

# 様々な前処理付き CGS に対する分析と それに基づく新アルゴリズムの提案

伊藤 祥司<sup>1,a)</sup> 杉原 正顯<sup>2</sup>

概要：従来から用いられている典型的な 2 種類の前処理付き CGS(PCGS) アルゴリズムの長所と短所について分析し、各々の長所のみを有する改善版 PCGS アルゴリズムを提案する。本稿では、これら PCGS の前処理系における残差ベクトルの構造、および、生成されるクリロフ部分空間と解の構造について議論する。数値実験により、分析内容を数値的にも検証し、本提案手法の有効性も確認する。

## 1. はじめに

非対称な係数行列 (サイズ  $n \times n$ ) を持つ線形方程式

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

に対する解法の一つである CGS 法 (Conjugate Gradient Squared method, 自乗共役勾配法) [15] を基にした様々な前処理付きアルゴリズムについて議論する。CGS 法はクリロフ部分空間法の一系統である双ランチョス系統の解法であり、双ランチョス系アルゴリズムは、(1) の係数行列を転置して構成される双対系 (シャドウ)

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

を導入して構築された BiCG 法 (BiConjugate Gradient method, 双共役勾配法) [4], [13] から発展した解法である。同じくクリロフ部分空間法であるアーノルディ系統の解法と比較して一復当たりの演算量と所要メモリ量が少ないことが特長である。BiCG 法を基にして CGS 法が提案され、さらには、CGS 法導出の発想に基づいて BiCGStab 法 (BiConjugate Gradient Stabilized method, 安定化双共役勾配法) [17] をはじめとする様々な解法へと発展してきている。これらのクリロフ部分空間法は、大規模な数値シミュレーションで頻りに用いられる重要な解法群である。

双ランチョス系統も含めた多くの反復解法では、それらの収束性を向上させるよう前処理付きアルゴリズムが用いられる。したがって、CGS 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、CGS 法だけの影響に留まらず、双ランチョス系統のアルゴリズム全般に対して同様の影響をもたらす得る、重要な位置付けである。

一般に、初期解  $x_0$  を用い初期残差ベクトルを  $r_0 = b - Ax_0$  と算出し、 $A$  と  $r_0$  から生成されるクリロフ列が張る  $k$  次元のクリロフ部分空間は  $\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{k-1}r_0\}$  と表わされる。前処理無しのクリロフ部分空間法においては、生成されるクリロフ部分空間  $\mathcal{K}_k(A, r_0)$  が、

$$x_k \in x_0 + \mathcal{K}_k(A, r_0) \quad (3)$$

を満たすように近似解  $x_k$  を作り出す。ところが、前処理付きのクリロフ部分空間法では、アルゴリズム中の行列とベクトルの前処理変換次第では異なるアルゴリズムが導出され、生成されるクリロフ部分空間も異なる場合がある。作り出される近似解も  $x_k$  の形として素直な構造となるアルゴリズムもあれば、それとは異なる構造のものもある。

本稿では、様々な前処理付き CGS (PCGS: Preconditioned CGS) アルゴリズムについて議論する。まず、従来から用いられてきている典型的な 2 種類の PCGS アルゴリズムを取り上げ、両者が生成するクリロフ部分空間と解の構造、および、その構造に基づく長所と短所について議論する。その上で、近年提案された 2 種類の改善版 PCGS[12] は、前述の両アルゴリズムの長所を兼ね備えていることを、クリロフ部分空間の構造の解析により示す。さらに、これら 4 種類の PCGS 間の体系的な解析を行う。

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)  $M$  を伴い記述された求解アルゴリズム、および、 $M$  に基づく演算子により変換された系を指すこととし、“コレスキー分解”や“不完全 LU”などに基づく、いわゆる、“前処理演算のアル

<sup>1</sup> 東京大学 情報基盤センター  
Information Technology Center, The University of Tokyo  
<sup>2</sup> 青山学院大学 理工学部 物理・数理学科  
Department of Physics and Mathematics, Aoyama Gakuin University  
a) itosho@cc.u-tokyo.ac.jp

ゴリズム”とは異なる．一例として，前処理系においては，元の線形方程式 (1) に対し， $A \approx M = M_L M_R$  の前処理行列を用い，

$$\begin{aligned} \tilde{A}\tilde{x} &= \tilde{b}, \\ \tilde{A} &\Rightarrow M_L^{-1} A M_R^{-1}, \quad \tilde{x} \Rightarrow M_R x, \quad \tilde{b} \Rightarrow M_L^{-1} b \end{aligned} \quad (4)$$

などと変換することに相当する．しかし，実際には (4) の変換自体は行なわず，(1) を求解するアルゴリズムに対し，(4) 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，前処理変換される行列やベクトルに対しては  $\tilde{A}$  のように “~ (チルダ)” を付して表す．

本論文の構成として，2 節では議論の背景である CGS 法導出過程と前処理付き BiCG の導出について概説し，BiCG 法に現れる  $A$  と  $A^T$  から構成される双対構造とアルゴリズム中のスカラー係数  $\alpha_k$ ， $\beta_k$  の特徴について説明する．3 節にて本稿で扱う様々な前処理付き CGS アルゴリズムについて議論する．まず，従来から用いられている典型的な 2 種類の PCGS アルゴリズムの長所と短所について説明した上で，文献 [12] で提案された 2 種類の改善版 PCGS は，これらの長所を併せ持つアルゴリズムであることを示す．4 節にて，3 節における議論を数値的に確認する．5 節でまとめる．

## 2. CGS 法の導出過程と前処理付き BiCG の導出

本節では，次節以後の議論の準備として，前処理無し BiCG 法と前処理無し CGS 法の導出，および，BiCG 法の前処理付きアルゴリズムの導出の概要を述べる．

### 2.1 BiCG 法のアルゴリズム構成

BiCG 法 [4], [13] 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

$$\mathbf{r}_k = R_k(A)\mathbf{r}_0, \quad (5)$$

$$\mathbf{r}_k^\sharp = R_k(A^T)\mathbf{r}_0^\sharp \quad (6)$$

と探索方向ベクトル

$$\mathbf{p}_k = P_k(A)\mathbf{r}_0, \quad (7)$$

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

について，

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

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

を満たす． $(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}$  はベクトル  $\mathbf{u}$  とベクトル  $\mathbf{v}$  の内積を表す．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) であり，

$\left(\mathbf{r}_0^\sharp, \mathbf{r}_0\right) \neq 0$  の条件を満たすよう  $\mathbf{r}_0^\sharp$  を設定する．本稿の議論では，この条件を確実に満たす一般的な方法として  $\mathbf{r}_0^\sharp = \mathbf{r}_0$  と設定する．これ以外の設定方法については別の議論である．

(5) ~ (8) で示した  $k$  次多項式は，

$$\begin{aligned} R_0(\lambda) &= 1, \quad P_0(\lambda) = 1, \\ R_k(\lambda) &= R_{k-1}(\lambda) - \alpha_{k-1} \lambda P_{k-1}(\lambda), \\ P_k(\lambda) &= R_k(\lambda) + \beta_{k-1} P_{k-1}(\lambda) \end{aligned}$$

の漸化式を満たす．具体的な BiCG 法のアルゴリズム記述は次のとおりである．

Algorithm 1. BiCG method:

$\mathbf{x}_0$  is an initial guess,  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ , set  $\beta_{-1} = 0$ ,  
 $\left(\mathbf{r}_0^\sharp, \mathbf{r}_0\right) \neq 0$ , e.g.,  $\mathbf{r}_0^\sharp = \mathbf{r}_0$ ,  
For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned} \mathbf{p}_k &= \mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1}, \quad \mathbf{p}_k^\sharp = \mathbf{r}_k^\sharp + \beta_{k-1} \mathbf{p}_{k-1}^\sharp, \\ \alpha_k &= \frac{\left(\mathbf{r}_k^\sharp, \mathbf{r}_k\right)}{\left(\mathbf{p}_k^\sharp, A\mathbf{p}_k\right)}, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k, \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A\mathbf{p}_k, \quad \mathbf{r}_{k+1}^\sharp = \mathbf{r}_k^\sharp - \alpha_k A^T \mathbf{p}_k^\sharp, \\ \beta_k &= \frac{\left(\mathbf{r}_{k+1}^\sharp, \mathbf{r}_{k+1}\right)}{\left(\mathbf{r}_k^\sharp, \mathbf{r}_k\right)}, \end{aligned}$$

End Do

本稿中では，特に断りの無い場合の収束判定方法は，

$$\frac{\|\mathbf{r}_k\|}{\|\mathbf{b}\|} \leq \varepsilon. \quad (9)$$

### 2.2 CGS 法の導出

CGS 法は BiCG 法に現れる  $A^T$  に伴う演算を回避するよう BiCG の中の変数を変換して得られた解法であり [15]，多項式表現 (5) ~ (8) を BiCG の  $\alpha_k$  と  $\beta_k$  を構成する演算に代入し，

$$\begin{aligned} \left(\mathbf{r}_k^\sharp, \mathbf{r}_k\right) &= \left(R_k(A^T)\mathbf{r}_0^\sharp, R_k(A)\mathbf{r}_0\right) = \left(\mathbf{r}_0^\sharp, R_k^2(A)\mathbf{r}_0\right), \\ \left(\mathbf{p}_k^\sharp, A\mathbf{p}_k\right) &= \left(P_k(A^T)\mathbf{r}_0^\sharp, AP_k(A)\mathbf{r}_0\right) = \left(\mathbf{r}_0^\sharp, AP_k^2(A)\mathbf{r}_0\right) \end{aligned}$$

