1}(z), \quad P_0(z) = 1$$

の交代漸化式を満たす。ただし、実際のアルゴリズム中では、これらの多項式は用いず、 $k$  回目の反復において、

$$\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1}\mathbf{p}_{k-1}, \quad \mathbf{p}_k^\sharp = \mathbf{r}_k^\sharp + \beta_{k-1}\mathbf{p}_{k-1}^\sharp,$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k, \quad \mathbf{r}_{k+1}^\sharp = \mathbf{r}_k^\sharp - \alpha_k A^T\mathbf{p}_k^\sharp$$

として記述される。式 (10), (11) において  $\langle \mathbf{u}, \mathbf{v} \rangle$  はスカラー積であり [4], [15], 本稿では特に、双対関係にあるベクトル同士の積を指し、内積  $(\mathbf{u}, \mathbf{v})$  とは区別して表記する。

本稿における議論の基となる BiCG 法のアルゴリズムに現れる係数  $\alpha_k$  と  $\beta_k$  の記述を示す。

$$\alpha_k = \frac{\langle \mathbf{r}_k^\sharp, \mathbf{r}_k \rangle}{\langle \mathbf{p}_k^\sharp, A\mathbf{p}_k \rangle}, \quad (12)$$

$$\beta_k = \frac{\langle \mathbf{r}_{k+1}^\sharp, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k^\sharp, \mathbf{r}_k \rangle}. \quad (13)$$

BiCG 法に対する前処理付きアルゴリズムを導出するため、次のように BiCG 法の形式的な導出を行う。まず、式 (1) と (2) とを一括して表現し、

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}},$$

$$\tilde{A} = \begin{bmatrix} A & O \\ O & A^T \end{bmatrix}, \tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^\sharp \end{bmatrix}, \tilde{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ \mathbf{b}^\sharp \end{bmatrix}$$

を CG 法に適用する [13], [18]. この係数行列から次の前処理行列

$$\begin{aligned} \check{A} \approx \check{K} &= \begin{bmatrix} K & O \\ O & K^T \end{bmatrix} = \begin{bmatrix} K_L & O \\ O & K_R^T \end{bmatrix} \begin{bmatrix} K_R & O \\ O & K_L^T \end{bmatrix} \\ &\equiv \check{K}_L \check{K}_R \end{aligned}$$

が得られ、このとき、

$$\begin{aligned} \tilde{\check{A}} &\Rightarrow \check{K}_L^{-1} \tilde{\check{A}} \check{K}_R^{-1} \\ &= \begin{bmatrix} K_L^{-1} A K_R^{-1} & O \\ O & K_R^{-T} A^T K_L^{-T} \end{bmatrix} \equiv \begin{bmatrix} \tilde{A} & O \\ O & \tilde{A}^T \end{bmatrix}, \end{aligned} \quad (14)$$

$$\tilde{\check{\mathbf{x}}}_k \Rightarrow \check{K}_R \check{\mathbf{x}}_k, \quad \tilde{\check{\mathbf{b}}} \Rightarrow \check{K}_L^{-1} \check{\mathbf{b}}.$$

これにともない、 $\check{\mathbf{p}}_k$  と  $\check{\mathbf{r}}_k$  は次のとおり変換される。

$$\begin{aligned} \tilde{\check{\mathbf{p}}}_k &\Rightarrow \check{K}_R \check{\mathbf{p}}_k = \begin{bmatrix} K_R \mathbf{p}_k \\ K_L^T \mathbf{p}_k^\sharp \end{bmatrix}, \\ \tilde{\check{\mathbf{r}}}_k &\Rightarrow \check{K}_L^{-1} \check{\mathbf{r}}_k = \begin{bmatrix} K_L^{-1} \mathbf{r}_k \\ K_R^{-T} \mathbf{r}_k^\sharp \end{bmatrix}. \end{aligned}$$

これらの前処理変換を基にして、線形系 (1) と双対系 (2) の各々の系に対する前処理変換を BiCG 法に適用することで PBiCG のアルゴリズムが得られる。

線形系の残差ベクトルと、双対系の残差ベクトルの前処理変換の記述は、

$$K_L^{-1} \mathbf{r}_k = K_L^{-1} \mathbf{b} - (K_L^{-1} A K_R^{-1}) (K_R \mathbf{x}_k), \quad (15)$$

$$K_R^{-T} \mathbf{r}_k^\sharp = K_R^{-T} \mathbf{b}^\sharp - (K_R^{-T} A^T K_L^{-T}) (K_L^T \mathbf{x}_k^\sharp). \quad (16)$$

これらを式変形すると次のとおり、 $\mathbf{r}_k$ ,  $\mathbf{r}_k^\sharp$  とともに前処理行列が現れず、本来の残差ベクトルの情報を保持する記述となる\*1。

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k, \quad (17)$$

$$\mathbf{r}_k^\sharp = \mathbf{b}^\sharp - A^T\mathbf{x}_k^\sharp. \quad (18)$$

この前処理系における双直交性と双共役性は、

$$\begin{aligned} \langle \tilde{\check{\mathbf{r}}}_i^\sharp, \tilde{\check{\mathbf{r}}}_j \rangle &= \langle K_R^{-T} \mathbf{r}_i^\sharp, K_L^{-1} \mathbf{r}_j \rangle \\ &= \langle \mathbf{r}_i^\sharp, K^{-1} \mathbf{r}_j \rangle = 0 \quad (i \neq j), \end{aligned} \quad (19)$$

$$\begin{aligned} \langle \tilde{\check{\mathbf{p}}}_i^\sharp, \tilde{\check{A}} \tilde{\check{\mathbf{p}}}_j \rangle &= \langle K_L^T \mathbf{p}_i^\sharp, (K_L^{-1} A K_R^{-1}) (K_R \mathbf{p}_j) \rangle \\ &= \langle \mathbf{p}_i^\sharp, A\mathbf{p}_j \rangle = 0 \quad (i \neq j). \end{aligned} \quad (20)$$

これらと同様に式 (12), (13) を前処理変換すると、

$$\alpha_k^{\text{PBiCG}} = \frac{\langle \tilde{\check{\mathbf{r}}}_k^\sharp, \tilde{\check{\mathbf{r}}}_k \rangle}{\langle \tilde{\check{\mathbf{p}}}_k^\sharp, \tilde{\check{A}} \tilde{\check{\mathbf{p}}}_k \rangle} = \frac{\langle \mathbf{r}_k^\sharp, K^{-1} \mathbf{r}_k \rangle}{\langle \mathbf{p}_k^\sharp, A\mathbf{p}_k \rangle}, \quad (21)$$

\*1 本稿では、線形系 (1) において  $\tilde{\check{\mathbf{r}}}_k \Rightarrow \mathbf{r}_k = K^{-1}(\mathbf{b} - A\mathbf{x}_k)$  など、前処理行列により残差ベクトルの情報が異なる系 [5], [6], [7] については議論しない。

$$\beta_k^{\text{PBICG}} = \frac{\langle \tilde{\mathbf{r}}_{k+1}^\#, \tilde{\mathbf{r}}_{k+1} \rangle}{\langle \tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k \rangle} = \frac{\langle \mathbf{r}_{k+1}^\#, K^{-1} \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k^\#, K^{-1} \mathbf{r}_k \rangle}. \quad (22)$$

式 (21), (22) に, 左前処理変換 (4) や右前処理変換 (5) を適用した場合でも, 最終的に得られる  $\alpha_k^{\text{PBICG}}$ ,  $\beta_k^{\text{PBICG}}$  は同一であるので前処理変換の方向に対する一致性がある.

