Scholars Journal of Physics, Mathematics and Statistics OPEN ACCESS
Abbreviated Key Title: Sch J Phys Math Stat
ISSN 2393-8056 (Print) | ISSN 2393-8064 (Online)
Journal homepage: https://saspublishers.com

Shifted Global BiCG-Type Methods for Solving the Large Stein Equation X + AXB = C

Ting Zhang, Sheng-Kun Li, Gang Liu

College of Applied Mathematics, Chengdu University of Information Technology, Chengdu, Sichuan, 610225, P. R. China

DOI: 10.36347/sjpms.2022.v09i09.001 | **Received:** 07.11.2022 | **Accepted:** 15.12.2022 | **Published:** 19.12.2022

***Corresponding author:** Sheng-Kun Li
College of Applied Mathematics, Chengdu University of Information Technology, Chengdu, Sichuan, 610225, P. R. China

Abstract **Original Research Article**

The Stein matrix equation plays an essential role in control and communications theory, linear algebra, image restoration, and so on. In this paper, we propose two shifted variants of global BiCG-type methods to solve the large Stein matrix equation which make full use of the shifted structure of the matrix equation. These modifications will not add more matrix-matrix multiplications and inner products. Finally, numerical examples are given to illustrate the effectiveness of the presented methods.

Keywords: Shifted system, BiCG-type method, Krylov subspace method, Stein matrix equation.

1 INTRODUCTION

In this paper, we consider the numerical solution of the large Stein matrix equation of the form:

$$AXB + X = C, \quad (1)$$

Where $A \in R^{n\times n}$, $B \in R^{s\times s}$ and $C \in R^{n\times s}$. The Stein matrix equation has a unique solution if and only if

$$\lambda u \neq 1, \forall \lambda \in \sigma(A), \forall u \in \sigma(B),$$

Where $\sigma(A)$ and $\sigma(B)$ are the sets of all the eigenvalues of matrices A and B, respectively. This paper only considers the case that the Stein matrix equation has a unique solution. In fact, the Stein matrix equation often arises in numerical methods for ordinary differential equations [1], Probability [2], model reduction problems [3], neural network [4] and image restoration [5, 6], and so on. This matrix equation is also a calculation tool for control system design [7, 8].

There are different methods for different types of the Stein matrix equation. When the matrices A and B are of small size orders, the direct methods are attractive, such as Hessenberg-Schur method [9] and Bartels-Stewart method [10]. This type of methods is based on matrix decomposition. When the orders of A and B are both large, Galerkin algorithm and minimal residual algorithm [11] which are based on Arnoldi process are proposed. When A is large and B is of small size, the block Arnoldi and the block Lanczos algorithms [12] have been investigated. For the large Stein equation with low rank right hand sides, the global Arnoldi method [13, 14], the block Arnoldi method [15] and the block Hessenberg method [16] are developed to compute the low rank solution. In addition, the extended block Krylov subspace methods [17, 18] are successively used to solve this type of matrix equation.

This paper mainly deals with matrix equation (1) with large A and B. In order to make full use of the shifted structure, we develop two shifted variants of global BiCG-type methods. This shifted idea was first proposed by Frommer and Gl¨assner [19] for solving $\sigma x + Ax = b$. In the sequel, the shifted variants of short term iterative methods have been studied in [20, 21].

To derive our methods, we call the matrix equation (1) a shifted system and the matrix equation:

$$AXB = C, \quad (2)$$

A seed system for convenience, let Λ be the linear operator defined as follows:

$$\Lambda : R^{n\times s} \to R^{n\times s}, \quad X \to \Lambda X = AXB$$

Correspondingly,

$$\Lambda^T : R^{n\times s} \to R^{n\times s}$$
$$X \to \Lambda^T X = A^T X B^T$$

Then the shifted system (1) can be written as:

$$X + \Lambda X = C, \quad (3)$$

And the seed system (2) can be written as

$$\Lambda X = C, \quad (4)$$

For a given matrix $V \in R^{n\times s}$, we define the matrix Krylov subspace as follows

$$\mathrm{K}_m(\Lambda, V) = span\{V, \Lambda V, \cdots, \Lambda^{m-1} V\}, \quad (5)$$

Where $\Lambda^i V$ is defined recursively as $\Lambda\left(\Lambda^{i-1} V\right)$. Because of the well-known shift-invariance property of Krylov subspaces, we have:

$$\mathrm{K}_m(\Lambda, V) = \mathrm{K}_m(\Lambda + I, V).$$

The following symbols will be used latter. We define the inner product $\langle K, M\rangle = tr\left(K^T M\right)$ where K^T is the transpose of matrix K. The $\|K\|_F = \sqrt{\langle K, K\rangle}$ represents the Frobenius norm. I_m represents the identity matrix of m rows and m columns.

