Japan Journal of Industrial and Applied Mathematics (2022) 39:19–74
https://doi.org/10.1007/s13160-021-00480-0

ORIGINAL PAPER

Improvement of preconditioned bi-Lanczos-type algorithms with residual norm minimization for the stable solution of systems of linear equations

Shoji Itoh[1]

Received: 3 February 2021 / Revised: 25 June 2021 / Accepted: 3 August 2021 /
Published online: 2 September 2021

Abstract

In this paper, improved algorithms are proposed for preconditioned bi-Lanczos-type methods with residual norm minimization for the stable solution of systems of linear equations. In particular, preconditioned algorithms pertaining to the bi-conjugate gradient stabilized method (BiCGStab) and the generalized product-type method based on the BiCG (GPBiCG) have been improved. These algorithms are more stable compared to conventional alternatives. Further, a stopping criterion changeover is proposed for use with these improved algorithms. This results in higher accuracy (lower true relative error) compared to the case where no changeover is done. Numerical results confirm the improvements with respect to the preconditioned BiCGStab, the preconditioned GPBiCG, and stopping criterion changeover. These improvements could potentially be applied to other preconditioned algorithms based on bi-Lanczos-type methods.

Keywords Congruence of preconditioning conversion · Direction of preconditioned system · Changeover of stopping criterion · Preconditioned BiCGStab · Preconditioned GPBiCG

Mathematics Subject Classification 15A06 · 65F10

This work was supported by KAKENHI (18K11342).

✉ Shoji Itoh
itosho@acm.org

[1] Department of Engineering Science, Faculty of Engineering, Osaka Electro-Communication University, 18-8 Hatsucho, Neyagawa-shi, Osaka 572-8530, Japan

1 Introduction

In scientific and engineering computation, natural phenomena and engineering problems are described by mathematical models. These models are often reduced to a system of linear equations:

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1.1}$$

where A is a large, sparse nonsymmetric coefficient matrix of size $n \times n$, $\boldsymbol{x}$ is the solution vector, and $\boldsymbol{b}$ is the right-hand side (RHS) vector. In this paper, we assume A, $\boldsymbol{x}$ and $\boldsymbol{b}$ to be real.

Preconditioned algorithms based on the bi-Lanczos-type Krylov subspace method are often adopted. These algorithms achieve satisfactory convergence with few mathematical operations and limited computational resources compared to algorithms based on the Arnoldi-type Krylov subspace method. The bi-Lanczos-type methods are based on the bi-conjugate gradient (BiCG) method [3, 10]. For instance, the bi-conjugate gradient stabilized method (BiCGStab) [18] and the generalized product-type method based on the BiCG (GPBiCG) [19] have been proposed.

Further, preconditioning is effective for improving the convergence of these algorithms. However, if a preconditioned algorithm is poorly designed, there may be no beneficial effect from the preconditioning operation. Consequently, it is essential to carefully design preconditioned algorithms.

Recently, an improved preconditioned algorithm for the conjugate gradient squared (CGS) method was proposed [8]. The algorithm retains some mathematical properties that are associated with the derivation of CGS from the BiCG method for the non-preconditioned system. It performs better than the conventional preconditioned CGS (PCGS) algorithm. A mathematical characteristic of this algorithm is that it corresponds to the left-preconditioned system, with $\alpha_k^{\mathrm{PCGS}} = \alpha_k^{\mathrm{PBiCG}}$, $\beta_k^{\mathrm{PCGS}} = \beta_k^{\mathrm{PBiCG}}$ for the standard preconditioned BiCG (PBiCG) algorithm [7]. Here, we seek to derive improved preconditioned algorithms for the BiCGStab and the GPBiCG which adopt the above characteristic. Importantly, there is an essential difference in preconditioning conversion between the BiCG, the CGS, and the BiCGStab, the GPBiCG. We have already shown that the direction of the preconditioned system for the CGS depends on the direction of the BiCG by focusing on the congruence property with respect to the direction of the preconditioning conversion [8]. In contrast, there is no congruence property on the minimal residual operation (MR part) for the BiCGStab and the GPBiCG. In what follows, we prove that the PCGS improvement mechanism can be extended to the preconditioned BiCGStab (PBiCGStab) and preconditioned GPBiCG (PGPBiCG) without the congruence property, by analyzing the structure of both preconditioned algorithms.

Moreover, we generate a further improvement via introducing a stopping criterion changeover [9]. Doing so realizes a highly accurate numerical solution with advantages for the right- and left-preconditioned systems for the CGS method [8].

This paper is organized as follows. Section 2 provides relevant definitions, theorems and notations. Section 3 focusses on various PBiCGStab including improved algorithms. Section 4 focusses on various PGPBiCG including improved algorithms. In Sect. 5, we present the results of numerical experiments. Finally, conclusions are offered in Sect. 6.

2 Preliminaries

In this study, *preconditioned algorithm* and *preconditioned system* refer to solving algorithms described with some preconditioning operator M (or preconditioner/preconditioning matrix) and the system converted by the operator based on M, respectively. Here, M is not a specific preconditioner, but a general preconditioner.

Definition 1 The matrix, vector and scalar variable for a preconditioned system are denoted by tilde ($\tilde{}$). Then, for the system and solution

$$\tilde{A}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}, \tag{2.1}$$

$$\tilde{A} = M_l^{-1}AM_r^{-1}, \;\; \tilde{\boldsymbol{x}} = M_r\boldsymbol{x}, \;\; \tilde{\boldsymbol{b}} = M_l^{-1}\boldsymbol{b}, \tag{2.2}$$

we define the direction of the preconditioned system of linear equations as follows:

- *The two-sided preconditioned system:* Equation (2.2);
- *The right-preconditioned system:* $M_l = I$ and $M_r = M$ in (2.2);
- *The left-preconditioned system:* $M_l = M$ and $M_r = I$ in (2.2),

where M is the preconditioner $M = M_lM_r$ ($M \approx A$) and I is the identity matrix.

Other vectors in the solution method are not preconditioned. The initial guess is given as $\boldsymbol{x}_0$, and $\tilde{\boldsymbol{x}}_0 = M_r\boldsymbol{x}_0$.

In terms of bi-Lanczos-type methods like the BiCG and the CGS, we assume the existence of a dual system $A^{\mathrm{T}}\boldsymbol{x}^\sharp = \boldsymbol{b}^\sharp$ and we will refer to this as the "shadow system". It has the initial shadow residual vector (ISRV: $\boldsymbol{r}_0^\sharp$ $(= \boldsymbol{b}^\sharp - A^{\mathrm{T}}\boldsymbol{x}_0^\sharp)$ assuming an initial guess). References [7, 8] have reported the following theorem concerning the relation between the construction and setting of ISRV and preconditioned BiCG, and CGS methods.

Theorem 1 (Itoh and Sugihara [7, 8]) *The direction of a preconditioned system for the BiCG method and CGS method is switched by the construction and setting of ISRV.*

After preliminaries, Example 1 in Definition 8 is used to illuminate Theorem 1.

Definition 2 Notation for the direction of preconditioning conversion:
$\tilde{A}_w (= M_l^{-1}AM_r^{-1})$: the matrix with two-sided preconditioning conversion.
$\tilde{A}_r (= AM^{-1})$: the matrix with right-preconditioning conversion.

$\tilde{A}_l (= M^{-1}A)$: the matrix with left-preconditioning conversion.

Definition 3 (*Congruence* [8]) Let the term “all directions on preconditioning conversions” be a synthesis of the preconditioning conversion for “the two-sided direction”, “the right direction”, and “the left direction”, not only for the system and solution (2.1), but also for other vectors in the solution method.

If all directions on the preconditioning conversions to the solution method are reduced to one and the same algorithm description, then we refer to this as “congruence” in the direction of the preconditioning conversion (See [8] for an example). Furthermore, the term “congruency” refers to the congruence property.

Proposition 1 (Congruency: Itoh and Sugihara [7, 8]) *There is congruence in the PBiCG and PCGS algorithms concerning the direction of the preconditioning conversion.*

At first glance, the above Definitions 1 and 2 may be misinterpreted as the same matter. Importantly, they differ because of the assertions of Theorem 1 and Proposition 1 concerning the PBiCG and PCGS algorithms, and these algorithms do not have residual norm minimization operations. In this paper, Theorem 1 will be extended to bi-Lanczos-type methods with residual norm minimization.

Definition 4 (*Notation concerning the inner product and norm.*) Let V be a linear space. Then, we describe $(\boldsymbol{u}, \boldsymbol{v})$ as the Euclidean inner product, $\boldsymbol{u}, \boldsymbol{v} \in V$. $\|\cdot\|$ means the 2-norm (the Euclidean vector norm and the corresponding matrix norm), and $\kappa(A)(= \|A\| \|A^{-1}\|)$ means the condition number of matrix A. Let $V^\sharp$ be the dual space of V. Then, we describe $\langle \boldsymbol{u}^\sharp, \boldsymbol{v} \rangle$ as the inner product expressing the duality of V and $V^\sharp$, $\boldsymbol{v} \in V, \boldsymbol{u}^\sharp \in V^\sharp$ [13].

Definition 5 (*Coordinative to the left-preconditioned algorithm*) If a preconditioning bi-Lanczos-based algorithm corresponds to the left-preconditioned system with mathematical equivalence, then we state that “the preconditioning bi-Lanczos-type algorithm is coordinative to the left-preconditioned algorithm.” Let $\boldsymbol{r}_k^+$ be the residual vector of the left-preconditioned (Left) bi-Lanczos-type algorithm in Definition 1, then a coordinative to the left-preconditioned (coLeft) algorithm has a residual vector $\boldsymbol{r}_k$, here, $\boldsymbol{r}_k^+ \equiv M^{-1}\boldsymbol{r}_k$. Further, the relative residual norm of the algorithm is $\|\boldsymbol{r}_k\| / \|\boldsymbol{b}\|$ for coLeft, but $\|\boldsymbol{r}_k^+\| / \|M^{-1}\boldsymbol{b}\|$ for Left.

Proposition 2 (Itoh and Sugihara [9]) *In the solution algorithms for a nonsymmetric system, there exist the following relations* (2.3) *for the right-preconditioned system and* (2.4) *for the left-preconditioned system. k denotes the iteration number.* $\boldsymbol{x}_{\mathrm{exact}}$ *means the exact solution for the system of linear equations* (1.1).

$$\frac{\|M(\boldsymbol{x}_{k+1} - \boldsymbol{x}_{\mathrm{exact}})\|}{\|M\boldsymbol{x}_{\mathrm{exact}}\|} \le \kappa(AM^{-1}) \frac{\|\boldsymbol{r}_{k+1}\|}{\|\boldsymbol{b}\|}, \tag{2.3}$$

$$\frac{\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_{\mathrm{exact}}\|}{\|\boldsymbol{x}_{\mathrm{exact}}\|} \le \kappa(M^{-1}A)\frac{\|\boldsymbol{r}^{+}_{k+1}\|}{\|M^{-1}\boldsymbol{b}\|}. \tag{2.4}$$

Definition 6 Let a stopping criterion $\|\boldsymbol{r}_{k+1}\|/\|\boldsymbol{b}\|$ be a benchmark. We define the following relation between the benchmark and $\|\boldsymbol{r}^{+}_{k+1}\|/\|M^{-1}\boldsymbol{b}\|$ for the left-preconditioned algorithm [9].

$$\frac{\|\boldsymbol{r}^{+}_{k+1}\|}{\|M^{-1}\boldsymbol{b}\|} < \frac{\|\boldsymbol{r}_{k+1}\|}{\|\boldsymbol{b}\|} \quad \text{(LTB: less than the benchmark)}, \tag{2.5}$$

$$\frac{\|\boldsymbol{r}_{k+1}\|}{\|\boldsymbol{b}\|} < \frac{\|\boldsymbol{r}^{+}_{k+1}\|}{\|M^{-1}\boldsymbol{b}\|} \quad \text{(GTB: greater than the benchmark)}\,. \tag{2.6}$$

The recurrences of the BiCG for the preconditioned system are

$$R_k(\tilde{\lambda}) = R_{k-1}(\tilde{\lambda}) - \alpha^{\mathrm{PBiCG}}_{k-1}\tilde{\lambda}P_{k-1}(\tilde{\lambda}), \quad R_0(\tilde{\lambda}) = 1, \tag{2.7}$$

$$P_k(\tilde{\lambda}) = R_k(\tilde{\lambda}) + \beta^{\mathrm{PBiCG}}_{k-1}P_{k-1}(\tilde{\lambda}), \quad P_0(\tilde{\lambda}) = 1. \tag{2.8}$$

Here, $R_k(\tilde{\lambda})$ is the residual polynomial of degree k. Further, $P_k(\tilde{\lambda})$ is the probing direction polynomial of degree k, that is, $\tilde{\boldsymbol{r}}^{\mathrm{BiCG}}_k = R_k(\tilde{A})\tilde{\boldsymbol{r}}_0$ and $\tilde{\boldsymbol{p}}^{\mathrm{BiCG}}_k = P_k(\tilde{A})\tilde{\boldsymbol{r}}_0$. For example, in the left-PBiCG, (2.7) is shown as $R^{\mathrm{L}}_k(\tilde{\lambda}) = R^{\mathrm{L}}_{k-1}(\tilde{\lambda}) - \alpha^{\mathrm{L}}_{k-1}\tilde{\lambda}P^{\mathrm{L}}_{k-1}(\tilde{\lambda})$. We denote these polynomials with superscript “L”[1], to indicate that these polynomials pertain to the left-preconditioned system [7].

Definition 7 (*Notation for the preconditioned vector*) The symbol $\boldsymbol{v}^{+}_k$ means $\boldsymbol{v}^{+}_k \equiv M^{-1}\boldsymbol{v}_k$. To distinguish between the direction of the preconditioned system, we use $\boldsymbol{v}^{+}_k (= Pol^{\mathrm{L}}_k(M^{-1}A)M^{-1}\boldsymbol{r}_0 = M^{-1}Pol^{\mathrm{L}}_k(AM^{-1})\boldsymbol{r}_0 = M^{-1}\boldsymbol{v}^{\mathrm{L}}_k)$ for the left-preconditioned system (L). Here, $Pol_k(\tilde{A})\tilde{\boldsymbol{r}}_0$ means the polynomial of degree k for $\tilde{\boldsymbol{v}}_k$. On the other hand[2], for the right-preconditioned system (R), we use $\check{\boldsymbol{v}}_k$ to distinguish with $\boldsymbol{v}^{+}_k$. Here, $\check{\boldsymbol{v}}_k \equiv M^{-1}Pol^{\mathrm{R}}_k(AM^{-1})\boldsymbol{r}_0 = M^{-1}\boldsymbol{v}^{\mathrm{R}}_k$.

Theorem 2 (Lanczos [10], Fletcher [3], Itoh and Sugihara [7]) *The BiCG method for the preconditioned system satisfies the following conditions:*

$$\langle \tilde{\boldsymbol{r}}^{\sharp}_i, \tilde{\boldsymbol{r}}^{\mathrm{BiCG}}_j \rangle = 0 \quad (i \neq j), \quad \text{(biorthogonality)}, \tag{2.9}$$

$$\langle \tilde{\boldsymbol{p}}^{\sharp}_i, \tilde{A}\tilde{\boldsymbol{p}}^{\mathrm{BiCG}}_j \rangle = 0 \quad (i \neq j), \quad \text{(biconjugacy)}. \tag{2.10}$$

Here, $\tilde{\boldsymbol{r}}^{\sharp}_i$ and $\tilde{\boldsymbol{p}}^{\sharp}_i$ are the vectors in the dual shadow system of $\tilde{\boldsymbol{r}}^{\mathrm{BiCG}}_j$ and $\tilde{\boldsymbol{p}}^{\mathrm{BiCG}}_j$, respectively.

[1] In a similar manner, we use “R” to indicate the right-preconditioned system and “W” to indicated the two-sided preconditioned system. On the other hand, we denote the direction of the preconditioning conversion in lower case.

[2] For coding purposes, we do not need to specifically distinguish $\boldsymbol{v}^{+}_k$ and $\check{\boldsymbol{v}}_k$.

Definition 8 (*Abbreviation of initial shadow residual vectors* [7, 8]) We assume that $\langle \tilde{\boldsymbol{r}}_0^\sharp, \tilde{\boldsymbol{r}}_0 \rangle \neq 0$, e.g., $\tilde{\boldsymbol{r}}_0^\sharp = \tilde{\boldsymbol{r}}_0 \neq \boldsymbol{0}$, regarding the construction and setting of the ISRV for the preconditioned system. Here, $\tilde{\boldsymbol{r}}_0$ means the initial residual vector for the preconditioned system of $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$. If $\tilde{\boldsymbol{r}}_0^\sharp = \boldsymbol{r}_0^\sharp$ and $\tilde{\boldsymbol{r}}_0 = M^{-1}\boldsymbol{r}_0$ are given, then it reduces to $\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$ (ISRV1). This construction and setting of ISRV brings coLeft to the left-preconditioned system. If $\tilde{\boldsymbol{r}}_0^\sharp = M^{-\mathrm{T}}\boldsymbol{r}_0^\sharp$ and $\tilde{\boldsymbol{r}}_0 = \boldsymbol{r}_0$ are given, then it reduces to $\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$ ($\boldsymbol{r}_0^\flat \equiv M^{-\mathrm{T}}\boldsymbol{r}_0^\sharp$) or $\boldsymbol{r}_0^\sharp = M^{\mathrm{T}}\boldsymbol{r}_0$ (ISRV2). This brings coLeft to the right-preconditioned system. Further, if $\tilde{\boldsymbol{r}}_0^\sharp = M^{-\mathrm{T}}\boldsymbol{r}_0^\sharp \equiv \boldsymbol{r}_0^\flat$, $\boldsymbol{r}_0^\flat = M^{-\mathrm{T}}M^{-1}\boldsymbol{r}_0$ and $\tilde{\boldsymbol{r}}_0 = \boldsymbol{r}_0$ are given, it reduces to $\boldsymbol{r}_0^\flat = M^{-\mathrm{T}}M^{-1}\boldsymbol{r}_0$ (ISRV9[3]), this brings the right-preconditioned algorithm to the left-preconditioned system.

Example 1 It is sufficient to prove the following cases regarding the biorthogonality. The biconjugacy can be proven in a similar manner [7, 8].

(1) Applying ISRV1 to coLeft:

$$\begin{aligned}(\tilde{\boldsymbol{r}}_k^\sharp, \tilde{\boldsymbol{r}}_k) &= (R_k(A^{\mathrm{T}}M^{-\mathrm{T}})\,\boldsymbol{r}_0^\sharp,\ M^{-1}R_k(AM^{-1})\boldsymbol{r}_0)\\ &= (R_k^{\mathrm{L}}(A^{\mathrm{T}}M^{-\mathrm{T}})(M^{-1}\boldsymbol{r}_0),\ R_k^{\mathrm{L}}(M^{-1}A)(M^{-1}\boldsymbol{r}_0)).\end{aligned}$$

(2) Applying ISRV2 to coLeft:

$$\begin{aligned}(\tilde{\boldsymbol{r}}_k^\sharp, \tilde{\boldsymbol{r}}_k) &= (R_k(A^{\mathrm{T}}M^{-\mathrm{T}})\,\boldsymbol{r}_0^\sharp,\ M^{-1}R_k(AM^{-1})\boldsymbol{r}_0)\\ &= (M^{-\mathrm{T}}R_k^{\mathrm{R}}(A^{\mathrm{T}}M^{-\mathrm{T}})(M^{\mathrm{T}}\boldsymbol{r}_0),\ R_k^{\mathrm{R}}(AM^{-1})\boldsymbol{r}_0)\\ &= (R_k^{\mathrm{R}}(M^{-\mathrm{T}}A^{\mathrm{T}})\boldsymbol{r}_0,\ R_k^{\mathrm{R}}(AM^{-1})\boldsymbol{r}_0).\end{aligned}$$

(3) Applying ISRV9 to right-preconditioned algorithm:

$$\begin{aligned}(\tilde{\boldsymbol{r}}_k^\sharp, \tilde{\boldsymbol{r}}_k) &= (R_k(M^{-\mathrm{T}}A^{\mathrm{T}})(M^{-\mathrm{T}}\boldsymbol{r}_0^\sharp),\ R_k(AM^{-1})\boldsymbol{r}_0)\\ &= (R_k(M^{-\mathrm{T}}A^{\mathrm{T}})\boldsymbol{r}_0^\flat,\ R_k(AM^{-1})\boldsymbol{r}_0)\\ &= (R_k^{\mathrm{L}}(M^{-\mathrm{T}}A^{\mathrm{T}})(M^{-\mathrm{T}}M^{-1}\boldsymbol{r}_0),\ R_k^{\mathrm{L}}(AM^{-1})\boldsymbol{r}_0)\\ &= (M^{-\mathrm{T}}R_k^{\mathrm{L}}(A^{\mathrm{T}}M^{-\mathrm{T}})(M^{-1}\boldsymbol{r}_0),\ R_k^{\mathrm{L}}(AM^{-1})\boldsymbol{r}_0)\\ &= (R_k^{\mathrm{L}}(A^{\mathrm{T}}M^{-\mathrm{T}})(M^{-1}\boldsymbol{r}_0),\ R_k^{\mathrm{L}}(M^{-1}A)(M^{-1}\boldsymbol{r}_0)).\end{aligned}$$

[3] Although ISRV9 is not sequential with respect to ISRV1 and ISRV2, the designation is consistent with our approach.

3 Analyses for various preconditioned BiCGStab algorithms

First, we outline how to derive the BiCGStab method for the preconditioned system, and propose relevant theorems concerning the characteristics of the PBiCGStab. Next, we give outlines of six types of PBiCGStab algorithms. Finally, we summarize the characteristics of their mathematical structures.