### 2.3 CGS 法における着眼点

CGS 法は, BiCG 法に現れる  $A^T$  を用いないよう BiCG の中の変数を変換して得られた解法である [14]. その導出過程では, 式 (1), (2) に対する残差ベクトルの多項式表現 (6), (7), および, 探索方向ベクトルの多項式表現 (8), (9) を BiCG の  $\alpha_k$  と  $\beta_k$  の演算に代入し,

$$\begin{aligned} \langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle &= \langle R_k(A^T) \mathbf{r}_0^\#, R_k(A) \mathbf{r}_0 \rangle \\ &= \langle \mathbf{r}_0^\#, R_k^2(A) \mathbf{r}_0 \rangle, \end{aligned} \quad (23)$$

$$\begin{aligned} \langle \mathbf{p}_k^\#, A \mathbf{p}_k \rangle &= \langle P_k(A^T) \mathbf{r}_0^\#, A P_k(A) \mathbf{r}_0 \rangle \\ &= \langle \mathbf{r}_0^\#, A P_k^2(A) \mathbf{r}_0 \rangle. \end{aligned} \quad (24)$$

ここで,

$$\mathbf{r}_k^{\text{CGS}} \equiv R_k^2(A) \mathbf{r}_0, \quad \mathbf{p}_k^{\text{CGS}} \equiv A P_k^2(A) \mathbf{r}_0$$

とおくと,

$$\alpha_k^{\text{BiCG}} = \frac{\langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle}{\langle \mathbf{p}_k^\#, A \mathbf{p}_k \rangle} \equiv \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_k^{\text{CGS}} \rangle}{\langle \mathbf{r}_0^\#, A \mathbf{p}_k^{\text{CGS}} \rangle} = \alpha_k^{\text{CGS}}, \quad (25)$$

$$\beta_k^{\text{BiCG}} = \frac{\langle \mathbf{r}_{k+1}^\#, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle} \equiv \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_{k+1}^{\text{CGS}} \rangle}{\langle \mathbf{r}_0^\#, \mathbf{r}_k^{\text{CGS}} \rangle} = \beta_k^{\text{CGS}} \quad (26)$$

が得られる. つまり, BiCG と CGS に現れる係数  $\alpha_k$  と  $\beta_k$  は, 式変形と変数の置き換えによる等価 ( $\equiv$ ) な関係である.

CGS に現れるその他のベクトルも, 式 (6), (8) を用いた式変形により導出される.

#### 2.3.1 従来の PCGS における問題点

CGS に対し従来から用いられている前処理変換は,

$$\begin{aligned} \tilde{A} &\Rightarrow K_L^{-1} A K_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow K_R \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow K_L^{-1} \mathbf{b}, \\ \tilde{\mathbf{p}}_k^{\text{CGS}} &\Rightarrow K_L^{-1} \mathbf{p}_k^{\text{CGS}}, \quad \tilde{\mathbf{r}}_k^{\text{CGS}} \Rightarrow K_L^{-1} \mathbf{r}_k^{\text{CGS}}, \\ \tilde{\mathbf{r}}_0^\# &\Rightarrow K_L^T \mathbf{r}_0^\# \end{aligned} \quad (27)$$

であり, CGS 中の係数  $\alpha_k$  の前処理変換は

$$\begin{aligned} \frac{\langle \tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_k^{\text{CGS}} \rangle}{\langle \tilde{\mathbf{r}}_0^\#, \tilde{A} \tilde{\mathbf{p}}_k^{\text{CGS}} \rangle} &= \frac{\langle K_L^T \mathbf{r}_0^\#, K_L^{-1} \mathbf{r}_k^{\text{CGS}} \rangle}{\langle K_L^T \mathbf{r}_0^\#, (K_L^{-1} A K_R^{-1}) (K_L^{-1} \mathbf{p}_k^{\text{CGS}}) \rangle} \\ &= \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_k^{\text{CGS}} \rangle}{\langle \mathbf{r}_0^\#, A K^{-1} \mathbf{p}_k^{\text{CGS}} \rangle} \end{aligned} \quad (28)$$

である [1], [3], [17]. この分子から CGS の  $\beta_k$  も同様に変換できる. 式 (4), (5) を適用し, 左・右前処理変換した場合でも最終的に式 (28) であるので一致性がある.

ところが, 式 (27) による双対系の残差ベクトルとは,

$$K_L^T \mathbf{r}_k^\# = K_R^{-T} \mathbf{b}^\# - (K_R^{-T} A^T K_L^{-T}) (K_L^T \mathbf{x}_k^\#)$$

である. これを式変形すると,

$$\mathbf{r}_k^\# = K^{-T} (\mathbf{b}^\# - A^T \mathbf{x}_k^\#) \quad (29)$$

となり, 双対系の本来の残差ベクトル (18) と比べて,  $K^{-T}$  が作用している. この点では, 前処理付き BiCG との等価な関係が成り立っていない.

一方で, 式 (18) となるよう双対系の方程式を前処理変換すると,

$$K_L^T \mathbf{r}_k^\# = K_L^T \mathbf{b}^\# - (K_L^T A^T X) (X^{-1} \mathbf{x}_k^\#) \quad (30)$$

であり\*2, この係数行列は式 (14) で示した  $\tilde{A}^T$  とは異なる. このとき, 前処理系において式 (23), (24) の式変形が成立しない. したがって,  $\alpha_k^{\text{PBICG}} \neq \alpha_k^{\text{PCGS}}$ ,  $\beta_k^{\text{PBICG}} \neq \beta_k^{\text{PCGS}}$  であるため, 前処理なしのときの式 (25), (26) の性質を有していない [9].

#### 2.3.2 PCGS 導出に対する改善

これに対し, PBiCG 自体に CGS 導出の手順を適用するとき, 前処理変換は,

$$\begin{aligned} \tilde{A} &\Rightarrow K_L^{-1} A K_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow K_R \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow K_L^{-1} \mathbf{b}, \\ \tilde{\mathbf{p}}_k^{\text{CGS}} &\Rightarrow K_R \mathbf{p}_k^{\text{CGS}}, \quad \tilde{\mathbf{r}}_k^{\text{CGS}} \Rightarrow K_L^{-1} \mathbf{r}_k^{\text{CGS}}, \\ \tilde{\mathbf{r}}_0^\# &\Rightarrow K_R^{-T} \mathbf{r}_0^\# \end{aligned} \quad (31)$$

である [9]. このとき, 双対系の残差ベクトルは,

$$K_R^{-T} \mathbf{r}_k^\# = K_R^{-T} \mathbf{b}^\# - (K_R^{-T} A^T K_L^{-T}) (K_L^T \mathbf{x}_k^\#)$$

であり, 式 (18) の関係を満たし, かつ, 係数行列は  $\tilde{A}^T$  である. 式 (31) を式 (25), (26) の前処理系に代入すると,

$$\begin{aligned} \alpha_k^{\text{PBICG}} &= \frac{\langle \tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k \rangle}{\langle \tilde{\mathbf{p}}_k^\#, \tilde{A} \tilde{\mathbf{p}}_k \rangle} \equiv \frac{\langle \tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_k^{\text{CGS}} \rangle}{\langle \tilde{\mathbf{r}}_0^\#, \tilde{A} \tilde{\mathbf{p}}_k^{\text{CGS}} \rangle} \\ &= \frac{\langle K_R^{-T} \mathbf{r}_0^\#, K_L^{-1} \mathbf{r}_k^{\text{CGS}} \rangle}{\langle K_R^{-T} \mathbf{r}_0^\#, (K_L^{-1} A K_R^{-1}) (K_R \mathbf{p}_k^{\text{CGS}}) \rangle} \\ &= \frac{\langle \mathbf{r}_0^\#, K^{-1} \mathbf{r}_k^{\text{CGS}} \rangle}{\langle \mathbf{r}_0^\#, K^{-1} A \mathbf{p}_k^{\text{CGS}} \rangle} = \alpha_k^{\text{PCGS}} \end{aligned}$$

であり, 式 (21) と同値の関係であり合理的である. この分子を用いて  $\beta_k^{\text{PBICG}} \equiv \beta_k^{\text{PCGS}}$  である. これらの前処理変換も一致性を有する.