The rest of this paper is organized as follows. In Section 2, we shortly review the global Bi-Conjugate Gradient (GBiCG) method and the global Bi-Conjugate Gradient Stabilized (GBiCGStab) method for solving the seed system (4). Section 3 and Section 4 deal with the shifted variants of GBiCG method and GBiCGStab method for solving the shifted linear systems (3). Some numerical examples are given to illustrate the efficiency of the proposed methods in Section 5. Finally, the paper is ended with a brief conclusion in Section 6.

2 The Global BiCG-Type Method

In this section, we briefly review the GBiCG method and GBiCGStab method for solving the seed system (4).

The BiCG method [22] is powerful for the solution of nonsymmetric linear system. A disadvantage of the BiCG method is that it needs to calculate A^T per iteration. To void computing the A^T, Van der Vorst [23] proposed a smoother convergence method, known as BiCGStab method. In [24], Jbilou *et al.,* proposed the GBiCG method and the GBiCGStab method for solving $AX = B$. This kind of global methods is based on oblique projections onto a matrix Krylov subspace. Just replace each operator A or A^T with a new Λ or Λ^T in the methods of Jbilou *et al.,* we can use the GBiCG method and the GBiCGStab method to solve the seed system (4). Thus, the corresponding pseudo-codes of the GBiCG and GBiCGStab methods can be given as follows, respectively.

Algorithm 1: The GBiCG method for the seed system (4)

1.	X_0 is an initial guess, $R_0 = C - \Lambda X_0$
2.	Choose $\tilde{R}_0$ (for example $\tilde{R}_0 = R_0$)
3.	Set $\rho_{-1} = 1, U_0 = R_0, \tilde{U}_{-1} = U_{-1} = 0$
4.	For $k = 0,1,2,\ldots,$ until convergence, do
5.	$\rho_k = \langle R_k, \tilde{R}_k\rangle, \beta_k = -\dfrac{\rho_k}{\rho_{k-1}}$
6.	$U_k = R_k - \beta_k U_{k-1}$
7.	$\tilde{U}_k = \tilde{R}_k - \beta_k \tilde{U}_{k-1}$
8.	$\gamma_k = \langle \Lambda U_k, \tilde{U}_k\rangle, \alpha_k = \dfrac{\rho_k}{\gamma_k}$
9.	$X_{k+1} = X_k + \alpha_k U_k$
10	$R_{k+1} = R_k - \alpha_k \Lambda U_k$
11.	$\tilde{R}_{k+1} = \tilde{R}_k - \alpha_k \Lambda^T \tilde{U}_k$
12.	EndDo

Algorithm 2: The Global Bi-Conjugate Gradient Stabilized (GBICGStab) Method

1.	X_0 is an initial guess, $R_0 = C - \Lambda X_0$
2.	Choose $\tilde{R}_0$ (for example $\tilde{R}_0 = R_0$)
3.	Set $\rho_{-1} = 1, \alpha_{-1} = 1, U_{-1} = 0$, $w_{-1} = 1 V_{-1} = 0$
4.	For $k = 0,1,2,\ldots,$ until convergence
5.	$\rho_k = \langle R_k, \tilde{R}_0 \rangle, \beta_{k,b} = -\frac{\alpha_{k-1}}{w_{k-1}} \frac{\rho_k}{\rho_{k-1}}$
6.	$U_k = R_k - \beta_k (U_{k-1} - w_{k-1} V_{k-1})$
7.	$V_k = \Lambda U_k$
8.	$\alpha_k = \frac{\rho_k}{\langle V_k, \tilde{R}_0 \rangle}$
9.	$S_k = R_k - \alpha_k V_k$
10.	$P_k = \Lambda S_k$
11.	$w_k = \frac{\langle S_k, P_k \rangle}{\langle P_k, P_k \rangle}$
12.	$X_{k+1} = X_k + \alpha_k U_k + w_k S_k$
13.	$R_{k+1} = S_k - w_k P_k$
14.	EndDo

3 The Shifted Global BiCG Method

In this section, we describe the process to solve the shifted system $(\Lambda + I)X = C$ by using the information obtained from the GBiCG method for the seed system $\Lambda X = C$.