とする．この変形は，線形系の係数行列  $A$  に対し，双対系の係数行列が  $A^T$  であるため成り立つ．ここで，

$$\mathbf{r}_k^{\text{CGS}} \equiv R_k^2(A)\mathbf{r}_0, \quad \mathbf{p}_k^{\text{CGS}} \equiv P_k^2(A)\mathbf{r}_0 \quad (10)$$

とおくと，

$$\alpha_k^{\text{BiCG}} = \frac{\left(\mathbf{r}_k^\sharp, \mathbf{r}_k\right)}{\left(\mathbf{p}_k^\sharp, A\mathbf{p}_k\right)} = \frac{\left(\mathbf{r}_0^\sharp, R_k^2(A)\mathbf{r}_0\right)}{\left(\mathbf{r}_0^\sharp, AP_k^2(A)\mathbf{r}_0\right)}$$

$$\begin{aligned} &= \frac{\left( \mathbf{r}_0^\#, \mathbf{r}_k^{\text{CGS}} \right)}{\left( \mathbf{r}_0^\#, A\mathbf{p}_k^{\text{CGS}} \right)} \equiv \alpha_k^{\text{CGS}}, \\ \beta_k^{\text{BiCG}} &= \frac{\left( \mathbf{r}_{k+1}^\#, \mathbf{r}_{k+1} \right)}{\left( \mathbf{r}_k^\#, \mathbf{r}_k \right)} = \frac{\left( \mathbf{r}_0^\#, R_{k+1}^2(A)\mathbf{r}_0 \right)}{\left( \mathbf{r}_0^\#, R_k^2(A)\mathbf{r}_0 \right)} \\ &= \frac{\left( \mathbf{r}_0^\#, \mathbf{r}_{k+1}^{\text{CGS}} \right)}{\left( \mathbf{r}_0^\#, \mathbf{r}_k^{\text{CGS}} \right)} \equiv \beta_k^{\text{CGS}} \end{aligned}$$

が得られる。つまり、BiCG と CGS のスカラー係数  $\alpha_k$  と  $\beta_k$  は、式変形と変数の置き換えによる等価な関係であることが分かる。最終的に次に表す CGS 法が得られる\*1。

Algorithm 2. CGS method:

$\mathbf{x}_0$  is an initial guess,  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ , set  $\beta_{-1} = 0$ ,  
 $\left( \mathbf{r}_0^\#, \mathbf{r}_0 \right) \neq 0$ , e.g.,  $\mathbf{r}_0^\# = \mathbf{r}_0$ ,  
For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned} \mathbf{u}_k &= \mathbf{r}_k + \beta_{k-1}\mathbf{q}_{k-1}, \\ \mathbf{p}_k &= \mathbf{u}_k + \beta_{k-1}(\mathbf{q}_{k-1} + \beta_{k-1}\mathbf{p}_{k-1}), \\ \alpha_k &= \frac{\left( \mathbf{r}_0^\#, \mathbf{r}_k \right)}{\left( \mathbf{r}_0^\#, A\mathbf{p}_k \right)}, \\ \mathbf{q}_k &= \mathbf{u}_k - \alpha_k A\mathbf{p}_k, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k(\mathbf{u}_k + \mathbf{q}_k), \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A(\mathbf{u}_k + \mathbf{q}_k), \\ \beta_k &= \frac{\left( \mathbf{r}_0^\#, \mathbf{r}_{k+1} \right)}{\left( \mathbf{r}_0^\#, \mathbf{r}_k \right)}, \end{aligned}$$

End Do

CGS 法では、(10) により、1 反復当たりクリロフ部分空間が 2 次元拡張されるため、 $k$  回目の反復においては

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_{2k}(A, \mathbf{r}_0) \quad (11)$$

を満たす近似解  $\mathbf{x}_k$  が算出される [2], [6]。

### 2.3 前処理付き BiCG アルゴリズムの導出

(1) と (2) に対する前処理変換

$$\left( M_L^{-1} A M_R^{-1} \right) (M_R \mathbf{x}) = M_L^{-1} \mathbf{b}, \quad (12)$$

$$\left( M_R^{-T} A^T M_L^{-T} \right) \left( M_L^T \mathbf{x}^\# \right) = M_R^{-T} \mathbf{b}^\# \quad (13)$$

に基づき、各々の系のベクトルに対する前処理変換

$$\tilde{\mathbf{r}}_k \Rightarrow M_L^{-1} \mathbf{r}_k = R_k(\tilde{A})\tilde{\mathbf{r}}_0, \quad (14)$$

$$\tilde{\mathbf{p}}_k \Rightarrow M_R \mathbf{p}_k = P_k(\tilde{A})\tilde{\mathbf{p}}_0, \quad (15)$$

$$\tilde{\mathbf{r}}_k^\# \Rightarrow M_R^{-T} \mathbf{r}_k^\# = R_k(\tilde{A}^T)\tilde{\mathbf{r}}_0^\#, \quad (16)$$

\*1 本稿における各々のアルゴリズム記述では、右上の CGS 等の記号は省略する。

$$\tilde{\mathbf{p}}_k^\# \Rightarrow M_L^T \mathbf{p}_k^\# = P_k(\tilde{A}^T)\tilde{\mathbf{p}}_0^\# \quad (17)$$

を Alg.1 に適用\*2し式変形すると、前処理付き BiCG (PBiCG : Preconditioned BiCG) が得られる。

ここで、線形系の残差ベクトルと、双対系の残差ベクトル(シャドウ残差ベクトル)の前処理変換について確認すると、

$$\begin{aligned} M_L^{-1} \mathbf{r}_k &= M_L^{-1} \mathbf{b} - \left( M_L^{-1} A M_R^{-1} \right) (M_R \mathbf{x}_k), \\ M_R^{-T} \mathbf{r}_k^\# &= M_R^{-T} \mathbf{b}^\# - \left( M_R^{-T} A^T M_L^{-T} \right) \left( M_L^T \mathbf{x}_k^\# \right) \end{aligned}$$

であるが、これらの式変形を進めると次のとおり、 $\mathbf{r}_k$  自体は前処理行列の影響を受けておらず、残差ベクトルの本来の構造に帰着されることが分かる。 $\mathbf{r}_k^\#$  も同様である。

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k, \quad \mathbf{r}_k^\# = \mathbf{b}^\# - A^T \mathbf{x}_k^\#.$$

この前処理系における双直交性と双共役性は、

$$\left( \tilde{\mathbf{r}}_i^\#, \tilde{\mathbf{r}}_j \right) = 0 \quad (i \neq j) \quad (\text{前処理系の双直交性}) \quad (18)$$

$$\left( \tilde{\mathbf{p}}_i^\#, \tilde{A}\tilde{\mathbf{p}}_j \right) = 0 \quad (i \neq j) \quad (\text{前処理系の双共役性}) \quad (19)$$

であり、各々に (12) ~ (17) の前処理変換を適用し変形すると、

$$\begin{aligned} \left( \tilde{\mathbf{r}}_i^\#, \tilde{\mathbf{r}}_j \right) &\Rightarrow \left( M_R^{-T} \mathbf{r}_i^\#, M_L^{-1} \mathbf{r}_j \right) = \left( \mathbf{r}_i^\#, M^{-1} \mathbf{r}_j \right), \\ \left( \tilde{\mathbf{p}}_i^\#, \tilde{A}\tilde{\mathbf{p}}_j \right) &\Rightarrow \left( M_L^T \mathbf{p}_i^\#, \left( M_L^{-1} A M_R^{-1} \right) (M_R \mathbf{p}_j) \right) \\ &= \left( \mathbf{p}_i^\#, A\mathbf{p}_j \right). \end{aligned}$$

これらを用いて

$$\alpha_k^{\text{PBiCG}} = \frac{\left( \tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k \right)}{\left( \tilde{\mathbf{p}}_k^\#, \tilde{A}\tilde{\mathbf{p}}_k \right)} = \frac{\left( \mathbf{r}_k^\#, M^{-1} \mathbf{r}_k \right)}{\left( \mathbf{p}_k^\#, A\mathbf{p}_k \right)}, \quad (20)$$

$$\beta_k^{\text{PBiCG}} = \frac{\left( \tilde{\mathbf{r}}_{k+1}^\#, \tilde{\mathbf{r}}_{k+1} \right)}{\left( \tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k \right)} = \frac{\left( \mathbf{r}_{k+1}^\#, M^{-1} \mathbf{r}_{k+1} \right)}{\left( \mathbf{r}_k^\#, M^{-1} \mathbf{r}_k \right)}. \quad (21)$$