3.1 Derivation of the BiCGStab method from PBiCG

The shadow residual vector in the PBiCGStab adopts $\tilde{\boldsymbol{s}}_k^\sharp = S_k(\tilde{A}^T)\tilde{\boldsymbol{r}}_0^\sharp$ using the polynomial $S_k(\tilde{\lambda})$ of degree k, *the stabilized polynomial*. $\tilde{\boldsymbol{s}}_k^\sharp$ is expanded as follows[4]:

$$\tilde{\boldsymbol{s}}_k^\sharp = \frac{\mathrm{lc}(S_k)}{\mathrm{lc}(R_k)}(\tilde{\boldsymbol{r}}_k^\sharp + d_{k-1}\tilde{\boldsymbol{r}}_{k-1}^\sharp + \cdots + d_1\tilde{\boldsymbol{r}}_1^\sharp + d_0\tilde{\boldsymbol{r}}_0^\sharp). \tag{3.1}$$

where, “lc” is an abbreviation of *the leading coefficient*;

$$\mathrm{lc}(R_{k+1}) = -\alpha_k^{\mathrm{PSTAB}}\ \mathrm{lc}(R_k), \quad \mathrm{lc}(S_{k+1}) = -\omega_k^{\mathrm{PSTAB}}\ \mathrm{lc}(S_k).$$

Deriving the bi-Lanczos-type method from the PBiCG, we apply (2.9) to the following inner product:

$$\langle \tilde{\boldsymbol{s}}_k^\sharp, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle = \frac{\mathrm{lc}(S_k)}{\mathrm{lc}(R_k)}\langle \tilde{\boldsymbol{r}}_k^\sharp, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle \equiv \frac{1}{c_k}\langle \tilde{\boldsymbol{r}}_k^\sharp, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle.$$

Therefore,

$$\begin{aligned} \langle \tilde{\boldsymbol{r}}_k^\sharp, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle &= c_k\langle \tilde{\boldsymbol{s}}_k^\sharp, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle = c_k\langle S_k(\tilde{A}^T)\tilde{\boldsymbol{r}}_0^\sharp, R_k(\tilde{A})\tilde{\boldsymbol{r}}_0 \rangle \\ &= c_k\langle \tilde{\boldsymbol{r}}_0^\sharp, S_k(\tilde{A})R_k(\tilde{A})\tilde{\boldsymbol{r}}_0 \rangle. \end{aligned} \tag{3.2}$$

Next, the inner product (2.10) is transformed by $\tilde{\boldsymbol{p}}_k^\sharp = \tilde{\boldsymbol{r}}_k^\sharp + \beta_{k-1}^{\mathrm{PBiCG}}\tilde{\boldsymbol{p}}_{k-1}^\sharp$ of PBiCG

$$\begin{aligned} \langle \tilde{\boldsymbol{p}}_k^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle &= \langle \tilde{\boldsymbol{r}}_k^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle + \beta_{k-1}^{\mathrm{PBiCG}}\langle \tilde{\boldsymbol{p}}_{k-1}^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle \\ &= \langle \tilde{\boldsymbol{r}}_k^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle. \end{aligned} \tag{3.3}$$

Further, like deriving (3.2), the inner product of $\tilde{\boldsymbol{s}}_k^\sharp$ and $\tilde{A}\tilde{\boldsymbol{p}}_k$ reduces the following with $\langle \tilde{\boldsymbol{r}}_i^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_j^{\mathrm{BiCG}} \rangle = \langle \tilde{\boldsymbol{p}}_i^\sharp - \beta_{i-1}^{\mathrm{PBiCG}}\tilde{\boldsymbol{p}}_{i-1}^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_j^{\mathrm{BiCG}} \rangle = 0$ $(i \neq j)$:

$$\begin{aligned} \langle \tilde{\boldsymbol{r}}_k^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle &= c_k\langle \tilde{\boldsymbol{s}}_k^\sharp, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle = c_k\langle S_k(\tilde{A}^T)\tilde{\boldsymbol{r}}_0^\sharp, \tilde{A}P_k(\tilde{A})\tilde{\boldsymbol{r}}_0 \rangle \\ &= c_k\langle \tilde{\boldsymbol{r}}_0^\sharp, \tilde{A}S_k(\tilde{A})P_k(\tilde{A})\tilde{\boldsymbol{r}}_0 \rangle. \end{aligned} \tag{3.4}$$

We state the polynomials of the residual vector and probing direction vector as

$$\tilde{\boldsymbol{r}}_k^{\mathrm{STAB}} \equiv S_k(\tilde{A})R_k(\tilde{A})\tilde{\boldsymbol{r}}_0, \quad \tilde{\boldsymbol{p}}_k^{\mathrm{STAB}} \equiv S_k(\tilde{A})P_k(\tilde{A})\tilde{\boldsymbol{r}}_0. \tag{3.5}$$

[4] Coefficients of d_i $(i = k-1, \dots, 0)$ inside the brackets are calculated by dividing each coefficient of $\tilde{\boldsymbol{r}}_i^\sharp$ by $\mathrm{lc}(S_k)/\mathrm{lc}(R_k)$.

Then, the following Proposition consists of (3.2) to (3.5):

Proposition 3

$$\alpha_k^{\text{PBiCG}} = \frac{\langle \tilde{\boldsymbol{r}}_k^{\sharp}, \tilde{\boldsymbol{r}}_k^{\text{BiCG}} \rangle}{\langle \tilde{\boldsymbol{p}}_k^{\sharp}, \tilde{A}\tilde{\boldsymbol{p}}_k^{\text{BiCG}} \rangle} \equiv \frac{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{\boldsymbol{r}}_k^{\text{STAB}} \rangle}{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{A}\tilde{\boldsymbol{p}}_k^{\text{STAB}} \rangle} = \alpha_k^{\text{PSTAB}},$$

$$\beta_k^{\text{PBiCG}} = \frac{\langle \tilde{\boldsymbol{r}}_{k+1}^{\sharp}, \tilde{\boldsymbol{r}}_{k+1}^{\text{BiCG}} \rangle}{\langle \tilde{\boldsymbol{r}}_k^{\sharp}, \tilde{\boldsymbol{r}}_k^{\text{BiCG}} \rangle} \equiv \frac{\alpha_k^{\text{PSTAB}}}{\omega_k^{\text{PSTAB}}} \times \frac{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{\boldsymbol{r}}_{k+1}^{\text{STAB}} \rangle}{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{\boldsymbol{r}}_k^{\text{STAB}} \rangle} = \beta_k^{\text{PSTAB}}.$$

These α_k^{PSTAB} and β_k^{PSTAB} compose the BiCG part of the PBiCGStab, and they determine the direction of the preconditioned system [7]. This is the common characteristic which exists in all bi-Lanczos-type methods.

The residual vector and polynomial of the BiCGStab method for the preconditioned system, are expressed as follows:

$$\begin{aligned} \tilde{\boldsymbol{r}}_{k+1}^{\text{STAB}} &= \tilde{\boldsymbol{t}}_k - \omega_k^{\text{PSTAB}} \tilde{A}\tilde{\boldsymbol{t}}_k \\ &= (I - \omega_k^{\text{PSTAB}} \tilde{A})\tilde{\boldsymbol{t}}_k \; = \; S_{k+1}^{\text{STAB}}(\tilde{A}) R_{k+1}(\tilde{A})\tilde{\boldsymbol{r}}_0, \end{aligned} \tag{3.6}$$

$$\tilde{\boldsymbol{t}}_k = \tilde{\boldsymbol{r}}_k^{\text{STAB}} - \alpha_k^{\text{PSTAB}} \tilde{A}\tilde{\boldsymbol{p}}_k^{\text{STAB}} \; = \; S_k^{\text{STAB}}(\tilde{A}) R_{k+1}(\tilde{A})\tilde{\boldsymbol{r}}_0. \tag{3.7}$$

The stabilized polynomial of PBiCGStab is

$$S_k^{\text{STAB}}(\tilde{\lambda}) = (1 - \omega_{k-1}^{\text{PSTAB}} \tilde{\lambda})(1 - \omega_{k-2}^{\text{PSTAB}} \tilde{\lambda}) \cdots (1 - \omega_0^{\text{PSTAB}} \tilde{\lambda}),$$

and contains no information regarding the direction of the preconditioned system, because α_k^{PSTAB} and β_k^{PSTAB} are not present.

Next, we consider PBiCGStab from the viewpoint of congruency [8]. β_k^{PSTAB} of the PBiCGStab has the following operation,

$$\omega_k^{\text{PSTAB}} = \frac{(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{\boldsymbol{t}}_k)}{(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)}. \tag{3.8}$$

The coefficient (3.8) defines the minimal residual (MR) operation of the polynomial of degree 1 that minimizes the norm of the residual vector in PBiCGStab, $||\tilde{\boldsymbol{r}}_{k+1}^{\text{STAB}}||$ [5, 18].

Theorem 3 ω_k^{PSTAB} *has no congruency on the direction of the preconditioning conversion, but its direction of the preconditioned system depends on the PBiCG regardless of the direction of the preconditioning conversion.*

Proof The three directions of the preconditioning conversion of (3.6), the two-sided (w), the left (l) and the right (r), are as follows, respectively:

$$M_l^{-1}\boldsymbol{r}_{k+1}^{\text{PSTAB}} = M_l^{-1}\boldsymbol{t}_k - \omega_k^w \tilde{A}_w (M_l^{-1}\boldsymbol{t}_k), \tag{3.9}$$

$$M^{-1}\boldsymbol{r}_{k+1}^{\mathrm{PSTAB}} = M^{-1}\boldsymbol{t}_k - \omega_k^l \tilde{A}_l (M^{-1}\boldsymbol{t}_k), \tag{3.10}$$

$$\boldsymbol{r}_{k+1}^{\mathrm{PSTAB}} = \boldsymbol{t}_k - \omega_k^r \tilde{A}_r \boldsymbol{t}_k. \tag{3.11}$$

These $\omega_k^{\mathrm{PSTAB}}$ corresponding to the above are

$$\omega_k^w = \frac{(\tilde{A}_w M_l^{-1}\boldsymbol{t}_k, M_l^{-1}\boldsymbol{t}_k)}{(\tilde{A}_w(M_l^{-1}\boldsymbol{t}_k), \tilde{A}_w(M_l^{-1}\boldsymbol{t}_k))} = \frac{(M_l^{-1}AM^{-1}\boldsymbol{t}_k, M_l^{-1}\boldsymbol{t}_k)}{(M_l^{-1}AM^{-1}\boldsymbol{t}_k, M_l^{-1}AM^{-1}\boldsymbol{t}_k)}, \tag{3.12}$$

$$\omega_k^l = \frac{(\tilde{A}_l M^{-1}\boldsymbol{t}_k, M^{-1}\boldsymbol{t}_k)}{(\tilde{A}_l(M^{-1}\boldsymbol{t}_k), \tilde{A}_l(M^{-1}\boldsymbol{t}_k))} = \frac{(M^{-1}AM^{-1}\boldsymbol{t}_k, M^{-1}\boldsymbol{t}_k)}{(M^{-1}AM^{-1}\boldsymbol{t}_k, M^{-1}AM^{-1}\boldsymbol{t}_k)}, \tag{3.13}$$

$$\omega_k^r = \frac{(\tilde{A}_r\boldsymbol{t}_k, \boldsymbol{t}_k)}{(\tilde{A}_r\boldsymbol{t}_k, \tilde{A}_r\boldsymbol{t}_k)} = \frac{(AM^{-1}\boldsymbol{t}_k, \boldsymbol{t}_k)}{(AM^{-1}\boldsymbol{t}_k, AM^{-1}\boldsymbol{t}_k)}. \tag{3.14}$$

Therefore, there is no congruency on the direction of the preconditioning conversion.

Next, we discuss the component of $M^{-1}\boldsymbol{t}_k$ in (3.12), which is the two-sided preconditioning conversion of (3.8), to analyze the direction of the preconditioned system. The initial residual vector of $\tilde{\boldsymbol{t}}_k$ is also $\tilde{\boldsymbol{r}}_0 = M_l^{-1}\boldsymbol{r}_0$. We assume the BiCG part of (3.7) to be the right preconditioned system (R), that is $R_{k+1}^{\mathrm{R}}(\tilde{A}_w)\tilde{\boldsymbol{r}}_0$. Then

$$\tilde{\boldsymbol{t}}_k^{\mathrm{R}} = S_k(\tilde{A}_w)R_{k+1}^{\mathrm{R}}(\tilde{A}_w)\tilde{\boldsymbol{r}}_0 = M_l^{-1}S_k(AM^{-1})R_{k+1}^{\mathrm{R}}(AM^{-1})\boldsymbol{r}_0 \equiv M_l^{-1}\boldsymbol{t}_k^{\mathrm{R}}.$$

Here, the superscript STAB of the polynomial S_k is omitted. In contrast, we assume the BiCG part of (3.7) to be the left preconditioned system (L). Then

$$\begin{aligned}\tilde{\boldsymbol{t}}_k^{\mathrm{L}} &= S_k(\tilde{A}_w)R_{k+1}^{\mathrm{L}}(\tilde{A}_w)\tilde{\boldsymbol{r}}_0 = M_r S_k(M^{-1}A)R_{k+1}^{\mathrm{L}}(M^{-1}A)M^{-1}\boldsymbol{r}_0\\ &= M_l^{-1}S_k(AM^{-1})R_{k+1}^{\mathrm{L}}(AM^{-1})\boldsymbol{r}_0 \equiv M_l^{-1}\boldsymbol{t}_k^{\mathrm{L}}.\end{aligned}$$

This is as per $\tilde{A}_w\tilde{\boldsymbol{t}}_k^{\mathrm{R}}$, $\tilde{A}_w\tilde{\boldsymbol{t}}_k^{\mathrm{L}}$ of (3.8). In any preconditioned system, they will reduce to ω_k^w of (3.12) superficially. Further, ω_k^r and ω_k^l also depend on the direction of the preconditioned system of the BiCG part. □

Remark 1 $\omega_k^{\mathrm{PSTAB}}$ plays a function in $\min\|M_l^{-1}\boldsymbol{r}_{k+1}^{\mathrm{PSTAB}}\|$, $\min\|M^{-1}\boldsymbol{r}_{k+1}^{\mathrm{PSTAB}}\|$ or $\min\|\boldsymbol{r}_{k+1}^{\mathrm{PSTAB}}\|$ for the residual vectors with preconditioning conversion of (3.9) to (3.11), respectively. Here, only $\min\|\boldsymbol{r}_{k+1}^{\mathrm{PSTAB}}\|$ of the right preconditioning conversion (r) minimizes "the original residual vector" without an operating preconditioner of (3.11).

However, it is not problematic to display only ω_k or $\omega_k^{\mathrm{PSTAB}}$ by broadly interpreting it as "ω_k with preconditioning conversion" when referring to ω_k in other algorithms. Therefore, only the MR coefficient in the PBiCGStab, that is ω_k^w, ω_k^l or ω_k^r, has no congruence property in a narrow sense. Further, these directions of the preconditioned system depend on the BiCG part.

Theorem 4 *The direction of a preconditioned system which depends on the BiCG part is determined by the coefficients of α_k and β_k in each PBiCGStab algorithm. These intrinsic operations are based on biorthogonality and biconjugacy. Further, the direction of PBiCGStab is switched by the construction and setting of the ISRV.*

Proof In the polynomial expressions of vectors $\tilde{\boldsymbol{r}}_k^{\mathrm{STAB}} = S_k^{\mathrm{STAB}}(\tilde{A})R_k(\tilde{A})\tilde{\boldsymbol{r}}_0$, $\tilde{\boldsymbol{p}}_k^{\mathrm{STAB}} = S_k^{\mathrm{STAB}}(\tilde{A})P_k(\tilde{A})\tilde{\boldsymbol{r}}_0$ and $\tilde{\boldsymbol{t}}_k = S_k^{\mathrm{STAB}}(\tilde{A})R_{k+1}(\tilde{A})\tilde{\boldsymbol{r}}_0$, the polynomial with the direction of the preconditioned system is only the BiCG part. That is $R_k(\tilde{A})\tilde{\boldsymbol{r}}_0$ and $P_k(\tilde{A})\tilde{\boldsymbol{r}}_0$. Here, the recurrences of $R_k(\tilde{\lambda})$, $P_k(\tilde{\lambda})$ contain $\alpha_k^{\mathrm{PSTAB}}$ and β_k^{PSTAB} composed by the operations of biorthogonality or biconjugacy; these are switched by the construction and setting of ISRV. □

The assertions in Theorem 4 were also noted in References [7, 8] where the PCGS and the PBiCG were analyzed with the congruence property (see Theorem 1 in this paper). Therein, it was proclaimed that the direction of the preconditioned system is not determined only by the linear system, but also depends on the construction and setting of the ISRV, for the preconditioned algorithms or the bi-Lanczos-type system. In this paper, we have confirmed this for the preconditioned algorithms of interest without the congruence property.

We verify Theorem 4 by various PBiCGStab algorithms in the next subsection and provide numerical results in Sect. 5.

3.2 Various preconditioned BiCGStab algorithms

In this subsection, six types of PBiCGStab algorithms are outlined. The first two algorithms are the right-PBiCGStab and left-PBiCGStab. These are based on Definition 1 with an extended application of Theorem 1 to the PBiCGStab. Subsequently, new variants of algorithms are presented. The first two algorithms are coordinative to the left-PBiCGStab (coLeft), and the algorithm switches the right-PBiCGStab to the left-preconditioned system by the construction and setting of ISRV9. To analyze algorithms with different directions between the BiCG and MR parts, the final two algorithms are given as follows. One is an algorithm with the left-preconditioned system of the BiCG part and the right-preconditioning conversion of the MR part that is mathematically equivalent to ISRV9 (Case1)[5]. The other is an algorithm with the right-preconditioned system of the BiCG part and the left-preconditioning conversion of the MR part (Case2). These six algorithms have two kinds of preconditioning operations in their iterative components. The Right and Left algorithms are both derived trivially. Only the coLeft algorithm is presented on the new variant PBiCGStab in this manuscript. However, Case1 with the polynomial expression is presented at the end of this section as a reference. Descriptions of all algorithms are presented in Appendix A. In the subsection, we only discuss algorithms based on BiCGStab, and we do not display " ˜ " (tilde) or the superscript of "(P)STAB".

[5] This algorithm has been reported as an implementation technique in Reference [6], and its point at issue was different from conventional (the right-) preconditioned BiCGStab.

3.2.1 Right- and Left-PBiCGStab algorithm

It is straightforward to derive both the Right and Left algorithms by the preconditioning conversion based on Definition 1, and by expressly providing the construction and setting of the ISRV.

Algorithm 1 Right-PBiCGStab algorithm (Right): The right-PBiCGStab is derived by Definition 1. This is the conventional PBiCGStab algorithm provided in References [1, 12, 18]. The construction and setting of the ISRV is $\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$, and the polynomials of the right preconditioned system are formed [8]. The standard stopping criteria are

$$\frac{\|\boldsymbol{t}_k\|}{\|\boldsymbol{b}\|} \leq \varepsilon \ \text{(Early check)}, \quad \frac{\|\boldsymbol{r}_{k+1}\|}{\|\boldsymbol{b}\|} \leq \varepsilon \ \text{(Convergence check)}. \tag{3.15}$$

Algorithm 2 Left-PBiCGStab algorithm (Left): The left-PBiCGStab is derived by Definition 1. The construction and setting of the ISRV is $\boldsymbol{r}_0^\sharp = \boldsymbol{r}_0^+$, and the polynomials of the left preconditioned system are formed [8]. The standard stopping criteria are

$$\frac{\|\boldsymbol{t}_k^+\|}{\|M^{-1}\boldsymbol{b}\|} \leq \varepsilon \ \text{(Early check)}, \quad \frac{\|\boldsymbol{r}_{k+1}^+\|}{\|M^{-1}\boldsymbol{b}\|} \leq \varepsilon \ \text{(Convergence check)}. \tag{3.16}$$

3.2.2 New variants of PBiCGStab algorithms

In this subsection, new variants of PBiCGStab algorithms are provided based on the foregoing Right and Left algorithms. For all algorithms, the standard stopping criteria are (3.15).

Algorithm 3 is derived as follows. Applying the same preconditioning conversion as Algorithm 2, splitting the residual vector $\boldsymbol{r}_k^+$ of Algorithm 2 into $M^{-1}\boldsymbol{r}_k$, and adding the recurrences for the residual vectors of $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k^+$ and $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega_k A\boldsymbol{t}_k^+$. In this iterative part, both $\boldsymbol{r}_{k+1}$ and $\boldsymbol{r}_{k+1}^+$ are calculated. The construction and setting of the ISRV is $\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$. This algorithm is coordinative to the Left PBiCGStab, and this BiCG part is the left preconditioned system. The MR part is the left preconditioning conversion.

