Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 6, No. 2, 2018, pp. 111-127

New variants of the global Krylov type methods for linear systems with multiple right-hand sides arising in elliptic PDEs

Ghodrat Ebadi*
Faculty of Mathematical Sciences,
University of Tabriz, 51666-14766 Tabriz, Iran.
E-mail: ghodrat_ebadi@yahoo.com

Somaiyeh Rashedi
Faculty of Mathematical Sciences,
University of Tabriz, 51666-14766 Tabriz, Iran.
E-mail: s_rashedi_t@yahoo.com

Abstract In this paper, we present new variants of global bi-conjugate gradient (Gl-BiCG) and global bi-conjugate residual (Gl-BiCR) methods for solving nonsymmetric linear systems with multiple right-hand sides. These methods are based on global oblique projections of the initial residual onto a matrix Krylov subspace. It is shown that these new algorithms converge faster and more smoothly than the Gl-BiCG and Gl-BiCR methods. The preconditioned versions of these methods are also explored in this study. Eventually, the efficiency of these approaches are demonstrated through numerical experimental results arising from two and three-dimensional advection dominated elliptic PDE.

Keywords. Matrix Krylov subspaces, Elliptic partial differential equation, Nonsymmetric linear systems, Global iterative methods, Multiple right-hand sides.

2010 Mathematics Subject Classification. 65F10, 65F30, 65F50, 65H10.

1. Introduction

We consider simultaneous linear systems with the same coefficient matrix A and s right-hand sides (r.h.s.) b_i,

$$Ax_i = b_i,\ i = 1, \ldots, s, \tag{1.1}$$

which can be written in the block form as the matrix equation

$$AX = B,$$

where the matrix $A \in \mathbb{R}^{n\times n}$ is a real sparse nonsymmetric matrix, $X = [x_1, \ldots, x_s]$ and $B = [b_1, \ldots, b_s] \in \mathbb{R}^{n\times s}$ with usually $s \ll n$. Such linear systems with multiple right-hand sides arise frequently in computational science and engineering, such as Quantum Chromo Dynamics (QCD) [13, 19], dynamics of structures [2], wave propagation phenomena [22]. For the other types of matrix equations, many studies have been developed [3, 12, 16].

Received: 20 April 2017 ; Accepted: 7 January 2018.
∗ Corresponding author.

When the coefficient matrix is small, the problem (1.1) is usually solved by a direct method, such as LU- factorization. For large ones, many iterative methods, which are the generalizations of the classical Krylov subspace methods [18], have been proposed in recent years. A class of solvers for solving the problem (1.1) is the block methods. The first development and analysis of the block methods were introduced by O'Leary in [14] where Block BiCG (Bl-BiCG) was considered together with block conjugate gradients (Bl-CG). The block generalized minimal residual (Bl-GMRES) algorithm can be found in [20], the block quasi minimum residual (Bl-QMR) method was suggested in [4] and the block bi-conjugate stabilized algorithm (Bl-BiCGStab) was proposed in [6]. Since some breakdowns may occur in the block methods, due to very ill-conditioned matrices produced in computing the coefficients by imposing the bi-orthogonality condition, one can avoid this by using deflation process which means reducing the dimension of block Krylov subspaces [4].

Another class is the global methods, which are based on the use of a global projection process onto a matrix Krylov subspace. Global full orthogonalization (Gl-FOM) and global generalized minimal residual (Gl-GMRES) based on global Arnoldi process were introduced in [10]. Later, in 2005, global bi-conjugate gradient method (Gl-BiCG) method and global BiCGStab algorithm (Gl-BiCGStab) based on global Lanczos process were introduced in [11]. Some other global variants of Krylov subspace methods can be found in [8] (Gl-CMRH), [24, 25] (Gl-CGS) and [5] (Gl-SCD). In 2016, improved versions of the block and global methods for the simultaneous solutions of large and sparse linear systems of equations with non-Hermitian coefficient matrices were proposed in [15].

Recently Zhang et al. extracted the global bi-conjugate residual (Gl-BiCR) and its improved version global conjugate residual squared algorithm (Gl-CRS) [26]. This approach leads to smoother convergence behavior and converge faster than the Gl-BiCG as well as its corresponding variants. In this work, motivated by the same idea, we derive global bi-conjugate residual stabilized (Gl-BiCRStab) method to improve the effectiveness of Gl-BiCR. Moreover, as observed in [7], for some problems arising from PDEs the convergence of Gl-BiCGStab and Gl-BiCRStab stagnate. We therefore generalize the minimal residual BiCGStab (MRBiCGStab) [23], known for single right hand sides, to solve linear systems with multiple right-hand sides. The new algorithm is called the global minimal residual BiCGStab (Gl-MRBiCGStab). In the same way, we obtain the global minimal residual BiCRStab (Gl-MRBiCRStab). We expect that these two methods perform well, since the F- norm (Frobenius norm) of residual matrix is minimized over $\mathbb{R}^2$ [23].

The outline of this paper is organized as follows. Sections 2 is a preliminary section, describing Gl-BiCG, Gl-BiCR and scalar polynomial interpretation of these algorithms in some details, due to [11, 26]. In section 3, Gl-BiCRStab algorithm is presented by analogously using Gl-BiCGStab derivation process. Then we devote section 4 to deriving Gl-MRBiCGStab and its preconditioned version. In section 5, we report some numerical examples, which show the superior performance of our new improvements. Finally, conclusions are summarized in section 6.

Throughout this paper, we use the following notations. For two matrices Y and Z in $\mathbb{R}^{n\times s}$, we define the inner product $< Y, Z >_F = tr(Z^T Y)$ where $tr(Z^T Y)$ denotes

the trace of the matrix Z^TY. The associated norm is the Frobenius norm denoted by $\|.\|_F$. A system of matrices of $\mathbb{R}^{n\times s}$ is said to be F-orthonormal if it is orthonormal with respect to $<.,.>_F$.

2. Preliminaries

In this section, we review Gl-BiCG and Gl-BiCR methods for solving the matrix equation (1.1). Then we give the scalar polynomial interpretation of these methods using formal orthogonal polynomials. These methods reduce to the single r.h.s. method for $s = 1$.

2.1. GL-BiCG and GL-BiCR.

In global methods we take the s systems (1.1) into one big, tensorized $sn \times sn$ system

$$(I \otimes A)x = b, \text{ where } b = \begin{bmatrix} b_1 \\ \vdots \\ b_s \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_s \end{bmatrix}, \tag{2.1}$$

and then apply a standard Krylov subspace method to the system (2.1).