以上から、前処理付き BiCG アルゴリズムが得られる。

### 3. 様々な前処理付き CGS アルゴリズム

本節では、まず、従来の典型的な 2 種類の前処理付き CGS アルゴリズムを示し、各々の長所と短所について述べる (3.1 節)。次に、文献 [12] で提案された 2 種類の改善版 PCGS のアルゴリズムについて説明する (3.2 節)。これら改善版 PCGS は従来から用いられてきた両アルゴリズムの長所のみを兼ね備えたアルゴリズムであることを示し、本節で取り上げる 4 種類の PCGS 間の関係について述べる (3.3 節)。なお、本稿で議論する全 PCGS のいづれにおいても、反復部分の前処理演算は 2 つのみであるため、計算時間面については考察しない。

\*2 正確には Alg.1 を構成する行列とベクトルにチルダ記号を付けた上で、これらの変換を適用する。

### 3.1 従来から用いられている典型的な PCGS アルゴリズム

#### 3.1.1 右前処理系の従来版 PCGS アルゴリズム (Conventional PCGS)

文献 [1], [5], [17] をはじめ多くの文献や数値計算ライブラリ等で採用されている従来の前処理付き CGS アルゴリズムは, 前節で示した CGS 法の行列とベクトルにチルダ記号を付けた上で

$$\begin{aligned}\tilde{A} &\Rightarrow M_L^{-1} A M_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow M_R \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow M_L^{-1} \mathbf{b}, \quad (22) \\ \tilde{\mathbf{r}}_k &\Rightarrow M_L^{-1} \mathbf{r}_k, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow M_L^T \mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow M_L^{-1} \mathbf{p}_k, \quad \tilde{\mathbf{u}}_k \Rightarrow M_L^{-1} \mathbf{u}_k, \quad \tilde{\mathbf{q}}_k \Rightarrow M_L^{-1} \mathbf{q}_k\end{aligned}$$

と前処理変換を施し, 次のアルゴリズムとして記述される.

Algorithm 3. Conventional PCGS algorithm (Conventional):

$\mathbf{x}_0$  is an initial guess,  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ , set  $\beta_{-1} = 0$ ,  
 $(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0) \Rightarrow (\mathbf{r}_0^\#, \mathbf{r}_0) \neq 0$ , e.g.,  $\mathbf{r}_0^\# = \mathbf{r}_0$ ,  
For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned}\mathbf{u}_k &= \mathbf{r}_k + \beta_{k-1} \mathbf{q}_{k-1}, \\ \mathbf{p}_k &= \mathbf{u}_k + \beta_{k-1} (\mathbf{q}_{k-1} + \beta_{k-1} \mathbf{p}_{k-1}), \\ \alpha_k &= \frac{(\mathbf{r}_0^\#, \mathbf{r}_k)}{(\mathbf{r}_0^\#, A M^{-1} \mathbf{p}_k)}, \\ \mathbf{q}_k &= \mathbf{u}_k - \alpha_k A M^{-1} \mathbf{p}_k, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k M^{-1} (\mathbf{u}_k + \mathbf{q}_k), \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A M^{-1} (\mathbf{u}_k + \mathbf{q}_k), \\ \beta_k &= \frac{(\mathbf{r}_0^\#, \mathbf{r}_{k+1})}{(\mathbf{r}_0^\#, \mathbf{r}_k)},\end{aligned}$$

End Do

この Alg.3 は, 前処理無し CGS アルゴリズム中の全ての係数行列  $A$  に対し右から  $M^{-1}$  を作用させ, 全ての解ベクトル  $\mathbf{x}_k$  に対し左から  $M$  を作用させた上で, 必要な式変形を行なっても導出される. すなわち,

$$\begin{aligned}\tilde{A} &\Rightarrow A M^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow M \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow \mathbf{b}, \\ \tilde{\mathbf{r}}_k &\Rightarrow \mathbf{r}_k, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow \mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow \mathbf{p}_k, \quad \tilde{\mathbf{u}}_k \Rightarrow \mathbf{u}_k, \quad \tilde{\mathbf{q}}_k \Rightarrow \mathbf{q}_k\end{aligned} \quad (23)$$

であり, これは, (22) において  $M_L = I, M_R = M$  ( $I$  は単位行列) としたときの, 前処理系の方程式  $(A M^{-1})(M \mathbf{x}) = \mathbf{b}$  をそのまま当てはめたものと同じである<sup>\*3</sup>. また, 線形系の残差ベクトルは  $\mathbf{r}_k = \mathbf{b} - (A M^{-1})(M \mathbf{x}_k)$  と構成されて

<sup>\*3</sup> PCGS では前処理変換の方向に対する一致性があるため, (22) において, “ $M_L = I, M_R = M$ ” や “ $M_L = M, M_R = I$ ” の前処理変換を施しても最終的に得られる PCGS のアルゴリズムは, Alg.3 と同一である [7], [8], [10].

いる. したがって, Alg.3 は “右前処理系” である.

Alg.3 において, 初期シャドウ残差ベクトルを  $\tilde{\mathbf{r}}_0^\# \Rightarrow M_L^T \mathbf{r}_0^\#$  と変換し, 前処理系で異なる記号 (右上の  $\#$  を  $\flat$ ) へ置き換える背景としては, 前処理系の初期シャドウ残差ベクトルは  $\mathbf{r}_0^\# = \mathbf{b}^\# - A^T \mathbf{x}_0^\#$  と異なる構造の,  $\mathbf{r}_0^\flat \equiv M^{-T} \mathbf{r}_0^\#$  であることがすでに確認されているためである [12]. 初期シャドウ残差ベクトルの設定においては下記の変換を行う.

$$(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0) \Rightarrow (M_L^T \mathbf{r}_0^\#, M_L^{-1} \mathbf{r}_0) = (\mathbf{r}_0^\flat, \mathbf{r}_0)$$

これは, (14) と (16) の変換を用いた,

$$\begin{aligned}(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0) &\Rightarrow (M_R^{-T} \mathbf{r}_0^\#, M_L^{-1} \mathbf{r}_0) \\ &= (M^{-T} \mathbf{r}_0^\#, \mathbf{r}_0) \equiv (\mathbf{r}_0^\flat, \mathbf{r}_0)\end{aligned} \quad (24)$$

と同じことである<sup>\*4</sup>.

このアルゴリズムは, 従来から国際的にも標準として用いられてきている前処理付きアルゴリズムである. これは, Van der Vorst, H. A. が BiCGStab 法を提案した文献 [17] などでの記述であり, Sonneveld, P. が CGS 法を提案した文献 [15] 中の前処理付き CGS を基にして, さらに変形したものである.

#### 3.1.2 左前処理系の PCGS アルゴリズム (Left system PCGS)

Sonneveld, P. による文献 [15] に基づく PCGS は,

$$\begin{aligned}\tilde{A} &\Rightarrow M_L^{-1} A M_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow M_R \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow M_L^{-1} \mathbf{b}, \\ \tilde{\mathbf{r}}_k &\Rightarrow \mathbf{r}_k, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow \mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow \mathbf{p}_k, \quad \tilde{\mathbf{u}}_k \Rightarrow \mathbf{u}_k, \quad \tilde{\mathbf{q}}_k \Rightarrow \mathbf{q}_k\end{aligned} \quad (25)$$

として,  $A, \mathbf{x}, \mathbf{b}$  のみに対し前処理変換が施されたものであるが, この場合 線形系の残差ベクトルは  $\mathbf{r}_k = M_L^{-1} (\mathbf{b} - A \mathbf{x}_k)$  であり, 本来の  $\mathbf{r}_k = \mathbf{b} - A \mathbf{x}_k$  とは異なる, 前処理行列 (この場合は  $M_L^{-1}$ ) の影響を受けた構造である. 特に従来から用いられているのは,  $M_L = M, M_R = I$  とした “左前処理系” であり,  $A$  と  $\mathbf{b}$  のみに左から  $M^{-1}$  を作用させることに等しい<sup>\*5</sup>. 本稿では, この左前処理系の残差ベクトルを  $\mathbf{r}_k^+$  と表記することにする. つまり,  $\mathbf{r}_k^+ \equiv M^{-1} \mathbf{r}_k (= M^{-1} (\mathbf{b} - A \mathbf{x}_k))$  である. また, 同様の意図で左前処理系においては  $\mathbf{p}_k^+, \mathbf{u}_k^+, \mathbf{q}_k^+$  と表記できる.

<sup>\*4</sup> 前処理系において  $\mathbf{r}_0^\flat \equiv M^{-T} \mathbf{r}_0^\#$  の関係を有することを前提としていれば,  $(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0)$  を  $(\mathbf{r}_0^\flat, \mathbf{r}_0)$  と表現しても問題は無い. 前者は前処理無しアルゴリズムからの変換を強調した記述である. 後者は前処理付きアルゴリズムに閉じた記述であり,  $(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0) \Rightarrow (M_L^T \mathbf{r}_0^\#, M_L^{-1} \mathbf{r}_0) = (\mathbf{r}_0^\flat, \mathbf{r}_0)$  である.