Algorithm 3 Coordinative to left-PBiCGStab algorithm (coLeft) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$,
$\boldsymbol{r}_0^+ \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \ldots,$ Do:
```

$$\begin{aligned}
\boldsymbol{p}_k^+ &= \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \omega_{k-1} M^{-1} A \boldsymbol{p}_{k-1}^+),\\
\alpha_k &= \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1} A \boldsymbol{p}_k^+ \rangle},\\
\boldsymbol{t}_k &= \boldsymbol{r}_k - \alpha_k A \boldsymbol{p}_k^+,\\
\boldsymbol{t}_k^+ &= \boldsymbol{r}_k^+ - \alpha_k M^{-1} A \boldsymbol{p}_k^+,\\
\omega_k &= \frac{(M^{-1} A \boldsymbol{t}_k^+, \boldsymbol{t}_k^+)}{(M^{-1} A \boldsymbol{t}_k^+, M^{-1} A \boldsymbol{t}_k^+)},\\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \omega_k \boldsymbol{t}_k^+,\\
\boldsymbol{r}_{k+1} &= \boldsymbol{t}_k - \omega_k A \boldsymbol{t}_k^+,\\
\boldsymbol{r}_{k+1}^+ &= \boldsymbol{t}_k^+ - \omega_k M^{-1} A \boldsymbol{t}_k^+,\\
\beta_k &= \frac{\alpha_k}{\omega_k} \times \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle},
\end{aligned}$$

End Do

Algorithm 4 Right-PBiCGStab with ISRV9 (ISRV9) : This algorithm is derived by replacing the ISRV of the Right (Algorithm 1) with $\boldsymbol{r}_0^\flat = M^{-\mathrm{T}} M^{-1} \boldsymbol{r}_0$ (ISRV9). This BiCG part is the left preconditioned system and the MR part is the right preconditioning conversion.

Algorithm 5 BiCG-left and MR-right PBiCGStab algorithm (Case1): This algorithm is mathematically equivalent to ISRV9 (Algorithm 4). That is, the BiCG part is the left preconditioned system and the MR part is the right preconditioning conversion.

Algorithm 6 BiCG-right and MR-left PBiCGStab algorithm (Case2): This algorithm is composed of the opposite combination to Case1 (Algorithm 5). That is, the BiCG part is the right preconditioned system and the MR part is the left preconditioning conversion.

3.2.3 Features of various preconditioned BiCGStab algorithms

Table 1 shows the characteristics of the PBiCGStab algorithms in this section.

There are differences between the Left (Algorithm 2) and the coLeft (Algorithm 3). The residual vector $\boldsymbol{r}_k^+$ of the Left is split into $M^{-1}\boldsymbol{r}_k$ of the coLeft, and the stopping criteria are different (see, Definition 5). However, the true residual vector $\boldsymbol{b} - A\boldsymbol{x}_k$ and the true error $\boldsymbol{x}_k - \boldsymbol{x}_{\text{exact}}$ are mathematically equivalent between both algorithms. ISRV9 (Algorithm 4) is mathematically equivalent to Case1 (Algorithm 5). The Right is the same algorithm as ISRV9, except for the construction and setting of the ISRV. Therefore, we may confirm that the direction of a preconditioned system for the BiCGStab is switched by the construction and setting of the ISRV.

As a reference, Case1 (Algorithm 5) with the polynomial expression is shown; double underline is used to indicate the polynomials.

Table 1 Characteristics of the six types of PBiCGStab algorithms. In the column of α_k, β_k, "R" indicates the right-preconditioned system and "L" indicates the left-preconditioned system. In the column of ω_k, "r" indicates the right-direction of preconditioning conversion and "l" indicates the left-direction of preconditioning conversion

Name (Alg. no.)	ISRV	α_k, β_k	ω_k	Standard stopping criterion
Right (Alg. 1)	$\boldsymbol{r}^\flat = \boldsymbol{r}_0$	R	r	(3.15)
Left (Alg. 2)	$\boldsymbol{r}^\sharp = \boldsymbol{r}_0^+$	L	l	(3.16)
coLeft (Alg. 3)	$\boldsymbol{r}^\sharp = M^{-1}\boldsymbol{r}_0$	L	l	(3.15)
ISRV9 (Alg. 4)	$\boldsymbol{r}^\flat = M^{-\mathrm{T}}M^{-1}\boldsymbol{r}_0$	L	r	(3.15)
Case 1 (Alg. 5)	$\boldsymbol{r}^\sharp = M^{-1}\boldsymbol{r}_0$	L	r	(3.15)
Case 2 (Alg. 6)	$\boldsymbol{r}^\flat = \boldsymbol{r}_0$	R	l	(3.15)

Algorithm 5′ Polynomial description of the BiCG-left and MR-right PBiCGStab algorithm (Case 1) :

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta^{\mathrm{L}}_{-1} = 0$,

$\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$,

$\boldsymbol{r}_0^+ \equiv M^{-1}\boldsymbol{r}_0$,

For $k = 0, 1, 2, 3, \dots,$ Do:

$$\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta^{\mathrm{L}}_{k-1}(\boldsymbol{p}_{k-1}^+ - \omega^r_{k-1}M^{-1}A\boldsymbol{p}_{k-1}^+) = \underline{\underline{S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},$$

$$\alpha^{\mathrm{L}}_k = \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+\rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1}A\boldsymbol{p}_k^+\rangle} = \underline{\underline{\frac{\langle M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}{\langle M^{-1}\boldsymbol{r}_0, \tilde{A}_l S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}}},$$

$$\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha^{\mathrm{L}}_k A\boldsymbol{p}_k^+ = \underline{\underline{MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},$$

$$\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha^{\mathrm{L}}_k M^{-1}A\boldsymbol{p}_k^+ = \underline{\underline{S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},$$

$$\begin{aligned}\omega^r_k &= \frac{(AM^{-1}\boldsymbol{t}_k, \boldsymbol{t}_k)}{(AM^{-1}\boldsymbol{t}_k, AM^{-1}\boldsymbol{t}_k)} \equiv \frac{(A\boldsymbol{t}_k^+, \boldsymbol{t}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)} \\ &= \underline{\underline{\frac{(\tilde{A}_r MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0)}{(\tilde{A}_r MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0, \tilde{A}_r MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0)}}},\end{aligned}$$

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha^{\mathrm{L}}_k\boldsymbol{p}_k^+ + \omega^r_k\boldsymbol{t}_k^+ \\ &= \boldsymbol{x}_k + \alpha^{\mathrm{L}}_k\underline{\underline{S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}} + \omega^r_k\underline{\underline{S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\end{aligned}$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega^r_k A\boldsymbol{t}_k^+ = \underline{\underline{MS_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},$$

$$\boldsymbol{r}_{k+1}^+ = M^{-1}\boldsymbol{r}_{k+1} = \underline{\underline{M^{-1}\big(MS_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\big)}},$$

$$\beta^{\mathrm{L}}_k = \frac{\alpha^{\mathrm{L}}_k}{\omega^r_k} \times \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+\rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+\rangle} = \frac{\alpha^{\mathrm{L}}_k}{\omega^r_k} \times \underline{\underline{\frac{\langle M^{-1}\boldsymbol{r}_0, S_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}{\langle M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}}},$$

End Do

4 Analyses for various preconditioned GPBiCG algorithms

In this section, first we provide an outline for deriving the GPBiCG method for the preconditioned system, and propose some corollaries on the characteristics of the PGPBiCG. Next, we give outlines of the six types of PGPBiCG algorithms. These mathematical structure of the characteristics correspond to the six types of PBiCG-Stab in the former section.

4.1 Derivation of the GPBiCG method from PBiCG

Derivation of the GPBiCG for the preconditioned system proceeds in a similar manner as the derivation of the PBiCGStab in section 3. This is particularly the case with respect to the derivation procedure from (3.1) to (3.4); except that the superscripts of $\alpha_k^{\mathrm{PSTAB}}$, β_k^{PSTAB}, and $\omega_k^{\mathrm{PSTAB}}$ are changed of PGP.

Here, we state the polynomials of the residual vector and probing direction vector as

$$\tilde{\boldsymbol{r}}_k^{\mathrm{GP}} \equiv S_k(\tilde{A})R_k(\tilde{A})\tilde{\boldsymbol{r}}_0, \quad \tilde{\boldsymbol{p}}_k^{\mathrm{GP}} \equiv S_k(\tilde{A})P_k(\tilde{A})\tilde{\boldsymbol{r}}_0.$$

Then, the following Corollary uses (3.2) to (3.4):

Corollary 1

$$\alpha_k^{\mathrm{PBiCG}} = \frac{\langle \tilde{\boldsymbol{r}}_k^{\sharp}, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle}{\langle \tilde{\boldsymbol{p}}_k^{\sharp}, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{BiCG}} \rangle} \equiv \frac{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{\boldsymbol{r}}_k^{\mathrm{GP}} \rangle}{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{GP}} \rangle} = \alpha_k^{\mathrm{PGP}},$$

$$\beta_k^{\mathrm{PBiCG}} = \frac{\langle \tilde{\boldsymbol{r}}_{k+1}^{\sharp}, \tilde{\boldsymbol{r}}_{k+1}^{\mathrm{BiCG}} \rangle}{\langle \tilde{\boldsymbol{r}}_k^{\sharp}, \tilde{\boldsymbol{r}}_k^{\mathrm{BiCG}} \rangle} \equiv \frac{\alpha_k^{\mathrm{PGP}}}{\omega_k^{\mathrm{PGP}}} \times \frac{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{\boldsymbol{r}}_{k+1}^{\mathrm{GP}} \rangle}{\langle \tilde{\boldsymbol{r}}_0^{\sharp}, \tilde{\boldsymbol{r}}_k^{\mathrm{GP}} \rangle} = \beta_k^{\mathrm{PGP}}.$$

The residual vector and polynomial of the GPBiCG method for the preconditioned system are expressed as follows:

$$\begin{aligned} \tilde{\boldsymbol{r}}_{k+1}^{\mathrm{GP}} &= \tilde{\boldsymbol{t}}_k - \eta_k^{\mathrm{PGP}}\tilde{\boldsymbol{y}}_k - \omega_k^{\mathrm{PGP}}\tilde{A}\tilde{\boldsymbol{t}}_k = S_{k+1}^{\mathrm{GP}}(\tilde{A})R_{k+1}(\tilde{A})\tilde{\boldsymbol{r}}_0, \\ \tilde{\boldsymbol{t}}_k &= \tilde{\boldsymbol{r}}_k^{\mathrm{GP}} - \alpha_k^{\mathrm{PGP}}\tilde{A}\tilde{\boldsymbol{p}}_k^{\mathrm{GP}} = S_k^{\mathrm{GP}}(\tilde{A})R_{k+1}(\tilde{A})\tilde{\boldsymbol{r}}_0. \end{aligned} \tag{4.1}$$

The stabilized polynomial of PGPBiCG is

$$\begin{aligned} &S_0^{\mathrm{GP}}(\tilde{\lambda}) = 1, \quad S_1^{\mathrm{GP}}(\tilde{\lambda}) = (1 - \omega_0^{\mathrm{PGP}}\tilde{\lambda})S_0^{\mathrm{GP}}(\tilde{\lambda}), \\ &S_{k+1}^{\mathrm{GP}}(\tilde{\lambda}) = (1 + \eta_k^{\mathrm{PGP}} - \omega_k^{\mathrm{PGP}}\tilde{\lambda})S_k^{\mathrm{GP}}(\tilde{\lambda}) - \eta_k^{\mathrm{PGP}}S_{k-1}^{\mathrm{GP}}(\tilde{\lambda}), \quad (k \geq 1). \end{aligned} \tag{4.2}$$

The coefficients of η_k^{PGP}, ω_k^{PGP} compose the minimal residual (MR) operation of the polynomial of degree 2 that minimizes the norm of the residual vector in PGPBiCG, $||\tilde{\boldsymbol{r}}_{k+1}^{\mathrm{GP}}||$ [4, 5, 19]. Therefore, the effect of the MR part in PGPBiCG is higher than in PBiCGStab.

For projection onto the two-dimensional subspace spanned by $\tilde{\boldsymbol{y}}_k$ and $\tilde{A}\tilde{\boldsymbol{t}}_k$, we define the $n \times 2$ matrix $\tilde{B} \equiv \left[\tilde{\boldsymbol{y}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k\right]$. Hence, the optimal coefficients η_k^{PGP} and ω_k^{PGP} for (4.1) are calculated as follows:

$$\tilde{B}^{\mathrm{T}}\tilde{B}\begin{bmatrix} \eta_k^{\mathrm{PGP}} \\ \omega_k^{\mathrm{PGP}} \end{bmatrix} = \tilde{B}^{\mathrm{T}}\,\tilde{\boldsymbol{t}}_k, \quad \text{therefore,} \quad \begin{bmatrix} \eta_k^{\mathrm{PGP}} \\ \omega_k^{\mathrm{PGP}} \end{bmatrix} = (\tilde{B}^{\mathrm{T}}\tilde{B})^{-1}\tilde{B}^{\mathrm{T}}\,\tilde{\boldsymbol{t}}_k.$$

After all, we obtain

$$\omega_k^{\mathrm{PGP}} = \frac{(\tilde{\boldsymbol{y}}_k, \tilde{\boldsymbol{y}}_k)(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{\boldsymbol{t}}_k) - (\tilde{\boldsymbol{y}}_k, \tilde{\boldsymbol{t}}_k)(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{\boldsymbol{y}}_k)}{(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)(\tilde{\boldsymbol{y}}_k, \tilde{\boldsymbol{y}}_k) - (\tilde{\boldsymbol{y}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{\boldsymbol{y}}_k)}, \tag{4.3}$$

$$\eta_k^{\mathrm{PGP}} = \frac{(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)(\tilde{\boldsymbol{y}}_k, \tilde{\boldsymbol{t}}_k) - (\tilde{\boldsymbol{y}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{\boldsymbol{t}}_k)}{(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)(\tilde{\boldsymbol{y}}_k, \tilde{\boldsymbol{y}}_k) - (\tilde{\boldsymbol{y}}_k, \tilde{A}\tilde{\boldsymbol{t}}_k)(\tilde{A}\tilde{\boldsymbol{t}}_k, \tilde{\boldsymbol{y}}_k)}. \tag{4.4}$$

Here, $\tilde{\boldsymbol{y}}_k = \tilde{A}Q_{k-1}(\tilde{A})R_{k+1}(\tilde{A})\tilde{\boldsymbol{r}}_0$, $Q_{k-1}(\tilde{\lambda}) = (S_{k-1}^{\mathrm{GP}}(\tilde{\lambda}) - S_k^{\mathrm{GP}}(\tilde{\lambda}))/\tilde{\lambda}$. Then, the recurrences of (4.2) are transformed into the following: $S_0^{\mathrm{GP}}(\tilde{\lambda}) = 1$, $Q_0(\tilde{\lambda}) = \omega_0^{\mathrm{PGP}}$, $S_k^{\mathrm{GP}}(\tilde{\lambda}) = S_{k-1}^{\mathrm{GP}}(\tilde{\lambda}) - \tilde{\lambda}Q_{k-1}(\tilde{\lambda})$, $Q_k(\tilde{\lambda}) = \omega_k^{\mathrm{PGP}}S_k^{\mathrm{GP}}(\tilde{\lambda}) + \eta_k^{\mathrm{PGP}}Q_{k-1}(\tilde{\lambda})\,(k \geq 1)$.

The following Corollary resonates with Theorem 3 of the PBiCGStab.

Corollary 2 *ω_k^{PGP} and η_k^{PGP} have no congruency on the direction of the preconditioning conversion, but the direction of the preconditioned system depends on the PBiCG regardless of the direction of the preconditioning conversion.*

The following are the coefficients of (4.3), (4.4) after preconditioning conversion to the right (r) and the left (l).

$$\omega_k^r = \frac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(A\boldsymbol{t}_k^+, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}, \tag{4.5}$$

$$\eta_k^r = \frac{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{t}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}. \tag{4.6}$$

$$\omega_k^l = \frac{(\boldsymbol{y}_k^+, \boldsymbol{y}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{t}_k^+) - (\boldsymbol{y}_k^+, \boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{y}_k^+)}{(M^{-1}A\boldsymbol{t}_k^+, M^{-1}A\boldsymbol{t}_k^+)(\boldsymbol{y}_k^+, \boldsymbol{y}_k^+) - (\boldsymbol{y}_k^+, M^{-1}A\boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{y}_k^+)}, \tag{4.7}$$

$$\eta_k^l = \frac{(M^{-1}A\boldsymbol{t}_k^+, M^{-1}A\boldsymbol{t}_k^+)(\boldsymbol{y}_k^+, \boldsymbol{t}_k^+) - (\boldsymbol{y}_k^+, M^{-1}A\boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{t}_k^+)}{(M^{-1}A\boldsymbol{t}_k^+, M^{-1}A\boldsymbol{t}_k^+)(\boldsymbol{y}_k^+, \boldsymbol{y}_k^+) - (\boldsymbol{y}_k^+, M^{-1}A\boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{y}_k^+)}. \tag{4.8}$$

Remark 2 The ω_k^r, η_k^r of the right preconditioning conversion minimizes the norm of the original residual vector $\|\boldsymbol{r}_{k+1}^{\mathrm{PGP}}\|$ without operating a preconditioner, as with *Remark* 1 for the PBiCGStab.

The following Corollary resonates with Theorem 4 of the PBiCGStab. We verify Corollary 3 by various PGPBiCG algorithms in the next subsection with numerical results provided in section 5.

Corollary 3 *The direction of a preconditioned system which depends on the BiCG part is determined by the coefficients of α_k and β_k in each PGPBiCG algorithm. These intrinsic operations are based on biorthogonality and biconjugacy. Further, the direction of PGPBiCG is switched by the construction and setting of the ISRV.*

4.2 Various preconditioned GPBiCG algorithms

In this subsection, the six types of PGPBiCG algorithms are shown and their mathematical structures are summarized in Sect. 4.2.3. These six algorithms have two kinds of preconditioning operations in their iterative part. Descriptions of all algorithms are presented in the Appendix B. We only discuss algorithms based on GPBiCG, and do not display " ˜ (tilde)" or the superscript of "(P)GP".

4.2.1 Right- and Left-PGPBiCG algorithm

The Right and Left algorithms are derived as follows.

Algorithm 7 Right-PGPBiCG algorithm (Right): The "Right-PGPBiCG algorithm" based on Definition 1 has three kinds of preconditioning operations ($M^{-1}\boldsymbol{p}_k$, $M^{-1}\boldsymbol{t}_k$, and $M^{-1}\boldsymbol{z}_k$). We state $\check{\boldsymbol{z}}_k \equiv M^{-1}\boldsymbol{z}_k$, and use $\check{\boldsymbol{r}}_k(\equiv M^{-1}\boldsymbol{r}_k)$, $\check{\boldsymbol{u}}_k(\equiv M^{-1}\boldsymbol{u}_k)$. Further, we apply the necessary transformation. The construction and setting of the ISRV is $\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$ and the polynomials of the right preconditioned system are formed [8]. The standard stopping criteria are (3.15); $\|\boldsymbol{t}_k\|/\|\boldsymbol{b}\| \le \varepsilon, \|\boldsymbol{r}_{k+1}\|/\|\boldsymbol{b}\| \le \varepsilon$.

Algorithm 8 Left-PGPBiCG algorithm (Left): We derive the left-PGPBiCG by Definition 1. The construction and setting of the ISRV is $\boldsymbol{r}_0^\sharp = \boldsymbol{r}_0^+$ and the polynomials of the left preconditioned system are formed [8]. The standard stopping criteria are (3.16); $\|\boldsymbol{t}_k^+\|/\|M^{-1}\boldsymbol{b}\| \le \varepsilon, \|\boldsymbol{r}_{k+1}^+\|/\|M^{-1}\boldsymbol{b}\| \le \varepsilon$.

4.2.2 New variants of PGPBiCG algorithms

In this subsection, new variants of PGPBiCG algorithms are provided. In the same manner as deriving new PBiCGStab variants, they are also based on the former Right (Algorithm 7) or Left (Algorithm 8) algorithms. For all algorithms, the standard stopping criteria are (3.15).

Algorithm 9 Coordinative to left-PGPBiCG algorithm (coLeft): This algorithm is coordinative to the Left PGPBiCG and the BiCG part is the left preconditioned system. The MR part is the left preconditioning conversion. The construction and setting of the ISRV is $\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$.

Algorithm 10 Right-PGPBiCG with ISRV9 (ISRV9): This algorithm is derived by replacing the ISRV of the Right (Algorithm 7) with $\boldsymbol{r}_0^\flat = M^{-\mathrm{T}}M^{-1}\boldsymbol{r}_0$ (ISRV9). The BiCG part is the left preconditioned system and the MR part is the right preconditioning conversion.

Algorithm 11 BiCG-left and MR-right PGPBiCG algorithm (Case 1): This algorithm is mathematically equivalent to ISRV9 (Algorithm 10). That is, the BiCG part is the left preconditioned system and the MR part is the right preconditioning conversion.

Algorithm 12 BiCG-right and MR-left PGPBiCG algorithm (Case 2): This algorithm is composed of the opposite combination to Case1 (Algorithm 11). That is, the BiCG part is the right preconditioned system and the MR part is the left preconditioning conversion.

4.2.3 Features of various preconditioned GPBiCG algorithms

Table 2 shows the characteristics of the PGPBiCG algorithms in this section.

Table 2 Characteristics of the six types of PGPBiCG algorithms. In the column of α_k, β_k, "R" indicates the right-preconditioned system and "L" indicates the left-preconditioned system. In the column of ω_k, η_k, " r " indicates the right-direction of preconditioning conversion and " l " indicates the left-direction of preconditioning conversion

Name (Alg. no.)	ISRV	α_k, β_k	ω_k, η_k	Standard stopping criterion
Right (Alg. 7)	$\boldsymbol{r}^\flat = \boldsymbol{r}_0$	R	r	(3.15)
Left (Alg. 8)	$\boldsymbol{r}^\sharp = \boldsymbol{r}_0^+$	L	l	(3.16)
coLeft (Alg. 9)	$\boldsymbol{r}^\sharp = M^{-1}\boldsymbol{r}_0$	L	l	(3.15)
ISRV9 (Alg. 10)	$\boldsymbol{r}^\flat = M^{-\mathrm{T}}M^{-1}\boldsymbol{r}_0$	L	r	(3.15)
Case1 (Alg. 11)	$\boldsymbol{r}^\sharp = M^{-1}\boldsymbol{r}_0$	L	r	(3.15)
Case2 (Alg. 12)	$\boldsymbol{r}^\flat = \boldsymbol{r}_0$	R	l	(3.15)

5 Numerical experiments

In Sect. 5.1, we verify that the numerical values of α_k and β_k of PBiCG, PBiCGStab and PGPBiCG are equivalent for each direction of the preconditioned system (Proposition 3, Corollary 1). Further, we confirm the characteristics of the six types of algorithms (Right, Left, coLeft, ISRV9, Case1, and Case2), by the numerical values of α_k, β_k, ω_k, and η_k. We also confirm that the direction of the preconditioned system which depends on the BiCG part is determined by the coefficients of α_k and β_k in each PBiCGStab, PGPBiCG algorithm, and the direction of PBiCGStab is switched by the construction and setting of the ISRV (Theorem 4, Corollary 3).

In Sect. 5.2, we present numerical results associated with the six types of PBiCGStab and PGPBiCG in Sects. 3.2 and 4.2, and the numerical results of the three types of PBiCGStab and PGPBiCG with changing over of the stopping criterion [9].

The test problems were generated by building real nonsymmetric matrices corresponding to linear systems taken from the University of Florida Sparse Matrix Collection [2] and the Matrix Market [11]. The RHS vector $\boldsymbol{b}$ of (1.1) was generated in two ways:

- RHSone] : Setting all elements of $\boldsymbol{x}_{\text{exact}}$ as 1.0 and substituting this into (1.1).
- RHS `+ nn`] : Setting all elements of $\boldsymbol{x}_{\text{exact}}$ as Mersenne Twister pseudo-random numbers and substituting this into (1.1). Here, we term this RHS "RHS `+ nn`", because the random numbers are generated using a seed number (`nn`). Ex.) RHS `+ 10` means that $\boldsymbol{b} = A\boldsymbol{x}_{\text{exact}}$ and $\boldsymbol{x}_{\text{exact}}$ is generated by `seed = 10`.

The numerical experiments were executed on a DELL PowerEdge R515 (AMD Opteron 4133) running Cent OS (kernel 2.6.32) and Intel compiler 16.0.2. The solution algorithm was implemented using the sequential mode of the Lis numerical computation library 1.1.2 [16] in double precision, with compiler options registered in the Lis "Makefile." In all tests, the initial solution was set to $\boldsymbol{x}_0 = \boldsymbol{0}$. ILU(0) was adopted as a preconditioning operation with PBiCG, PBiCGStab and PGPBiCG algorithms; here, the value "zero" means the *fill-in* level. We evaluated the algorithm's relative residual 2-norm (ARR). We adopted (3.15) as the stopping criterion in all cases except for the left-preconditioned algorithms where we used (3.16). We set $\varepsilon = 10^{-12}$ in all cases. In all tables, the true relative residual (TRR) $\log_{10}$ 2-norm ($\|\boldsymbol{b} - A\hat{\boldsymbol{x}}\|/\|\boldsymbol{b}\|$) and the true relative error (TRE) $\log_{10}$ 2-norm ($\|\hat{\boldsymbol{x}} - \boldsymbol{x}_{\text{exact}}\|/\|\boldsymbol{x}_{\text{exact}}\|$) at convergence are stated. Here, $\hat{\boldsymbol{x}}$ means the numerical solution. The maximum number of iterations (mx) was set to 1000, except for `viscoplastic2` (mx = 2000) and `young3c` (mx = 2500) matrices solved by PBiCGStab, and `young3c` (mx = 5000) matrix solved by PGPBiCG.

5.1 Verification of PBiCG, PBiCGStab and PGPBiCG algorithms

The `add32` matrix was used and the linear equations were generated using RHSone.

We confirm Proposition 3 and Corollary 1. Figure 1 plots values of α_k and Fig. 2 plots values of β_k for PBiCG, PBiCGStab and PGPBiCG. In both figures, the upper displays the values of the right-preconditioned algorithms and the lower displays the values of the left-preconditioned algorithms.

We plotted the values of α_k and β_k for five algorithm types, except for coLeft[6] of PBiCGStab algorithms or PGPBiCG algorithms presented in Sects. 3.2 or 4.2; these are shown in Figs. 3, 4, 5, 6. From these figures[7], we confirm the correspondence relations between the behavior of α_k and β_k and the direction of the preconditioned system summarized in Tables 1, 2. That is to say, behaviors of α_k and β_k are the same between Right and Case2 (right-preconditioned system algorithms), and the behaviors of α_k and β_k are the same among Left, ISRV9 and Case1 (left-preconditioned system algorithms). On the other hand, behaviors of α_k and β_k are not the same between the right-preconditioned system algorithms and the left-preconditioned system algorithms. It is very important that the directions of the preconditioned system for ISRV9, Case1 and Case2 accord with the directions of each α_k and β_k in spite of the fact that the direction of the MR operators in these algorithms differ from the directions of each α_k and β_k; see Theorem 3 and Corollary 2. Further, we have numerically confirmed Theorem 4 and Corollary 3, because the ISRV9 and Right algorithms are identical, except for the construction and setting of the ISRV.

We have also confirmed that both ω_k values of the ISRV9 and Case1 algorithms of the PBiCGStab, the PGPBiCG, and η_k values of the PGPBiCG were the same in Figs. 7, 8 and 9; these algorithms were equivalent. In contrast, both values of the other algorithms of the above ω_k and η_k differ from each other. We have again numerically confirmed Theorems 3 and 4, and Corollary 2, 3.

As reference information, other numerical results for `add32` using the relative residual of each algorithm are listed in Appendix C.

In the next subsection, we will discuss various results by solving the six types of PBiCGStab and PGPBiCG.

5.2 PBiCGStab and PGPBiCG results