Let X_0 be an initial $n \times s$ matrix guess to the solution X of (1.1) and $R_0 = B - AX_0$ its associated residual. Then the approximate solution X_k generated by global iterative methods satisfies

$$X_k - X_0 \in \mathcal{K}_k(A, R_0),$$

where $\mathcal{K}_k(A, R_0)$ is the matrix Krylov subspace defined as:

$$\begin{aligned} \mathcal{K}_k(A, R_0) &= span\{R_0, AR_0, \ldots, A^{k-1}R_0\} \\ &= \Big\{ \sum_{i=0}^{k-1} \alpha_i A^i R_0, \ \alpha_i \in \mathbb{R}, \ i = 0, 1, \ldots, k-1 \Big\}. \end{aligned}$$

Although the recurrence formulas for updating an approximation and a residual matrix are the same in Gl-BiCG and Gl-BiCR approaches, the iteration coefficients α_j and β_j are different. These coefficients are determined by the following orthogonality conditions:

$$R_j \perp_F W, \quad AP_j \perp_F W.$$

Choosing $W = \mathcal{K}_j(A^T, \widetilde{R}_0)$ leads to Gl-BiCG given in the following Algorithm. The algorithm breaks down if the nominator in α_j or β_j is zero. In a similar spirit, Gl-BiCR algorithm can be derived by choosing $W = A^T\mathcal{K}_j(A^T, \widetilde{R}_0)$ and replacing the coefficients α_j and β_j of Gl-BiCG with the following coefficients

$$\alpha_j = \frac{\langle \widetilde{R}_j, AR_j \rangle_F}{\langle A^T\widetilde{P}_j, AP_j \rangle_F}, \ \beta_j = \frac{\langle \widetilde{R}_{j+1}, AR_{j+1} \rangle_F}{\langle \widetilde{R}_j, AR_j \rangle_F}.$$

Note that one can use $AP_{j+1} = AR_{j+1} + \beta_j AP_j$ in the Gl-BiCR algorithm to reduce the number of matrix-matrix multiplications per iteration step.

Algorithm 1: Global BiCG (Gl-BiCG)[11]

choose $X_0, \widetilde{R}_0 \in \mathbb{R}^{n\times s}$
put $R_0 = B - AX_0$, $P_0 = R_0, \widetilde{P}_0 = \widetilde{R}_0$
for $j = 0, 1, 2, \ldots$ until convergence **do**
  $\alpha_j = \frac{\langle \widetilde{R}_j, R_j\rangle_F}{\langle \widetilde{P}_j, AP_j\rangle_F}$
  $X_{j+1} = X_j + \alpha_j P_j$
  $R_{j+1} = R_j - \alpha_j AP_j,\ \widetilde{R}_{j+1} = \widetilde{R}_j - \alpha_j A^T \widetilde{P}_j$
  $\beta_j = \frac{\langle \widetilde{R}_{j+1}, R_{j+1}\rangle_F}{\langle \widetilde{R}_j, R_j\rangle_F}$
  $P_{j+1} = R_{j+1} + \beta_j P_j,\ \widetilde{P}_{j+1} = \widetilde{R}_{j+1} + \beta_j \widetilde{P}_j$

2.2. **Connection with scalar polynomials.** The residual R_j of the Gl-BiCG and Gl-BiCR algorithms is defined as

$$R_j = \phi_j(A)R_0, \tag{2.2}$$

where ϕ_j is a polynomial of degree j with scalar coefficients satisfying $\phi_j(0) = 1$ and the direction P_j is defined as

$$P_j = \psi_j(A)R_0, \tag{2.3}$$

where ψ_j is a polynomial with scalar coefficients. Note that $\widetilde{R}_j$ and $\widetilde{P}_j$ can also be expressed as $\widetilde{R}_j = \phi_j(A^T)\widetilde{R}_0$ and $\widetilde{P}_j = \psi_j(A^T)\widetilde{R}_0$. These scalar polynomials are related by the following recurrence formulas

$$\phi_{j+1}(t) = \phi_j(t) - \alpha_j t\psi_j(t), \tag{2.4}$$

and

$$\psi_{j+1}(t) = \phi_{j+1}(t) + \beta_j\psi_j(t), \tag{2.5}$$

with $\phi_0(t) = \psi_0(t) = 1$, for $t \in \mathbb{R}$.

For Gl-BiCG and its variants the notation $\mathcal{C}$ is considered as the functional defined on the set of scalar polynomials with coefficients in $\mathbb{R}$ and given by Jbilou et al. [11]

$$\mathcal{C}(\phi) = \langle \phi(A)R_0, \widetilde{R}_0\rangle_F,$$

where ϕ is a scalar polynomial. We also define the functional

$$\mathcal{C}^{(1)}(\phi) = \langle A\phi(A)R_0, \widetilde{R}_0\rangle_F.$$

With these definitions, it is easy to prove the following properties:

$$\mathcal{C}(\phi + \psi) = \mathcal{C}(\phi) + \mathcal{C}(\psi),$$
$$\mathcal{C}(t\phi) = t\mathcal{C}(\phi),\ \ for\ t \in R.$$

The same relations are also satisfied by $\mathcal{C}^{(1)}$. The scalar polynomials ϕ_j and ψ_j associated by (2.2) and (2.3) with the matrix residual and the matrix direction polynomials generated by Gl-BiCG algorithm belong to families of formal orthogonal polynomials with respect to $\mathcal{C}$ and $\mathcal{C}^{(1)}$. These results are generalized in the next proposition.

Proposition 2.1. *Let ϕ_j and ψ_j, for $j \geq 1$ be the sequences of scalar polynomial defined by (2.4) and (2.5). If U_i is an arbitrary scalar polynomial of degree i, $i = 0, 1, \ldots, j-1$, then we have the following orthogonality properties:*

$$\mathcal{C}(\phi_j U_i) = 0, \text{ for } i = 0, \ldots, j-1,$$
$$\mathcal{C}^{(1)}(\psi_j U_i) = 0, \text{ for } i = 0, \ldots, j-1.$$

Proof. Refer to [25]. □

Using the relations (2.4), (2.5), Proposition 2.1 and the fact that $\widetilde{\phi}_j$ and $\widetilde{\psi}_j$ are some sequence of polynomials of degree j, we have

$$\mathcal{C}(\widetilde{\phi}_j \phi_j) = \alpha_j \mathcal{C}^{(1)}(\widetilde{\phi}_j \psi_j), \tag{2.6}$$

and

$$\mathcal{C}^{(1)}(\widetilde{\psi}_j \phi_{j+1}) = -\beta_j \mathcal{C}^{(1)}(\widetilde{\psi}_j \psi_j). \tag{2.7}$$

Therefore, by the definition of the functionals $\mathcal{C}$ and $\mathcal{C}^{(1)}$, the relations (2.6) and (2.7) become

$$\alpha_j = \frac{\langle R_j, \widetilde{\phi}_j(A^T)\widetilde{R}_0\rangle_F}{\langle AP_j, \widetilde{\phi}_j(A^T)\widetilde{R}_0\rangle_F}, \tag{2.8}$$

$$\beta_j = -\frac{\langle R_{j+1}, A^T\widetilde{\psi}_j(A^T)\widetilde{R}_0\rangle_F}{\langle AP_j, \widetilde{\psi}_j(A^T)\widetilde{R}_0\rangle_F}. \tag{2.9}$$

For the Gl-BiCR and its variants the functional $\mathcal{C}$ and $\mathcal{C}^{(1)}$ are defined as

$$\begin{aligned} \mathcal{C}(\phi) &= \langle \phi(A)R_0, A^T\widetilde{R}_0\rangle_F, \\ \mathcal{C}^{(1)}(\phi) &= \langle A\phi(A)R_0, A^T\widetilde{R}_0\rangle_F. \end{aligned}$$

Therefore, α_j and β_j can be obtained as

$$\alpha_j = \frac{\langle R_j, \widetilde{\phi}_j(A^T)A^T\widetilde{R}_0\rangle_F}{\langle AP_j, \widetilde{\phi}_j(A^T)A^T\widetilde{R}_0\rangle_F}, \tag{2.10}$$

$$\beta_j = -\frac{\langle R_{j+1}, A^T\widetilde{\psi}_j(A^T)A^T\widetilde{R}_0\rangle_F}{\langle AP_j, \widetilde{\psi}_j(A^T)A^T\widetilde{R}_0\rangle_F}. \tag{2.11}$$

3. Gl-BiCR stabilized algorithm (Gl-BiCRStab)

In this section, we derive a new variant of Gl-BiCR, called Gl-BiCRStab, which speeds up the convergence of the Gl-BiCR.