<sup>\*5</sup> (25) で残差ベクトルが前処理行列の影響を受けぬよう  $M_L = I, M_R = M$  として前処理変換すると Alg.3 となる. この変換は, (22) において  $M_L = I, M_R = M$  とすることも等しい. (25) を (22) のように両側変換に基づき一般化して表すと下記のとおりである.

$$\begin{aligned}\tilde{A} &\Rightarrow M_L^{-1} A M_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow M_R \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow M_L^{-1} \mathbf{b}, \\ \tilde{\mathbf{r}}_k &\Rightarrow M_R \mathbf{r}_k^+, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow M_R^{-T} \mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow M_R \mathbf{p}_k^+, \quad \tilde{\mathbf{u}}_k \Rightarrow M_R \mathbf{u}_k^+, \quad \tilde{\mathbf{q}}_k \Rightarrow M_R \mathbf{q}_k^+.\end{aligned}$$

表 1 従来から用いられている 2 種類の PCGS アルゴリズムの問題点

	残差ベクトルの構造	生成されるクリロフ部分空間と解の構造
従来版 PCGS (Alg.3)	$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$	$M\mathbf{x}_k \in M\mathbf{x}_0 + \mathcal{K}_{2k}(AM^{-1}, \mathbf{r}_0)$
左前処理系 PCGS (Alg.4)	$\mathbf{r}_k^+ = M^{-1}(\mathbf{b} - A\mathbf{x}_k)$	$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_{2k}(M^{-1}A, \mathbf{r}_0^+)$

最終的に導出される左前処理系 PCGS アルゴリズムは下記のとおりである。これも従来から良く見かける典型的な PCGS アルゴリズムの一つであり、

$$\begin{aligned} \tilde{A} &\Rightarrow M^{-1}A, \quad \tilde{\mathbf{x}}_k \Rightarrow \mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow M^{-1}\mathbf{b}, \\ \tilde{\mathbf{r}}_k &\Rightarrow \mathbf{r}_k^+, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow \mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow \mathbf{p}_k^+, \quad \tilde{\mathbf{u}}_k \Rightarrow \mathbf{u}_k^+, \quad \tilde{\mathbf{q}}_k \Rightarrow \mathbf{q}_k^+ \end{aligned} \quad (26)$$

と前処理変換を施し、次のアルゴリズムとして記述される<sup>\*6</sup>。

Algorithm 4. Left PCGS algorithm (Left system):

$\mathbf{x}_0$  is an initial guess,  $\mathbf{r}_0^+ = M^{-1}(\mathbf{b} - A\mathbf{x}_0)$ , set  $\beta_{-1} = 0$ ,  
( $\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0$ )  $\Rightarrow$  ( $\mathbf{r}_0^\#, \mathbf{r}_0^+$ )  $\neq 0$ , e.g.,  $\mathbf{r}_0^\# = \mathbf{r}_0^+$ ,  
For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned} \mathbf{u}_k^+ &= \mathbf{r}_k^+ + \beta_{k-1}\mathbf{q}_{k-1}^+, \\ \mathbf{p}_k^+ &= \mathbf{u}_k^+ + \beta_{k-1}(\mathbf{q}_{k-1}^+ + \beta_{k-1}\mathbf{p}_{k-1}^+), \\ \alpha_k &= \frac{(\mathbf{r}_0^\#, \mathbf{r}_k^+)}{(\mathbf{r}_0^\#, M^{-1}A\mathbf{p}_k^+)}, \\ \mathbf{q}_k^+ &= \mathbf{u}_k^+ - \alpha_k M^{-1}A\mathbf{p}_k^+, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k(\mathbf{u}_k^+ + \mathbf{q}_k^+), \\ \mathbf{r}_{k+1}^+ &= \mathbf{r}_k^+ - \alpha_k M^{-1}A(\mathbf{u}_k^+ + \mathbf{q}_k^+), \\ \beta_k &= \frac{(\mathbf{r}_0^\#, \mathbf{r}_{k+1}^+)}{(\mathbf{r}_0^\#, \mathbf{r}_k^+)}, \end{aligned}$$

End Do

ただし、Alg.4 の収束判定においては、(9) をそのまま適用して  $\|\mathbf{r}_k^+\|/\|\mathbf{b}\| \leq \varepsilon$  とした場合には、極めて不完全な収束判定となる。ここで、

$$\frac{\|\mathbf{r}_k^+\|}{\|M^{-1}\mathbf{b}\|} \leq \varepsilon \quad (27)$$

と、分母に対しても  $M^{-1}$  を乗ずると概ねは改善されるが、あくまでも、残差ベクトルは本来の  $\mathbf{r}_k$  とは異なり、 $\mathbf{r}_k^+$  に対する収束判定である [9], [11]。

### 3.1.3 2 種類の従来版 PCGS アルゴリズムの比較

ここでは、従来版の Alg.3 と左前処理系の Alg.4 が生成

<sup>\*6</sup> 一般的なアルゴリズム記述では、 $\mathbf{r}_k^+, \mathbf{p}_k^+, \mathbf{u}_k^+, \mathbf{q}_k^+$  は単に  $\mathbf{r}_k, \mathbf{p}_k, \mathbf{u}_k, \mathbf{q}_k$  と記述すれば良い。本稿ではアルゴリズムの数理構造に注目した議論を行うために右上に + を付している。同様に、シャドウの # と  $\tilde{\cdot}$  についても、一般的なアルゴリズム記述では一種類の記号で表記すれば十分である。

するクリロフ部分空間と解の構造に注目し、各々の特徴と問題点についてまとめる。

Alg.3 は右前処理系  $(AM^{-1})(M\mathbf{x}) = \mathbf{b}, \mathbf{r}_k = \mathbf{b} - (AM^{-1})(M\mathbf{x}_k)$  であり、これが生成するクリロフ部分空間と解の構造との関係は、

$$M\mathbf{x}_k \in M\mathbf{x}_0 + \mathcal{K}_{2k}(AM^{-1}, \mathbf{r}_0).$$

Alg.4 は  $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$  に基づく左前処理系であり、 $\mathbf{r}_k^+ = M^{-1}(\mathbf{b} - A\mathbf{x}_k)$  のため、

$$\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{K}_{2k}(M^{-1}A, \mathbf{r}_0^+). \quad (28)$$

3.1 節の議論をまとめたものが表 1 であり、従来から用いられてきている両アルゴリズムには一長一短あることが確認できる。Alg.3 では、アルゴリズム中の残差ベクトルは  $\mathbf{b} - A\mathbf{x}_k$  に相当する情報を保持しており、収束判定も (9) で良いが、生成されるクリロフ部分空間に対する解の構造は本来の  $\mathbf{x}_k$  とは異なるものである。一方、Alg.4 では、生成されるクリロフ部分空間に対する解の構造は  $\mathbf{x}_k$  であるものの、アルゴリズム中の残差ベクトルは  $\mathbf{r}_k^+$  であり、収束判定では (27) を用いるため、本来の残差を評価していない。

## 3.2 改善版 PCGS アルゴリズム

文献 [12] では、BiCG 法から CGS 法を導出した手順に基づき、前処理付き BiCG (PBiCG) に対して CGS 法の導出手順を適用した改善版 PCGS が提案された。さらに改善版の変種も提案された。これら改善版 PCGS の  $\alpha_k$  と  $\beta_k$  は、2.3 節の前処理付き BiCG アルゴリズム中の  $\alpha_k$  と  $\beta_k$  と等価な関係である。本小節では、これら 2 種類の改善版 PCGS について説明する。

### 3.2.1 改善版 PCGS アルゴリズム 1

#### (Improved1 PCGS)

あらためて、双対系 (2) の前処理変換を確認すると、線形系の前処理変換 (4) の係数行列の転置に基づき、

$$\begin{aligned} \tilde{A}^T \tilde{\mathbf{x}}^\# &= \tilde{\mathbf{b}}^\#, \\ \tilde{A}^T &\Rightarrow M_R^{-T} A^T M_L^{-T}, \quad \tilde{\mathbf{x}}^\# \Rightarrow M_L^T \mathbf{x}^\#, \quad \tilde{\mathbf{b}}^\# \Rightarrow M_R^{-T} \mathbf{b}^\# \end{aligned}$$

である。BiCG の線形系と双対系における残差ベクトルと探索方向ベクトルの前処理変換後の多項式表現は (14) ~ (17) であるから、(20), (21) に現れる演算に代入すると、

$$\begin{aligned} (\tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k) &= (R_k(\tilde{A}^T) \tilde{\mathbf{r}}_0^\#, R_k(\tilde{A}) \tilde{\mathbf{r}}_0) \\ &= (\tilde{\mathbf{r}}_0^\#, R_k^2(\tilde{A}) \tilde{\mathbf{r}}_0), \end{aligned} \quad (29)$$

$$\begin{aligned} \left( \tilde{\mathbf{p}}_k^\#, \tilde{A}\tilde{\mathbf{p}}_k \right) &= \left( P_k(\tilde{A}^T)\tilde{\mathbf{r}}_0^\#, \tilde{A}P_k(\tilde{A})\tilde{\mathbf{r}}_0 \right) \\ &= \left( \tilde{\mathbf{r}}_0^\#, \tilde{A}P_k^2(\tilde{A})\tilde{\mathbf{r}}_0 \right) \end{aligned} \quad (30)$$