In this subsection, we compare and evaluate the performance of various PBiCGStab and PGPBiCG algorithms for solving the linear equations of RHSone. The results with the standard stopping criterion are shown in Tables 3, 4. Further, the results from using three types of algorithms (RtL, coLtL, C1tL) with stopping criterion changeover [9], are shown in Tables 5, 6. Here, "RtL" means the *Right* (*Algorithm* 1, 7) *with stopping criterion* (3.15) *to the Left criterion* (3.16)[8] by the changeover. In a similar way, "coLtL" means the *coLeft to the Left criterion* by the changeover, and "C1tL" means the *Case1 to the Left criterion* by the changeover.

Further, we show results from replacing the RHS for the latter three types of algorithms (PBiCGStab: Tables 7, 8, 9; PGPBiCG: Tables 10, 11 12).

In these tables, significant disadvantages of one algorithm over the other are emphasized by **bold font**. For example, poor accuracy over 10^{-8} on the TRR or the

[6] coLeft was excluded in this validation because α_k, β_k, ω_k and η_k are identical with Left mathematically and in programming code.

[7] The same β_k values are plotted twice at the right-end in the figures, because these program codes log twice at the iteration stop; there is no mathematical basis for this duplication.

[8] The right-PBiCGStab algorithm changed over only at the early check, that is, changing $\|\boldsymbol{t}_k\|/\|\boldsymbol{b}\|$ to $\|M^{-1}\boldsymbol{t}_k\|/\|M^{-1}\boldsymbol{b}\|$, because this algorithm has no $M^{-1}\boldsymbol{r}_k$ operation and we avoid excessive calculation on the preconditioning operation.

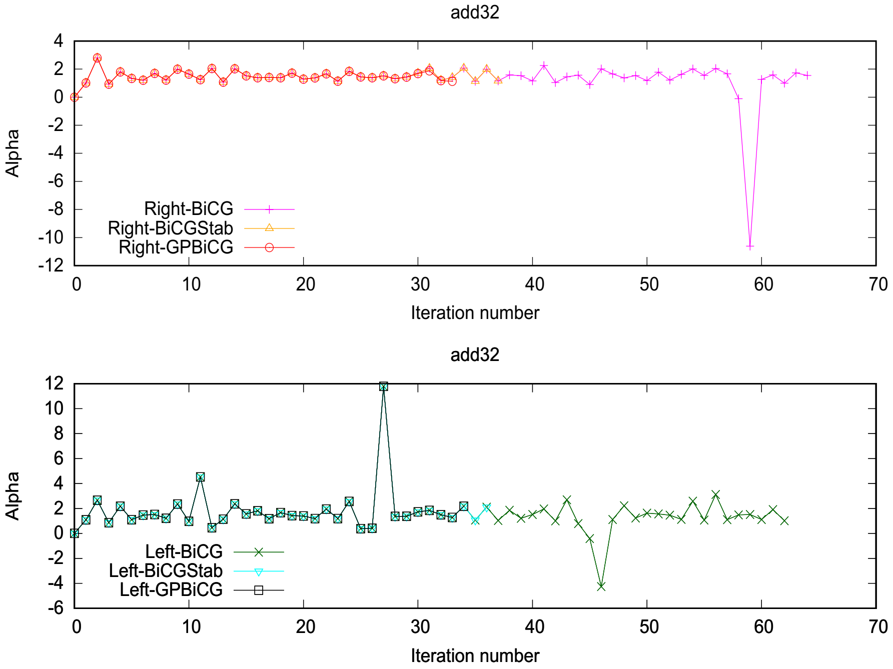

Fig. 1 Values of α_k for the right- (upper) and left- (lower) PBiCG, PBiCGStab and PGPBiCG

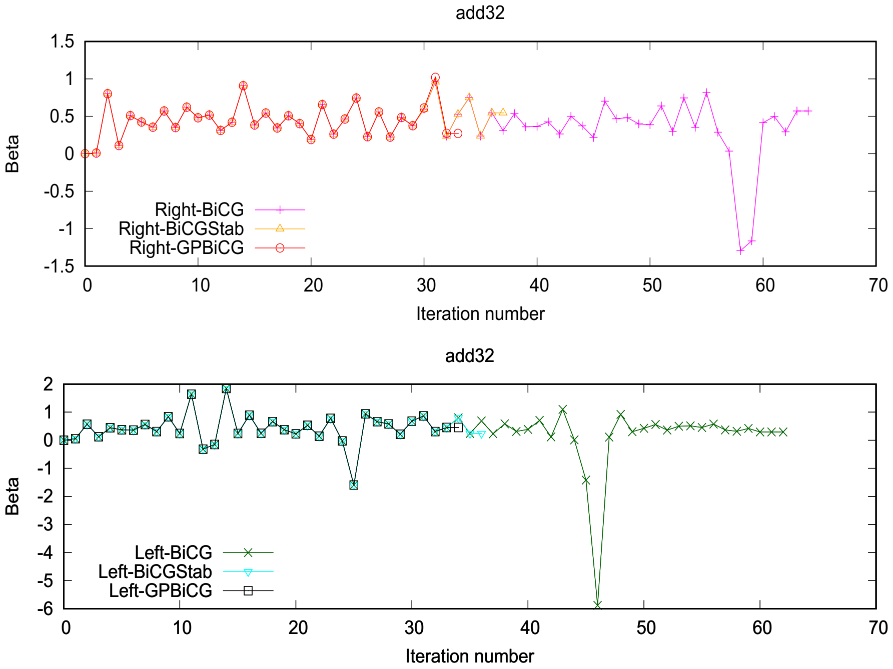

Fig. 2 Values of β_k for the right- (upper) and left- (lower) PBiCG, PBiCGStab and PGPBiCG

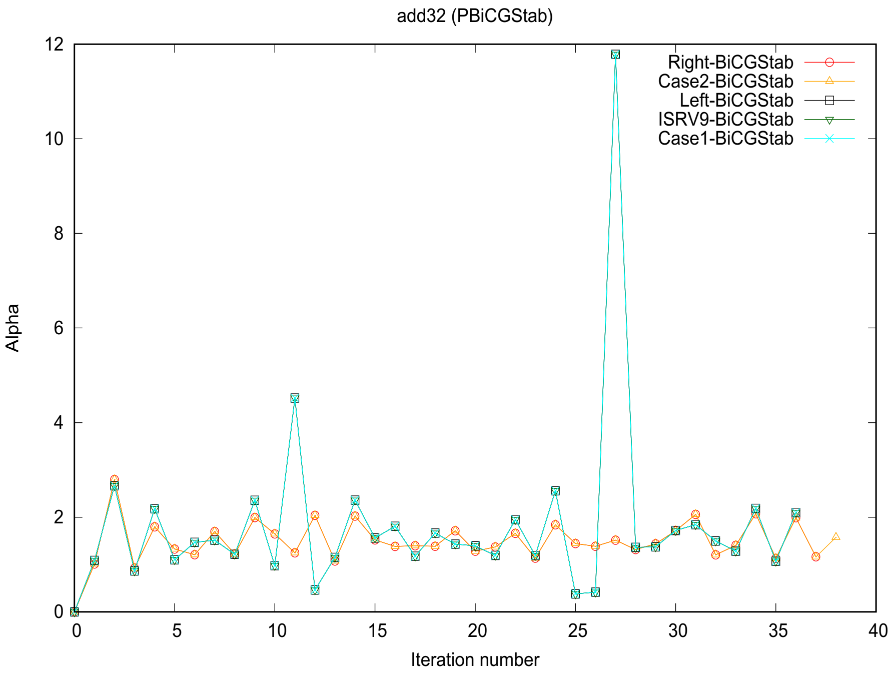

Fig. 3 Values of α_k for each PBiCGStab

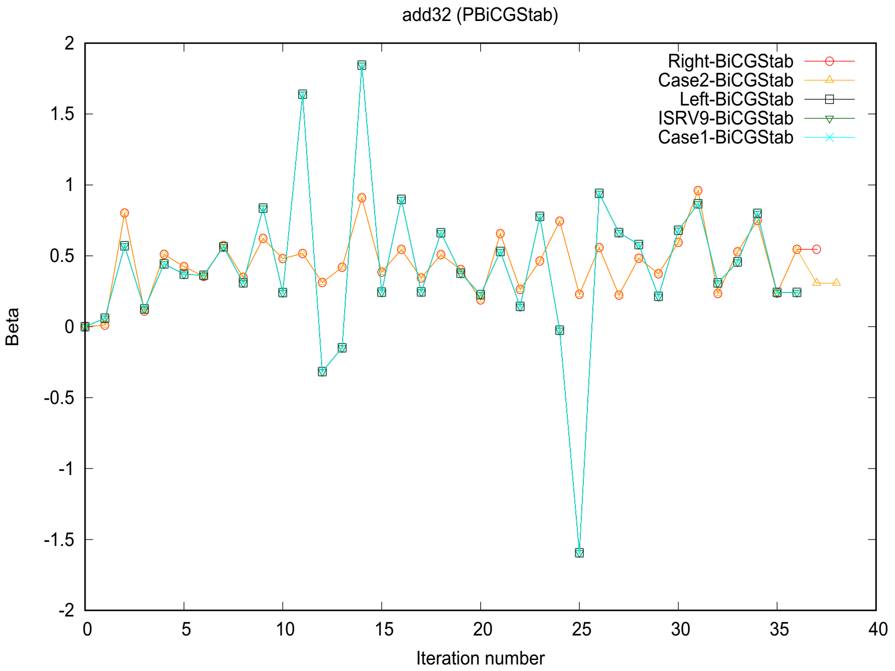

Fig. 4 Values of β_k for each PBiCGStab

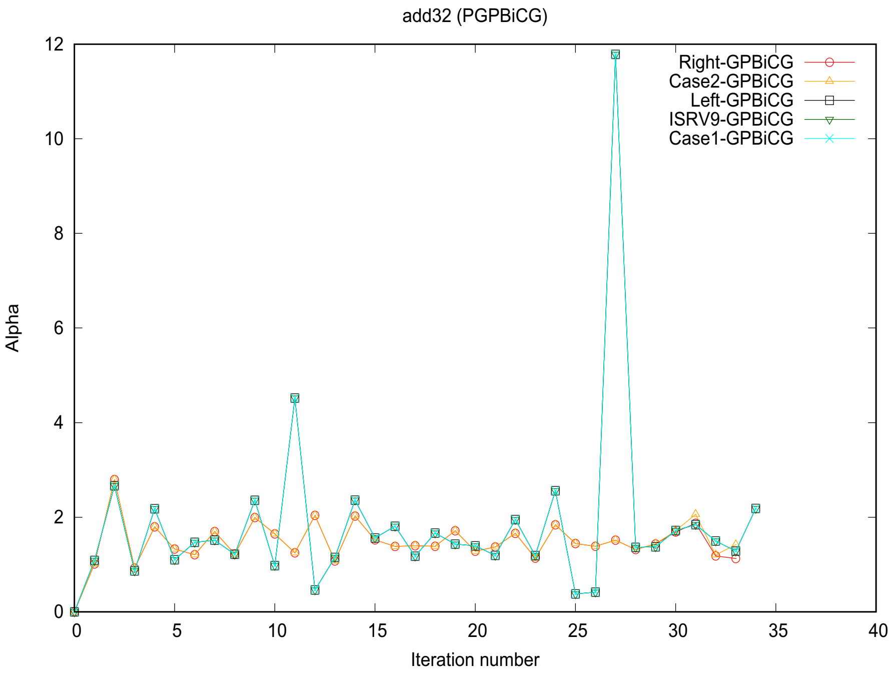

Fig. 5 Values of α_k for each PGPBiCG

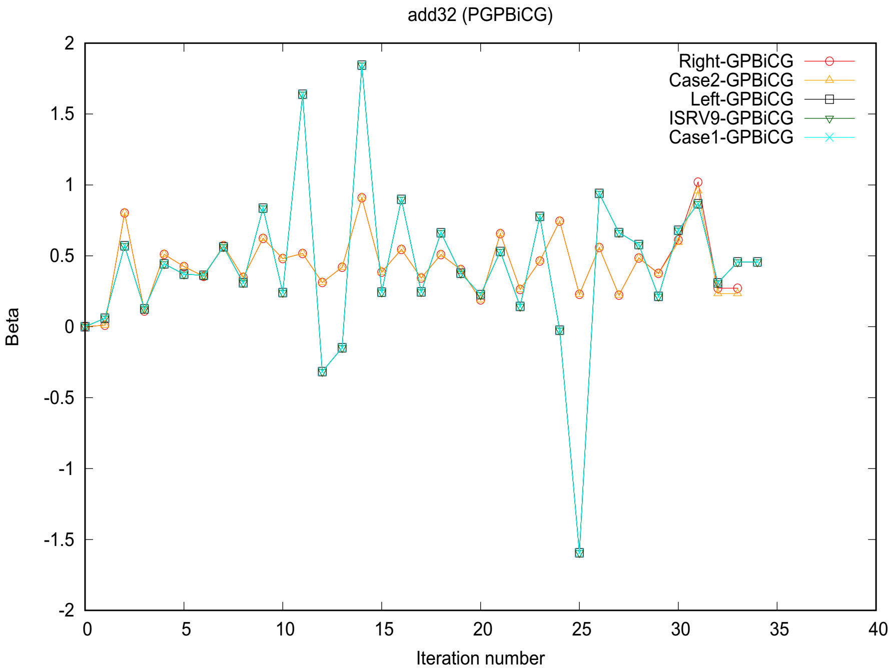

Fig. 6 Values of β_k for each PGPBiCG

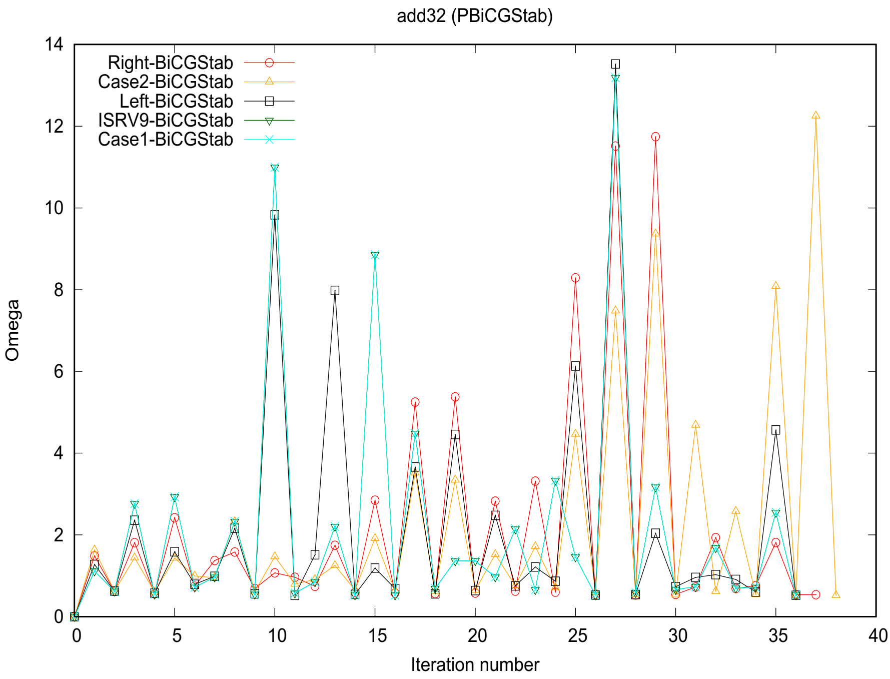

Fig. 7 Values of ω_k for each PBiCGStab

TRE, or too many iterations until convergence. Further, “Max” means reaching the maximum number of iterations with no convergence by stagnation or divergence. “BD” means breakdown at the operation of α_k or β_k. “NaN” means that “not a number” occurs at MR operations by zero division.

For Right, Left, coLeft, Case1, RtL, coLtL and C1tL of PBiCGStab, we present convergence history graphs for two kinds of solving problems with RHSone (Figs. 10, 11, 12, 13, 14, 15).

In Tables 3, 4, the results of Case2 are similar to Right in terms of breakdown and poor accuracy on the TRR and TRE. Here, Case2 is a kind of right-preconditioned system. We did not apply stopping criterion changeover to Case2, because the results using Case2 did not suggest an adequate solution. We applied stopping criterion changeover to coLeft and Case1 of the left-preconditioned system, and they are labeled “coLtL” and “C1tL”, respectively. Because ISRV9 and Case1 are equivalent algorithms, we applied the criterion changeover only to Case1. Further, we applied the changeover to Right just for form, and this was labeled “RtL”.

In Tables 5, 6, RtL is shown to yield improved accuracy for TRR and TRE compared to Right in many cases, but the result of `viscoplastic2` in Table 6

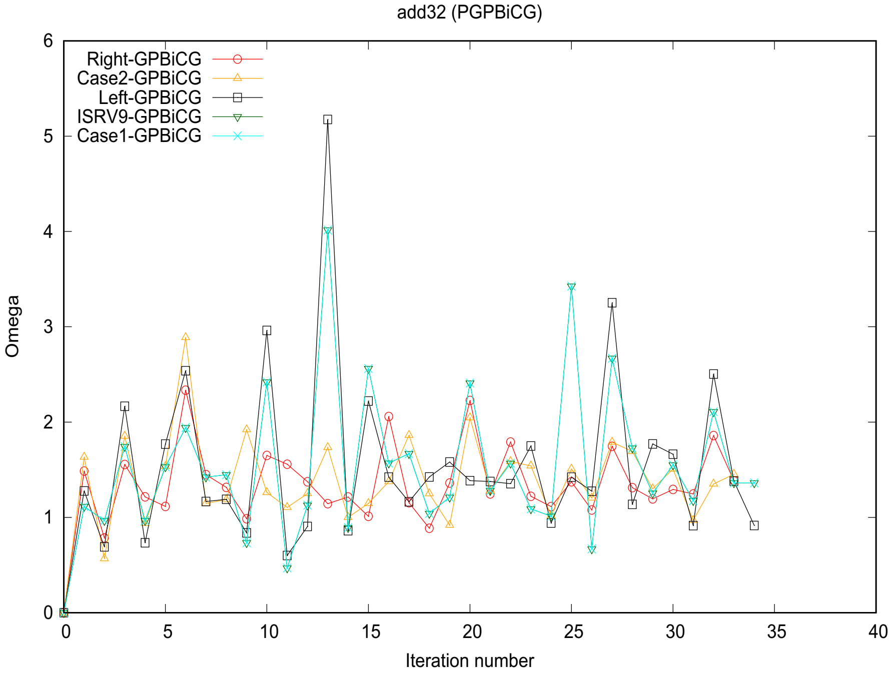

Fig. 8 Values of ω_k for each PGPBiCG

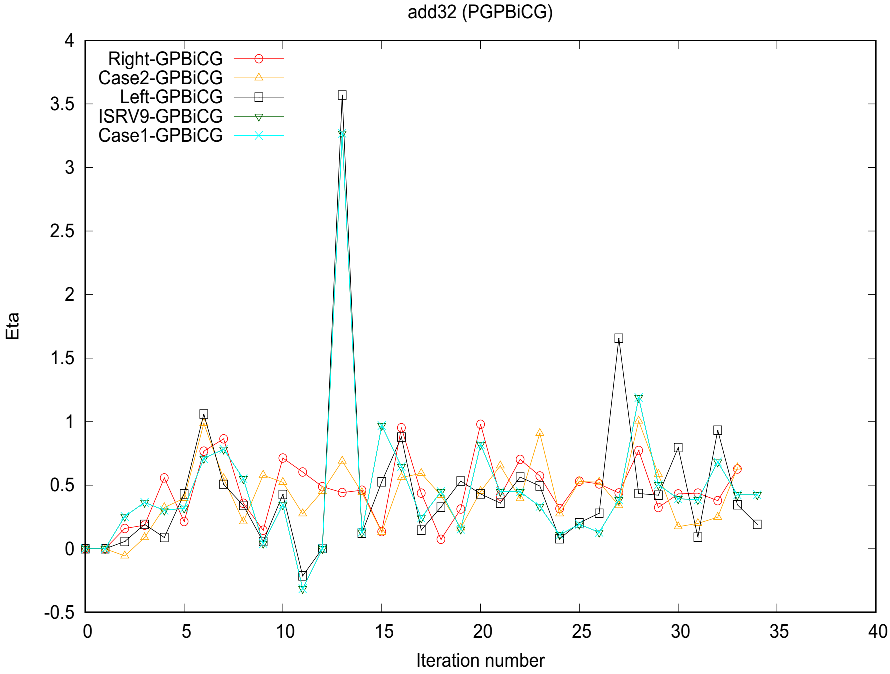

Fig. 9 Values of η_k for each PGPBiCG

Table 3 Numerical results using the relative residual of each algorithm. (PBiCGStab, RHSone); N is the problem size. The three numbers in each row for the column pertaining to each method are as follows: the leftmost number is $\log_{10}$ of the TRR 2-norm, the number in parentheses is the required number of iterations before convergence, and the lower number is $\log_{10}$ of the TRE 2-norm. The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Matrix	N	Right (Alg. 1)	Left (Alg. 2)	coLeft (Alg. 3)	ISRV9 (Alg. 4)	Case1 (Alg. 5)	Case2 (Alg. 6)
arc130	130	– 13.11 (2) **– 3.96**	– 15.76 (5) – 10.52	– 15.19 (5) – 8.83	– 13.22 (2) **– 5.96**	– 13.22 (2) **– 5.96**	– 12.20 (**21**) **– 4.78**
jpwh_991	991	**– 0.58** (2) **– 0.18 BD**	– 11.68 (16) – 12.14	– 13.33 (18) – 13.44	– 13.35 (18) – 13.45	– 13.35 (18) – 13.45	**– 0.53** (2) **– 0.20 BD**
lop163	163	– 10.29 (38) – 8.87	– 8.43 (37) **– 6.82**	– 10.79 (**185**) – 9.41 **NaN**	– 10.74 (32) – 9.19	– 10.82 (37) – 9.67	– 8.96 (**Max**) **– 7.32**
olm2000	2000	**3.26** (**739**) **7.04 BD**	– 13.07 (39) – 9.00	– 12.54 (38) – 8.38	– 12.22 (33) – 8.12	– 12.51 (34) – 8.39	**24.45** (**600**) **28.23 BD**
olm5000	5000	**56.35** (**Max**) **60.75**	– 13.17 (38) – 10.02	– 12.78 (33) **– 7.71**	– 12.11 (31) **– 7.54**	– 12.06 (25) **– 6.91**	**57.31** (**Max**) **61.73**
pores_ 3	532	– 12.25 (24) – 9.27	– 10.04 (26) **– 7.21**	– 12.40 (28) – 9.38	– 12.35 (25) – 9.47	– 12.34 (25) – 9.47	– 12.07 (27) – 9.50
raefsky3	21200	– 12.17 (118) **– 4.50**	– 14.97 (172) – 11.30	– 12.10 (92) **– 5.61**	– 12.13(94) **– 5.72**	– 12.35 (93) **– 5.82**	**– 6.99** (**Max**) **1.31**
viscoplastic1	4326	– 12.43 (682) – 8.88	– 9.94 (518) **– 6.32**	– 12.16 (651) – 8.57	– 12.03 (627) – 8.42	– 12.12 (665) – 8.42	**– 5.26** (**Max**) **– 1.72**
(mx = 2000) viscoplastic2	32769	– 12.07 (966) – 8.64	– 10.06 (761) **– 6.62**	– 12.21 (**1281**) – 8.78	– 12.02 (996) – 8.58	– 12.57 (1027) – 9.14	**– 1.39** (**Max**) **2.05**
(mx = 2500) young3c	841	– 12.07 (**1583**) – 10.49	– 9.16 (1132) – 8.04	– 11.04 (**Max**) – 10.87	– 12.16 (**1409**) – 10.59	– 12.14 (1155) – 10.54	– 8.70 (**Max**) **– 7.03**

Table 4 Numerical results using the relative residual of each algorithm. (PGPBiCG, RHSone)

Matrix	N	Right (Alg. 7)	Left (Alg. 8)	coLeft (Alg. 9)	ISRV9 (Alg. 10)	Case1 (Alg. 11)	Case2 (Alg. 12)
arc130	130	– 12.74 (2) **– 2.78**	**– 5.25** (44) **– 0.28**	**– 5.25** (57) **– 0.28 NaN**	– 13.22 (2) **– 5.96**	– 13.22 (2) **– 5.96**	– 13.51 (2) **– 6.18**
jpwh_991	991	**– 0.58** (1) **– 0.18 BD**	– 12.02 (14) – 12.23	– 12.02 (14) – 12.23	– 12.26 (14) – 12.26	– 12.26 (14) – 12.26	**– 0.53** (1) **– 0.20 BD**
lop163	163	**– 7.97** (31) **– 6.23**	**– 7.42** (22) **– 5.53**	– 10.09 (**110**) – 8.55 **NaN**	– 10.72 (23) – 9.20	– 10.73 (25) – 9.68	– 8.29 (**Max**) **– 7.05**
olm2000	2000	**– 3.01** (**Max**) **0.77**	– 11.29 (56) – 9.33	– 11.29 (**289**) – 9.56 **NaN**	– 11.53 (32) – 8.30	– 12.38 (38) – 8.62	**71.50** (**419**) **75.30 NaN**
olm5000	5000	**– 4.62** (**Max**) **– 0.08**	– 12.63 (39) – 9.71	– 12.43 (36) **– 7.80**	– 12.42 (35) – 9.44	– 12.64 (29) – 8.04	**68.65** (**625**) **73.01 NaN**
pores_3	532	– 12.69 (25) – 9.50	**– 5.51** (25) **– 2.93**	**– 5.51** (**114**) **– 2.93 NaN**	– 12.22 (25) – 9.51	– 12.23 (25) – 9.51	– 12.32 (25) – 9.53
raefsky3	21200	– 12.14 (113) **– 4.49**	– 13.67 (158) – 10.29	– 12.01 (94) **– 5.19**	– 12.06 (95) **– 5.27**	– 12.25 (92) **– 5.49**	– 12.02 (129) **– 5.22**
viscoplastic1	4326	– 12.01 (694) – 8.42	– 9.84 (498) **– 6.21**	– 12.21 (731) – 8.62	– 12.05 (638) – 8.34	– 12.01 (619) – 8.43	**– 6.35** (**Max**) **– 2.79**
viscoplastic2	32769	– 12.19 (814) – 8.11	– 9.29 (664) **– 5.85**	– 12.28 (895) – 8.85	– 12.31 (908) – 8.40	– 12.31 (873) – 8.38	**– 4.43** (**Max**) **– 0.99**
(mx = 5000) young3c	841	– 10.64 (**2928**) – 10.29	**– 6.91** (**2457**) **– 6.73**	**– 6.91** (**Max**) **– 6.74**	– 11.46 (**2832**) – 10.87	– 9.14 (1882) – 9.63	– 8.00 (**Max**) **– 6.33**

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 5 Numerical results using the relative residual of each algorithm with stopping criterion changeover. (PBiCGStab, RHSone); COND is the condition number

Matrix	COND	Right (Alg. 1)	Left (Alg. 2)	RtL (Alg. 1 to L)	coLtL (Alg. 3 to L)	C1tL (Alg. 5 to L)
arc130	1.1e+10	– 13.11 (2) **– 3.96**	– 15.76 (5) – 10.52	– 15.96 (3) – 10.25	– 15.76 (5) – 10.52	– 20.88 (3) – 11.45
jpwh_991	7.3e+02	**– 0.58** (2) **– 0.18 BD**	– 11.68 (16) – 12.14	**– 0.58** (2) **– 0.18 BD**	– 13.33 (18) – 13.44	– 13.35 (18) – 13.45
lop163	2.8e+07	– 10.29 (38) – 8.87	– 8.43 (37) **– 6.82**	– 10.29 (38) – 8.87	– 10.79 (**185**) – 9.41 **NaN**	– 10.82 (37) – 9.67
olm2000	5.9e+06	**3.26** (**739**) **7.04 BD**	– 13.07 (39) – 9.00	**3.26** (**739**) **7.04 BD**	– 13.07 (39) – 9.00	– 13.51 (37) – 11.38
olm5000	3.7e+07	**56.35** (**Max**) **60.75**	– 13.17 (38) – 10.02	**56.35** (**Max**) **60.75**	– 13.17 (38) – 10.02	– 13.18 (30) – 9.36
pores_3	5.6e+05	– 12.25 (24) – 9.27	– 10.04 (26) **– 7.21**	– 13.04 (25) – 10.01	– 12.40 (28) – 9.38	– 12.34 (25) – 9.47
raefsky3	2.0e+11	– 12.17 (118) **– 4.50**	– 14.97 (172) – 11.30	– 15.09 (204) – 11.08	– 14.97 (172) – 11.30	– 15.11 (180) – 10.84
viscoplastic1	1.4e+05	– 12.43 (682) – 8.88	– 9.94 (518) **– 6.32**	– 12.43 (682) – 8.88	– 12.16 (651) – 8.57	– 12.23 (666) – 8.64
(mx = 2000) viscoplastic2	2.3e+05	– 12.07 (966) – 8.64	– 10.06 (761) **– 6.62**	– 12.07 (966) – 8.64	– 12.21 (**1281**) – 8.78	– 12.57 (1027) – 9.14
(mx = 2500) young3c	9.3e+03	– 12.07 (**1583**) – 10.49	– 9.16 (1132) – 8.04	– 12.07 (**1583**) – 10.49	– 11.04 (**Max**) – 10.87	– 12.14 (1155) – 10.54

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 6 Numerical results using the relative residual of each algorithm with stopping criterion changeover. (PGPBiCG, RHSone)

Matrix	COND	Right (Alg. 7)	Left (Alg. 8)	RtL (Alg. 7 to L)	coLtL (Alg. 9 to L)	C1tL (Alg. 11 to L)
arc130	1.1e+10	– 12.74 (2) **– 2.78**	**– 5.25** (**44**) **– 0.28**	– 15.96 (3) – 11.02	**– 5.25** (**57**) **– 0.28 NaN**	– 20.86 (3) – 11.45
jpwh_991	7.3e+02	**– 0.58** (1) **– 0.18 BD**	– 12.02 (14) – 12.23	**– 0.58** (1) **– 0.18 BD**	– 12.02 (14) – 12.23	– 12.26 (14) – 12.26
lop163	2.8e+07	**– 7.97** (31) **– 6.23**	**– 7.42** (22) **– 5.53**	**– 7.97** (31) **– 6.23**	– 10.09 (**110**) – 8.55 **NaN**	– 10.73 (25) – 9.68
olm2000	5.9e+06	**– 3.01** (**Max**) **0.77**	– 11.29 (56) – 9.33	**– 3.01** (**Max**) **0.77**	– 11.29 (**289**) – 9.56 **NaN**	– 12.63 (44) – 10.65
olm5000	3.7e+07	**– 4.62** (**Max**) **– 0.08**	– 12.63 (39) – 9.71	**– 4.62** (**Max**) **– 0.08**	– 12.63 (39) – 9.71	– 12.77 (36) – 9.86
pores_3	5.6e+05	– 12.69 (25) – 9.50	**– 5.51** (25) **– 2.93**	– 12.69 (25) – 9.50	**– 5.51** (**114**) **– 2.93 NaN**	– 12.23 (25) – 9.51
raefsky3	2.0e+11	– 12.14 (113) **– 4.49**	– 13.67 (158) – 10.29	– 15.04 (175) – 10.99	– 13.67 (158) – 10.29	– 15.09 (151) – 11.13
viscoplastic1	1.4e+05	– 12.01 (694) – 8.42	– 9.84 (498) **– 6.21**	– 12.05 (694) – 8.43	– 12.21 (731) – 8.62	– 12.01 (619) – 8.43
viscoplastic2	2.3e+05	– 12.19 (814) – 8.11	– 9.29 (664) **– 5.85**	– 11.54 (816) **– 7.73**	– 12.28 (895) – 8.85	– 11.88 (875) – 8.43
(mx = 5000) young3c	9.3e+03	– 10.64 (**2928**) – 10.29	**– 6.91** (**2457**) **– 6.73**	– 10.64 (**2928**) – 10.29	**– 6.91** (**Max**) **– 6.74**	– 9.14 (1882) – 9.63

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 7 Numerical results using the relative residual of each algorithm with stopping criterion changeover for many RHSs. (olm5000, PBiCGStab, mx = 1000)

RHS ID	Right (Alg. 1)	Left (Alg. 2)	RtL (Alg. 1 to L)	coLtL (Alg. 3 to L)	C1tL (Alg. 5 to L)
RHS + 10	– 10.22 (**Max**) **– 3.41**	– 14.50 (41) – 9.09	– 10.22 (**Max**) **– 3.41**	– 14.50 (41) – 9.09	– 14.50 (39) – 9.09
RHS + 20	– 12.39 (**111**) **– 5.72**	– 14.80 (39) – 9.99	– 14.76 (**170**) – 10.73	– 14.80 (39) – 9.99	– 14.59 (44) – 10.14
RHS + 30	– 8.66 (**Max**) **– 1.84**	– 14.87 (41) – 11.33	– 8.66 (**Max**) **– 1.84**	– 14.87 (41) – 11.33	– 14.97 (39) – 11.11
RHS + 40	– 12.14 (**470**) **– 5.40**	– 14.41 (38) – 9.78	– 14.22 (**691**) – 9.34	– 14.41 (38) – 9.78	– 14.45 (34) – 11.17
RHS + 50	– 12.42 (**199**) **– 6.02**	– 14.40 (41) – 9.45	– 14.41 (**466**) – 10.49	– 14.40 (41) – 9.45	– 14.37 (38) – 10.21

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 8 Numerical results using the relative residual of each algorithm with stopping criterion changeover for many RHSs. (viscoplastic1, PBiCGStab, mx = 1000)

RHS ID	Right (Alg. 1)	Left (Alg. 2)	RtL (Alg. 1 to L)	coLtL (Alg. 3 to L)	C1tL (Alg. 5 to L)
RHS + 10	– 12.56 (702) – 9.25	– 9.72 (533) **– 6.22**	– 10.81 (704) **– 7.23**	– 12.39 (719) – 8.82	– 12.01 (681) – 8.47
RHS + 20	– 12.28 (724) – 8.77	– 9.59 (620) **– 6.14**	– 11.85 (725) – 8.27	– 12.37 (729) – 8.93	– 12.07 (682) – 8.57
RHS + 30	– 12.03 (634) – 8.45	– 10.01 (502) **– 6.48**	– 11.97 (635) – 8.38	– 12.41 (744) – 8.89	– 12.10 (711) – 8.48
RHS + 40	– 12.15 (730) – 8.67	– 9.68 (502) **– 6.16**	– 11.81 (731) – 8.32	– 12.23 (709) – 8.67	– 12.06 (757) – 8.48
RHS + 50	– 12.37 (635) – 8.61	– 9.58 (549) **– 6.02**	– 12.37 (635) – 8.61	– 12.17 (684) – 8.67	– 12.53 (702) – 8.92

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 9 Numerical results using the relative residual of each algorithm with stopping criterion changeover for many RHSs. (young3c, PBiCGStab, mx = 2500)

RHS ID	Right (Alg. 1)	Left (Alg. 2)	RtL (Alg. 1 to L)	coLtL (Alg. 3 to L)	C1tL (Alg. 5 to L)
RHS + 10	– 11.92 (**1630**) – 10.24	– 9.73 (1115) – 8.36	– 11.92 (**1630**) – 10.24	– 11.57 (**Max**) – 10.96	– 12.22 (1101) – 10.23
RHS +20	– 12.08 (**1509**) – 10.34	– 9.42 (1157) – 8.06	– 12.08 (**1509**) – 10.34	– 11.24 (**Max**) – 10.65	– 12.03 (1130) – 10.60
RHS + 30	– 12.00 (**1361**) – 10.18	– 9.58 (1160) **– 7.61**	– 12.00 (**1361**) – 10.18	– 11.49 (**Max**) – 10.88	– 12.32 (1082) – 10.46
RHS + 40	– 12.16 (1349) – 10.36	– 9.67 (1134) – 8.24	– 12.16 (1349) – 10.36	– 11.96 (**Max**) – 11.38	– 12.02 (1132) – 10.25
RHS + 50	– 12.31 (1375) – 10.65	– 9.67 (1168) **– 7.69**	– 12.31 (1375) – 10.65	– 11.85 (**Max**) – 11.26	– 12.02 (1202) – 10.09

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

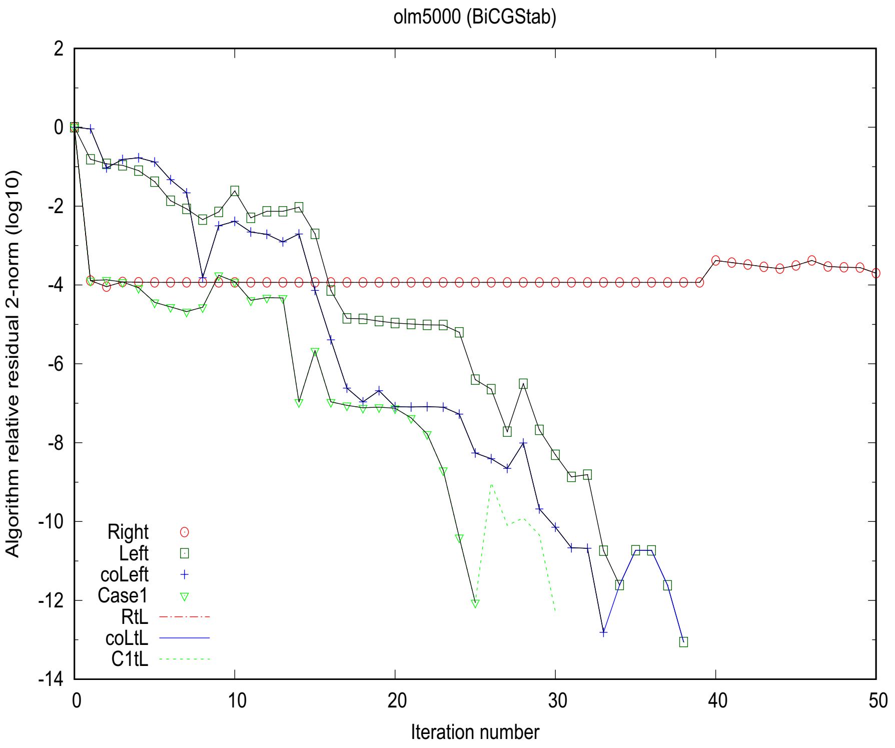

Fig. 10 Convergence histories for the ARR 2-norm (olm5000, RHSone)

suggests decreased accuracy for both. This is confirmed in Table 6 of PBiCGStab. Further, RtL did not lead to improved accuracy in Table 10. It was confirmed that RtL never solved equations that were not solved as BD or Max by Right (Tables 7, 11).

Next, we consider the results pertaining to LTB in Definition 6 and list the problems for which Left was weak in Tables 3, 4. The LTB case is the result of poor accuracy on the TRE, in spite of convergence with a smaller or equal iteration number to coLeft, pores_3, viscoplastic{1, 2} were concerned in Table 3 of PBiCGStab. Here, we extend the comparing target to ISRV9 and Case1, then young3c was also concerned[9]. Further, viscoplastic{1, 2} were also concerned in Table 4 of PGPBiCG. We extend the comparing target to ISRV9 and Case1, then lop163, pores_3 were also concerned, and arc130 exhibited poor accuracy for the results of the six types of algorithms in Table 4. In terms of the other cases,

[9] However, the TRE of young3c by coLeft satisfied 10^{-8}.

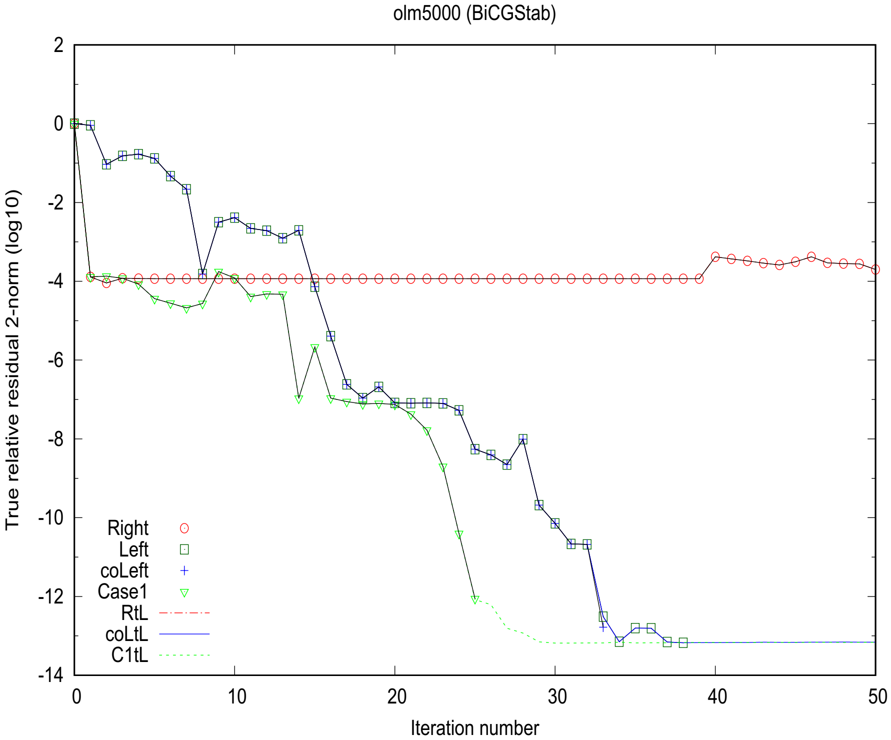

Fig. 11 Convergence histories for the TRR 2-norm (olm5000, RHSone)

results according to Left were poor on the TRR and the TRE in Tables 8, 9, 10, 11, 12.

Next, we consider results pertaining to GTB in Definition 6, and list the solving problems to work out by Left in Tables 3, 4. In Table 3 of PBiCGStab, `arc130`, `olm{2000,5000}`, `raefsky3` were concerned, and Left exhibited superior accuracy compared to the other five types of algorithms[10]. In Table 4 of PGPBiCG, `olm{2000,5000}`, `raefsky3` were concerned, and Left was the most accurate among the six types of algorithms. However, coLtL and C1tL converged roughly with fewer iterations and higher accuracy compared to Left in some cases as per Tables 5, 6.

coLeft and coLtL are mathematically equivalent algorithms to Left, except for the stopping criterion being caused by splitting $\boldsymbol{r}_k^+$ of Left into $M^{-1}\boldsymbol{r}_k$. Comparing the results in Tables 5, 6, the case of the GTB and converging without NaN or Max, Left and coLtL exhibited mathematically identical accuracy; for

[10] The difference on the TRR and the TRE occurred for `arc130` between the Left and coLeft in spite of the fact that the required numbers were both 5, because coLeft converged at the early check.

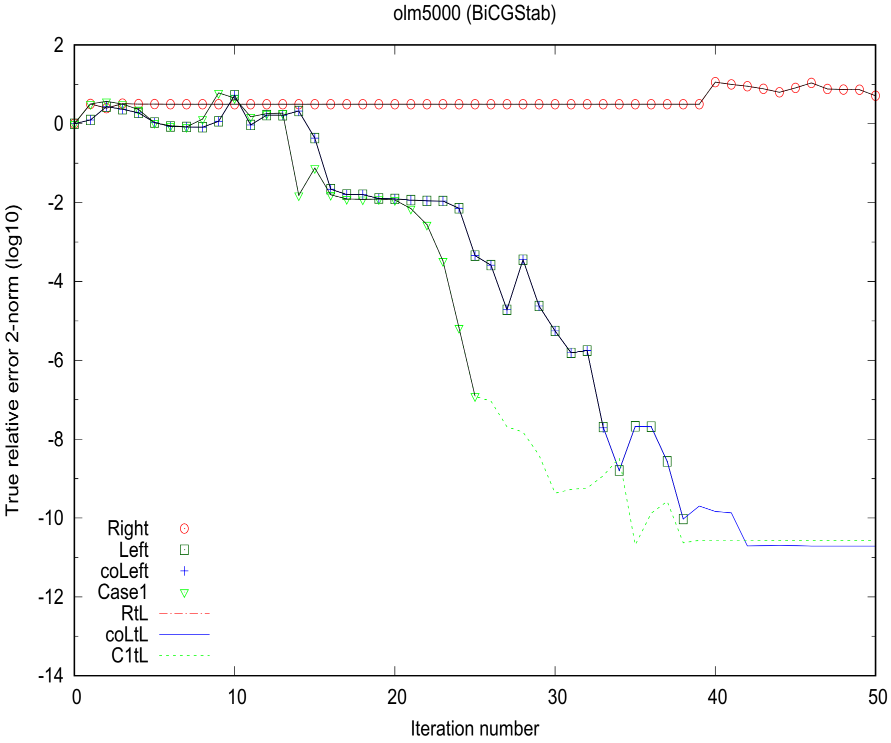

Fig. 12 Convergence histories for the TRE 2-norm (olm5000, RHSone)

example, `arc130`, `olm{2000,5000}`, `raefsky3` in Table 5 (PBiCGStab), `jpwh_991`, `olm5000`, `raefsky3` in Table 6 (PGPBiCG). This was also confirmed by the fact that the values after changing over coLeft to coLtL were the same as per Left in Fig. 10 (`olm5000`). We might roughly rank as “the coLeft ≤ the Left = the coLtL” on accuracy, except for the LTB cases. In the case of LTB, “the Left ≤ the coLeft = the coLtL”, because there were many cases where coLeft achieved higher accuracy.

The Case1 and ISRV9 algorithms of the left-preconditioned system directly minimize the norm of the residual vector which is not operating M^{-1} in the MR part. These algorithms generated stable solutions without incurring breakdown, NaN or Max. However, a major gap between TRR and TRE was recognized at `arc130`, `olm5000`, `raefsky3` in Table 3 (PBiCGStab) and at `arc130`,

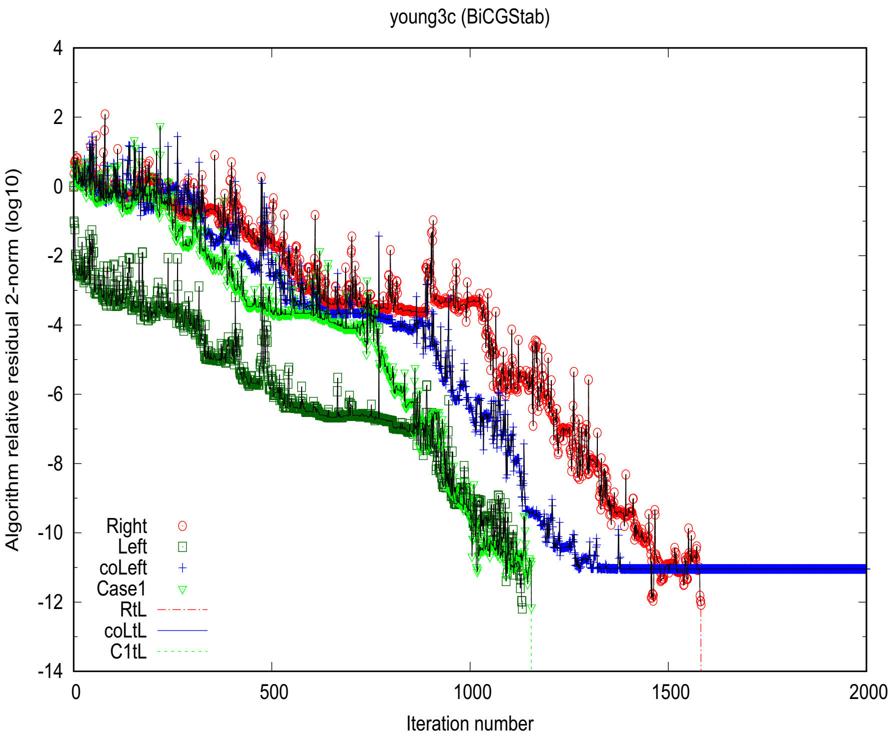

Fig. 13 Convergence histories for the ARR 2-norm (young3c, RHSone)

`raefsky3` in Table 4 (PGPBiCG). This is because Case1 and ISRV9 adopted $\|\boldsymbol{r}_{k+1}\|/\|\boldsymbol{b}\|$ as the stopping criterion, in spite of the relations between the TRE and the ARR of the algorithms of the left-preconditioned system, shown as (2.4) in Proposition 2. The above issue was resolved in the C1tL by the changeover. Further, improved accuracy was recognized. The issue of the major gap in the former case was also recognized in Left, however Left was not able to change over the stopping criterion.

Figures 10, 11, 12, 13, 14, 15 show the behavior of the ARR, TRR and TRE for $\hat{\boldsymbol{x}} = \boldsymbol{x}_{k+1}$. In these figures, iterations of the TRR and the TRE continued irrespective of the second convergence of the ARR after changeover, to observe the converging status of the TRR and the TRE. From these results, we confirmed stagnation of

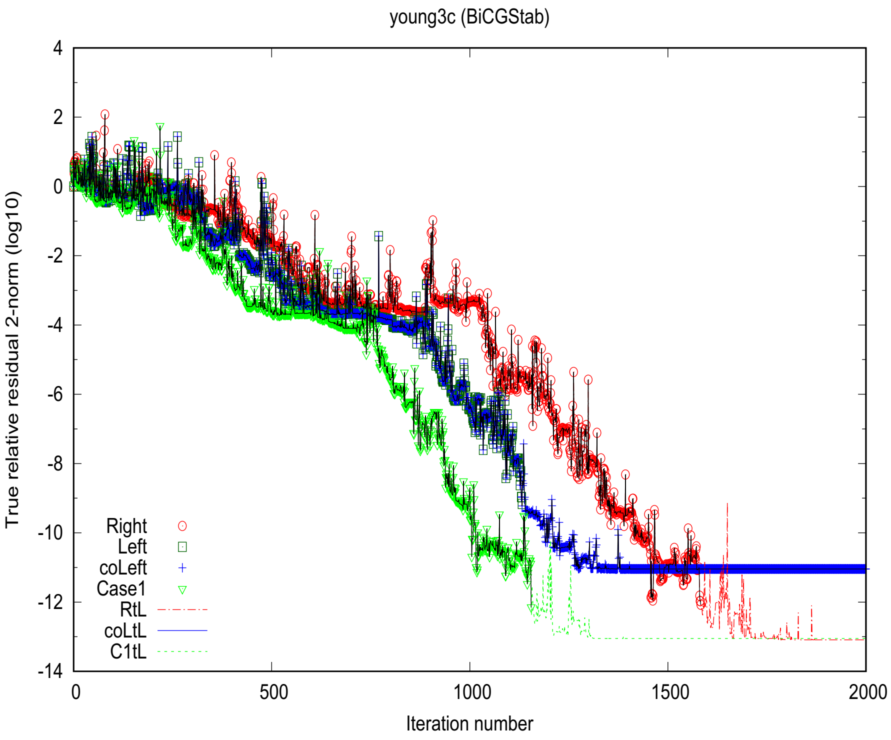

Fig. 14 Convergence histories for the TRR 2-norm (young3c, RHSone)

the TRR and TRE with high accuracy. We may expect high accuracy on the TRE without iteration stopping, but in practical terms, we need a mathematically based stopping criterion to avoid excessive iterative calculations, and the changeover was confirmed as an effective measure in this respect.

Finally, we summarize these numerical results. The changeover was very effective on the RtL, the coLtL and the C1tL. The results for Case1 and ISRV9, the algorithms composed by the BiCG part of the left-preconditioned system and the MR part of the right-preconditioning conversion, were stable in terms of accurately solving linear equations. C1tL further improved the accuracy of the TRE. From the above, C1tL is considered to be the best algorithm. We may think that "ISRV9 to the Left stopping criterion" is also promising because ISRV9 is equivalent to Case1. However, after the changeover, an early check is only available in the case of PBiCGStab, because the iteration part of ISRV9 is identical to Right.

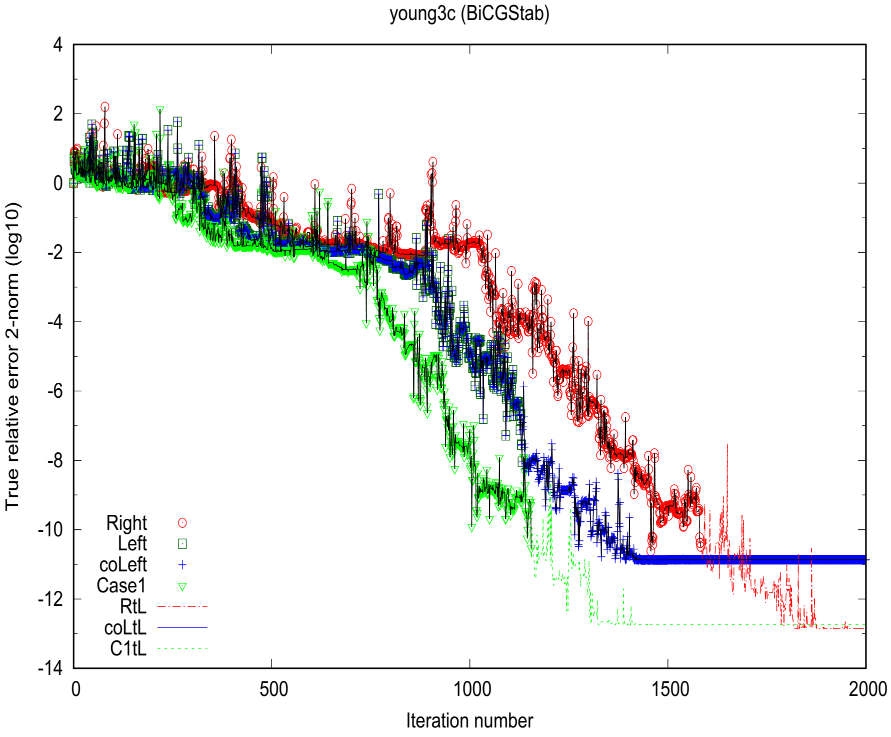

Fig. 15 Convergence histories for the TRE 2-norm (young3c, RHSone)

6 Conclusions

In this paper, we have proposed improved PBiCGStab and PGPBiCG algorithms as preconditioned bi-Lanczos-type algorithms with residual norm minimization. It was confirmed that the direction of the preconditioned system which depends on the BiCG part is determined by the coefficients of α_k and β_k, in each PBiCGStab and PGPBiCG, similar to PBiCG or PCGS [7, 8], in spite of the fact that both methods have MR operators of ω_k and η_k with no congruency in terms of the direction of preconditioning convergence (Theorem 3, Corollary 2). We have numerically confirmed that the direction of the preconditioned system which depends on the BiCG part for both methods is switched by the construction and setting of the ISRV. That is, Theorem 1 for PBiCG and PCGS is extended to PBiCGStab and PGPBiCG

Table 10 Numerical results using the relative residual of each algorithm with stopping criterion changeover for many RHSs. (lop163, PGPBiCG, mx = 1000)

RHS ID	Right	Left	RtL	coLtL	C1tL
	(Alg. 7)	(Alg. 8)	(Alg. 7 to L)	(Alg. 9 to L)	(Alg. 11 to L)
RHS + 10	**– 4.90** (35)	– 8.05 (20)	**– 4.90** (35)	– 10.89 (**99**)	– 11.27 (19)
	– 2.96	**– 5.66**	**– 2.96**	– 9.57 **NaN**	– 10.44
RHS + 20	– 8.47 (27)	– 8.48 (20)	– 8.47 (27)	– 9.56 (**97**)	– 10.47 (18)
	– 7.49	**– 6.04**	**– 7.49**	– 8.22 **NaN**	– 8.58
RHS + 30	**– 6.39** (31)	**– 7.67** (22)	**– 6.39** (31)	– 10.53 (**100**)	– 11.05 (21)
	– 4.40	**– 6.24**	**– 4.40**	– 9.57 **NaN**	– 10.94
RHS + 40	**– 6.33** (31)	– 8.15 (22)	**– 6.33** (31)	– 10.06 (**115**)	– 11.08 (24)
	– 4.60	**– 5.70**	**– 4.60**	– 8.09 **NaN**	– 10.22
RHS + 50	**– 5.42** (31)	**– 7.92** (23)	**– 5.42** (31)	– 10.23 (**109**)	– 10.83 (18)
	– 4.33	**– 5.74**	**– 4.33**	– 8.46 **NaN**	– 8.83

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 11 Numerical results using the relative residual of each algorithm with stopping criterion changeover for many RHSs. (olm5000, PGPBiCG, mx = 1000)

RHS ID	Right (Alg. 7)	Left (Alg. 8)	RtL (Alg. 7 to L)	coLtL (Alg. 9 to L)	C1tL (Alg. 11 to L)
RHS + 10	**11.55** (**Max**) **18.39**	– 13.14 (42) – 9.25	**11.55** (**Max**) **18.39**	– 13.14 (42) – 9.25	– 12.78 (45) – 8.27
RHS + 20	– 12.08 (**216**) **– 5.25**	– 9.94 (45) **– 6.41**	– 13.03 (**537**) – 8.88	– 9.94 (**211**) **– 6.41 NaN**	– 12.96 (40) – 8.67
RHS + 30	**– 7.26** (**Max**) **– 0.60**	– 12.12 (45) – 8.24	**– 7.26** (**Max**) **– 0.60**	– 12.12 (45) – 8.24	– 12.84 (44) – 8.76
RHS + 40	**– 4.82** (**Max**) **1.92**	– 12.65 (41) – 9.07	**– 4.82** (**Max**) **1.92**	– 12.65 (41) – 9.07	– 14.21 (38) – 10.69
RHS + 50	**– 6.90** (**Max**) **– 0.10**	– 10.44 (46) **– 6.19**	**– 6.90** (**Max**) **– 0.10**	– 10.44 (229) **– 6.19 NaN**	– 12.58 (38) – 9.41

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

Table 12 Numerical results using the relative residual of each algorithm with stopping criterion changeover for many RHSs. (young3c, PGPBiCG, mx = 5000)

RHS ID	Right (Alg. 7)	Left (Alg. 8)	RtL (Alg. 7 to L)	coLtL (Alg. 9 to L)	C1tL (Alg. 11 to L)
RHS + 10	– 11.23 (**2939**) – 10.39	– 9.05 (**2590**) **– 7.85**	– 11.23 (**2939**) – 10.39	– 9.05 (**Max**) – 8.44	– 10.65 (1826) – 9.91
RHS + 20	– 9.83 (**3034**) – 8.66	**– 6.38** (2422) **– 5.79**	– 9.83 (**3034**) – 8.66	**– 6.38** (**Max**) **– 5.79**	– 10.89 (2143) – 10.01
RHS + 30	– 10.61 (**3193**) – 9.88	**– 7.12** (**2668**) **– 6.51**	– 10.61 (**3193**) – 9.88	**– 7.12** (**Max**) **– 6.51**	– 11.18 (2119) – 10.65
RHS + 40	– 11.00 (**2533**) – 10.04	**– 7.60** (**2358**) **– 6.97**	– 11.00 (**2533**) – 10.04	**– 7.60** (**Max**) **– 6.97**	– 11.20 (1892) – 10.71
RHS + 50	– 10.59 (**3172**) – 9.49	– 9.37 (**2387**) **– 7.93**	– 10.59 (**3172**) – 9.49	– 9.38 (**Max**) – 8.81	– 10.87 (1712) – 10.05

The data in **bold font** shows a disadvantage, poor accuracy on TRR or TRE (over 10^{-8}), too many iterations required until convergence compared to the others, and BD, NaN, Max

which contain the MR operation (Theorem 4, Corollary 3). Therefore, we may expect that the left-preconditioned system is associated with higher accuracy on the TRE than the right-preconditioned system from Proposition 2 and Reference [9]. Further, we have numerically confirmed that Case1 of the right-preconditioning conversion for the MR operator yields better results than coLeft of the left-preconditioning for the MR.

We have also numerically confirmed that changing over stopping criterion improves the TRE.

Based on the foregoing, we concluded that the C1tL algorithm composed of Case1 with changeover is preferable for both PBiCGStab and PGPBiCG. Some numerical results showed the stability of the C1tL, the accuracy on the TRE and the state of affairs for solving linear equations.

The fundamentals of improvement put forward in this paper could be applied to other preconditioned bi-Lanczos-type methods, for example, BiCGStab(*L*) [15], GBiCGStab(*s*, *L*) [17] and Ritz-IDR [14].

Appendix

This appendix shows various preconditioned algorithms analyzed in the paper.

Various preconditioned BiCGStab algorithms

We describe the six types of preconditioned BiCGStab algorithms.