The residual of Gl-BiCRStab is defined by

$$R_j = \widetilde{\phi}_j(A)\phi_j(A)R_0,$$

where $\widetilde{\phi}_j$ is a polynomial defined by a coupled two-term recurrence and ϕ_j is the Gl-BiCR polynomial. The polynomial recurrence for the polynomial $\widetilde{\phi}_{j+1}$ is defined

as

$$\begin{aligned}\widetilde{\phi}_0(A) &= I,\\ \widetilde{\phi}_{j+1}(A) &= \widetilde{\phi}_j(A) - w_j A\widetilde{\phi}_j(A),\end{aligned} \tag{3.1}$$

for scalar sequence w_j. One can write (3.1) as

$$\widetilde{\phi}_{j+1}(A) = (I - w_0 A)(I - w_1 A)\dots(I - w_j A).$$

We have

$$R_{j+1} = \widetilde{\phi}_{j+1}(A) R_{j+1}^{Gl-BiCR},$$

where $R_j^{Gl-BiCR}$ and $P_j^{Gl-BiCR}$ are defined in (2.2) and (2.3), respectively. Thus, we have

$$R_{j+1}^{Gl-BiCR} = R_j^{Gl-BiCR} - \alpha_j A P_j^{Gl-BiCR}, \tag{3.2}$$

$$P_{j+1}^{Gl-BiCR} = R_{j+1}^{Gl-BiCR} + \beta_j P_j^{Gl-BiCR}. \tag{3.3}$$

The appropriate recurrence relations for Gl-BiCRStab can be derived. Ignoring the scalar coefficients at first, we start with a relation for the residual polynomial $\widetilde{\phi}_{j+1} R_{j+1}^{Gl-BiCR}$. We immediately obtain

$$\widetilde{\phi}_{j+1} R_{j+1}^{Gl-BiCR} = \widetilde{\phi}_j R_{j+1}^{Gl-BiCR} - w_j A\widetilde{\phi}_j R_{j+1}^{Gl-BiCR}, \tag{3.4}$$

which is updatable if a recurrence relation is found for the $\widetilde{\phi}_j R_{j+1}^{Gl-BiCR}$. For this, we write

$$\widetilde{\phi}_j R_{j+1}^{Gl-BiCR} = \widetilde{\phi}_j R_j^{Gl-BiCR} - \alpha_j A\widetilde{\phi}_j P_j^{Gl-BiCR}. \tag{3.5}$$

$\widetilde{\phi}_{j+1} P_{j+1}^{GL-BiCR}$ may be updated in a similar way, i.e.,

$$\widetilde{\phi}_{j+1} P_{j+1}^{Gl-BiCR} = \widetilde{\phi}_{j+1} R_{j+1}^{Gl-BiCR} + \beta_j A\widetilde{\phi}_{j+1} P_j^{Gl-BiCR}, \tag{3.6}$$

where

$$\widetilde{\phi}_{j+1} P_j^{Gl-BiCR} = \widetilde{\phi}_j P_j^{Gl-BiCR} - w_j A\widetilde{\phi}_j P_j^{Gl-BiCR}. \tag{3.7}$$

Combining (3.4) and (3.5) leads to

$$\widetilde{\phi}_{j+1} R_{j+1}^{Gl-BiCR} = \widetilde{\phi}_j R_j^{Gl-BiCR} - \alpha_j A\widetilde{\phi}_j P_j^{Gl-BiCR} - w_j A\widetilde{\phi}_j R_{j+1}^{Gl-BiCR}. \tag{3.8}$$

Similarly, combining (3.6) and (3.7) leads to

$$\widetilde{\phi}_{j+1} P_{j+1}^{Gl-BiCR} = \widetilde{\phi}_{j+1} R_{j+1}^{GL-BiCR} + \beta_j (I - w_j A)\widetilde{\phi}_j P_j^{Gl-BiCR}. \tag{3.9}$$

The new matrices are defined as follows:

$$R_{j+1} = \widetilde{\phi}_{j+1} R_{j+1}^{Gl-BiCR}, \quad P_{j+1} = \widetilde{\phi}_{j+1} P_{j+1}^{Gl-BiCR}, \quad S_j = \widetilde{\phi}_j R_{j+1}^{Gl-BiCR}.$$

The above recurrences (3.4), (3.8), (3.9) and (3.5) are respectively converted into

$$\begin{aligned}R_{j+1} &= (I - w_j A) S_j,\\ R_{j+1} &= R_j - \alpha_j A P_j - w_j A S_j,\\ P_{j+1} &= R_{j+1} + \beta_j (I - w_j A) P_j,\\ S_j &= R_j - \alpha_j A P_j.\end{aligned}$$

The iteration coefficients α_j and β_j can be computed in the way that R_j and AP_j are orthogonal to the Krylov subspace $A^T K_j(A^T, \widetilde{R}_0)$. From (2.10) and (2.11) these coefficients are determined

$$\alpha_j = \frac{\langle R_j, A^T \widetilde{R}_0 \rangle_F}{\langle AP_j, A^T \widetilde{R}_0 \rangle_F},$$

$$\beta_j = -\frac{\langle AS_j, A^T \widetilde{R}_0 \rangle_F}{\langle AP_j, A^T \widetilde{R}_0 \rangle_F}.$$

Also the parameter w_j is selected to minimize the $F-$norm of R_j, so we have

$$w_j = \frac{\langle S_j, AS_j \rangle_F}{\langle AS_j, AS_j \rangle_F}.$$

Using the above results, the resulting Gl-BiCRStab algorithm can be summarized as follows. When $s = 1$, the Gl-BiCRStab algorithm reduces to the BiCRStab algorithm

Algorithm 2: Global BiCRStab (Gl-BiCRStab)

choose $X_0, \widetilde{R}_0 \in \mathbb{R}^{n\times s}$
put $R_0 = B - AX_0$, $P_0 = R_0, \widetilde{P}_0 = \widetilde{R}_0$
for $j = 0, 1, 2, \ldots$ until convergence **do**
- $Q_j = AP_j$
- $\alpha_j = \frac{\langle R_j,\ A^T \widetilde{R}_0 \rangle_F}{\langle Q_j,\ A^T \widetilde{R}_0 \rangle_F}$
- $S_j = R_j - \alpha_j Q_j$
- $T_j = AS_j$
- $w_j = \frac{\langle S_j, T_j \rangle_F}{\langle T_j,\ T_j \rangle_F}$
- $X_{j+1} = X_j + \alpha_j P_j + w_j S_j$
- $R_{j+1} = S_j - w_j T_j$
- $\beta_j = -\frac{\langle T_j,\ A^T \widetilde{R}_0 \rangle_F}{\langle Q_j,\ A^T \widetilde{R}_0 \rangle_F}$
- $P_{j+1} = R_{j+1} + \beta_j (P_j - w_j Q_j)$

[1]. Note that the Gl-BiCRStab algorithm does not suffer from dependence of vectors during the iterations until a matrix invariant subspace is obtained. Therefore, no deflation procedure to delete linearly or almost linearly dependent vectors is used. However, a breakdown may occur in Algorithm 2 if $\langle Q_j, A^T \widetilde{R}_0 \rangle_F = 0$. In [11], Jbilou et al. gave a look-ahead technique to avoid this problem. This problem is not treated in this study.