Let $X_m^s \in \mathrm{K}_m(\Lambda + I, C)$ be the approximate solution of the $m-th$ iterate of the shifted system and $X_m \in \mathrm{K}_m(\Lambda, C)$ be the approximate solution of the $m-th$ iterate of the seed system with the initial guesses $X_0^s = 0_{n\times s}$ and $X_0 = 0_{n\times s}$. Then we can get the residual of the $m-th$ iterate of the seed system $R_m = C - \Lambda X_m$ which is from $\mathrm{K}_{m+1}(\Lambda, C)$ and $R_0 = C$. Note that:

$$X_m \in \mathrm{K}_m(\Lambda, C) \Leftrightarrow X_m = \sum_{i=0}^{m-1} k_i \Lambda^i C;\ k_i \in R, i = 0,\cdots,m-1.$$

If we express:

$X_m = T_{m-1}(\Lambda)C, T_{m-1} \in \Pi_{m-1}$ (polynomials of degree $\le m-1$)

Then

$R_m = C - \Lambda X_m = D_m(\Lambda)C, D_m \in \Pi_m$ (polynomials of degree $\le m-1$)

Similarly, $R_m^s = C - (\Lambda + I)X_m^s$, where $R_m^s \in \mathrm{K}_m(\Lambda + I, C)$ and $R_0^s = C$. And we also can have:

$$X_m^s \in \mathrm{K}_m(\Lambda + I, C) \Leftrightarrow X_m = \sum_{i=0}^{m-1} k_i^s (\Lambda + I)^i C;\ k_i^s \in R, i = 0,\cdots,m-1.$$

So we can get:

$$X_m^s \in \mathrm{K}_m(\Lambda + I, C) \Leftrightarrow X_m = \sum_{i=0}^{m-1} k_i^s (\Lambda + I)^i C;\ k_i^s \in R, i = 0,\cdots,m-1.$$

$$X_m^s = T_{m-1}^s(\Lambda + I)C, T_{m-1}^s \in \Pi_{m-1},$$

Then

$$R_m^s = C - (\Lambda + I)X_m^s = D_m^s(\Lambda + I)C, D_m^s \in \Pi_m.$$

Hence, using the relationship between X_m^s and R_m^s, we have following results,

$$D_m^s(\varepsilon + 1) = 1 - (\varepsilon + 1)T_m^s(\varepsilon + 1),$$

Where the sign ε replaces operator Λ. Then let $\varepsilon = -1$, we obtain that

$$D_m^s(0) = 1.$$

After m iterations, X_m of the seed system and X_m^s of the shifted system belong to the same Krylov subspace $\mathrm{K}_m(\Lambda, C)$, and they have similar form. We

know that the seed system and the shifted system have collinear residuals, so we assume a collinear factor π_m^s and let:

$$R_m^s = \left(\frac{1}{\pi_m^s}\right) R_m, \pi_m^s \in R, \tag{6}$$

Which can be converted to;

$$D_m^s(\Lambda + I)C = \left(\frac{1}{\pi_m^s}\right) D_m(\Lambda)C,$$

$$D_m^s(\varepsilon + 1) = \left(\frac{1}{\pi_m^s}\right) D_m(\varepsilon), \tag{7}$$

Since $D_m^s(0) = 1$, we can get the equation of the collinear factor:

$$\pi_m^s = D_m(-1). \tag{8}$$

According Algorithm 1, we can obtain following equations:

$$U_{k-1} = \frac{1}{\beta_k}(R_k - U_k),$$

$$\Lambda U_k = \frac{1}{\alpha_k}(R_k - R_{k+1}),$$

Then R_k can be written as

$$\begin{aligned} R_k &= R_{k-1} - \alpha_{k-1}\Lambda U_{k-1} \\ &= R_{k-1} - \frac{\alpha_{k-1}}{\beta_k}\Lambda(R_k - U_k) \\ &= R_{k-1} - \frac{\alpha_{k-1}}{\beta_k}\Lambda R_k + \frac{\alpha_{k-1}}{\beta_k}\Lambda U_k \\ &= R_{k-1} - \frac{\alpha_{k-1}}{\beta_k}\Lambda R_k + \frac{\alpha_{k-1}}{\alpha_k\beta_k}(R_k - R_{k+1}) \end{aligned},$$