```
Algorithm 1. Right-PBiCGStab algorithm (Right) :
    $\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta_{-1} = 0$,
    $\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$,
    For $k = 0, 1, 2, 3, \cdots$, Do :
        $\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_{k-1}(\boldsymbol{p}_{k-1} - \omega_{k-1} AM^{-1}\boldsymbol{p}_{k-1})$,
        $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\flat, AM^{-1}\boldsymbol{p}_k \rangle}$,
        $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k AM^{-1}\boldsymbol{p}_k$,
        $\omega_k = \dfrac{(AM^{-1}\boldsymbol{t}_k, \boldsymbol{t}_k)}{(AM^{-1}\boldsymbol{t}_k, AM^{-1}\boldsymbol{t}_k)}$,
        $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k M^{-1}\boldsymbol{p}_k + \omega_k M^{-1}\boldsymbol{t}_k$,
        $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega_k AM^{-1}\boldsymbol{t}_k$,
        $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}$,
    End Do
```

Algorithm 2. Left-PBiCGStab algorithm (Left) :

```
$\boldsymbol{x}_0$ is an initial guess,  $\boldsymbol{r}_0^+ = M^{-1}(\boldsymbol{b} - A\boldsymbol{x}_0)$,   set  $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\sharp = \boldsymbol{r}_0^+$,
For   $k = 0, 1, 2, 3, \cdots$ , Do :
    $\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \omega_{k-1} M^{-1} A \boldsymbol{p}_{k-1}^+)$,
    $\alpha_k = \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1} A \boldsymbol{p}_k^+ \rangle}$,
    $\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha_k M^{-1} A \boldsymbol{p}_k^+$,
    $\omega_k = \frac{(M^{-1} A \boldsymbol{t}_k^+, \boldsymbol{t}_k^+)}{(M^{-1} A \boldsymbol{t}_k^+, M^{-1} A \boldsymbol{t}_k^+)}$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \omega_k \boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1}^+ = \boldsymbol{t}_k^+ - \omega_k M^{-1} A \boldsymbol{t}_k^+$,
    $\beta_k = \frac{\alpha_k}{\omega_k} \times \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}$,
End Do
```

Algorithm 3. Coordinative to left-PBiCGStab algorithm (coLeft) :