4. Global minimal residual BiCGStab algorithm (Gl-MRBiCGStab)

Although Gl-BiCGStab is now considered as one of the most effective methods for solving nonsymmetric linear systems with multiple right-hand sides, unfortunately for some problems specifically in case of discretized advection dominated PDEs., where,

for instance, Gl-BiCG performs well, the convergence of Gl-BiCGStab stagnates. In order to overcome this weakness, we incorporate a quadratic stabilization polynomial into Gl-BiCGStab algorithm. This method is expected to reveal better result.

As was suggested in [23] for one right-hand side linear system, we use a polynomial $\widetilde{\Phi}_{2j}$ which is a product of k quadratic polynomials. The residual of Gl-MRBiCGStab is denoted by R_{2j}^{Gl-MR} and defined as

$$R_{2j}^{Gl-MR} = \widetilde{\Phi}_{2j}(A)R_{2j}^{Gl-BiCG},$$

where $R_j^{Gl-BiCG}$ is the residual of Gl-BiCG defined in (2.2) and $\widetilde{\Phi}_{2j}(A)$ satisfies

$$\begin{aligned}\widetilde{\Phi}_0(A) &= I,\\ \widetilde{\Phi}_{2j}(A) &= (I + w_1A + w_2A^2)(I + w_3A + w_3A^2)\dots(I + w_{2j-1}A + w_{2j}A^2)\\ &= (I + w_{2j-1}A + w_{2j}A^2)\widetilde{\Phi}_{2j-2}. \end{aligned} \tag{4.1}$$

The matrix residuals R_{2j}^{Gl-MR} and the search matrix directions P_{2j}^{Gl-MR} can be taken

$$\begin{aligned}R_{2j}^{Gl-MR} &= \widetilde{\Phi}_{2j}(A)R_{2j}^{Gl-BiCG},\\ P_{2j}^{Gl-MR} &= \widetilde{\Phi}_{2j}(A)P_{2j}^{Gl-BiCG},\end{aligned}$$

where $P_j^{Gl-BiCG}$ is defined in (2.3). We will concentrate on the matrix updates first. Assume that the iteration coefficients α_j, β_j and w_j are explicitly given. Following from (3.2) and (3.3), we have

$$\widetilde{\Phi}_{2j}R_{2j+1}^{Gl-BiCG} = \widetilde{\Phi}_{2j}R_{2j}^{Gl-BiCG} - \alpha_{2j}A\widetilde{\Phi}_{2j}P_{2j}^{Gl-BiCG}, \tag{4.2}$$

$$\widetilde{\Phi}_{2j}P_{2j+1}^{Gl-BiCG} = \widetilde{\Phi}_{2j}R_{2j+1}^{Gl-BiCG} + \beta_{2j}\widetilde{\Phi}_{2j}P_{2j}^{Gl-BiCG}, \tag{4.3}$$

$$\widetilde{\Phi}_{2j}R_{2j+2}^{Gl-BiCG} = \widetilde{\Phi}_{2j}R_{2j+1}^{Gl-BiCG} - \alpha_{2j+1}A\widetilde{\Phi}_{2j}P_{2j+1}^{Gl-BiCG}. \tag{4.4}$$

From (4.1), we have

$$\begin{aligned}\widetilde{\Phi}_{2j+2}R_{2j+2}^{Gl-BiCG} &= \widetilde{\Phi}_{2j}R_{2j+2}^{Gl-BiCG} + w_{2j+1}A\widetilde{\Phi}_{2j}R_{2j+2}^{Gl-BiCG}\\ &\quad + w_{2j+2}A^2\widetilde{\Phi}_{2j}R_{2j+2}^{Gl-BiCG}, \end{aligned} \tag{4.5}$$

$$\begin{aligned}\widetilde{\Phi}_{2j+2}P_{2j+2}^{Gl-BiCG} &= \widetilde{\Phi}_{2j+2}R_{2j+2}^{Gl-BiCG} + \beta_{2j+1}\{\widetilde{\Phi}_{2j}P_{2j+1}^{Gl-BiCG}\\ &\quad + w_{2j+1}A\widetilde{\Phi}_{2j}P_{2j+1}^{Gl-BiCG} + w_{2j+2}A^2\widetilde{\Phi}_{2j}P_{2j+1}^{Gl-BiCG}\}. \end{aligned} \tag{4.6}$$

Let

$$\begin{aligned}\bar{R}_{2j+1} &= \widetilde{\Phi}_{2j}R_{2j+1}^{Gl-BiCG},\\ \bar{R}_{2j+2} &= \widetilde{\Phi}_{2j}R_{2j+2}^{Gl-BiCG},\\ \bar{P}_{2j+1} &= \widetilde{\Phi}_{2j}P_{2j+1}^{Gl-BiCG},\end{aligned}$$

then the above recurrences (4.2)-(4.6) are respectively converted into

$$\begin{aligned}
\bar{R}_{2j+1} &= R_{2j}^{Gl-MR} - \alpha_{2j} A P_{2j}^{Gl-MR},\\
\bar{P}_{2j+1} &= \bar{R}_{2j+1} + \beta_{2j} P_{2j}^{Gl-MR},\\
\bar{R}_{2j+2} &= \bar{R}_{2j+1} - \alpha_{2j+1} A \bar{P}_{2j+1},\\
R_{2j+2}^{Gl-MR} &= \bar{R}_{2j+2} + w_{2j+1} A \bar{R}_{2j+2} + w_{2j+2} A^2 \bar{R}_{2j+2},\\
P_{2j+2}^{Gl-MR} &= R_{2j+2}^{Gl-MR} + \beta_{2j+1}\{\bar{P}_{2j+1} + w_{2j+1} A \bar{P}_{2j+1} + w_{2j+2} A^2 \bar{P}_{2j+1}\}.
\end{aligned}$$