と変形できる． $\tilde{\mathbf{r}}_0^\# \Rightarrow M_R^{-T}\mathbf{r}_0^\#$ ，および，(29)，(30) においても (10) と同様

$$\begin{aligned} \tilde{\mathbf{r}}_k^{\text{CGS}} &\Rightarrow M_L^{-1}\mathbf{r}_k^{\text{CGS}} \equiv R_k^2(\tilde{A})\tilde{\mathbf{r}}_0, \\ \tilde{\mathbf{p}}_k^{\text{CGS}} &\Rightarrow M_R\mathbf{p}_k^{\text{CGS}} \equiv P_k^2(\tilde{A})\tilde{\mathbf{r}}_0 \end{aligned}$$

とすると，2.3 節にて述べた PBiCG の  $\alpha^{\text{PBiCG}}$ ， $\beta^{\text{PBiCG}}$ ，すなわち，(20)，(21) は各々，

$$\begin{aligned} \alpha_k^{\text{PBiCG}} &= \frac{\left( \tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k \right)}{\left( \tilde{\mathbf{p}}_k^\#, \tilde{A}\tilde{\mathbf{p}}_k \right)} = \frac{\left( \mathbf{r}_0^\#, M^{-1}\mathbf{r}_k^{\text{CGS}} \right)}{\left( \mathbf{r}_0^\#, M^{-1}A\mathbf{p}_k^{\text{CGS}} \right)} \equiv \alpha_k^{\text{PCGS}}, \\ \beta_k^{\text{PBiCG}} &= \frac{\left( \tilde{\mathbf{r}}_{k+1}^\#, \tilde{\mathbf{r}}_{k+1} \right)}{\left( \tilde{\mathbf{r}}_k^\#, \tilde{\mathbf{r}}_k \right)} = \frac{\left( \mathbf{r}_0^\#, M^{-1}\mathbf{r}_{k+1}^{\text{CGS}} \right)}{\left( \mathbf{r}_0^\#, M^{-1}\mathbf{r}_k^{\text{CGS}} \right)} \equiv \beta_k^{\text{PCGS}} \end{aligned}$$

であり，PBiCG と PCGS の各々のスカラー係数  $\alpha_k$ ， $\beta_k$  は等価な関係である．その他のベクトルも同様に式変形すると，次の前処理付き CGS アルゴリズムが得られる [12] ．

Algorithm 5. Improved PCGS algorithm (Improved1):

$\mathbf{x}_0$  is an initial guess,  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ , set  $\beta_{-1} = 0$ ,  
 $(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0) \Rightarrow (\mathbf{r}_0^\#, M^{-1}\mathbf{r}_0) \neq 0$ , e.g.,  $\mathbf{r}_0^\# = M^{-1}\mathbf{r}_0$ ,  
For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned} \mathbf{u}_k^+ &= M^{-1}\mathbf{r}_k + \beta_{k-1}\mathbf{q}_{k-1}^+, \\ \mathbf{p}_k^+ &= \mathbf{u}_k^+ + \beta_{k-1}(\mathbf{q}_{k-1}^+ + \beta_{k-1}\mathbf{p}_{k-1}^+), \\ \alpha_k &= \frac{\left( \mathbf{r}_0^\#, M^{-1}\mathbf{r}_k \right)}{\left( \mathbf{r}_0^\#, M^{-1}A\mathbf{p}_k^+ \right)}, \\ \mathbf{q}_k^+ &= \mathbf{u}_k^+ - \alpha_k M^{-1}A\mathbf{p}_k^+, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k(\mathbf{u}_k^+ + \mathbf{q}_k^+), \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A(\mathbf{u}_k^+ + \mathbf{q}_k^+), \\ \beta_k &= \frac{\left( \mathbf{r}_0^\#, M^{-1}\mathbf{r}_{k+1} \right)}{\left( \mathbf{r}_0^\#, M^{-1}\mathbf{r}_k \right)}, \end{aligned}$$

End Do

この Alg.5 は，(12) と等価となるよう，CGS 法に対し，

$$\begin{aligned} \tilde{A} &\Rightarrow M_L^{-1}AM_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow M_R\mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow M_L^{-1}\mathbf{b}, \\ \tilde{\mathbf{r}}_k &\Rightarrow M_L^{-1}\mathbf{r}_k, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow M_R^{-T}\mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow M_R\mathbf{p}_k^+, \quad \tilde{\mathbf{u}}_k \Rightarrow M_R\mathbf{u}_k^+, \quad \tilde{\mathbf{q}}_k \Rightarrow M_R\mathbf{q}_k^+ \end{aligned}$$

と前処理変換したことに等しい．ここでも， $\mathbf{p}_k^+$ ， $\mathbf{u}_k^+$ ， $\mathbf{q}_k^+$  と表記できる．

### 3.2.2 改善版 PCGS アルゴリズム 2

(Improved2 PCGS)

従来版の Alg.3 における前処理変換 (22) に対して初期

シャドウ残差ベクトルの前処理変換のみを変更し，

$$\begin{aligned} \tilde{A} &\Rightarrow M_L^{-1}AM_R^{-1}, \quad \tilde{\mathbf{x}}_k \Rightarrow M_R\mathbf{x}_k, \quad \tilde{\mathbf{b}} \Rightarrow M_L^{-1}\mathbf{b}, \\ \tilde{\mathbf{r}}_k &\Rightarrow M_L^{-1}\mathbf{r}_k, \quad \tilde{\mathbf{r}}_0^\# \Rightarrow M_R^{-T}\mathbf{r}_0^\#, \\ \tilde{\mathbf{p}}_k &\Rightarrow M_L^{-1}\mathbf{p}_k, \quad \tilde{\mathbf{u}}_k \Rightarrow M_L^{-1}\mathbf{u}_k, \quad \tilde{\mathbf{q}}_k \Rightarrow M_L^{-1}\mathbf{q}_k \end{aligned}$$

とすると次の前処理付き CGS アルゴリズムが得られる．これは Alg.5 と等価である [12] ．

Algorithm 6. Another improved PCGS algorithm (Improved2):

$\mathbf{x}_0$  is an initial guess,  $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ , set  $\beta_{-1} = 0$ ,  
 $(\tilde{\mathbf{r}}_0^\#, \tilde{\mathbf{r}}_0) \Rightarrow (\mathbf{r}_0^\#, M^{-1}\mathbf{r}_0) \neq 0$ , e.g.,  $\mathbf{r}_0^\# = M^{-1}\mathbf{r}_0$ ,  
For  $k = 0, 1, 2, \dots$ , until convergence, Do:

$$\begin{aligned} \mathbf{u}_k &= \mathbf{r}_k + \beta_{k-1}\mathbf{q}_{k-1}, \\ \mathbf{p}_k &= \mathbf{u}_k + \beta_{k-1}(\mathbf{q}_{k-1} + \beta_{k-1}\mathbf{p}_{k-1}), \\ \alpha_k &= \frac{\left( M^{-T}\mathbf{r}_0^\#, \mathbf{r}_k \right)}{\left( M^{-T}\mathbf{r}_0^\#, AM^{-1}\mathbf{p}_k \right)}, \\ \mathbf{q}_k &= \mathbf{u}_k - \alpha_k AM^{-1}\mathbf{p}_k, \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k M^{-1}(\mathbf{u}_k + \mathbf{q}_k), \\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k AM^{-1}(\mathbf{u}_k + \mathbf{q}_k), \\ \beta_k &= \frac{\left( M^{-T}\mathbf{r}_0^\#, \mathbf{r}_{k+1} \right)}{\left( M^{-T}\mathbf{r}_0^\#, \mathbf{r}_k \right)}, \end{aligned} \quad (31)$$

End Do

### 3.3 左前処理系を含めた 4 種類の前処理付き CGS アルゴリズムに対する解析

左前処理系 (Alg.4) の線形系の残差ベクトル  $\mathbf{r}_k^+$  を

$$\mathbf{r}_k^+ \mapsto M^{-1}\mathbf{r}_k, \quad (k = 0, 1, 2, \dots) \quad (33)$$

と展開し\*7，必要な式変形を施すと改善版 1 (Alg.5) に帰着される．また，(28) に対し (33) を適用すると，

$$\begin{aligned} \mathcal{K}_{2k}(M^{-1}A, \mathbf{r}_0^+) &\mapsto \mathcal{K}_{2k}(M^{-1}A, M^{-1}\mathbf{r}_0) \\ &= M^{-1}\mathcal{K}_{2k}(AM^{-1}, \mathbf{r}_0), \end{aligned}$$

したがって，改善版 1 (Alg.5) で生成されるクリロフ部分空間と解の構造は

$$\begin{aligned} \mathbf{x}_k &\in \mathbf{x}_0 + \mathcal{K}_{2k}(M^{-1}A, \mathbf{r}_0^+) \\ \mapsto \mathbf{x}_k &\in \mathbf{x}_0 + M^{-1}\mathcal{K}_{2k}(AM^{-1}, \mathbf{r}_0). \end{aligned} \quad (34)$$