```
$\boldsymbol{x}_0$ is an initial guess,  $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,   set  $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$,  $\boldsymbol{r}_0^+ \equiv M^{-1}\boldsymbol{r}_0$,
For   $k = 0, 1, 2, 3, \cdots$ , Do :
    $\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \omega_{k-1} M^{-1} A \boldsymbol{p}_{k-1}^+)$,
    $\alpha_k = \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1} A \boldsymbol{p}_k^+ \rangle}$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A \boldsymbol{p}_k^+$,
    $\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha_k M^{-1} A \boldsymbol{p}_k^+$,
    $\omega_k = \frac{(M^{-1} A \boldsymbol{t}_k^+, \boldsymbol{t}_k^+)}{(M^{-1} A \boldsymbol{t}_k^+, M^{-1} A \boldsymbol{t}_k^+)}$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \omega_k \boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega_k A \boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1}^+ = \boldsymbol{t}_k^+ - \omega_k M^{-1} A \boldsymbol{t}_k^+$,
    $\beta_k = \frac{\alpha_k}{\omega_k} \times \frac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}$,
End Do
```

Algorithm 4. Right-PBiCGStab with ISRV9 (ISRV9) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\flat = M^{-\mathrm{T}} M^{-1} \boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\boldsymbol{p}_k = \boldsymbol{r}_k + \beta_{k-1}(\boldsymbol{p}_{k-1} - \omega_{k-1} A M^{-1} \boldsymbol{p}_{k-1})$,
    $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\flat, A M^{-1} \boldsymbol{p}_k \rangle}$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A M^{-1} \boldsymbol{p}_k$,
    $\omega_k = \dfrac{(A M^{-1} \boldsymbol{t}_k, \boldsymbol{t}_k)}{(A M^{-1} \boldsymbol{t}_k, A M^{-1} \boldsymbol{t}_k)}$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k M^{-1} \boldsymbol{p}_k + \omega_k M^{-1} \boldsymbol{t}_k$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega_k A M^{-1} \boldsymbol{t}_k$,
    $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}$,
End Do
```