And thus:

$$R_{k+1} = -\alpha_k \Lambda R_k + \left(1 - \frac{\alpha_k\beta_k}{\alpha_{k-1}}\right) R_k + \frac{\alpha_k\beta_k}{\alpha_{k-1}} R_{k-1}. \tag{9}$$

Finally, we get the recurrence relation of the residuals, with $R_m = D_m(\Lambda)C$, the equation (9) becomes;

$$D_{k+1}(\varepsilon) = -\alpha_k \varepsilon D_k(\varepsilon) + \left(1 - \frac{\alpha_k\beta_k}{\alpha_{k-1}}\right) D_k(\varepsilon) + \frac{\alpha_k\beta_k}{\alpha_{k-1}} D_{k-1}(\varepsilon). \tag{10}$$

With $\pi_m^s = D_m(-1)$ and $\varepsilon = -1$, the equation (10) can be expressed as;

$$\pi_{k+1}^s = (1 + \alpha_k)\pi_k^s + \frac{\alpha_k\beta_k}{\alpha_{k-1}}(\pi_{k-1}^s - \pi_k^s), \tag{11}$$

Where $\pi_{-1}^s = \pi_0^s = 1$. This is the three-term recurrence relation for the collinearity factor. On the other hand, from the relations (6) and (9), the three-term recurrence for the residuals of the shifted system can be reached and its formula is expressed as follows:

$$\begin{aligned} R_{k+1}^s &= \frac{1}{\pi_{k+1}^s} R_{k+1} \\ &= \frac{1}{\pi_{k+1}^s}\left[-\alpha_k \Lambda R_k + \left(1 - \frac{\alpha_k\beta_k}{\alpha_{k-1}}\right) R_k + \frac{\alpha_k\beta_k}{\alpha_{k-1}} R_{k-1}\right] \\ &= -\frac{\pi_k^s \alpha_k}{\pi_{k+1}^s}(\Lambda + I) R_k^s + \left(\frac{\pi_k^s}{\pi_{k+1}^s} - \frac{\pi_k^s \alpha_k \beta_k}{\pi_{k+1}^s \alpha_{k-1}} + \frac{\pi_k^s \alpha_k}{\pi_{k+1}^s}\right) R_k^s + \frac{\pi_{k-1}^s \alpha_k \beta_k}{\pi_{k+1}^s \alpha_{k-1}} R_{k-1}^s \end{aligned},$$

Let

$$\alpha_k^s = \frac{\pi_k^s}{\pi_{k+1}^s}\alpha_k, \tag{12}$$

$$\beta_k^s = \left(\frac{\pi_{k-1}^s}{\pi_k^s}\right)^2 \beta_k, \tag{13}$$

So, the residuals of the shifted system which has the resemble equation expression as the residuals of the seed system can be written as:

$$R_{k+1}^s = -\alpha_k^s(\Lambda + I) R_k^s + \left(1 - \frac{\alpha_k^s \beta_k^s}{\alpha_{k-1}^s}\right) R_k^s + \frac{\alpha_k^s \beta_k^s}{\alpha_{k-1}^s} R_{k-1}^s, \tag{14}$$

Now, according to the equation (14) and $U_{k-1}^s = \frac{(\Lambda + I)^{-1}(R_{k-1}^s - R_k^s)}{\alpha_{k-1}^s}$, we have;

$$R_k^s = R_{k-1}^s - \alpha_{k-1}^s(\Lambda + I) U_{k-1}^s, \tag{15}$$

And the search direction of the shifted system is updated to;

$$U_k^s = R_k^s - \beta_k^s U_{k-1}^s. \tag{16}$$

Since the relation (6), the equation (16) becomes;

$$U_k^s = \frac{1}{\pi_k^s} R_k - \beta_k^s U_{k-1}^s, \tag{17}$$

Where $U_{-1}^s = 0$. Then it follows from the relations $R_k^s = C - (\Lambda + I) X_k^s$ and relation (15) that;

$$X_{k+1}^s = X_k^s + \alpha_k^s U_k^s, \tag{18}$$

From Algorithm 1 and the above derivation, the shifted GBiCG (SGBiCG) method is summarized as follows.