また，Alg.5 と Alg.6 とは等価であり，共に残差ベクトルは  $\mathbf{r}_k$  であるため，収束判定は基本的な (9) であり，本来の残差ベクトル  $\mathbf{r}_k$  に対し評価できる．

\*7 本稿中の  $\mapsto$  記号は，前処理行列が潜在的に作用しているベクトル  $\mathbf{r}_k^+$ ， $\mathbf{r}_k^b$  を，各々， $M^{-1}\mathbf{r}_k$ ， $M^{-T}\mathbf{r}_k^\#$  と展開することを表している．

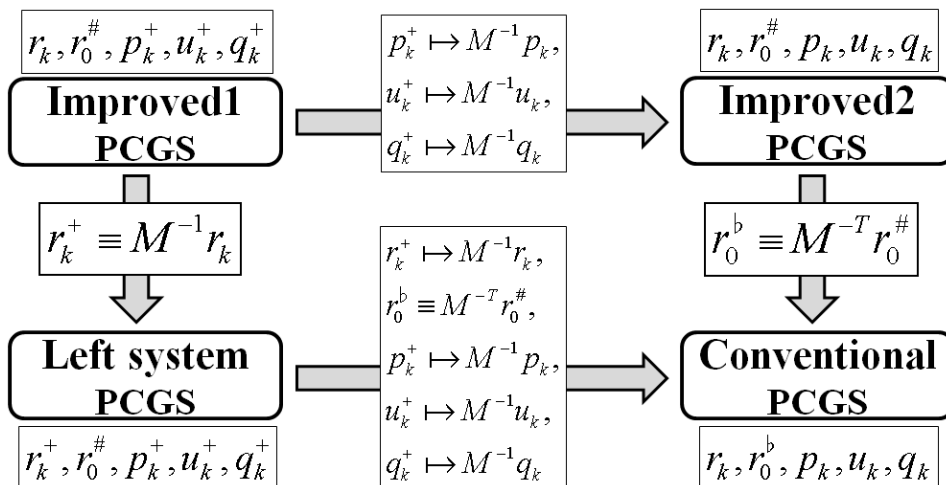


図 1 4 種類の前処理付き CGS アルゴリズム間の関係

以上から，両改善版 PCGS は，表 1 で示した従来版 (Alg.3) と左前処理系 (Alg.4) の両方の長所を兼ね備えていることが確認された。

本節で議論した 4 種類の PCGS の関係を体系化したものが図 1 である。Improved1 PCGS (Alg.5) から Improved2 PCGS (Alg.6) への変換で表している  $p_k^+ \mapsto M^{-1}p_k$  とは，Improved1 中のベクトル  $p_k^+$  を Improved2 では  $M^{-1}p_k$  と展開することを示している。改善版 2 (Alg.6) において，双対系の初期シャドウ残差ベクトル  $M^{-T}r_0^#$  を  $r_0^b$  と置き換え，(24) としてアルゴリズムを書き換えたものが従来版 (Alg.3) である [12]。

各 PCGS の残差ベクトルの構造，および，生成されるクリロフ部分空間と解の構造を以下にまとめる。これら 4 種類の PCGS の特性は 4 節の数値実験でも確認する。

- Conventional PCGS (Alg.3) :

$$r_k = b - (AM^{-1})(Mx_k),$$

$$Mx_k \in Mx_0 + \mathcal{K}_{2k}(AM^{-1}, r_0).$$

- Left system PCGS (Alg.4) :

$$r_k^+ = M^{-1}(b - Ax_k),$$

$$x_k \in x_0 + \mathcal{K}_{2k}(M^{-1}A, r_0^+).$$

- Improved1 PCGS (Alg.5) :

$$M^{-1}r_k = M^{-1}(b - (AM^{-1})(Mx_k)),$$

$$x_k \in x_0 + M^{-1}\mathcal{K}_{2k}(AM^{-1}, r_0).$$

- Improved2 PCGS (Alg.6) :

$$M^{-1}r_k = M^{-1}(b - (AM^{-1})(Mx_k)),$$

$$x_k \in x_0 + M^{-1}\mathcal{K}_{2k}(AM^{-1}, r_0).$$

文献 [12] では Alg.3 と Alg.6 は初期シャドウ残差ベクトルの構造の違いにより切り替わっていることについて述べられ

ている。つまり， $(\tilde{r}_0^#, \tilde{r}_0) \Rightarrow (M_R^{-T}r_0^#, M_L^{-1}r_0) \neq 0$  に対し， $(M^{-T}r_0^#, r_0) \equiv (r_0^b, r_0) \neq 0$  としたものが Alg.3 である。  $(r_0^#, M^{-1}r_0) \neq 0$  としたものが Alg.6，および，それと等価な Alg.5 である。さらに， $(r_0^#, M^{-1}r_0) \equiv (r_0^#, r_0^+) \neq 0$  としたものが Alg.4 である。

#### 4. 数値実験

3 節で述べた 4 種類の PCGS アルゴリズム (Alg.3~6) の特性について，収束性に関する 3 通りの評価により数値的に確認する。

線形方程式のテスト問題として，Matrix Market[14] と The University of Florida Sparse Matrix Collection[3] の中から線形方程式向きの実非対称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 とした厳密解 ( $\equiv x_{\text{exact}}$ ) を用い  $b = Ax_{\text{exact}}$  として生成した。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) [16] のバージョン 1.1.2 逐次コードを倍精度で使用し，各アルゴリズムに対応したコードを作成した。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。併用する前処理演算は ILU(0) を用い PCGS で各問題を求解した。アルゴリズムに与える初期解は  $x_0 = 0$ 。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内の一つである。最大反復回数は 1000 回とした (ほとんどの求解問題とアルゴリズムにて，1000 回以内で収束，または，Breakdown が生じている)。

初期シャドウ残差ベクトルは，前処理系における  $(\tilde{r}_0^#, \tilde{r}_0) \neq 0$  の関係に基づき，各アルゴリズム中に記載したとおり，Alg.3 では  $r_0^b = r_0$ ，Alg.5 と Alg.6 では  $r_0^b = M^{-1}r_0$ ，Alg.4 では  $r_0^b = r_0^+$  とした。

数値実験に用いた計算サーバの環境は，計算機は DELL Precision T7400，CPU は Intel Xeon E5420(2.5GHz)，メ

メモリサイズは16GB, OSはCent OS(Kernel 2.6.18), コンパイラはIntel icc 10.1, ifort 10.1 である.

本小節における数値実験では,

- (a) アルゴリズム中の残差ベクトルを用いた相対残差 (algorithm relative residual) に対する評価 [ 図 2, 図 5, 表 2 ],
- (b) アルゴリズム中の数値解を用いて算出した真の残差ベクトルを用いた真の相対残差 (true relative residual) に対する評価 [ 図 3, 図 6, 表 3 ]

の他に, 厳密解 ( $x_{\text{exact}}$ ) があらかじめ分かっているため,

- (c) アルゴリズム中の数値解を用いて算出した真の相対誤差 (true relative error) に対する評価 [ 図 4, 図 7, 表 4 ]

も行った. 収束判定は, (a) の場合, Alg.3, 5, 6 では (9), Alg.4 では (27) を用い, ともに 2 ノルムである. (b) の場合は全アルゴリズムとも  $\|b - Ax_k\|_2 / \|b\|_2 \leq \varepsilon$  であり, (c) の場合は全アルゴリズムとも  $\|x_k - x_{\text{exact}}\|_2 / \|x_{\text{exact}}\|_2 \leq \varepsilon$  である. また, (a) ~ (c) 全てにおいて  $\varepsilon = 10^{-12}$  である.

結果は図 2 ~ 図 4, 図 5 ~ 図 7 のグラフ, および, 表 2 ~ 表 4 のとおりである. “Conventional” は Alg.3 で示した従来の前処理変換に基づく PCGS を示しており, “Left system” は Alg.4 で示した左前処理系の PCGS, “Improved1” は Alg.5 で示した本提案の前処理変換に基づく PCGS, “Improved2” は Alg.6 で示した PCGS である.

図 2 ~ 図 4, および, 図 5 ~ 図 7 のグラフから, (a) アルゴリズム中の残差ベクトルによる相対残差の評価では Left system と Improved1, Improved2 との PCGS で収束の振る舞いは異なるが, (b) 真の相対残差と (c) 真の相対誤差では同じ振る舞いである. したがって, 3.3 節で解析した, 左前処理系 PCGS の Alg.4 と改善版 PCGS の Alg.5 とは, アルゴリズムの残差ベクトルは  $r_k^+$  と  $r_k$  であり異なるものの, 生成されるクリロフ部分空間に関して Alg.5 は Alg.4 に準じていることが数値的にも確認された. 一方で, Conventional のみ, (a) ~ (c) の全てに亘り振る舞いが異なる. 以上から, 表 1 で示した, 各 PCGS により生成されるクリロフ部分空間に対する解の構造の違いも数値的に確認された.