\*2  $X$  は  $\mathbf{r}_k^\# \Rightarrow K_L^T \mathbf{r}_k^\#$  を満たすよう導出すると, 一例として,  $X = K^{-T} K_L^{-T}$  である [9].

### 3. 前処理付き BiCGStab 法の問題点を改善したアルゴリズムの提案

前章で確認した PBiCG と PCGS, および, 本章で議論する PBiCGStab のアルゴリズムに現れる係数の特性をまとめたものが表 1 である.

PBiCG と PCGS は“一致性” (式 (3), (4), (5) の各前処理変換に対し, 最終的に得られる演算の記述が一致する特性 [5], [6]) を有することに対し, PBiCGStab に現れる係数では一致性を有していない. また, PBiCG と PCGS の  $\alpha_k$  と  $\beta_k$  には等価の関係があったが, PBiCGStab には新たに一致性のない係数  $\omega_k$  が現れる. このようなことに注目し, PBiCGStab を分析のうえ, 改善アルゴリズムを提案する.

#### 3.1 BiCGStab 法における着眼点

BiCGStab 法では, 係数  $\omega$  を有する多項式

$$S_k(z) = (1 - \omega_{k-1}z)(1 - \omega_{k-2}z) \cdots (1 - \omega_0z) \quad (32)$$

を用いる. ここで, BiCGStab のシャドウ残差ベクトルでは  $S_k(A^T)$  を用いて,  $\mathbf{s}_k^\# = S_k(A^T)\mathbf{r}_0^\#$  を生成し, BiCG の  $\mathbf{r}_k^\#$  の代わりに用いる. 式 (32) を展開し,  $\mathbf{s}_k^\#$  は,

$$\mathbf{s}_k^\# = \frac{\text{lc}(S_k)}{\text{lc}(R_k)} \left( \mathbf{r}_k^\# + d_{k-1}\mathbf{r}_{k-1}^\# + \cdots + d_1\mathbf{r}_1^\# + d_0\mathbf{r}_0^\# \right) \quad (33)$$

と表せる\*3. 記号 lc とは, “leading coefficient, 多項式の最高次係数” の略であり,

$$\frac{\text{lc}(R_{k+1})}{\text{lc}(R_k)} = -\alpha_k, \quad \frac{\text{lc}(S_{k+1})}{\text{lc}(S_k)} = -\omega_k$$

の関係である.

BiCG からの導出は, 次のとおり, 式 (33) と  $\mathbf{r}_k$  とのスカラ積に対し, 双直交性 (10) を適用すると,

$$\langle \mathbf{s}_k^\#, \mathbf{r}_k \rangle = \frac{\text{lc}(S_k)}{\text{lc}(R_k)} \langle \mathbf{r}_k^\# + d_{k-1}\mathbf{r}_{k-1}^\# +$$

表 1 各前処理付きアルゴリズムに表れる係数の一致性

Table 1 Congruency of coefficients in preconditioned algorithms.

	$\alpha_k$	$\beta_k$	$\omega_k$
PBiCG	○	○	-
PCGS	○	○	-
PBiCGStab	○	△	×

○: 一致性あり, ×: 一致性なし, △: 広義の一致性あり (一致性なしの係数  $\omega_k$  をともなう), -: アルゴリズム中に存在しない係数

\*3 括弧内の係数  $d_i$  ( $i = k-1, \dots, 0$ ) は, 各  $\mathbf{r}_i^\#$  の係数を  $\text{lc}(S_k)/\text{lc}(R_k)$  で割ったものであるが,  $\mathbf{r}_k^\#$  以外の項は, 双直交性 (10) と双共役性 (11) により考慮する必要はないため, 特に定式化しない.

$$\begin{aligned} & \cdots + d_1\mathbf{r}_1^\# + d_0\mathbf{r}_0^\#, \mathbf{r}_k \rangle \\ & = \frac{\text{lc}(S_k)}{\text{lc}(R_k)} \langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle \equiv \frac{1}{c_k} \langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle \end{aligned}$$

の関係である. したがって,

$$\begin{aligned} \langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle & = c_k \langle \mathbf{s}_k^\#, \mathbf{r}_k \rangle = c_k \langle S_k(A^T)\mathbf{r}_0^\#, R_k(A)\mathbf{r}_0 \rangle \\ & = c_k \langle \mathbf{r}_0^\#, S_k(A)R_k(A)\mathbf{r}_0 \rangle. \end{aligned} \quad (34)$$

次に, BiCG のアルゴリズム記述  $\mathbf{p}_k^\# = \mathbf{r}_k^\# + \beta_{k-1}\mathbf{p}_{k-1}^\#$  と式 (11) から,

$$\begin{aligned} \langle \mathbf{p}_k^\#, A\mathbf{p}_k \rangle & = \langle \mathbf{r}_k^\#, A\mathbf{p}_k \rangle + \beta_{k-1} \langle \mathbf{p}_{k-1}^\#, A\mathbf{p}_k \rangle \\ & = \langle \mathbf{r}_k^\#, A\mathbf{p}_k \rangle \end{aligned} \quad (35)$$

および,  $\langle \mathbf{r}_i^\#, A\mathbf{p}_j \rangle = \langle \mathbf{p}_i^\# - \beta_{i-1}\mathbf{p}_{i-1}^\#, A\mathbf{p}_j \rangle = 0$  ( $i \neq j$ ) であるので,  $\mathbf{s}_k^\#$  と  $A\mathbf{p}_k$  とのスカラ積に対し式 (33) を適用し式変形すると, 最終的に

$$\begin{aligned} \langle \mathbf{r}_k^\#, A\mathbf{p}_k \rangle & = c_k \langle \mathbf{s}_k^\#, A\mathbf{p}_k \rangle = c_k \langle S_k(A^T)\mathbf{r}_0^\#, AP_k(A)\mathbf{r}_0 \rangle \\ & = c_k \langle \mathbf{r}_0^\#, AS_k(A)P_k(A)\mathbf{r}_0 \rangle. \end{aligned} \quad (36)$$

ここで,

$$\mathbf{r}_k^{\text{STAB}} \equiv S_k(A)R_k(A)\mathbf{r}_0, \quad (37)$$

$$\mathbf{p}_k^{\text{STAB}} \equiv S_k(A)P_k(A)\mathbf{r}_0 \quad (38)$$

とおくと, 式 (34)~(36) により

$$\begin{aligned} \alpha_k^{\text{BiCG}} & = \frac{\langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle}{\langle \mathbf{p}_k^\#, A\mathbf{p}_k \rangle} \equiv \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_k^{\text{STAB}} \rangle}{\langle \mathbf{r}_0^\#, A\mathbf{p}_k^{\text{STAB}} \rangle} = \alpha_k^{\text{STAB}}, \quad (39) \\ \beta_k^{\text{BiCG}} & = \frac{\langle \mathbf{r}_{k+1}^\#, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k^\#, \mathbf{r}_k \rangle} \equiv \frac{\alpha_k}{\omega_k} \times \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_{k+1}^{\text{STAB}} \rangle}{\langle \mathbf{r}_0^\#, \mathbf{r}_k^{\text{STAB}} \rangle} = \beta_k^{\text{STAB}} \end{aligned} \quad (40)$$

であり, BiCG と BiCGStab の係数  $\alpha_k$  と  $\beta_k$  も, 式変形と変数の置き換えによる等価な関係である [15].