Table 1 : Summary of operations per iteration step, where $AXPY : \alpha X + Y$, IP: Inner product, ΛX : Operator Λ acts on X

Name	AXPY	IP	ΛX
GBICG	7	2	2
SGBICG	7	2	2

Algorithm 3: The Shifted Global Bi-Conjugate Gradient (SGBICG) Method

1.	$X_0 = 0,\ X_0^s = X_0$
2.	$R_0 = C,\ \tilde{R}_0 = R_0$
3.	Set $\rho_{-1} = \rho_{-1}^s = 1,\ \tilde{U}_{-1} = U_{-1} = U_{-1}^s = 0$
4.	$\alpha_{-1} = 1,\ \pi_{-1}^s = \pi_0^s = 1$
5.	For $k = 0,1,2,\ldots,$ until convergence, do
6.	{seed system}
7.	$\rho_k = \langle R_k, \tilde{R}_k \rangle, \beta_k = -\dfrac{\rho_k}{\rho_{k-1}}$
8.	$U_k = R_k - \beta_k U_{k-1}$
9.	$\tilde{U}_k = \tilde{R}_k - \beta_k \tilde{U}_{k-1}$
10.	$Z_k = \Lambda U_k, \alpha_k = \dfrac{\rho_k}{\langle Z_k, \tilde{U}_k \rangle}$
11.	$X_{k+1} = X_k + \alpha_k U_k$
12.	{shifted system}
13.	$\pi_{k+1}^s = (1+\alpha_k)\pi_k^s + \dfrac{\alpha_k \beta_k}{\alpha_{k-1}}(\pi_{k-1}^s - \pi_k^s)$
14.	$\alpha_k^s = \dfrac{\pi_k^s}{\pi_{k+1}^s}\alpha_k$
15.	$\beta_k^s = \left(\dfrac{\pi_{k-1}^s}{\pi_k^s}\right)^2 \beta_k$
16.	$U_k^s = \dfrac{1}{\pi_k^s} R_k - \beta_k^s U_{k-1}^s$
17.	$X_{k+1}^s = X_k^s + \alpha_k^s U_k^s$
18.	{updating}
19.	$R_{k+1} = R_k - \alpha_k Z_k$
20.	$\tilde{R}_{k+1} = \tilde{R}_k - \alpha_k \Lambda^H \tilde{U}_k$
21.	$R_{k+1}^s = \left(\dfrac{1}{\pi_{k+1}^s}\right) R_{k+1}$
22.	EndDo

Observing Algorithm 3, we can notice that there is a great difference between the GBiCG method and the SGBiCG method for solving the shifted system (3). The computational cost of the GBiCG and SGBiCG methods at each iteration step is shown in Table 1. As seen from Table 1, the GBiCG method and the SGBiCG method need the same number of operations per iteration.

4 The Shifted Global BiCGStab Method

In this section, we drive the shifted GBiCGStab (SGBiCGStab) method for solving the shifted system (3) via applying the information sent from GBiCGStab method for the seed system (4). It is known to all that the GBiCGStab method combines the GBiCG method and the minimal residual (MR) idea, so naturally the SGBiCGStab method can be considered as a combination of SGBiCG method and the MR idea. Referring to the aforementioned idea, we can deduce the SGBiCGStab method for solving the shifted system (3).

According to GBiCGStab method, we can easily know the form of residue of seed system;

$$R_k = Q_k(\Lambda) D_k(\Lambda) R_0, \tag{19}$$

Where D_k is the residual polynomial of GBiCG method and Q_k is a new k-degree polynomial, which is used to correct the oscillation phenomenon of residual norm, and;

$$Q_{k+1}(\lambda) = (1 - w_k \lambda) Q_k(\lambda), \tag{20}$$

Where $Q_0(\lambda) = 1$ and w_k is a number used to minimize the residual. Then, we can get;

$$Q_{k+1}(\lambda) = \prod_{i=1}^{k} (1 - w_i \lambda). \tag{21}$$

From Algorithm 2, the formula about R_{k+1} and S_k is as follow:

$$R_{k+1} = S_k - w_k \Lambda S_k. \tag{22}$$

In addition, from the relation (20), we have;

$$Q_{k+1}(\Lambda) = (I - w_k \Lambda) Q_k(\Lambda). \tag{23}$$

Then substituting (19), (23) into (22), we get the following formulas;

$$\begin{aligned} &Q_{k+1}(\Lambda) D_{k+1}(\Lambda) R_0 = (I - W_k \Lambda) S_k \\ &(I - W_k \Lambda) Q_k(\Lambda) D_{k+1}(\Lambda) R_0 = (I - W_k \Lambda) S_k \end{aligned},$$