Now we determine the parameters α_{2j}, β_{2j}, α_{2j+1} and β_{2j+1} in terms of new matrices R_{2j}^{Gl-MR}, P_{2j}^{Gl-MR}, $\bar{R}_{2j+1}$, $\bar{P}_{2j+1}$ and $\bar{R}_{2j+2}$. Using (2.8), (2.9) and taking $\widetilde{\phi}_{2j} = \widetilde{\psi}_{2j} = \widetilde{\Phi}_{2j}$ and $\widetilde{\Phi}_{2j+1} = A\Phi_{2j}$ yield

$$\begin{aligned}
\alpha_{2j} &= \frac{\langle \widetilde{\Phi}_{2j}(A)\phi_{2j}(A)R_0, \widetilde{R}_0\rangle_F}{\langle A\widetilde{\Phi}_{2j}(A)\psi_{2j}(A)R_0, \widetilde{R}_0\rangle_F} = \frac{\langle R_{2j}^{Gl-MR}, \widetilde{R}_0\rangle_F}{\langle A P_{2j}^{Gl-MR}, \widetilde{R}_0\rangle_F},\\
\beta_{2j} &= -\frac{\langle A\widetilde{\Phi}_{2j}(A)\phi_{2j+1}(A)R_0, \widetilde{R}_0\rangle_F}{\langle A\widetilde{\Phi}_{2j}(A)\psi_{2j}(A)R_0, \widetilde{R}_0\rangle_F} = -\frac{\langle A\bar{R}_{2j+1}, \widetilde{R}_0\rangle_F}{\langle A P_{2j}^{Gl-MR}, \widetilde{R}_0\rangle_F}\\
&= -\alpha_{2j}\frac{\langle A\bar{R}_{2j+1}, \widetilde{R}_0\rangle_F}{\langle R_{2j}^{Gl-MR}, \widetilde{R}_0\rangle_F},\\
\alpha_{2j+1} &= \frac{\langle A\widetilde{\Phi}_{2j}(A)\phi_{2j+1}(A)R_0, \widetilde{R}_0\rangle_F}{\langle A^2\widetilde{\Phi}_{2j}(A)\psi_{2j+1}(A)R_0, \widetilde{R}_0\rangle_F} = \frac{\langle A\bar{R}_{2j+1}, \widetilde{R}_0\rangle_F}{\langle A^2\bar{P}_{2j+1}, \widetilde{R}_0\rangle_F},\\
\beta_{2j+1} &= -\frac{\langle A^2\widetilde{\Phi}_{2j}(A)\phi_{2j+2}(A)R_0, \widetilde{R}_0\rangle_F}{\langle A^2\widetilde{\Phi}_{2j}(A)\psi_{2j+1}(A)R_0, \widetilde{R}_0\rangle_F} = -\frac{\langle A^2\bar{R}_{2j+2}, \widetilde{R}_0\rangle_F}{\langle A^2\bar{P}_{2j+1}, \widetilde{R}_0\rangle_F}\\
&= -\alpha_{2j+1}\frac{\langle A^2\bar{R}_{2j+1}, \widetilde{R}_0\rangle_F}{\langle A\bar{R}_{2j+1}, \widetilde{R}_0\rangle_F},
\end{aligned}$$

The parameters w_{2j+1} and w_{2j+2} are chosen to minimize the $F-$ norm of the matrix residual $\|R_{2j+2}^{Gl-MR}\|$. Since the $\|R_{2j+2}^{Gl-MR}\|$ is minimized over two dimensional vector space $\mathbb{R}^2$, it may be expected that Gl-MRBiCGStab converges faster than the Gl-BiCGStab in which residual norm is minimized over one dimensional vector space $\mathbb{R}^2$ [23].

The computational costs of the above parameters are expensive. In practice, the recurrence relations of the Gl-MRBiCGStab is computed in different way to reduce the number of matrix-matrix multiplication as the following algorithm:

Algorithm 3: Global Minimal Residual BiCGStab (Gl-MRBiCGStab)

choose $X_0, \widetilde{R}_0 \in \mathbb{R}^{n\times s}$
put $R_0^* = B - AX_0$, $P_0 = R_0, \widetilde{P}_0^* = \widetilde{R}_0$
for $j = 0, 2, 4, \ldots$ until convergence **do**
Compute AP_j^*
$\alpha_j = \frac{\langle R_j^*, \widetilde{R}_0\rangle_F}{\langle AP_j^*, \widetilde{R}_0\rangle_F}$
$\bar{X}_{j+1} = X_j + \alpha_j P_j^*$
$\bar{R}_{j+1} = R_j^* - \alpha_j AP_j^*$
$\widehat{R}_{j+1} = A\bar{R}_{j+1}$
$\beta_j = -\alpha_j \frac{\langle \widehat{R}_{j+1}, \widetilde{R}_0\rangle_F}{\langle R_j^*, \widetilde{R}_0\rangle_F}$
$\bar{P}_{j+1} = \bar{R}_{j+1} + \beta_j P_j^*$
$\widehat{P}_{j+1} = \widehat{R}_{j+1} + \beta_j AP_j^*$
$\widehat{P}_{j+1}^* = A\widehat{P}_j$
$\alpha_{j+1} = \frac{\langle \widehat{R}_{j+1}, \widetilde{R}_0\rangle_F}{\langle \widehat{P}_{j+1}^*, \widetilde{R}_0\rangle_F}$
$\bar{X}_{j+2} = \bar{X}_{j+1} + \alpha_{j+1}\bar{P}_{j+1}$
$\bar{R}_{j+2} = \bar{R}_{j+1} - \alpha_{j+1}\widehat{P}_{j+1}$
$\widehat{R}_{j+2} = \widehat{R}_{j+1} - \alpha_{j+1}\widehat{P}_{j+1}^*$
$\widehat{R}_{j+1}^* = A\widehat{R}_{j+2}$
$w_{j+1} = \frac{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \bar{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F - \langle \bar{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \widehat{R}_{j+2}^*, \widehat{R}_{j+2}^*\rangle_F}{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \widehat{R}_{j+2}^*, \widehat{R}_{j+2}^*\rangle_F - \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F}$
$w_{j+2} = \frac{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \bar{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F - \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \bar{R}_{j+2}, \widehat{R}_{j+2}\rangle_F}{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F - \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F}$
$X_{j+2} = \bar{X}_{j+2} - w_{j+1}\bar{R}_{j+2} - w_{j+2}\widehat{R}_{j+2}$
$R_{j+2}^* = \bar{R}_{j+2} + w_{j+1}\widehat{R}_{j+2} + w_{j+2}\widehat{R}_{j+2}^*$
$\beta_{j+1} = -\alpha_{j+1}\frac{\langle \widehat{R}_{j+2}^*, \widetilde{R}_0\rangle_F}{\langle \widehat{R}_{j+1}, \widetilde{R}_0\rangle_F}$
$P_{j+2}^* = R_{j+2}^* + \beta_{j+1}(\bar{P}_{j+1} + W_{j+1}\widehat{P}_{j+1} + w_{j+2}\widehat{P}_{j+1}^*)$

The effectiveness of the iterative methods is usually achieved by using a suitable preconditioner. The (right) preconditioned version of Algorithm 3 is obtained by applying it to $AK^{-1}y = B$ and recovering X through $X = K^{-1}y$ where K is the preconditioner. The resulting preconditioned algorithm is summarized in Algorithm 4.