ところが, CGS とは異なり, 式 (40) は

$$\omega_k = \frac{(A\mathbf{s}_k, \mathbf{s}_k)}{(A\mathbf{s}_k, A\mathbf{s}_k)} \quad (41)$$

という係数をとともなう. ここで  $(\mathbf{u}, \mathbf{v})$  は内積であり, 前述のとおり, BiCG と CGS がスカラ積  $(\mathbf{u}, \mathbf{v})$  のみで構成されることに対する大きな相違点である. この係数 (41) は BiCGStab の中で  $\|\mathbf{r}_{k+1}^{\text{STAB}}\|$  を最小化する演算 (MR 演算) を構成する. ここで,

$$\mathbf{r}_{k+1}^{\text{STAB}} = \mathbf{s}_k - \omega_k A\mathbf{s}_k. \quad (42)$$

BiCGStab に現れるその他のベクトルは, 式 (37), (38) を用いた式変形により導出される.

BiCGStab に対して従来から用いられている前処理変

換は,

$$\begin{aligned} \tilde{A} &\Rightarrow K_L^{-1}AK_R^{-1}, \tilde{\mathbf{x}}_k \Rightarrow K_R\mathbf{x}_k, \tilde{\mathbf{b}} \Rightarrow K_L^{-1}\mathbf{b}, \\ \tilde{\mathbf{p}}_k^{\text{STAB}} &\Rightarrow K_L^{-1}\mathbf{p}_k^{\text{STAB}}, \tilde{\mathbf{s}}_k \Rightarrow K_L^{-1}\mathbf{s}_k, \\ \tilde{\mathbf{r}}_k^{\text{STAB}} &\Rightarrow K_L^{-1}\mathbf{r}_k^{\text{STAB}}, \tilde{\mathbf{r}}_0^\# \Rightarrow K_L^T\mathbf{r}_0^\# \end{aligned} \quad (43)$$

であり [3], [17], このとき  $\alpha_k$  の前処理変換は

$$\begin{aligned} \frac{\langle \tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_k^{\text{STAB}} \rangle}{\langle \tilde{\mathbf{r}}_0^\#, \tilde{A}\tilde{\mathbf{p}}_k^{\text{STAB}} \rangle} &= \frac{\langle K_L^T\mathbf{r}_0^\#, K_L^{-1}\mathbf{r}_k^{\text{STAB}} \rangle}{\langle K_L^T\mathbf{r}_0^\#, (K_L^{-1}AK_R^{-1})(K_L^{-1}\mathbf{p}_k^{\text{STAB}}) \rangle} \\ &= \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_k^{\text{STAB}} \rangle}{\langle \mathbf{r}_0^\#, AK^{-1}\mathbf{p}_k^{\text{STAB}} \rangle} \equiv \alpha_k^{\text{PSTAB}} \end{aligned} \quad (44)$$

であるため, PCGS の場合と同様, 一致性がある.

$\omega_k$  の前処理変換については,

$$\tilde{\omega}_k = \frac{(\tilde{A}\tilde{\mathbf{s}}_k, \tilde{\mathbf{s}}_k)}{(\tilde{A}\tilde{\mathbf{s}}_k, \tilde{A}\tilde{\mathbf{s}}_k)} \quad (45)$$

であり, 式 (4), (5) の前処理変換の方向に関して一致性がない. なぜならば, 式 (42) の前処理系

$$\tilde{\mathbf{r}}_{k+1}^{\text{STAB}} = \tilde{\mathbf{s}}_k - \tilde{\omega}_k \tilde{A}\tilde{\mathbf{s}}_k \quad (46)$$

の両側 (b) · 左 (l) · 右 (r) 方向の前処理変換は,

$$K_L^{-1}\mathbf{r}_{k+1}^{\text{STAB}} = K_L^{-1}\mathbf{s}_k - \tilde{\omega}_k^b (K_L^{-1}AK_R^{-1})(K_L^{-1}\mathbf{s}_k), \quad (47)$$

$$K^{-1}\mathbf{r}_{k+1}^{\text{STAB}} = K^{-1}\mathbf{s}_k - \tilde{\omega}_k^l (K^{-1}A)(K^{-1}\mathbf{s}_k), \quad (48)$$

$$\mathbf{r}_{k+1}^{\text{STAB}} = \mathbf{s}_k - \tilde{\omega}_k^r (AK^{-1})\mathbf{s}_k \quad (49)$$

であり, 各々に対応する  $\tilde{\omega}_k$  を素直に求めると,

$$\tilde{\omega}_k^b = \frac{(K_L^{-1}AK^{-1}\mathbf{s}_k, K_L^{-1}\mathbf{s}_k)}{(K_L^{-1}AK^{-1}\mathbf{s}_k, K_L^{-1}AK^{-1}\mathbf{s}_k)},$$

$$\tilde{\omega}_k^l = \frac{(K^{-1}AK^{-1}\mathbf{s}_k, K^{-1}\mathbf{s}_k)}{(K^{-1}AK^{-1}\mathbf{s}_k, K^{-1}AK^{-1}\mathbf{s}_k)},$$

$$\tilde{\omega}_k^r = \frac{(AK^{-1}\mathbf{s}_k, \mathbf{s}_k)}{(AK^{-1}\mathbf{s}_k, AK^{-1}\mathbf{s}_k)}$$

となるからである. あるいは, 式 (45) に対し, 直接, 両側 · 左 · 右前処理変換を適用しても同じである.

式 (47)~(49) には前処理変換にかかる冗長な演算を有するものもあり, 式変形によりそれらを消去し,  $\tilde{\omega}_k^b$ ,  $\tilde{\omega}_k^l$ ,  $\tilde{\omega}_k^r$  を単に “前処理変換された  $\omega_k$ ” として広義にとらえ,  $\tilde{\omega}_k$  と記述すると, 最終的に式 (47)~(49) は, いずれも

$$\mathbf{r}_{k+1}^{\text{STAB}} = \mathbf{s}_k - \tilde{\omega}_k (AK^{-1})\mathbf{s}_k \quad (50)$$

となる (広義の一致性がある). しかしながら,  $\tilde{\omega}_k^b$ ,  $\tilde{\omega}_k^l$ ,  $\tilde{\omega}_k^r$  自体は, 式変形を行っても同一の記述とはならない (一致性がない).

$\beta_k^{\text{PSTAB}}$  については,  $\tilde{\omega}_k$  をともなうため, 狭義には両側 ·

左 · 右前処理変換に対する一致性はないものの, 広義にとらえると一致性がある (表 1 では  $\Delta$ ).