Algorithm 5. BiCG-left and MR-right PBiCGStab algorithm (Case1) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\sharp = M^{-1} \boldsymbol{r}_0$, $\boldsymbol{r}_0^+ \equiv M^{-1} \boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \omega_{k-1} M^{-1} A \boldsymbol{p}_{k-1}^+)$,
    $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1} A \boldsymbol{p}_k^+ \rangle}$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A \boldsymbol{p}_k^+$,
    $\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha_k M^{-1} A \boldsymbol{p}_k^+$,
    $\omega_k = \dfrac{(A \boldsymbol{t}_k^+, \boldsymbol{t}_k)}{(A \boldsymbol{t}_k^+, A \boldsymbol{t}_k^+)}$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \omega_k \boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega_k A \boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1}^+ = M^{-1} \boldsymbol{r}_{k+1}$,
    $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}$,
End Do
```

Algorithm 6. BiCG-right and MR-left PBiCGStab algorithm (Case2) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$, $\breve{\boldsymbol{r}}_0 \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\breve{\boldsymbol{p}}_k = \breve{\boldsymbol{r}}_k + \beta_{k-1}(\breve{\boldsymbol{p}}_{k-1} - \omega_{k-1} M^{-1} A \breve{\boldsymbol{p}}_{k-1})$,
    $\alpha_k = \frac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\flat, A\breve{\boldsymbol{p}}_k \rangle}$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A \breve{\boldsymbol{p}}_k$,
    $\breve{\boldsymbol{t}}_k = \breve{\boldsymbol{r}}_k - \alpha_k M^{-1} A \breve{\boldsymbol{p}}_k$,
    $\omega_k = \frac{(M^{-1} A \breve{\boldsymbol{t}}_k, \breve{\boldsymbol{t}}_k)}{(M^{-1} A \breve{\boldsymbol{t}}_k, M^{-1} A \breve{\boldsymbol{t}}_k)}$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \breve{\boldsymbol{p}}_k + \omega_k \breve{\boldsymbol{t}}_k$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \omega_k A \breve{\boldsymbol{t}}_k$,
    $\breve{\boldsymbol{r}}_{k+1} = \breve{\boldsymbol{t}}_k - \omega_k M^{-1} A \breve{\boldsymbol{t}}_k$,
    $\beta_k = \frac{\alpha_k}{\omega_k} \times \frac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}$,
End Do
```

As a reference, the Case1 (Algorithm 5) with the polynomial expression is shown, the parts of the double underline indicate the polynomials.

Algorithm 5'. Polynomial description of the BiCG-left and MR-right PBiCGStab algorithm (Case1) :

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\beta^{\mathrm{L}}_{-1} = 0$,
$\boldsymbol{r}^\sharp_0 = M^{-1}\boldsymbol{r}_0$, $\boldsymbol{r}^+_0 \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :

$$\begin{aligned}
\boldsymbol{p}^+_k &= \boldsymbol{r}^+_k + \beta^{\mathrm{L}}_{k-1}(\boldsymbol{p}^+_{k-1} - \omega^r_{k-1}M^{-1}A\boldsymbol{p}^+_{k-1}) = \underline{\underline{S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\alpha^{\mathrm{L}}_k &= \frac{\langle \boldsymbol{r}^\sharp_0, \boldsymbol{r}^+_k\rangle}{\langle \boldsymbol{r}^\sharp_0, M^{-1}A\boldsymbol{p}^+_k\rangle} = \underline{\underline{\frac{\langle M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}{\langle M^{-1}\boldsymbol{r}_0, \tilde{A}_l S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}}},\\
\boldsymbol{t}_k &= \boldsymbol{r}_k - \alpha^{\mathrm{L}}_k A\boldsymbol{p}^+_k = \underline{\underline{MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{t}^+_k &= \boldsymbol{r}^+_k - \alpha^{\mathrm{L}}_k M^{-1}A\boldsymbol{p}^+_k = \underline{\underline{S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\omega^r_k &= \frac{(AM^{-1}\boldsymbol{t}_k, \boldsymbol{t}_k)}{(AM^{-1}\boldsymbol{t}_k, AM^{-1}\boldsymbol{t}_k)} \equiv \frac{(A\boldsymbol{t}^+_k, \boldsymbol{t}_k)}{(A\boldsymbol{t}^+_k, A\boldsymbol{t}^+_k)}\\
&= \underline{\underline{\frac{(\tilde{A}_r MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0)}{(\tilde{A}_r MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0, \tilde{A}_r MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0)}}},\\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha^{\mathrm{L}}_k\boldsymbol{p}^+_k + \omega^r_k\boldsymbol{t}^+_k\\
&= \boldsymbol{x}_k + \alpha^{\mathrm{L}}_k\underline{\underline{S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}} + \omega^r_k\underline{\underline{S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{r}_{k+1} &= \boldsymbol{t}_k - \omega^r_k A\boldsymbol{t}^+_k = \underline{\underline{MS_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{r}^+_{k+1} &= M^{-1}\boldsymbol{r}_{k+1} = \underline{\underline{M^{-1}\left(MS_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\right)}},\\
\beta^{\mathrm{L}}_k &= \frac{\alpha^{\mathrm{L}}_k}{\omega^r_k} \times \frac{\langle \boldsymbol{r}^\sharp_0, \boldsymbol{r}^+_{k+1}\rangle}{\langle \boldsymbol{r}^\sharp_0, \boldsymbol{r}^+_k\rangle} = \frac{\alpha^{\mathrm{L}}_k}{\omega^r_k} \times \underline{\underline{\frac{\langle M^{-1}\boldsymbol{r}_0, S_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}{\langle M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0\rangle}}},
\end{aligned}$$

End Do

Various preconditioned GPBiCG algorithms

We describe the six types of preconditioned GPBiCG algorithms.

```
**Algorithm 7. Right-PGPBiCG algorithm (Right) :**
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\boldsymbol{t}_{-1} = \boldsymbol{w}_{-1} = 0$, $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$, $\breve{\boldsymbol{r}}_0 \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\breve{\boldsymbol{p}}_k = \breve{\boldsymbol{r}}_k + \beta_{k-1}(\breve{\boldsymbol{p}}_{k-1} - \breve{\boldsymbol{u}}_{k-1})$,
    $\alpha_k = \frac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\flat, A\breve{\boldsymbol{p}}_k \rangle}$,
    $\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k \boldsymbol{w}_{k-1} + \alpha_k A\breve{\boldsymbol{p}}_k$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\breve{\boldsymbol{p}}_k$,
    $\breve{\boldsymbol{t}}_k = \breve{\boldsymbol{r}}_k - \alpha_k M^{-1}A\breve{\boldsymbol{p}}_k$,
    $\omega_k = \frac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(A\breve{\boldsymbol{t}}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(A\breve{\boldsymbol{t}}_k, \boldsymbol{y}_k)}{(A\breve{\boldsymbol{t}}_k, A\breve{\boldsymbol{t}}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\breve{\boldsymbol{t}}_k)(A\breve{\boldsymbol{t}}_k, \boldsymbol{y}_k)}$,
    $\eta_k = \frac{(A\breve{\boldsymbol{t}}_k, A\breve{\boldsymbol{t}}_k)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, A\breve{\boldsymbol{t}}_k)(A\breve{\boldsymbol{t}}_k, \boldsymbol{t}_k)}{(A\breve{\boldsymbol{t}}_k, A\breve{\boldsymbol{t}}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\breve{\boldsymbol{t}}_k)(A\breve{\boldsymbol{t}}_k, \boldsymbol{y}_k)}$,
        ( if $k = 0$, then $\omega_0 = \frac{(A\breve{\boldsymbol{t}}_0, \boldsymbol{t}_0)}{(A\breve{\boldsymbol{t}}_0, A\breve{\boldsymbol{t}}_0)}$, $\eta_0 = 0$ )
    $\breve{\boldsymbol{u}}_k = \omega_k M^{-1}A\breve{\boldsymbol{p}}_k + \eta_k(\breve{\boldsymbol{t}}_{k-1} - \breve{\boldsymbol{r}}_k + \beta_{k-1}\breve{\boldsymbol{u}}_{k-1})$,
    $\breve{\boldsymbol{z}}_k = \omega_k \breve{\boldsymbol{r}}_k + \eta_k \breve{\boldsymbol{z}}_{k-1} - \alpha_k \breve{\boldsymbol{u}}_k$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \breve{\boldsymbol{p}}_k + \breve{\boldsymbol{z}}_k$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k \boldsymbol{y}_k - \omega_k A\breve{\boldsymbol{t}}_k$,
    $\breve{\boldsymbol{r}}_{k+1} = M^{-1}\boldsymbol{r}_{k+1}$,
    $\beta_k = \frac{\alpha_k}{\omega_k} \times \frac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}$,
    $\boldsymbol{w}_k = A\breve{\boldsymbol{t}}_k + \beta_k A\breve{\boldsymbol{p}}_k$,
End Do
```

Algorithm 8. Left-PGPBiCG algorithm (Left) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0^+ = M^{-1}(\boldsymbol{b} - A\boldsymbol{x}_0)$, set $\boldsymbol{t}_{-1}^+ = \boldsymbol{w}_{-1}^+ = 0,\ \beta_{-1} = 0,$
$\boldsymbol{r}_0^\sharp = \boldsymbol{r}_0^+,$
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \boldsymbol{u}_{k-1}^+),$
    $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1}A\boldsymbol{p}_k^+ \rangle},$
    $\boldsymbol{y}_k^+ = \boldsymbol{t}_{k-1}^+ - \boldsymbol{r}_k^+ - \alpha_k \boldsymbol{w}_{k-1}^+ + \alpha_k M^{-1}A\boldsymbol{p}_k^+,$
    $\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha_k M^{-1}A\boldsymbol{p}_k^+,$
    $\omega_k = \dfrac{(\boldsymbol{y}_k^+, \boldsymbol{y}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{t}_k^+) - (\boldsymbol{y}_k^+, \boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{y}_k^+)}{(M^{-1}A\boldsymbol{t}_k^+, M^{-1}A\boldsymbol{t}_k^+)(\boldsymbol{y}_k^+, \boldsymbol{y}_k^+) - (\boldsymbol{y}_k^+, M^{-1}A\boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{y}_k^+)},$
    $\eta_k = \dfrac{(M^{-1}A\boldsymbol{t}_k^+, M^{-1}A\boldsymbol{t}_k^+)(\boldsymbol{y}_k^+, \boldsymbol{t}_k^+) - (\boldsymbol{y}_k^+, M^{-1}A\boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{t}_k^+)}{(M^{-1}A\boldsymbol{t}_k^+, M^{-1}A\boldsymbol{t}_k^+)(\boldsymbol{y}_k^+, \boldsymbol{y}_k^+) - (\boldsymbol{y}_k^+, M^{-1}A\boldsymbol{t}_k^+)(M^{-1}A\boldsymbol{t}_k^+, \boldsymbol{y}_k^+)},$
        ( if $k = 0$, then $\omega_0 = \dfrac{(M^{-1}A\boldsymbol{t}_0^+, \boldsymbol{t}_0^+)}{(M^{-1}A\boldsymbol{t}_0^+, M^{-1}A\boldsymbol{t}_0^+)},\ \eta_0 = 0$ )
    $\boldsymbol{u}_k^+ = \omega_k M^{-1}A\boldsymbol{p}_k^+ + \eta_k(\boldsymbol{t}_{k-1}^+ - \boldsymbol{r}_k^+ + \beta_{k-1}\boldsymbol{u}_{k-1}^+),$
    $\boldsymbol{z}_k^+ = \omega_k \boldsymbol{r}_k^+ + \eta_k \boldsymbol{z}_{k-1}^+ - \alpha_k \boldsymbol{u}_k^+,$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \boldsymbol{z}_k^+,$
    $\boldsymbol{r}_{k+1}^+ = \boldsymbol{t}_k^+ - \eta_k \boldsymbol{y}_k^+ - \omega_k M^{-1}A\boldsymbol{t}_k^+,$
    $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle},$
    $\boldsymbol{w}_k^+ = M^{-1}A\boldsymbol{t}_k^+ + \beta_k M^{-1}A\boldsymbol{p}_k^+,$
End Do
```

Algorithm 9. Coordinative to left-PGPBiCG algorithm (coLeft) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,   set $\boldsymbol{t}_{-1}^{+} = \boldsymbol{w}_{-1}^{+} = 0,\ \beta_{-1} = 0,$
$\boldsymbol{r}_0^{\sharp} = M^{-1}\boldsymbol{r}_0,\ \boldsymbol{r}_0^{+} \equiv M^{-1}\boldsymbol{r}_0,$
For   $k = 0, 1, 2, 3, \cdots$ , Do :
        $\boldsymbol{p}_k^{+} = \boldsymbol{r}_k^{+} + \beta_{k-1}(\boldsymbol{p}_{k-1}^{+} - \boldsymbol{u}_{k-1}^{+}),$
        $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^{\sharp}, \boldsymbol{r}_k^{+} \rangle}{\langle \boldsymbol{r}_0^{\sharp}, M^{-1}A\boldsymbol{p}_k^{+} \rangle},$
        $\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k \boldsymbol{w}_{k-1} + \alpha_k A\boldsymbol{p}_k^{+},$
        $\boldsymbol{y}_k^{+} = \boldsymbol{t}_{k-1}^{+} - \boldsymbol{r}_k^{+} - \alpha_k \boldsymbol{w}_{k-1}^{+} + \alpha_k M^{-1}A\boldsymbol{p}_k^{+},$
        $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k^{+},$
        $\boldsymbol{t}_k^{+} = \boldsymbol{r}_k^{+} - \alpha_k M^{-1}A\boldsymbol{p}_k^{+},$
        $\omega_k = \dfrac{(\boldsymbol{y}_k^{+}, \boldsymbol{y}_k^{+})(M^{-1}A\boldsymbol{t}_k^{+}, \boldsymbol{t}_k^{+}) - (\boldsymbol{y}_k^{+}, \boldsymbol{t}_k^{+})(M^{-1}A\boldsymbol{t}_k^{+}, \boldsymbol{y}_k^{+})}{(M^{-1}A\boldsymbol{t}_k^{+}, M^{-1}A\boldsymbol{t}_k^{+})(\boldsymbol{y}_k^{+}, \boldsymbol{y}_k^{+}) - (\boldsymbol{y}_k^{+}, M^{-1}A\boldsymbol{t}_k^{+})(M^{-1}A\boldsymbol{t}_k^{+}, \boldsymbol{y}_k^{+})},$
        $\eta_k = \dfrac{(M^{-1}A\boldsymbol{t}_k^{+}, M^{-1}A\boldsymbol{t}_k^{+})(\boldsymbol{y}_k^{+}, \boldsymbol{t}_k^{+}) - (\boldsymbol{y}_k^{+}, M^{-1}A\boldsymbol{t}_k^{+})(M^{-1}A\boldsymbol{t}_k^{+}, \boldsymbol{t}_k^{+})}{(M^{-1}A\boldsymbol{t}_k^{+}, M^{-1}A\boldsymbol{t}_k^{+})(\boldsymbol{y}_k^{+}, \boldsymbol{y}_k^{+}) - (\boldsymbol{y}_k^{+}, M^{-1}A\boldsymbol{t}_k^{+})(M^{-1}A\boldsymbol{t}_k^{+}, \boldsymbol{y}_k^{+})},$
              $(\text{ if } k = 0, \text{ then } \omega_0 = \dfrac{(M^{-1}A\boldsymbol{t}_0^{+}, \boldsymbol{t}_0^{+})}{(M^{-1}A\boldsymbol{t}_0^{+}, M^{-1}A\boldsymbol{t}_0^{+})},\ \eta_0 = 0\ )$
        $\boldsymbol{u}_k^{+} = \omega_k M^{-1}A\boldsymbol{p}_k^{+} + \eta_k(\boldsymbol{t}_{k-1}^{+} - \boldsymbol{r}_k^{+} + \beta_{k-1}\boldsymbol{u}_{k-1}^{+}),$
        $\boldsymbol{z}_k^{+} = \omega_k \boldsymbol{r}_k^{+} + \eta_k \boldsymbol{z}_{k-1}^{+} - \alpha_k \boldsymbol{u}_k^{+},$
        $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^{+} + \boldsymbol{z}_k^{+},$
        $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k \boldsymbol{y}_k - \omega_k A\boldsymbol{t}_k^{+},$
        $\boldsymbol{r}_{k+1}^{+} = \boldsymbol{t}_k^{+} - \eta_k \boldsymbol{y}_k^{+} - \omega_k M^{-1}A\boldsymbol{t}_k^{+},$
        $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^{\sharp}, \boldsymbol{r}_{k+1}^{+} \rangle}{\langle \boldsymbol{r}_0^{\sharp}, \boldsymbol{r}_k^{+} \rangle},$
        $\boldsymbol{w}_k = A\boldsymbol{t}_k^{+} + \beta_k A\boldsymbol{p}_k^{+},$
        $\boldsymbol{w}_k^{+} = M^{-1}A\boldsymbol{t}_k^{+} + \beta_k M^{-1}A\boldsymbol{p}_k^{+},$
End Do
```