Algorithm 4: Preconditioned Gl-MRBiCGStab

choose $X_0, \widetilde{R}_0 \in \mathbb{R}^{n\times s}$
put $R_0^* = B - AX_0$, $P_0^* = R_0$
for $j = 0, 2, 4, \ldots$ until convergence **do**
- Solve $K\widetilde{P}_j^* = P_j^*$ and compute $A\widetilde{P}_j^*$
- $\alpha_j = \frac{\langle R_j^*, \widetilde{R}_0\rangle_F}{\langle A\widetilde{P}_j^*, \widetilde{R}_0\rangle_F}$
- $\bar{X}_{j+1} = X_j + \alpha_j \widetilde{P}_j^*$
- $\bar{R}_{j+1} = R_j^* - \alpha_j A\widetilde{P}_j^*$
- Solve $K\widetilde{\bar{R}}_{j+1} = \bar{R}_{j+1}$ and compute $\widehat{R}_{j+1} = A\widetilde{\bar{R}}_{j+1}$
- $\beta_j = -\alpha_j \frac{\langle \widehat{R}_{j+1}, \widetilde{R}_0\rangle_F}{\langle R_j^*, \widetilde{R}_0\rangle_F}$
- $\bar{P}_{j+1} = \bar{R}_{j+1} + \beta_j P_j^*$
- $\widetilde{\bar{P}}_{j+1} = \widetilde{\bar{R}}_{j+1} + \beta_j \widetilde{P}_j^*$
- $\widehat{P}_{j+1} = \widehat{R}_{j+1} + \beta_j A\widetilde{P}_j^*$
- Solve $K\widetilde{\widehat{P}}_{j+1} = \widehat{P}_{j+1}$ and compute $\widehat{P}_{j+1}^* = A\widetilde{\widehat{P}}_{j+1}$
- $\alpha_{j+1} = \frac{\langle \widehat{R}_{j+1}, \widetilde{R}_0\rangle_F}{\langle \widehat{P}_{j+1}^*, \widetilde{R}_0\rangle_F}$
- $\bar{X}_{j+2} = \bar{X}_{j+1} + \alpha_{j+1}\widetilde{\bar{P}}_{j+1}$
- $\bar{R}_{j+2} = \bar{R}_{j+1} - \alpha_{j+1}\widehat{P}_{j+1}$
- $\widetilde{\bar{R}}_{j+2} = \widetilde{\bar{R}}_{j+1} - \alpha_{j+1}\widetilde{\widehat{P}}_{j+1}$
- $\widehat{R}_{j+2} = \widehat{R}_{j+1} - \alpha_{j+1}\widehat{P}_{j+1}^*$
- Solve $K\widetilde{\widehat{R}}_{j+2} = \widehat{R}_{j+2}$ and compute $\widehat{R}_{j+2}^* = A\widetilde{\widehat{R}}_{j+2}$
- $\widehat{R}_{j+2}^* = A\widetilde{\widehat{R}}_{j+2}$
- $w_{j+1} = \frac{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \bar{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F - \langle \bar{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \widehat{R}_{j+2}^*, \widehat{R}_{j+2}^*\rangle_F}{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \widehat{R}_{j+2}^*, \widehat{R}_{j+2}^*\rangle_F - \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F}$
- $w_{j+2} = \frac{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \bar{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F - \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \bar{R}_{j+2}, \widehat{R}_{j+2}\rangle_F}{\langle \widehat{R}_{j+2}, \widehat{R}_{j+2}\rangle_F \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F - \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F \langle \widehat{R}_{j+2}, \widehat{R}_{j+2}^*\rangle_F}$
- $X_{j+2} = \bar{X}_{j+2} - w_{j+1}\widetilde{\bar{R}}_{j+2} - w_{j+2}\widetilde{\widehat{R}}_{j+2}$
- $R_{j+2}^* = \bar{R}_{j+2} + w_{j+1}\widehat{R}_{j+2} + w_{j+2}\widehat{R}_{j+2}^*$
- $\beta_{j+1} = -\alpha_{j+1}\frac{\langle \widehat{R}_{j+2}^*, \widetilde{R}_0\rangle_F}{\langle \widehat{R}_{j+1}, \widetilde{R}_0\rangle_F}$
- $P_{j+2}^* = R_{j+2}^* + \beta_{j+1}(\bar{P}_{j+1} + W_{j+1}\widehat{P}_{j+1} + w_{j+2}\widehat{P}_{j+1}^*)$

To end this section, we remark that it is possible to, in a similar spirit, derive Gl-MRBiCRStab algorithm. We do not write down explicitly the resulting algorithm, Gl-MRBiCRStab, but we will report its performance in our numerical experiments. When solving one linear system of equations, this method reduces to the MRBiCRStab [27].

5. Numerical experiments

The purpose of this section is to assess the efficiency of the various global methods. All experiments were performed in double precision floating-point arithmetic on a PC with Intel(R)Core (TM) i5, CPU 2.67 GHz and 4.00 GB of RAM using MATLAB R2008. In all runs the iteration was started with $X_0 = 0$, $B = rand(n, 10)$, $\tilde{R}_0 = R_0$, the stopping criterion was $\frac{\|R^{(k)}\|_F}{\|R^{(0)}\|_F} \leq 10^{-10}$ or when the number of iterations exceeded 800. For all examples we will also report results using (right) no-fill ILU preconditioner. Note that Its, CPU and $\|R\|_F$ stand for the number of iterations, consuming time in seconds and final residual norm, respectively. If this final residual norm is larger than 1, we interpret this as divergence, noted as "div." in the tables. Smaller residual norms like 10^{-6} may be interpreted as an indicator of slow convergence.

Example 5.1. We consider the three-dimensional advection dominated elliptic PDE on the unit cube

$$u_{xx} + u_{yy} + u_{zz} + \nu \cdot u_x = f, \quad (x, y, z) \in \Omega = (0, 1)^3, \tag{5.1}$$

where $\nu > 0$ is a parameter which controls the influence of the convection term and f is defined such that for zero Dirichlet boundary conditions the exact solution is

$$u(x, y, z) = \exp(xyz)\sin(\pi x)\sin(\pi y)\sin(\pi z).$$

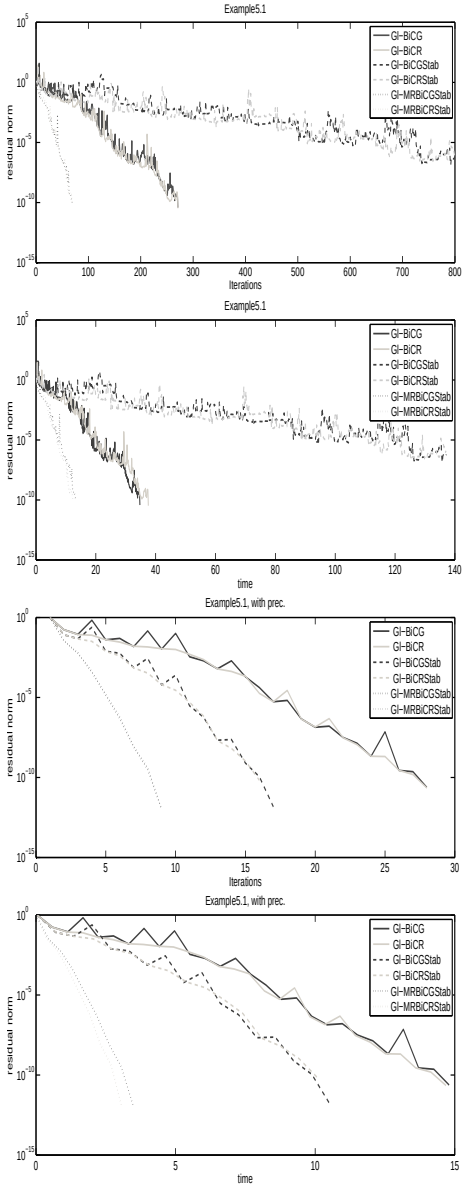
This PDE was considered in [21] as an example where BiCGStab encounters convergence problems for larger values of ν. We discretize (5.1) using second order finite differences on an equispaced grid with 50^3 interior grid points. This gives a linear system of size $n = 125,000$ with seven non-zeros per row.