PBiCGStab に現れるその他のベクトルの演算についても,  $\tilde{\omega}_k$  を広義にとらえると, 前処理変換の方向によらず一致性を有する. つまり, PBiCGStab において広義の一致性のない演算は, MR 演算の  $\tilde{\omega}_k^b$ ,  $\tilde{\omega}_k^l$ ,  $\tilde{\omega}_k^r$  のみである.

### 3.2 従来の前処理付き BiCGStab 法のアルゴリズム

通常,  $\tilde{\omega}_k$  では, 前処理変換された残差ベクトル (47)~(49) の中から,  $\|\mathbf{r}_{k+1}^{\text{STAB}}\|$  自体を最小化して, 前処理演算に関する不要な計算が生じない右変換 (5) を適用する [18].

線形方程式 (1) を右前処理変換すると

$$(AK^{-1})(K\mathbf{x}) = \mathbf{b}. \quad (51)$$

これに基づき, 従来から用いられている典型的な前処理付き BiCGStab 法 [1], [3], [17], [18] は, 式 (43) に対し, 次のとおり右前処理変換が施されたものである.

$$\begin{aligned} \tilde{\mathbf{p}}_k^{\text{STAB}} &\Rightarrow \mathbf{p}_k^{\text{STAB}}, \tilde{\mathbf{s}}_k \Rightarrow \mathbf{s}_k, \tilde{\mathbf{r}}_k^{\text{STAB}} \Rightarrow \mathbf{r}_k^{\text{STAB}}, \\ \tilde{\mathbf{r}}_0^\# &\Rightarrow \mathbf{r}_0^\#. \end{aligned} \quad (52)$$

このとき得られる前処理付き BiCGStab 法のアルゴリズム (従来版) は次のとおりである\*4. 本稿でのアルゴリズム記述は文献 [3] にならった. また, 本章の記述では, 1 つのアルゴリズムに閉じているため, 前処理変換の議論で用いた “~ (チルダ)” や右上の “STAB” などの記号は表記していない.

Algorithm 1. 前処理付き BiCGStab 法 (従来版):

$\mathbf{x}_0, \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0,$

$\langle \tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0 \rangle \neq 0, e.g., \mathbf{r}_0^\# = \mathbf{r}_0, \beta_{-1} = 0,$

For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1}(\mathbf{p}_{k-1} - \omega_{k-1}AK^{-1}\mathbf{p}_{k-1}),$$

$$\alpha_k = \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_k \rangle}{\langle \mathbf{r}_0^\#, AK^{-1}\mathbf{p}_k \rangle},$$

$$\mathbf{s}_k = \mathbf{r}_k - \alpha_k AK^{-1}\mathbf{p}_k,$$

$$\omega_k = \frac{(AK^{-1}\mathbf{s}_k, \mathbf{s}_k)}{(AK^{-1}\mathbf{s}_k, AK^{-1}\mathbf{s}_k)},$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k K^{-1}\mathbf{p}_k + \omega_k K^{-1}\mathbf{s}_k,$$

$$\mathbf{r}_{k+1} = \mathbf{s}_k - \omega_k AK^{-1}\mathbf{s}_k,$$

$$\beta_k = \frac{\alpha_k}{\omega_k} \times \frac{\langle \mathbf{r}_0^\#, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_0^\#, \mathbf{r}_k \rangle},$$

End Do

Alg.1 の反復部分における前処理演算は  $K^{-1}\mathbf{p}_k$  と  $K^{-1}\mathbf{s}_k$  の 2 つである.

\*4 本稿では収束判定の議論は扱わないので, その記述は省略する. 収束判定に関する議論は文献 [1], [7] などがある. 特に文献 [7] の研究から派生して, 文献 [5], [6], [9] などは, 本稿の議論に至った先行研究でもある.

表 2 BiCGStab の前処理変換

Table 2 Preconditioning transformation in BiCGStab method.

方程式	前処理変換		残差ベクトルの情報
	残差ベクトル	係数行列	
線形	$\mathbf{r}_k$	$\circ : AK^{-1}$	$\circ : \mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$
双対	$K^{-T}\mathbf{r}_k^\sharp$	$\circ : K^{-T}A^T$	$\circ : \mathbf{r}_k^\sharp = \mathbf{b}^\sharp - A^T\mathbf{x}_k^\sharp$
	$\mathbf{r}_k^\sharp$	$\circ : K^{-T}A^T$	$\times : \mathbf{r}_k^\sharp = K^{-T}(\mathbf{b}^\sharp - A^T\mathbf{x}_k^\sharp)$
	$\mathbf{r}_k^\sharp$	$\times : A^TK^{-T}$	$\circ : \mathbf{r}_k^\sharp = \mathbf{b}^\sharp - A^T\mathbf{x}_k^\sharp$

「係数行列の前処理変換」欄と「残差ベクトルの情報」欄の記号：  
 $\circ$  : 適切,  $\times$  : 不適切

### 3.3 従来のアルゴリズムの問題点とそれに対する改善

一方, BiCGStab 法をはじめとする“双ランチョス系統”のアルゴリズムでは, 線形方程式 (1) に対する双対系を基にアルゴリズムが構築されている. ここで, 双対系の方程式 (2) に対し両側から前処理を施したものは, BiCG の前処理変換 (14)~(16) のとおりであり, これらに対応する右変換は,

$$(K^{-T}A^T)\mathbf{x}^\sharp = K^{-T}\mathbf{b}^\sharp \quad (53)$$

と表される. ここで, PBiCG は一貫性があるため, どの方向から変換しても得られる前処理付きアルゴリズムは同一である. 式 (53) に基づく残差ベクトルは,

$$K^{-T}\mathbf{r}_k^\sharp = K^{-T}\mathbf{b}^\sharp - (K^{-T}A^T)\mathbf{x}_k^\sharp \quad (54)$$

である. つまり, (初期) シャドウ残差ベクトルに対する前処理変換が,

$$\tilde{\mathbf{r}}_k^\sharp \Rightarrow K^{-T}\mathbf{r}_k^\sharp \quad (55)$$

であれば, 式 (18) の関係を満たす. ところが, 従来の前処理変換 (52) に基づくと,

$$\tilde{\mathbf{r}}_k^\sharp \Rightarrow \mathbf{r}_k^\sharp \quad (56)$$

であり, (初期) シャドウ残差ベクトルは式 (29) の関係に帰着されてしまう.

式 (55) の変換と式 (56) との比較を表 2 としてまとめた. 特に双対系の係数行列を  $A^TK^{-T}$  と変換した場合 (表の最下行) には, 前処理系における式 (34), (36) の  $S_k(A^T)$  に関する式変形ができない. したがって,  $\alpha_k^{\text{PBiCG}} \neq \alpha_k^{\text{PSTAB}}$ ,  $\beta_k^{\text{PBiCG}} \neq \beta_k^{\text{PSTAB}}$  である.