Algorithm 10. Right-PGPBiCG with ISRV9 (ISRV9) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\boldsymbol{t}_{-1} = \boldsymbol{w}_{-1} = 0$, $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\flat = M^{-\mathrm{T}} M^{-1} \boldsymbol{r}_0$, $\boldsymbol{r}_0^+ \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \boldsymbol{u}_{k-1}^+)$,
    $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\flat, A\boldsymbol{p}_k^+ \rangle}$,
    $\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k \boldsymbol{w}_{k-1} + \alpha_k A\boldsymbol{p}_k^+$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k^+$,
    $\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha_k M^{-1} A\boldsymbol{p}_k^+$,
    $\omega_k = \dfrac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(A\boldsymbol{t}_k^+, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t} +_k)(\boldsymbol{y}_k^+, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}$,
    $\eta_k = \dfrac{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{t}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}$,
        ( if $k = 0$, then $\omega_0 = \dfrac{(A\boldsymbol{t}_0^+, \boldsymbol{t}_0)}{(A\boldsymbol{t}_0^+, A\boldsymbol{t}_0^+)}$, $\eta_0 = 0$ )
    $\boldsymbol{u}_k^+ = \omega_k M^{-1} A\boldsymbol{p}_k^+ + \eta_k(\boldsymbol{t}_{k-1}^+ - \boldsymbol{r}_k^+ + \beta_{k-1}\boldsymbol{u}_{k-1}^+)$,
    $\boldsymbol{z}_k^+ = \omega_k \boldsymbol{r}_k^+ + \eta_k \boldsymbol{z}_{k-1}^+ - \alpha_k \boldsymbol{u}_k^+$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \boldsymbol{z}_k^+$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k \boldsymbol{y}_k - \omega_k A\boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1}^+ = M^{-1}\boldsymbol{r}_{k+1}$,
    $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}$,
    $\boldsymbol{w}_k = A\boldsymbol{t}_k^+ + \beta_k A\boldsymbol{p}_k^+$,
End Do
```

Algorithm 11. BiCG-left and MR-right PGPBiCG algorithm (Case1) :

```
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$,   set $\boldsymbol{t}_{-1} = \boldsymbol{w}_{-1} = 0$, $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\sharp = M^{-1}\boldsymbol{r}_0$, $\boldsymbol{r}_0^+ \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
    $\boldsymbol{p}_k^+ = \boldsymbol{r}_k^+ + \beta_{k-1}(\boldsymbol{p}_{k-1}^+ - \boldsymbol{u}_{k-1}^+)$,
    $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, M^{-1}A\boldsymbol{p}_k^+ \rangle}$,
    $\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k \boldsymbol{w}_{k-1} + \alpha_k A\boldsymbol{p}_k^+$,
    $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k^+$,
    $\boldsymbol{t}_k^+ = \boldsymbol{r}_k^+ - \alpha_k M^{-1}A\boldsymbol{p}_k^+$,
    $\omega_k = \dfrac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(A\boldsymbol{t}_k^+, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}$,
    $\eta_k = \dfrac{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{t}_k)}{(A\boldsymbol{t}_k^+, A\boldsymbol{t}_k^+)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, A\boldsymbol{t}_k^+)(A\boldsymbol{t}_k^+, \boldsymbol{y}_k)}$,
        ( if $k = 0$, then $\omega_0 = \dfrac{(A\boldsymbol{t}_0^+, \boldsymbol{t}_0)}{(A\boldsymbol{t}_0^+, A\boldsymbol{t}_0^+)}$, $\eta_0 = 0$ )
    $\boldsymbol{u}_k^+ = \omega_k M^{-1}A\boldsymbol{p}_k^+ + \eta_k(\boldsymbol{t}_{k-1}^+ - \boldsymbol{r}_k^+ + \beta_{k-1}\boldsymbol{u}_{k-1}^+)$,
    $\boldsymbol{z}_k^+ = \omega_k \boldsymbol{r}_k^+ + \eta_k \boldsymbol{z}_{k-1}^+ - \alpha_k \boldsymbol{u}_k^+$,
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k^+ + \boldsymbol{z}_k^+$,
    $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k \boldsymbol{y}_k - \omega_k A\boldsymbol{t}_k^+$,
    $\boldsymbol{r}_{k+1}^+ = M^{-1}\boldsymbol{r}_{k+1}$,
    $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_{k+1}^+ \rangle}{\langle \boldsymbol{r}_0^\sharp, \boldsymbol{r}_k^+ \rangle}$,
    $\boldsymbol{w}_k = A\boldsymbol{t}_k^+ + \beta_k A\boldsymbol{p}_k^+$,
End Do
```

```
**Algorithm 12. BiCG-right and MR-left PGPBiCG algorithm (Case2) :**
$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\boldsymbol{t}_{-1} = \boldsymbol{w}_{-1} = \breve{\boldsymbol{t}}_{-1} = \breve{\boldsymbol{w}}_{-1} = 0$, $\beta_{-1} = 0$,
$\boldsymbol{r}_0^\flat = \boldsymbol{r}_0$, $\breve{\boldsymbol{r}}_0 \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :
        $\breve{\boldsymbol{p}}_k = \breve{\boldsymbol{r}}_k + \beta_{k-1}(\breve{\boldsymbol{p}}_{k-1} - \breve{\boldsymbol{u}}_{k-1})$,
        $\alpha_k = \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}{\langle \boldsymbol{r}_0^\flat, A\breve{\boldsymbol{p}}_k \rangle}$,
        $\boldsymbol{y}_k = \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha_k \boldsymbol{w}_{k-1} + \alpha_k A\breve{\boldsymbol{p}}_k$,
        $\breve{\boldsymbol{y}}_k = \breve{\boldsymbol{t}}_{k-1} - \breve{\boldsymbol{r}}_k - \alpha_k \breve{\boldsymbol{w}}_{k-1} + \alpha_k M^{-1}A\breve{\boldsymbol{p}}_k$,
        $\boldsymbol{t}_k = \boldsymbol{r}_k - \alpha_k A\breve{\boldsymbol{p}}_k$,
        $\breve{\boldsymbol{t}}_k = \breve{\boldsymbol{r}}_k - \alpha_k M^{-1}A\breve{\boldsymbol{p}}_k$,
        $\omega_k = \dfrac{(\breve{\boldsymbol{y}}_k, \breve{\boldsymbol{y}}_k)(M^{-1}A\breve{\boldsymbol{t}}_k, \breve{\boldsymbol{t}}_k) - (\breve{\boldsymbol{y}}_k, \breve{\boldsymbol{t}}_k)(M^{-1}A\breve{\boldsymbol{t}}_k, \breve{\boldsymbol{y}}_k)}{(M^{-1}A\breve{\boldsymbol{t}}_k, M^{-1}A\breve{\boldsymbol{t}}_k)(\breve{\boldsymbol{y}}_k, \breve{\boldsymbol{y}}_k) - (\breve{\boldsymbol{y}}_k, M^{-1}A\breve{\boldsymbol{t}}_k)(M^{-1}A\breve{\boldsymbol{t}}_k, \breve{\boldsymbol{y}}_k)}$,
        $\eta_k = \dfrac{(M^{-1}A\breve{\boldsymbol{t}}_k, M^{-1}A\breve{\boldsymbol{t}}_k)(\breve{\boldsymbol{y}}_k, \breve{\boldsymbol{t}}_k) - (\breve{\boldsymbol{y}}_k, M^{-1}A\breve{\boldsymbol{t}}_k)(M^{-1}A\breve{\boldsymbol{t}}_k, \breve{\boldsymbol{t}}_k)}{(M^{-1}A\breve{\boldsymbol{t}}_k, M^{-1}A\breve{\boldsymbol{t}}_k)(\breve{\boldsymbol{y}}_k, \breve{\boldsymbol{y}}_k) - (\breve{\boldsymbol{y}}_k, M^{-1}A\breve{\boldsymbol{t}}_k)(M^{-1}A\breve{\boldsymbol{t}}_k, \breve{\boldsymbol{y}}_k)}$,
            ( if $k = 0$, then $\omega_0 = \dfrac{(M^{-1}A\breve{\boldsymbol{t}}_0, \breve{\boldsymbol{t}}_0)}{(M^{-1}A\breve{\boldsymbol{t}}_0, M^{-1}A\breve{\boldsymbol{t}}_0)}$, $\eta_0 = 0$ )
        $\breve{\boldsymbol{u}}_k = \omega_k M^{-1}A\breve{\boldsymbol{p}}_k + \eta_k(\breve{\boldsymbol{t}}_{k-1} - \breve{\boldsymbol{r}}_k + \beta_{k-1}\breve{\boldsymbol{u}}_{k-1})$,
        $\breve{\boldsymbol{z}}_k = \omega_k \breve{\boldsymbol{r}}_k + \eta_k \breve{\boldsymbol{z}}_{k-1} - \alpha_k \breve{\boldsymbol{u}}_k$,
        $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \breve{\boldsymbol{p}}_k + \breve{\boldsymbol{z}}_k$,
        $\boldsymbol{r}_{k+1} = \boldsymbol{t}_k - \eta_k \boldsymbol{y}_k - \omega_k A\breve{\boldsymbol{t}}_k$,
        $\breve{\boldsymbol{r}}_{k+1} = \breve{\boldsymbol{t}}_k - \eta_k \breve{\boldsymbol{y}}_k - \omega_k M^{-1}A\breve{\boldsymbol{t}}_k$,
        $\beta_k = \dfrac{\alpha_k}{\omega_k} \times \dfrac{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_{k+1} \rangle}{\langle \boldsymbol{r}_0^\flat, \boldsymbol{r}_k \rangle}$,
        $\boldsymbol{w}_k = A\breve{\boldsymbol{t}}_k + \beta_k A\breve{\boldsymbol{p}}_k$,
        $\breve{\boldsymbol{w}}_k = M^{-1}A\breve{\boldsymbol{t}}_k + \beta_k M^{-1}A\breve{\boldsymbol{p}}_k$,
End Do
```

As a reference, the Case1 (Algorithm 11) with the polynomial expression is shown, the parts of the double underline indicate the polynomials.

Algorithm 11'. Polynomial description of the BiCG-left and MR-right PGP-BiCG algorithm (Case1) :

$\boldsymbol{x}_0$ is an initial guess, $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$, set $\boldsymbol{t}_{-1} = \boldsymbol{w}_{-1} = 0$, $\beta^{\mathrm{L}}_{-1} = 0$,
$\boldsymbol{r}^{\sharp}_0 = M^{-1}\boldsymbol{r}_0$, $\boldsymbol{r}^{+}_0 \equiv M^{-1}\boldsymbol{r}_0$,
For $k = 0, 1, 2, 3, \cdots$, Do :

$$
\begin{aligned}
\boldsymbol{p}^{+}_k &= \boldsymbol{r}^{+}_k + \beta^{\mathrm{L}}_{k-1}(\boldsymbol{p}^{+}_{k-1} - \boldsymbol{u}^{+}_{k-1}) = \underline{\underline{S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\alpha^{\mathrm{L}}_k &= \frac{\langle \boldsymbol{r}^{\sharp}_0, \boldsymbol{r}^{+}_k \rangle}{\langle \boldsymbol{r}^{\sharp}_0, M^{-1}A\boldsymbol{p}^{+}_k \rangle} = \frac{\langle M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0 \rangle}{\langle M^{-1}\boldsymbol{r}_0, \tilde{A}_l S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0 \rangle},\\
\boldsymbol{y}_k &= \boldsymbol{t}_{k-1} - \boldsymbol{r}_k - \alpha^{\mathrm{L}}_k \boldsymbol{w}_{k-1} + \alpha^{\mathrm{L}}_k A\boldsymbol{p}^{+}_k = \underline{\underline{M\tilde{A}_l Q_{k-1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{t}_k &= \boldsymbol{r}_k - \alpha^{\mathrm{L}}_k A\boldsymbol{p}^{+}_k = \underline{\underline{MS_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{t}^{+}_k &= \boldsymbol{r}^{+}_k - \alpha^{\mathrm{L}}_k M^{-1}A\boldsymbol{p}^{+}_k = \underline{\underline{S_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\omega^{r}_k &= \frac{(\boldsymbol{y}_k, \boldsymbol{y}_k)(\tilde{A}_r\boldsymbol{t}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \boldsymbol{t}_k)(\tilde{A}_r\boldsymbol{t}_k, \boldsymbol{y}_k)}{(\tilde{A}_r\boldsymbol{t}_k, \tilde{A}_r\boldsymbol{t}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, \tilde{A}_r\boldsymbol{t}_k)(\tilde{A}_r\boldsymbol{t}_k, \boldsymbol{y}_k)}\\
&\quad \rightarrow \text{substitute each vector}'\text{s polynomial},\\
\eta^{r}_k &= \frac{(\tilde{A}_r\boldsymbol{t}_k, \tilde{A}_r\boldsymbol{t}_k)(\boldsymbol{y}_k, \boldsymbol{t}_k) - (\boldsymbol{y}_k, \tilde{A}_l\boldsymbol{t}_k)(\tilde{A}_r\boldsymbol{t}_k, \boldsymbol{t}_k)}{(\tilde{A}_r\boldsymbol{t}_k, \tilde{A}_r\boldsymbol{t}_k)(\boldsymbol{y}_k, \boldsymbol{y}_k) - (\boldsymbol{y}_k, \tilde{A}_r\boldsymbol{t}_k)(\tilde{A}_r\boldsymbol{t}_k, \boldsymbol{y}_k)},\\
&\quad \rightarrow \text{substitute each vector}'\text{s polynomial},\\
&\quad \left(\text{ if } k = 0, \text{ then } \omega^{r}_0 = \frac{(\tilde{A}_r\boldsymbol{t}_0, \boldsymbol{t}_0)}{(\tilde{A}_r\boldsymbol{t}_0, \tilde{A}_r\boldsymbol{t}_0)}, \ \eta^{r}_0 = 0 \right)\\
\boldsymbol{u}^{+}_k &= \omega^{r}_k M^{-1}A\boldsymbol{p}^{+}_k + \eta^{r}_k(\boldsymbol{t}^{+}_{k-1} - \boldsymbol{r}^{+}_k + \beta^{\mathrm{L}}_{k-1}\boldsymbol{u}^{+}_{k-1}) = \underline{\underline{\tilde{A}_l Q_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{z}^{+}_k &= \omega^{r}_k\boldsymbol{r}^{+}_k + \eta^{r}_k\boldsymbol{z}^{+}_{k-1} - \alpha^{\mathrm{L}}_k\boldsymbol{u}^{+}_k = \underline{\underline{Q_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha^{\mathrm{L}}_k\boldsymbol{p}^{+}_k + \boldsymbol{z}^{+}_k\\
&= \boldsymbol{x}_k + \alpha^{\mathrm{L}}_k\underline{\underline{S_k(\tilde{A}_l)P^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}} + \underline{\underline{Q_k(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{r}_{k+1} &= \boldsymbol{t}_k - \eta^{r}_k\boldsymbol{y}_k - \omega^{r}_k A\boldsymbol{t}^{+}_k = \underline{\underline{MS_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\boldsymbol{r}^{+}_{k+1} &= M^{-1}\boldsymbol{r}_{k+1} = \underline{\underline{S_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},\\
\beta^{\mathrm{L}}_k &= \frac{\alpha^{\mathrm{L}}_k}{\omega^{r}_k} \times \frac{\langle \boldsymbol{r}^{\sharp}_0, \boldsymbol{r}^{+}_{k+1} \rangle}{\langle \boldsymbol{r}^{\sharp}_0, \boldsymbol{r}^{+}_k \rangle} = \frac{\alpha^{\mathrm{L}}_k}{\omega^{r}_k} \times \frac{\langle M^{-1}\boldsymbol{r}_0, S_{k+1}(\tilde{A}_l)R^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0 \rangle}{\langle M^{-1}\boldsymbol{r}_0, S_k(\tilde{A}_l)R^{\mathrm{L}}_k(\tilde{A}_l)M^{-1}\boldsymbol{r}_0 \rangle},\\
\boldsymbol{w}_k &= A\boldsymbol{t}^{+}_k + \beta^{\mathrm{L}}_k A\boldsymbol{p}^{+}_k = \underline{\underline{M\tilde{A}_l S_k(\tilde{A}_l)P^{\mathrm{L}}_{k+1}(\tilde{A}_l)M^{-1}\boldsymbol{r}_0}},
\end{aligned}
$$

End Do

Numerical results for `add32` using the relative residual of each algorithm.

Tables 13, 14, 15 show numerical results for `add32` as reference information[11].

[11] A difference in TRR and TRE occurred between the Left and the coLeft in spite of the fact that the required numbers were both 34 in Table 14. This is because coLeft converged at the early check.

Table 13 Numerical results by PBiCGStab using RHSone

Matrix	N	Right	Left	coLeft	ISRV9	Case1	Case2
		(Alg. 1)	(Alg. 2)	(Alg. 3)	(Alg. 4)	(Alg. 5)	(Alg. 6)
add32	4960	− 12.20 (37)	− 12.45 (36)	− 12.01 (35)	− 12.14 (36)	− 12.14 (36)	− 12.56 (38)
		− 11.03	− 11.27	− 10.94	− 11.03	− 11.03	− 11.38

Table 14 Numerical results by PGPBiCG using RHSone. COND is the condition number

Matrix	COND	Right (Alg. 7)	Left (Alg. 8)	coLeft (Alg. 9)	ISRV9 (Alg. 10)	Case1 (Alg. 11)	Case2 (Alg. 12)
add32	2.14e+02	− 12.06 (33) − 11.04	− 12.30 (34) − 11.32	− 12.06 (34) − 11.26	− 12.06 (34) − 11.23	− 12.06 (34) − 11.23	− 12.12 (33) − 11.21

Table 15 Numerical results by PBiCG using RHSone [7]

Matrix	Right- PBiCG (standard)	Left-PBiCG
add32	– 12.38 (64) – 12.19	– 12.27 (62) – 11.73

Acknowledgements The author gratefully acknowledges valuable discussions with the late professor Masaaki Sugihara. This work was supported by Grants-in-Aid for Scientific Research (KAKENHI 18K11342).

Declarations

Conflict of interest The author declares that he has no conflict of interest.

References

1. Barrett, R., Berry, M., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H.A.: Templates for the solution of linear systems: Building Blocks for Iterative Methods. SIAM (1994)
2. Davis, T.A.: The University of Florida sparse matrix collection. http://www.cise.ufl.edu/research/sparse/matrices/
3. Fletcher, R.: Conjugate gradient methods for indefinite systems. In: Watson, G. (ed.) Numerical Analysis Dundee 1975. Lecture Notes in Mathematics, vol. 506, pp. 73–89. Springer, Berlin (1976)
4. Gutknecht, M.H.: Variants of BICGSTAB for matrices with complex spectrum. SIAM J. Sci. Comput. 14, 1020–1033 (1993)
5. Gutknecht, M.H.: Lanczos-type solvers for nonsymmetric linear systems of equations. Acta Numerica. 6, 271–397 (1997)
6. Itoh, S., Katagiri, T., Sakurai, T., Igai, M., Ohshima, S., Kuroda, H., Naono, K.: An improvement in preconditioned algorithm of BiCGStab method. IPSJ Trans. Adv. Comput. Syst. 5(3), 11–21 (2012). ((**in Japanese**))
7. Itoh, S., Sugihara, M.: Structure of the polynomials in preconditioned BiCG algorithms and the switching direction of preconditioned systems. arXiv:1603.00175 [math.NA]. (2016)
8. Itoh, S., Sugihara, M.: Structure of the preconditioned system in various preconditioned conjugate gradient squared algorithms. Results Appl. Math. 3, 100008 (2019)
9. Itoh, S., Sugihara, M.: Changing over stopping criterion for stable solving nonsymmetric linear equations by preconditioned conjugate gradient squared method. Appl. Math. Lett. 102, 106088 (2020)
10. Lanczos, C.: Solution of systems of linear equations by minimized iterations. J. Res. Nat. Bur. Stand. 49, 33–53 (1952)
11. Matrix Market project: http://math.nist.gov/MatrixMarket/
12. Meurant, G.: Computer Solution of Large Linear Systems. Elsevier (2005)
13. Satake, I.: Linear Algebra. Marcel Dekker, Inc. (1975)
14. Simoncini, V., Szyld, D.B.: Interpreting IDR as a Petrov–Galerkin Method. SIAM J. Sci. Comput. 32, 1898–1912 (2010)
15. Sleijpen, G.L.G., Fokkema, D.R.: BiCGstab(L) for linear equations involving unsymmetric matrices with complex spectrum. Elec. Trans. Numer. Anal. 1, 11–32 (1993)

16. SSI project: Lis: Library of iterative solvers for linear systems. http://www.ssisc.org/lis/
17. Tanio, M., Sugihara, M.: GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials. J. Comput. Appl. Math. 235, 765–784 (2010)
18. Van der Vorst, H.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, 631–644 (1992)
19. Zhang, S.-L.: GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems. SIAM J. Sci. Comput. 18, 537–551 (1997)