Table 1: Numerical results of Example 5.1.

	no preconditioning			ILU preconditioning		
Methods	*Its*	$CPU(s)$	$\|R\|_F$	*Its*	$CPU(s)$	$\|R\|_F$
Gl-BiCG	271	34.71	1.19e-010	28	14.75	7.49e-011
Gl-BiCGStab	800	137.1	1.72e-006	17	10.48	4.67e-012
Gl-MRBiCGStab	69	13.36	3.09e-010	9	3.49	3.20e-012
Gl-BiCR	271	37.53	1.04e-010	28	14.63	6.97e-011
Gl-BiCRStab	800	137.0	3.95e-006	16	10.04	2.11e-010
Gl-MRBiCRStab	69	12.30	3.09e-010	9	3.07	3.19e-012

Figure 1 shows convergence plots for Example 5.1 without preconditioning (top row) and with a (right) no-fill ILU preconditioner (bottom row). The left plots show the relative Frobenius norm of the residual as a function of the total number of iterations. The right plots show the same residual norms, but now as a function of time. We can see from Figure 1 while Gl-BiCGStab works worse than Gl-BiCG and has slow convergence, Gl-MRBiCGStab converges nicely and almost twice as fast as Gl-BiCG. The same results can be observed for Gl-BiCR and its variants. Also, we see that Gl-MRBiCGStab and Gl-MRBiCRStab give much smoother convergence behavior than the other proposed methods.

FIGURE 1. Convergence plots for Gl-BiCG, Gl-BiCR and their variants. Top: Example 5.1, bottom: Example 5.1 with ILU preconditioning.



In the presence of preconditioning, where the preconditioned matrix is better conditioned and convergence is reached after a relatively small number of iterations,

Gl-BiCGStab and Gl-BiCRStab converge faster than Gl-BiCG and Gl-BiCR, respectively. But, Gl-MRBiCGStab and Gl-MRBiCRStab return the best results in this case. They converge faster and give much smoother convergence behavior than the others. Table 1 confirms our discussion of Figure 1: The convergence of Gl-MRBiCGStab and Gl-MRBiCRStab methods are superior to that of the other methods. Moreover, Gl-BiCR variants converge slightly faster than their Gl-BiCG counterparts in almost all cases. The known convergence problems of Gl-BiCGStab and Gl-BiCRStab are the fact that these methods are not able to efficiently accommodate the relevant spectral properties of the respective matrix [21].

Example 5.2. Same as Example 5.1, but now with $\nu = 10$.

Table 2: Numerical results of Example 5.2.

	no preconditioning			ILU preconditioning		
Methods	Its	$CPU(s)$	$\|R\|_F$	Its	$CPU(s)$	$\|R\|_F$
Gl-BiCG	227	29.24	2.60e-010	74	38.64	2.18e-010
Gl-BiCGStab	147	25.46	2.98e-010	53	33.89	2.62e-010
Gl-MRBiCGStab	75	14.35	2.085e-010	25	10.89	1.86e-010
Gl-BiCR	225	31.27	2.58e-010	75	39.93	1.68e-010
Gl-BiCRStab	140	23.22	3.01e-010	49	31.38	1.13e-010
Gl-MRBiCRStab	75	12.16	1.14e-010	24	10.49	2.01e-010

Table 2 shows that even if Gl-BiCGStab and Gl-BiCRStab converge well, Gl-MRBiCGStab and Gl-MRBiCRStab may be good competitors. These methods return the best performance with respect to the number of iterations and CPU time.

Example 5.3. We consider the elliptic PDE [9]

$$-u_{xx} - u_{yy} + 2\alpha_1 u_x + 2\alpha_2 u_y - 2\alpha_3 u = 0,$$

on the unit square with Dirichlet boundary conditions, where $\alpha_1 = \alpha_3 = 10$, $\alpha_2 = 20$. Applying the central finite differences on $[0, 1] \times [0, 1]$, with the mesh size $h = \frac{1}{201}$ yields the five-diagonal linear systems.

Table 3: Numerical results of Example 5.3.

	no preconditioning			ILU preconditioning		
Methods	Its	$CPU(s)$	$\|R\|_F$	Its	$CPU(s)$	$\|R\|_F$
Gl-BiCG	712	28.90	2.24e-010	208	29.56	2.49e-010
Gl-BiCGStab	423	35.81	2.25e-010	123	20.39	2.81e-010
Gl-MRBiCGStab	217	22.58	1.29e-010	66	6.89	6.37e-011
Gl-BiCR	678	31.61	2.65e-010	203	20.38	2.35e-010
Gl-BiCRStab	422	34.23	1.24010	115	16.28	2.47e-010
Gl-MRBiCRStab	216	22.40	2.25e-010	57	6.11	2.41e-010

As can be seen from Table 3, this example has the same discussions as Example 5.2.

Example 5.4. Next, we give the two-dimensional convection-diffusion equation [17]

$$-u_{xx} - u_{yy} + 1000(xu_x + yu_y) + 10u = f(x, y),$$

over the unit square $\Omega = (0, 1) \times (0, 1)$ with the Dirichlet boundary conditions. Using 5-point central differences to this equation, a five-diagonal linear system of order 4356 is obtained. The mesh size is chosen as $67(= M + 1)$ in both directions of Ω, so that the resulting system has the $M^2 \times M^2$ coefficient matrix.

Table 4: Numerical results of Example 5.4.

	no preconditioning			ILU preconditioning		
Methods	Its	$CPU(s)$	$\|R\|_F$	Its	$CPU(s)$	$\|R\|_F$
Gl-BiCG	534	1.90	2.18e-010	800	10.53	div.
Gl-BiCGStab	800	3.53	div.	60	0.76	1.6948e-010
Gl-MRBiCGStab	219	1.06	2.21e-010	29	0.25726	8.2333e-011
Gl-BiCR	506	2.01	3.05e-010	800	9.03	div.
Gl-BiCRStab	800	3.54	div.	55	0.70324	2.5463e-010
Gl-MRBiCRStab	204	0.98	2.72e-010	27	0.24263	2.0242e-010

From Figure 2 and Table 4 we observe that Gl-BiCGStab, Gl-BiCRStab, preconditioned Gl-BiCG and preconditioned Gl-BiCR fail to converge. To overcome this problem, Gl-MRBiCGStab and GL-MRBiCRStab are used. These two variants converge faster and give smoother convergence behavior than the other methods.