合理的な変換 (55) は, 双ランチョス系統の前処理付きアルゴリズム全般に適用でき, これまで, CGS 法の場合を取り上げて議論されてきた [6], [9]. 特に CGS 法の議論では, 前処理変換の方向に対する“一貫性”や  $\alpha_k, \beta_k$  が等価となることが特徴である. そこで提案された前処理変換を式 (51) のとおり右前処理変換された方程式の下, BiCGStab 法に適用すると,

$$\tilde{\mathbf{p}}_k^{\text{STAB}} \Rightarrow K\mathbf{p}_k^{\text{STAB}}, \quad \tilde{\mathbf{s}}_k \Rightarrow \mathbf{s}_k, \quad \tilde{\mathbf{r}}_k^{\text{STAB}} \Rightarrow \mathbf{r}_k^{\text{STAB}},$$

$$\tilde{\mathbf{r}}_0^\sharp \Rightarrow K^{-T}\mathbf{r}_0^\sharp \quad (57)$$

である. これらを式 (39), (40) の前処理系に代入すると,

$$\begin{aligned} \alpha_k^{\text{PBiCG}} &= \frac{\langle \tilde{\mathbf{r}}_k^\sharp, \tilde{\mathbf{r}}_k \rangle}{\langle \tilde{\mathbf{p}}_k^\sharp, \tilde{A}\tilde{\mathbf{p}}_k \rangle} \equiv \frac{\langle \tilde{\mathbf{r}}_0^\sharp, \tilde{\mathbf{r}}_k^{\text{STAB}} \rangle}{\langle \tilde{\mathbf{r}}_0^\sharp, \tilde{A}\tilde{\mathbf{p}}_k^{\text{STAB}} \rangle} \\ &= \frac{\langle K^{-T}\mathbf{r}_0^\sharp, \mathbf{r}_k^{\text{STAB}} \rangle}{\langle K^{-T}\mathbf{r}_0^\sharp, (AK^{-1})(K\mathbf{p}_k^{\text{STAB}}) \rangle} \\ &= \frac{\langle \mathbf{r}_0^\sharp, K^{-1}\mathbf{r}_k^{\text{STAB}} \rangle}{\langle \mathbf{r}_0^\sharp, K^{-1}A\mathbf{p}_k^{\text{STAB}} \rangle} = \alpha_k^{\text{PSTAB}}, \end{aligned} \quad (58)$$

$$\begin{aligned} \beta_k^{\text{PBiCG}} &= \frac{\langle \tilde{\mathbf{r}}_{k+1}^\sharp, \tilde{\mathbf{r}}_{k+1} \rangle}{\langle \tilde{\mathbf{r}}_k^\sharp, \tilde{\mathbf{r}}_k \rangle} \equiv \frac{\tilde{\alpha}_k}{\tilde{\omega}_k} \times \frac{\langle \tilde{\mathbf{r}}_0^\sharp, \tilde{\mathbf{r}}_{k+1}^{\text{STAB}} \rangle}{\langle \tilde{\mathbf{r}}_0^\sharp, \tilde{\mathbf{r}}_k^{\text{STAB}} \rangle} \\ &= \frac{\tilde{\alpha}_k}{\tilde{\omega}_k} \times \frac{\langle K^{-T}\mathbf{r}_0^\sharp, \mathbf{r}_{k+1}^{\text{STAB}} \rangle}{\langle K^{-T}\mathbf{r}_0^\sharp, \mathbf{r}_k^{\text{STAB}} \rangle} \\ &= \frac{\tilde{\alpha}_k}{\tilde{\omega}_k} \times \frac{\langle \mathbf{r}_0^\sharp, K^{-1}\mathbf{r}_{k+1}^{\text{STAB}} \rangle}{\langle \mathbf{r}_0^\sharp, K^{-1}\mathbf{r}_k^{\text{STAB}} \rangle} = \beta_k^{\text{PSTAB}} \end{aligned} \quad (59)$$

であり, 前処理付き BiCGStab においても,  $\alpha_k, \beta_k$  が前処理付き BiCG の式 (21), (22) と同値の関係であることが分かる. このとき, 次に示す前処理付き BiCGStab アルゴリズム (改善版) が得られる.

#### Algorithm 2. 前処理付き BiCGStab 法 (改善版) :

$\mathbf{x}_0, \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0,$   
 $\langle \tilde{\mathbf{r}}_0^\sharp, \tilde{\mathbf{r}}_0 \rangle \neq 0, \text{ e.g., } \mathbf{r}_0^\sharp = K^{-1}\mathbf{r}_0, \beta_{-1} = 0,$   
 For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned} \mathbf{p}_k &= K^{-1}\mathbf{r}_k + \beta_{k-1}(\mathbf{p}_{k-1} - \omega_{k-1}K^{-1}A\mathbf{p}_{k-1}), \\ \alpha_k &= \frac{\langle \mathbf{r}_0^\sharp, K^{-1}\mathbf{r}_k \rangle}{\langle \mathbf{r}_0^\sharp, K^{-1}A\mathbf{p}_k \rangle}, \end{aligned} \quad (60)$$

$$\begin{aligned} \mathbf{s}_k &= \mathbf{r}_k - \alpha_k A\mathbf{p}_k, \\ K^{-1}\mathbf{s}_k &= K^{-1}\mathbf{r}_k - \alpha_k K^{-1}A\mathbf{p}_k, \end{aligned} \quad (61)$$

$$\begin{aligned} \omega_k &= \frac{(AK^{-1}\mathbf{s}_k, \mathbf{s}_k)}{(AK^{-1}\mathbf{s}_k, AK^{-1}\mathbf{s}_k)}, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k + \omega_k K^{-1}\mathbf{s}_k, \\ \mathbf{r}_{k+1} &= \mathbf{s}_k - \omega_k AK^{-1}\mathbf{s}_k, \\ \beta_k &= \frac{\alpha_k}{\omega_k} \times \frac{\langle \mathbf{r}_0^\sharp, K^{-1}\mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_0^\sharp, K^{-1}\mathbf{r}_k \rangle}, \end{aligned} \quad (62)$$

End Do

Alg.2 の反復部分には 3 つの前処理演算  $K^{-1}\mathbf{s}_k, K^{-1}\mathbf{r}_k, K^{-1}A\mathbf{p}_k$  が表現されており, 一見, 前処理に関する演算量が増加したように見えるが,  $K^{-1}\mathbf{s}_k$  自体は式 (61) により計算不要となる.  $K^{-1}\mathbf{r}_k$  と  $K^{-1}A\mathbf{p}_k$  の前処理演算は, すでに式 (62) ( $k=0$  では反復前に計算) と式 (60) の部分で計算済みである.

表 3 各テスト問題に対する計算結果  
Table 3 Numerical results for each test problem.

Matrix	N	NNZ	Conventional (Alg.1)			Improved (Alg.2)		
			Iter.	TRR	Time	Iter.	TRR	Time
cryg2500	2500	12349	314	-7.88	7.46e-2	<b>119</b>	-10.62	3.02e-2
cryg10000	10000	49699	No convergence			524	-9.55	5.43e-1
fs_760_2	760	5739	<b>102</b>	-12.07	9.60e-3	149	-12.32	1.40e-2
fs_760_3	760	5816	1938	-12.77	1.71e-1	<b>1080</b>	-12.23	9.67e-2
jpwh_991	991	6027	Breakdown			18	-13.35	3.04e-3
memplus	17758	99147	376	-12.21	1.00e0	<b>342</b>	-12.00	9.10e-1
olm5000	5000	19996	No convergence			27	-12.07	1.21e-2
raefsky3	21200	1488768	120	-12.29	4.83e0	<b>92</b>	-12.35	3.83e0
watt_2	1856	11550	144	-12.38	3.05e-2	<b>139</b>	-12.01	3.01e-2

#### 4. 数値実験

前章までの議論で、本提案の改善版 PBiCGStab の方が合理的なアルゴリズムであることが確認された。本章では、従来版と改善版とで、どの程度、求解状況への影響があるのか確認する。