Later, we can obtain the equation of S_k:

$$S_k = Q_k(\Lambda) D_{k+1}(\Lambda) R_0. \tag{24}$$

Referring to Section 3, here R_{k+1}^s and S_k^s satisfy the following identities;

$$\begin{aligned} R_{k+1}^s &= Q_{k+1}^s(\Lambda + I) D_{k+1}^s(\Lambda + I) R_0, \\ S_k^s &= Q_k^s(\Lambda + I) D_{k+1}^s(\Lambda + I) R_0. \end{aligned}$$

According to the equation (7), R_{k+1}^s and S_k^s can also be written as follows:

$$R_{k+1}^s = Q_{k+1}^s(\Lambda + I) \frac{1}{\pi_{k+1}^s} D_{k+1}(\Lambda) R_0, \tag{25}$$

$$S_k^s = Q_k^s(\Lambda + I) \frac{1}{\pi_{k+1}^s} D_{k+1}(\Lambda) R_0. \tag{26}$$

Firstly, we discuss the part of shifted MR method. In order to obtain the computational formulas for the shifted MR polynomial, we consider a linear factor;

$$(1 - (\lambda + 1) w_k^s) = c(1 - w_k \lambda),$$

Which gives

$$\begin{aligned} c &= \frac{1}{1 + w_k}, \\ w_k^s &= \frac{w_k}{1 + w_k}. \end{aligned} \tag{27}$$

Since $Q_{k+1}(\lambda) = \prod_{i=1}^{k} (1 - w_i \lambda)$, we can write the matrix polynomial of the corresponding shifted system;

$$Q_{k+1}^s(\lambda + 1) = \prod_{i=1}^{k} (1 - (\lambda + 1) w_i^s). \tag{28}$$

Substituting the equation (27) into (28), we can get;

$$\begin{aligned} Q_k^s(\lambda + 1) &= \prod_{i=1}^{k-1} \left(1 - (\lambda + 1) \frac{w_i}{1 + w_i}\right), \\ &= \prod_{i=1}^{k-1} \frac{1}{1 + w_i} (1 - \lambda w_i) \end{aligned}$$

And $Q_k^s(\lambda + 1)$ can be written as;

$$Q_k^s(\lambda + 1) = d_k^s Q_k(\lambda). \tag{29}$$

Where $d_k^s = \prod_{i=1}^{k-1} \frac{1}{1 + w_i}$ and $d_0^s = 1$. Observing the equation of d_k^s, the expression of d_{k+1}^s is as follows:

$$d_{k+1}^s = \frac{d_k^s}{1 + w_k}. \tag{30}$$

Now, we consider the shifted GBiCG part. According to the derivation in Section 3, we can easily get the equations of π_{k+1}^s, α_k^s and β_k^s, where;

$$\pi_{k+1}^s = (1 + \alpha_k) \pi_k^s + \frac{\alpha_k \beta_k}{\alpha_{k-1}} (\pi_{k-1}^s - \pi_k^s),$$

$$\alpha_k^s = \frac{\pi_k^s}{\pi_{k+1}^s}\alpha_k,$$

$$\beta_k^s = \left(\frac{\pi_{k-1}^s}{\pi_k^s}\right)^2 \beta_k.$$

Correspondingly, in the light of the equation (29), we obtain

$$Q_k^s\left(\Lambda + I\right) = d_k^s Q_k\left(\Lambda\right). \tag{31}$$

Hence, we substitute the equation (31) into (25) and (26), resulting in

$$S_k^s = \frac{d_k^s}{\pi_{k+1}^s} S_k, \tag{32}$$

$$R_{k+1}^s = \frac{d_{k+1}^s}{\pi_{k+1}^s} R_{k+1}. \tag{33}$$

Following the previous derivation process, we readily have

$$U_k^s = R_k^s - \beta_k^s\left(U_{k-1}^s - w_{k-1}^s\left(\Lambda + I\right)U_{k-1}^s\right), \tag{34}$$

$$S_k^s = R_k^s - \alpha_k^s\left(\Lambda + I\right)U_k^s. \tag{35}$$

Let $V_k^s = \left(\Lambda + I\right)U_k^s$, , and substituting it into (34) and (35), it follows that,

$$V_k^s = \frac{1}{\alpha_k^s}\left(R_k^s - S_k^s\right). \tag{36}$$

From Algorithm 2 and the above derivation, we can get the following Algorithm 4.