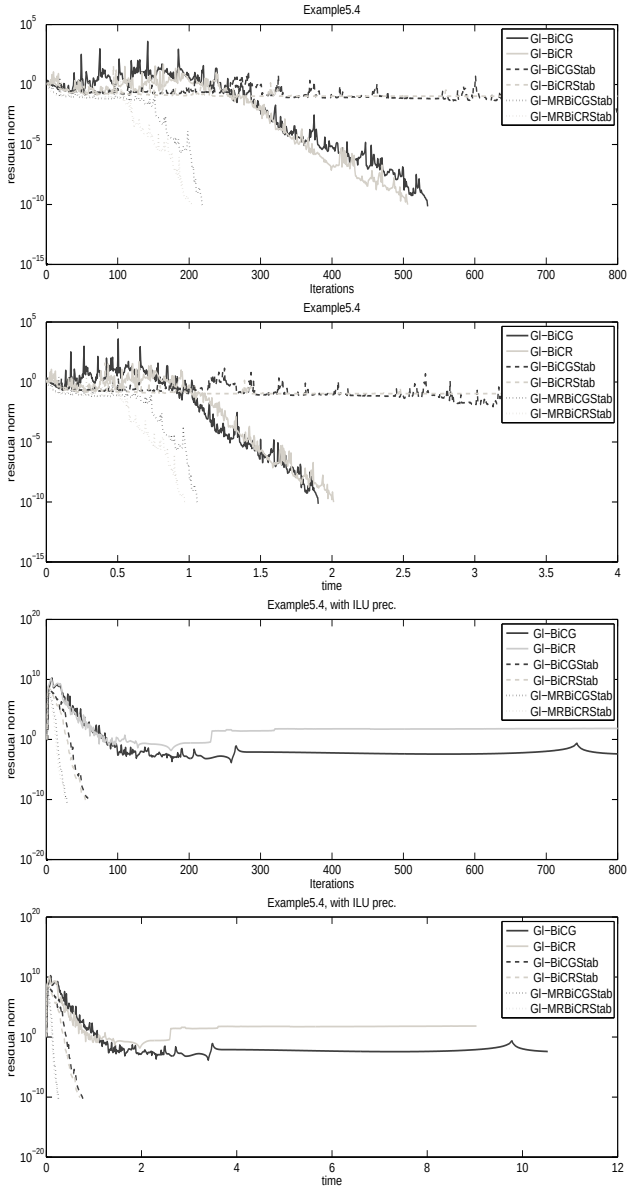
6. Conclusions

In this paper, we investigated improved variants of the Gl-BiCG and Gl-BiCR methods for solving nonsymmetric linear systems with multiple right-hand sides. Firstly, we presented Gl-BiCRStab which converges slightly faster than the Gl-BiCGStab for the problems tested in this paper. Then, we derived Gl-MRBiCGStab and Gl-MRBiCRStab with incorporating quadratic stabilization polynomial into the Gl-BiCGStab and Gl-BiCRStab, respectively. As clearly seen from the numerical examples, Gl-MRBiCGStab and Gl-MRBiCRStab have better convergence behavior than the other proposed algorithms. They are efficient in terms of the number of iterations and CPU-time, also they have the potential to turn divergent iterations into convergent ones. Moreover, when preconditioner is used, the MRBiCGStab and MR-BiCRStab are also effective and cheap. Therefore, we conclude that Gl-MRBiCGStab and Gl-MRBiCRStab as well as their preconditioned versions may be considered as competitive algorithms to solve nonsymmetric linear systems with multiple right-hand sides.

Acknowledgement

The authors thank the anonymous referee and editor-in-chief for their suggestions for revising this manuscript.

FIGURE 2. Convergence plots for Gl-BiCG, Gl-BiCR and their variants. Top: Example 5.4, bottom: Example 5.4 with ILU preconditioning.



REFERENCES

[1] K. Abe and G. L. Sleijpen, *BiCR variants of the hybrid BiCG methods for solving linear systems with nonsymmetric matrices*, J. Comput. Appl. Math., *234* (2010), 985–994.

[2] R. W. Clough and J. Penzien, *Dynamics of Structures*, McGraw-Hill, 1975.
[3] M. Dehghan and M. Hajarian, *The general coupled matrix equations over generalized bisymmetric matrices*, Linear Algebra Appl., *432* (2010), 1531–1552.
[4] R. W. Freund and M. Malhotra, *A block-QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl., *254* (1997), 119–157.
[5] C. Gu and Z. Yang, *Global SCD algorithm for real positive definite linear systems with multiple right-hand sides*, Appl. Math. Comput., *189* (2007), 59–67.
[6] A. E. Guennouni, K. Jbilou, and H. Sadok, *A block version of BiCGSTAB for linear systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., *16* (2003), 129–142.
[7] M. H. Gutknecht, *Variants of BiCGStab for matrices with complex spectrum*, SIAM J. Sci. Comput., *14* (1993), 1020–1033.
[8] M. Heyouni, *The global Hessenberg and global CMRH methods for linear systems with multiple right-hand sides*, Numer. Algo., *26* (2001), 317–332.
[9] D. Y. Hu and L. Reichel, *Krylov subspace methods for the Sylvester equations*, Linear Algebra Appl., *174* (1992), 283–314.
[10] K. Jbilou, A. Messaoudi, and H. Sadok, *Global FOM and GMRES algorithms for matrix equation*, Appl. Numer. Math, *31* (1999), 49–63.
[11] K. Jbilou, H. Sadok, and A. Tinzefte, *Oblique Projection Methods for Linear Sys- tems With Multiple Right-hand Sides*, Electron. Trans. Numer. Anal., *20* (2005), 119–138.
[12] S. K. Li and T. Z. Huang, *LSQR iterative method for generalized coupled Sylvester matrix equations*, Appl. Math. Model., *36* (2012), 3545–3554.
[13] Y. Nakamura, K. Ishikawa, Y. Kuramashi, T. Sakurai, and H. Tadano, *Modified block BiCGSTAB for lattice QCD*, Comput. Phys. Commun., *183* (2012), 34–37.
[14] D. P. O'Leary, *The Block Conjugate Gradient Algorithm and Related Methods*, Linear Algebra Appl., *29* (1980), 293–322.
[15] S. Rashedi, G. Ebadi, S. Birk, and A. Frommer, *On short recurrence krylov type methods for linear systems with many right-hand sides*, J. Comput. Appl. Math., *300* (2016), 18–29.
[16] S. Rashedi, G. Ebadi, and A. Biswas, *The maximal and minimal ranks of a quaternion matrix expression with applications*, J. Egypt. Math. Soc., *21* (2013), 175–183.
[17] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Stat. Comput., *14* (1993), 461–469.
[18] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
[19] T. Sakurai, H. Tadano, and Y. Kuramashi, *Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD*, Comput. Phys. Comm., *181* (2010), 113–117.
[20] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Statist. Comput., *16* (1995), 917–933.
[21] G. L. Sleijpen and D. R. Fokkema, *BICGSTAB(l) for linear equations involving unsymmetric matrices with complex spectrum*, Electron. Trans. Numer. Anal., *1* (1993), 11–32.
[22] J. Virieux and S. Operto, *An overview of full waveform inversion in exploration geophysics*, Geophysics, *74* (2009), 127–152.
[23] J. H. Yun and M. S. Joo, *BCG like methods for solving nonsymmetric linear systems*, Journal of The Chungcheong Mathematical Society, *8* (1995), 55–69.
[24] J. Zhang and H. Dai, *Global CGS algorithm for linear systems with multiple right-hand sides*, Numer. Math. J. Chin. Univ., *30* (2008), 390–399.
[25] J. Zhang, H. Dai, and J. Zhao, *Generalized global conjugate gradient squared algorithm*, Appl. Math. Comput., *216* (2010), 3694–3706.
[26] J. Zhang, H. Dai, and J. Zhao, *A new family of global methods for linear systems with multiple right-hand sides*, J. Comput. Appl. Math., *236* (2011),1562–1575.
[27] J. Zhang and J. Zhao, *A new variant of BiCRSTAB method for solving nonsymmetric indefinite linear systems*, J. Inf. Comput. Sci., *8* (2011), 2183–2195.