線形方程式のテスト問題として、The University of Florida Sparse Matrix Collection [16] と Matrix Market [12] の中から線形方程式向きの行列を用いて求解問題を用意した。右辺項のベクトルは、解ベクトルの全要素を 1.0 とし式 (1) に代入して生成した。求解アルゴリズムは、数値計算ライブラリ Lis (Library of Iterative Solvers for Linear Systems) [11] のバージョン 1.1.2 逐次コードを倍精度で使用し、コンパイルオプションは、Lis の Makefile に記載されている値を用いた。アルゴリズムに与える初期解は  $x_0 = 0$ 、初期シャドウ残差ベクトルは、前処理系における  $\langle \tilde{r}_0^\#, \tilde{r}_0 \rangle \neq 0$  の関係を有するよう、各アルゴリズム中に記載したとおり Alg.1 では  $r_0^\# = r_0$ 、Alg.2 では  $r_0^\# = K^{-1}r_0$  とした。収束判定は、アルゴリズム中の残差ベクトルに対する相対残差ノルムが  $\|r_k\|_2 / \|b\|_2 \leq 1.0 \times 10^{-12}$  を満たすときとした ( $r_k$  はアルゴリズム中の残差ベクトル、 $k$  は反復回数)。最大反復回数には基本的に行列のサイズを用いたが、問題によっては、収束の振舞いの確認のため、上限を増やしたものもある。これらのテスト条件は、数値計算アルゴリズムに対する典型的な評価方法のうちの 1 つである。数値実験に用いた計算サーバは、DELL Precision T7400 (Intel Xeon E5420, 2.5 GHz, MEM: 16 GB), Cent OS (Kernel 2.6.18) で、コンパイラは Intel icc 10.1 である。

ここでは、前処理演算は ILU(0) を用い BiCGStab で各問題を求解した。結果は表 3 のとおりである。N は問題サイズ、NNZ は非零要素数、従来版による結果は「Conventional (Alg.1)」、本提案の改善版による結果は「Improved (Alg.2)」の欄に示し、各々の欄中の項目は、左から「収束までの所要反復回数 (Iter.)」、「数値解を代入したときの相対残差ノルムの  $\log_{10}$  (TRR)」、参考情報としての「求解時間 [sec]

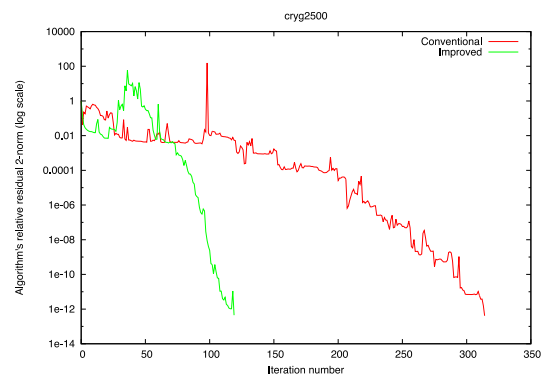


図 2 収束履歴グラフ (cryg2500)

Fig. 2 Convergence history graph (cryg2500).

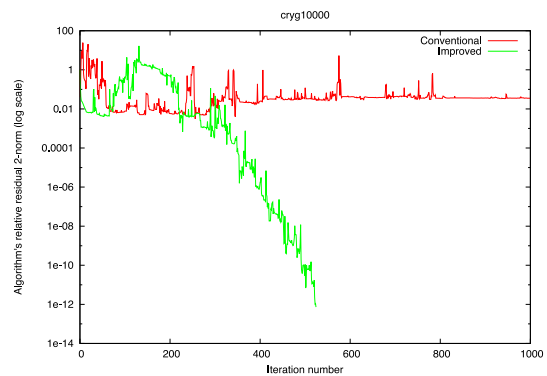


図 3 収束履歴グラフ (cryg10000)

Fig. 3 Convergence history graph (cryg10000).

(Time)」である。また、所要反復回数の比較では、少ない方を太字で示している\*5。

jpwh\_991 については、Alg.1 のみブレイクダウンし、Alg.2 では正常に収束している。この結果については、ILU(0)-BiCGStab に限らず、他の前処理演算や他の双ランチョス系アルゴリズムでも同様となること、体系的性能評価 [8]

\*5 本研究での議論は、前処理変換方法の違いによる前処理付き BiCGStab のアルゴリズム構造に対する議論が主であるため、求解時間や収束履歴グラフ (図 2~5 いずれも赤線が従来版、緑線が改善版である) は参考情報としてである。計算時間への影響が大きい反復部分の演算量については、前述のとおり、Alg.1 と Alg.2 とでほとんど違いはない。



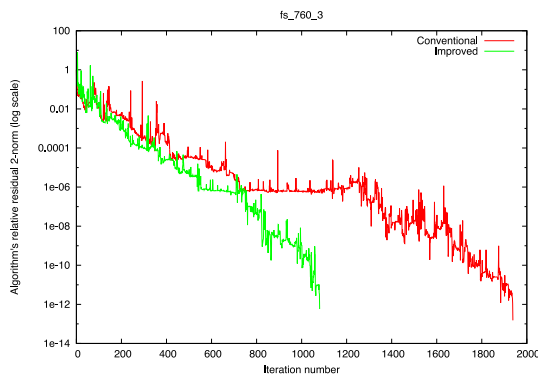


図 4 収束履歴グラフ (fs\_760\_3)

Fig. 4 Convergence history graph (fs\_760\_3).

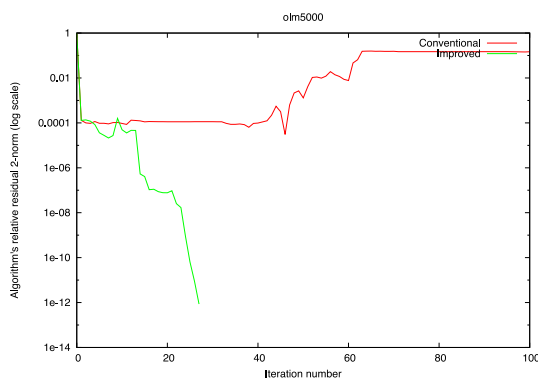


図 5 収束履歴グラフ (olm5000)

Fig. 5 Convergence history graph (olm5000).

により確認されている [9].

cryg10000, olm5000 に対し, Alg.1 では収束せず, Alg.2 では収束した. cryg2500 では, Alg.1 は, Alg.2 と比較して, Iter が多いうえに, TRR では劣っている.

これら以外は, 収束が速いか遅いかという点で若干の差異が確認される程度であった. 両アルゴリズムの求解状況に関する分析は, PBiCG に遡って解析する必要もあると考えており, 今後の課題である.

## 5. まとめ

本稿では前処理付き BiCGStab 法の改善アルゴリズムを提案した. 従来の研究成果として, PBiCG に CGS 導出の手順を適用し, アルゴリズム構築のロジックと照らし合わせたうえで合理的な PCGS が提案されている. ただし, PBiCG, PCGS とともに係数は  $\alpha_k, \beta_k$  のみであり, MR 演算をとまわず, 前処理変換方向に関して一致性を有する. それに対し, 本稿では, 係数  $\omega_k$  による MR 演算を有し一致性のない PBiCGStab に対しても, 改善版 PCGS の手法を拡張して適用でき, 数理的にも合理的であることを示した.

数値実験からも, 本提案の改善版アルゴリズムの求解状況における優れた結果が確認された. 改善版の反復部分の演算量は従来版とほぼ同じであるため, 求解時間の点でも

良好であると期待できる. しかしながら, 改善版であっても反復求解中のブレークダウンの可能性は残っている. これは前処理なしの BiCG, CGS, BiCGStab 自体の特性に起因するものである.