各表の  $n$  は係数行列のサイズ (問題サイズ), NNZ は非零要素数, 各々の欄に示した値は, 上段の数値が「真の相対残差ベクトルのノルム値の  $\log_{10}$ 」を示し, ( ) 内の数値が「収束までの所要反復回数」である. また, 下段には「真の相対誤差のノルム値の  $\log_{10}$ 」を示している\*8.

表 2 ~ 表 4 に亘り従来版 PCGS (Alg.3) の結果に注目すると, jpw\_h\_991 にて Breakdown, o1m5000 にて収束不

\*8 表 3, 表 4 中の Stagnation の内, 表 3 の poisson3Db の Improved2 や表 4 の o1m5000 の Left system については,  $\varepsilon = 10^{-12}$  の直前まで収束しており, ほぼ十分な精度に至っていると見做せる. これらは単に, 真の相対残差や相対誤差の評価に対する基準値としては,  $10^{-12}$  の精度は厳しい値というだけである.

十分なまま Stagnation (停滞)\*9 が生じているが, 他の 3 種類の PCGS (Alg.4 ~ 6) では求解されている. また, 厳密解が分かっても評価可能な (a), (b) (表 2, 3) にて bwf782, poisson3Db と watt\_\_1 の結果を比較すると, 従来版 PCGS (Alg.3) のみ, 真の相対残差や真の相対誤差の値が, 他の 3 種類の PCGS よりも劣る精度で収束, あるいは, 停滞している. 特に, watt\_\_1 では, 表 2 と表 3 において, 従来版 PCGS のみ他の 3 種類よりも速く収束しているが, 表 4 の相対誤差による評価と比べると, アルゴリズム中の残差も真の残差も早い段階で収束してしまっていることが分かる. 逆に, wang4 は, 表 2 と表 3 では従来版 PCGS の真の相対誤差の精度が 4 種類の PCGS の中で一番高いが, 表 4 では一番低い値の精度で (反復 55 回あたりから) 停滞している.

これらの結果は, 各 PCGS が生成するクリロフ部分空間と解の構造の議論とも整合する. すなわち, 表 1 で示したクリロフ部分空間と解の関係, および, 左前処理系における残差ベクトル  $r_k^+$  が  $r_k$  とは異なるということである. さらに, (34) で示した  $x_k \in x_0 + \mathcal{K}_{2k}(M^{-1}A, r_0^+) \mapsto x_k \in x_0 + M^{-1}\mathcal{K}_{2k}(AM^{-1}, r_0)$  についても確認された.

## 5. まとめ

本論文では, 従来から使用されてきている典型的な 2 種類の PCGS アルゴリズム (従来版 PCGS, 左前処理系 PCGS) について, “残差ベクトルの構造” および “生成されるクリロフ部分空間と解の構造” に注目して, 各々の長所と短所について説明した. それらに対し, 文献 [12] で提案された改善版 PCGS (改善版 1, 2) は, 従来からの 2 種類の PCGS の長所のみを併せ持つアルゴリズムであることを示した. 特に両改善版は左前処理系に準ずるものであり, 左前処理系の残差ベクトルの構造  $r_k^+$  を展開し, 本来の残差ベクトル  $r_k (= b - Ax_k)$  により記述されたアルゴリズムであることが確認された. そのため, 収束判定においても, アルゴリズム中の残差ベクトルの情報に基づいた適切な評価が可能である.

文献 [12] では  $(\tilde{r}_0^{\sharp}, \tilde{r}_0) \neq 0$  をどのように式変形し, 初期シャドウ残差ベクトルを構成するかに応じてクリロフ部分空間が切り替わることが示されているが, 本稿ではさらに, 左前処理系 PCGS (Alg.4) も交えて議論し, 改善版 1 (Alg.5) の  $M^{-1}r_k$  を  $r_k^+$  と置いたアルゴリズムが Alg.4 であることも示した.

以上の議論と数値実験による確認から, 生成されるクリロフ部分空間と解の構造は, 改善版 1, 2 では  $x_k \in x_0 + M^{-1}\mathcal{K}_{2k}(AM^{-1}, r_0)$ , 左前処理系では  $x_k \in x_0 + \mathcal{K}_{2k}(M^{-1}A, r_0^+)$  であり, 従来版の右前処理系では

\*9 o1m5000 については, Alg.3 のみ行列サイズの 5000 回まで反復させても停滞したままのため, 他の問題と同じく 1000 回における情報を掲載している.



$Mx_k \in Mx_0 + \mathcal{K}_{2k}(AM^{-1}, r_0)$  である。

謝辞 本研究は文部科学省科学研究費助成事業基盤研究(C)(課題番号:25390145)の支援を受けた。

## 参考文献

- [1] Barrett, R., et al., *Templates for the solution of linear systems: Building Blocks for Iterative Methods*, SIAM, 1994. 邦訳)長谷川里美, 長谷川秀彦, 藤野清次, 反復法 Templates, 朝倉書店, 1996.
- [2] Bruaset, A.M., *A Survey of Preconditioned Iterative Methods*, Longman Scientific & Technical, 1995.
- [3] Davis, T. A., The University of Florida Sparse Matrix Collection, <http://www.cise.ufl.edu/research/sparse/matrices/>
- [4] Fletcher, R., Conjugate Gradient Methods for Indefinite Systems, *Numerical Analysis Dundee 1975*, ed. by Watson, G., Lecture Notes in Mathematics, **506**, Springer-Verlag, pp.73–89, 1976.
- [5] 藤野清次, 張 紹良, 反復法の数理, 朝倉書店, 1996.
- [6] Gutknecht, M. H., On Lanczos-Type Methods for Wilson Fermions, *Numerical Challenges in Lattice Quantum Chromodynamics*, (eds. by Frommer, A., Lippert, T., Medeke, B. and Schilling, K.), Lecture Notes in Computational Science and Engineering **15**, Springer Berlin Heidelberg, pp.48–65, 2000.
- [7] 伊藤祥司, 杉原正顯, 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, 阪大, 9 月 (2009).
- [8] 伊藤祥司, 杉原正顯, 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, **15**, pp.171–174, 九大, 5 月 (2010).
- [9] 伊藤祥司, 杉原正顯, 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), **3(2)**, pp.9–19 (2010).
- [10] 伊藤祥司, 杉原正顯, クリロフ部分空間法に対する前処理方向とライブラリ実装における注意点, 日本応用数学会 2010 年度年会講演予稿集, 明大, 9 月 (2010).
- [11] 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*, (eds. Naono, K., Teranishi, K., Cavazos, J. and Suda, R.), pp. 135–152, Springer, 2010.
- [12] 伊藤祥司, 杉原正顯, 導出過程に着目した CGS 法の前処理付きアルゴリズム, 日本応用数学会論文誌, **23(2)**, pp.253–286 (2013).
- [13] Lanczos, C., Solution of Systems of Linear Equations by Minimized Iterations, *J. Res. Nat. Bur. of Standards*, **49**, pp. 33–53 (1952).
- [14] Matrix Market project, <http://math.nist.gov/MatrixMarket/>
- [15] Sonneveld, P., CGS, A fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, **10(1)**, pp. 36–52 (1989).
- [16] SSI project, Lis, <http://www.ssi.org/lis/>
- [17] Van der Vorst, Henk A., Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, **13(2)**, pp. 631–644 (1992).

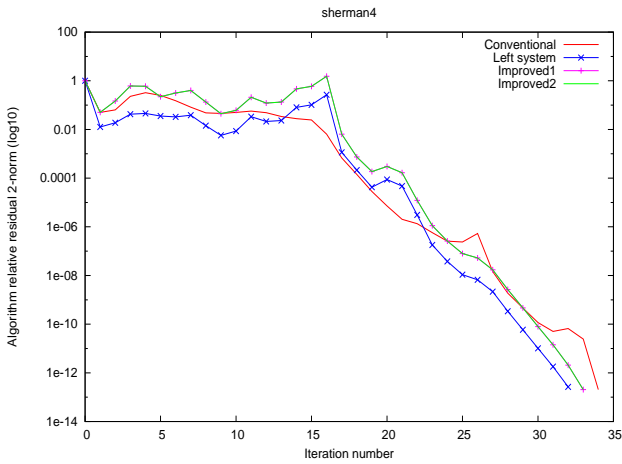


図 2 (a) アルゴリズム中の残差ベクトルを用いた相対残差の収束履歴グラフ (sherman4).

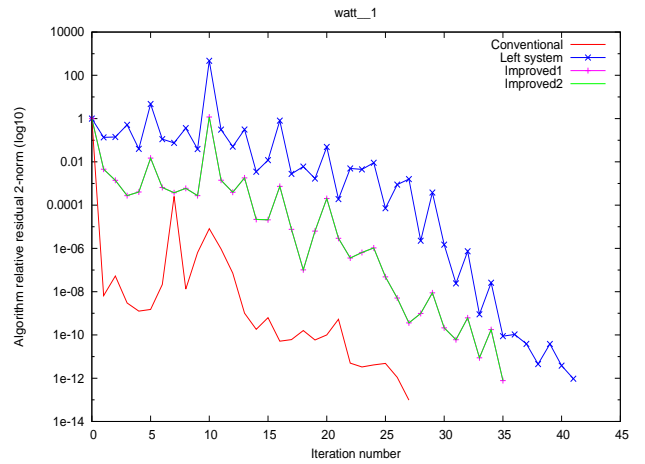


図 5 (a) アルゴリズム中の残差ベクトルを用いた相対残差の収束履歴グラフ (watt\_1).