本研究による改善手法は, PBiCGStab 以後に提案されてきている, MR 演算を有する多くのアルゴリズム (PGP-BiCG, PBiCGStab( $\ell$ ), ほか) にも適用できると考えられるため, 本稿での議論の影響範囲は大きいと期待できる. 前処理付き BiCGStab の Alg.1 と Alg.2 は, Xabclib [19] に実装されている.

謝辞 第 1 著者 (S.I.) は, 本研究の PCGS 部分をはじめ, 日頃から多岐にわたり議論していただいている東京大学大学院情報理工学系研究科の杉原正顕教授に感謝申し上げます. 本研究の一部は文部科学省「e-サイエンス実現のためのシステム統合・連携ソフトウェアの研究開発」シームレス高生産・高性能プログラミング環境, および, 文部科学省科学研究費補助金基盤研究 (B) (21300007, 21300017) の支援を受けている.

## 参考文献

- [1] Barrett, R. et al.: *Templates for the solution of linear systems: Building Blocks for Iterative Methods*, SIAM (1994). 長谷川里美, 長谷川秀彦, 藤野清次: 反復法 Templates, 朝倉書店 (1996).
- [2] Fletcher, R.: *Conjugate Gradient Methods for Indefinite Systems*, *Numerical Analysis Dundee 1975*, Watson, G. (Ed.), *Lecture Notes in Mathematics*, 506, pp.73-89, Springer-Verlag (1976).
- [3] 藤野清次, 張 紹良: 反復法の数理, 朝倉書店 (1996).
- [4] 伊理正夫: 岩波講座応用数学「線形代数 II」, 岩波書店 (1997).
- [5] 伊藤祥司, 杉原正顕: 線形方程式求解に対するクリロフ部分空間法の前処理系に着目した体系的な特性分析, 日本応用数学会 2009 年度年会講演予稿集, 大阪大学 (Sep. 2009).
- [6] 伊藤祥司, 杉原正顕: 双ランチョス系統の前処理付きアルゴリズムの改善, 計算工学講演会論文集, Vol.15, pp.171-174, 九州大学 (May 2010).
- [7] 伊藤祥司, 杉原正顕, 姫野龍太郎: クリロフ部分空間法に対する前処理方式と収束判定について, 情報処理学会論文誌: コンピューティングシステム (ACS), Vol.3, No.2, pp.9-19 (2010).
- [8] Itoh, S. and Sugihara, M.: Systematic performance evaluation of linear solvers using quality control techniques, *Software Automatic Tuning From Concepts to State-of-the-Art Results*, Naono, K., Teranishi, K., Cavazos, J. and Suda, R. (Eds.), pp.135-152, Springer (2010).
- [9] 伊藤祥司, 杉原正顕: 導出過程に着目した前処理付き CGS 法の適切なアルゴリズム, 情報処理学会研究報告, Vol.2011-HPC-130, No.46, pp.1-10 (2011).
- [10] Lanczos, C.: Solution of Systems of Linear Equations by Minimized Iterations, *J. Res. Nat. Bur. Standards*, Vol.49, pp.33-53 (1952).
- [11] Lis, available from <http://www.ssisec.org/lis/>.
- [12] Matrix Market, available from <http://math.nist.gov/MatrixMarket/>.
- [13] 名取 亮: BCG 法と CGS 法, 数理解析研究所講究録,

- Vol.613, pp.135-143 (1987).
- [14] Sonneveld, P.: CGS, A fast Lanczos-type solver for non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol.10, No.1, pp.36-52 (1989).
  - [15] 杉原正顕, 室田一雄: 線形計算の数理, 岩波書店 (2009).
  - [16] The University of Florida Sparse Matrix Collection, available from <http://www.cise.ufl.edu/research/sparse/matrices/>.
  - [17] Van der Vorst, H.A.: BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.*, Vol.13, No.2, pp.631-644 (1992).
  - [18] Van der Vorst, H.A.: *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press (2003).
  - [19] Xabclib project, available from <http://www.abc-lib.org/Xabclib/>.



伊藤 祥司 (正会員)

2001年3月筑波大学大学院工学研究科博士課程修了。博士(工学)。筑波大学大学院システム情報工学研究科, 理化学研究所を経て, 現在, 東京大学情報基盤センター特任准教授。数値計算アルゴリズムの体系的性能評価, 求解プロセスに対する品質管理の研究に従事。日本応用数理学会, SIAM, ACM 各会員。



片桐 孝洋 (正会員)

東京大学情報基盤センター准教授。1994年豊田工業高等専門学校情報工学科卒業。1996年京都大学工学部情報工学科卒業。2001年東京大学大学院理学系研究科情報科学専攻博士課程修了。博士(理学)。2001年4月日本学術振興会特別研究員PD, 12月科学技術振興機構研究者, 2002年6月電気通信大学大学院情報システム学研究科助手, 2005年3月から2006年1月米国カリフォルニア大学バークレー校コンピュータサイエンス学科訪問学者, 2007年4月東京大学情報基盤センター特任准教授を経て, 2011年12月より現職。超並列数値計算アルゴリズム, およびソフトウェア自動チューニングの研究に従事。2002年情報処理学会山下記念研究賞受賞。2011年文部科学大臣表彰若手科学者賞受賞。日本応用数理学会, ACM, IEEE-CS, SIAM 等各会員。



櫻井 隆雄 (正会員)

1980年生。2003年東京大学工学部電子情報工学科卒業。2005年同大学院情報理工学系研究科電子情報学専攻修士課程修了。同年(株)日立製作所入社。以来, 並列計算機向け行列ライブラリ, 自動チューニング, およびクライアントモニタリングシステムの研究開発に従事。日本応用数理学会会員。



猪貝 光祥

1963年生。1987年横浜市立大学文理学部物理課程卒業。同年現(株)日立超LSIシステムズ入社。以来, 科学技術計算用ソフトウェアおよびその並列化手法に関する研究開発に従事。



大島 聡史 (正会員)

東京大学情報基盤センター助教。2009年電気通信大学大学院情報システム学研究科博士課程修了。博士(工学)。同年4月同研究科にて博士研究員。同年10月より現職。GPGPU, 並列計算に関する研究に従事。ACM,

SIAM 等各会員。



黒田 久泰 (正会員)

愛媛大学大学院理工学研究科電子情報工学専攻准教授。1993年名古屋大学理学部物理学卒業。1995年京都大学大学院工学研究科応用システム科学専攻修士課程修了。2000年東京大学大学院理学系研究科情報科学専攻博士課程単位取得退学。同年同大学情報基盤センター助手。2007年同助教。2009年より現職。高性能計算, 自動チューニングに関する研究に従事。ACM, IEEE, SIAM, 日本応用数理学会, 人工知能学会各会員。



直野 健 (正会員)

1968年生. 1994年京都大学大学院理学研究科数理解析専攻修士課程修了. 同年(株)日立製作所中央研究所入社. 現在,(株)日立製作所中央研究所主任研究員,博士(工学). 並列計算機SR2201, SR8000, SR11000, SR16000

向け行列計算ライブラリの研究開発に従事. また,自動チューニング,業務モニタリングシステムの研究開発に従事. 2000年JSPP最優秀論文賞受賞,2006年日本計算工学会奨励賞受賞,2009年関東地方発明賞発明奨励賞受賞. 日本応用数学会,日本計算工学会各会員.