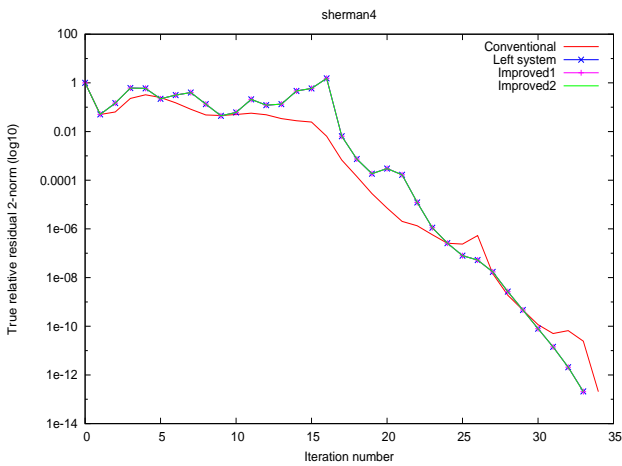


図 3 (b) 真の残差ベクトルによる真の相対残差の収束履歴グラフ (sherman4).

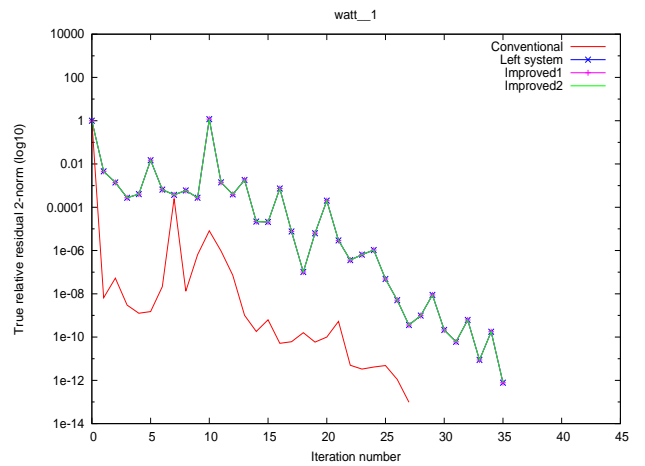


図 6 (b) 真の残差ベクトルによる真の相対残差の収束履歴グラフ (watt\_1).

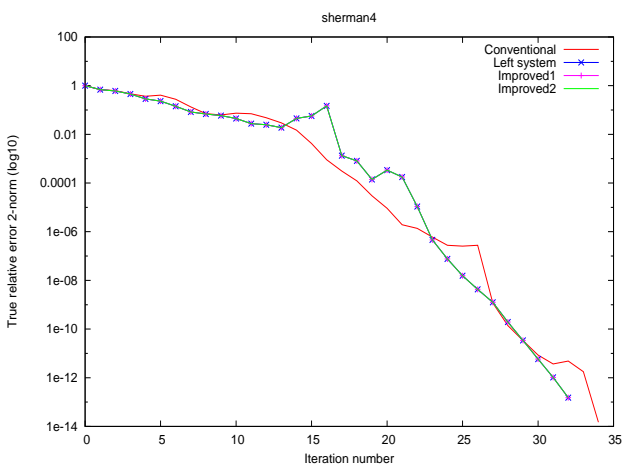


図 4 (c) 数値解を用いて算出した真の相対誤差の収束履歴グラフ (sherman4).

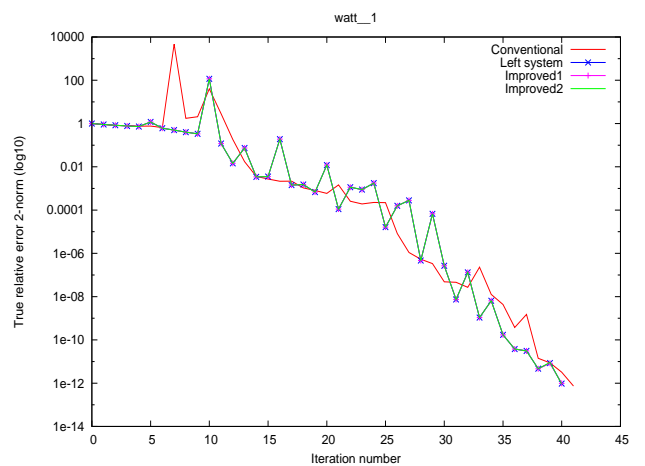


図 7 (c) 数値解を用いて算出した真の相対誤差の収束履歴グラフ (watt\_1).

表 2 (a) アルゴリズム中の残差ベクトルの相対残差により収束判定した結果

Matrix	$n$	NNZ	Conventional (Alg.3)	Left system (Alg.4)	Improved1 (Alg.5)	Improved2 (Alg.6)
add32	4960	19848	-12.17 (35) -12.17	-13.06 (37) -12.96	-12.04 (35) -11.96	-12.04 (35) -11.96
bfwa782	782	7514	-9.36 (93) -10.29	-12.37 (83) -12.09	-12.82 (78) -12.48	-12.17 (84) -12.22
jpwh_991	991	6027	Breakdown	-11.83 (15) -12.10	-12.44 (16) -12.53	-12.44 (16) -12.53
olm5000	5000	19996	-0.18 (Stag.) 4.22	-12.79 (38) -10.64	-12.20 (34) -8.05	-12.21 (33) -8.00
poisson3Db	85623	2374949	-10.14 (122) -10.33	-12.93 (119) -13.31	-12.49 (123) -13.39	-11.79 (117) -12.07
sherman4	1104	3786	-12.69 (34) -13.83	-11.68 (32) -12.82	-12.69 (33) -13.82	-12.69 (33) -13.83
wang4	26068	177196	-12.22 (52) -10.14	-10.96 (55) -9.66	-12.69 (56) -9.71	-12.70 (56) -9.71
watt_1	1856	11360	-13.01 (27) -5.96	-15.48 (41) -12.63	-12.11 (35) -9.77	-12.11 (35) -9.77

表 3 (b) 真の残差ベクトルによる真の相対残差により収束判定した結果

Matrix	$n$	NNZ	Conventional (Alg.3)	Left system (Alg.4)	Improved1 (Alg.5)	Improved2 (Alg.6)
add32	4960	19848	-12.17 (35) -12.17	-12.04 (35) -11.96	-12.04 (35) -11.96	-12.04 (35) -11.96
bfwa782	782	7514	-9.36 (Stag.) -10.29	-12.37 (83) -12.09	-12.82 (78) -12.48	-12.17 (84) -12.22
jpwh_991	991	6027	Breakdown	-12.44 (16) -12.53	-12.44 (16) -12.53	-12.44 (16) -12.53
olm5000	5000	19996	-0.18 (Stag.) 4.22	-12.49 (31) -8.28	-12.20 (34) -8.05	-12.21 (33) -8.00
poisson3Db	85623	2374949	-10.14 (Stag.) -10.33	-12.08 (113) -12.95	-12.49 (123) -13.39	-11.77 (Stag.) -12.06
sherman4	1104	3786	-12.69 (34) -13.83	-12.68 (33) -13.81	-12.69 (33) -13.82	-12.69 (33) -13.83
wang4	26068	177196	-12.22 (52) -10.14	-12.69 (56) -9.71	-12.69 (56) -9.71	-12.70 (56) -9.71
watt_1	1856	11360	-13.01 (27) -5.96	-12.11 (35) -9.77	-12.11 (35) -9.77	-12.11 (35) -9.77

表 4 (c) 数値解を用いて算出した真の相対誤差により収束判定した結果

Matrix	$n$	NNZ	Conventional (Alg.3)	Left system (Alg.4)	Improved1 (Alg.5)	Improved2 (Alg.6)
add32	4960	19848	-12.17 (35) -12.17	-12.00 (36) -12.29	-12.00 (36) -12.29	-12.00 (36) -12.29
bfwa782	782	7514	-9.36 (Stag.) -10.29	-12.37 (83) -12.09	-12.00 (77) -12.42	-12.17 (84) -12.22
jpwh_991	991	6027	Breakdown	-11.83 (15) -12.10	-11.83 (15) -12.10	-11.83 (15) -12.10
olm5000	5000	19996	-0.18 (Stag.) 4.22	-12.79 (Stag.) -11.23	-12.80 (49) -13.22	-12.59 (52) -13.09
poisson3Db	85623	2374949	-10.14 (Stag.) -10.33	-11.24 (111) -12.21	-11.61 (117) -12.57	-11.53 (116) -12.04
sherman4	1104	3786	-12.69 (34) -13.83	-11.68 (32) -12.82	-11.68 (32) -12.82	-11.68 (32) -12.82
wang4	26068	177196	-12.48 (Stag.) -10.66	-13.37 (68) -12.20	-12.80 (66) -12.57	-12.83 (66) -12.17
watt_1	1856	11360	-18.06 (41) -12.13	-14.28 (40) -12.02	-14.26 (40) -12.01	-14.26 (40) -12.05