Algorithm 4: The SGBiCGStab method for the shifted system (3)

1.	$X_0 = 0,\ X_0^s = X_0$
2.	$R_0 = C,\ \tilde{R}_0 = R_0$
3.	Set $\rho_{-1} = 1, \alpha_{-1} = 1,\ w_{-1} = 1$
4.	$V_{-1} = V_{-1}^s = 0,\ U_{-1} = U_{-1}^s = 0$
5.	$\pi_{-1}^s = \pi_0^s = d_0^s = 1$
6.	For $k = 0,1,2,\ldots,$ until convergence, do
7.	{Seed System}
8.	$\rho_k = \left\langle R_k, \tilde{R}_0\right\rangle, \beta_k = -\frac{\alpha_{k-1}}{w_{k-1}}\frac{\rho_k}{\rho_{k-1}}$
9.	$U_k = R_k - \beta_k\left(U_{k-1} - w_{k-1}V_{k-1}\right)$
10.	$V_k = \Lambda U_k$
11.	$\alpha_k = \frac{\rho_k}{\left\langle V_k, \tilde{R}_0\right\rangle}$
12.	$P_k = \Lambda S_k$
13.	$S_k = R_k - \alpha_k V_k$
14.	$w_k = \frac{\left\langle S_k, P_k\right\rangle}{\left\langle P_k, P_k\right\rangle}$
15.	$X_{k+1} = X_k + \alpha_k U_k + w_k S_k$
16.	$R_{k+1} = S_k - w_k P_k$
17.	{Shifted System}
18.	$\pi_{k+1}^s = \left(1 + \alpha_k\right)\pi_k^s + \frac{\alpha_k\beta_k}{\alpha_{k-1}}\left(\pi_{k-1}^s - \pi_k^s\right)$
19.	$\alpha_k^s = \frac{\pi_k^s}{\pi_{k+1}^s}\alpha_k$

20.	$\beta_k^s = \left(\frac{\pi_{k-1}^s}{\pi_k^s}\right)^2 \beta_k$
21.	$w_k^s = \frac{w_k}{1+w_k}$
22.	$S_k^s = \frac{d_k^s}{\pi_{k+1}^s} S_k$
23.	$U_k^s = R_k^s - \beta_k^s \left(U_{k-1}^s - w_{k-1}^s V_{k-1}^s\right)$
24.	$X_{k+1}^s = X_k^s + \alpha_k^s U_k^s + w_k^s S_k^s$
25.	{Updating}
26.	$d_{k+1}^s = \frac{d_k^s}{1+w_k}$
27.	$V_k^s = \frac{R_k^s - S_k^s}{\alpha_k^s}$
28.	$R_{k+1}^s = \frac{d_{k+1}^s}{\pi_{k+1}^s} R_{k+1}$
29.	Enddo

Table 2 : Summary of operations per iteration step, where $AXPY : \alpha X + Y$ **, IP: Inner product,** ΛX **: Operator** Λ **acts on** X

Name	AXPY	IP	ΛX
GBICGStab	7	4	2
SGBICGStab	8	4	2

At the end of this section, the computational cost per iteration step for the GBiCGStab and SGBiCGStab methods is given in Table 2. In Table 2, we find that the computational cost per iteration step of the GBiCGStab and SGBiCGStab methods is almost the same in solving the shifted system (3).

5 NUMERICAL EXPERIMENTS

In this section, two numerical examples are presented. In order to illustrate the effectiveness of the shifted global BiCG-type methods, we compare the SGBiCG method and the SGBiCGStab method with the GBiCG method and the GBiCGStab method for solving the stein matrix equation (1), respectively. We evaluate the proposed methods with aspects of the number of iterations (Its) and computational time in seconds (CPU). All numerical calculations are performed on Window 10 (64bit), and MATLAB 2016a running on a laptop with an Intel Core i7-6700HQ 2.60 GHz CPU and 8.00 GB memory. In all runs, the initial guess is zero matrix with suitable size and the matrix C satisfies that the exact solution is $X = \left(x_{ij}\right)_{n\times s}$ with $x_{ij} = 1$. The condition for stopping the iteration is $\frac{\|R_{k+1}\|_F}{\|R_0\|_F} < 10^{-10}$.

Example 4.1 for the first experiment, we compare the SGBiCG method with the GBiCG method for the stein matrix equation $AXB + X = C$ where;

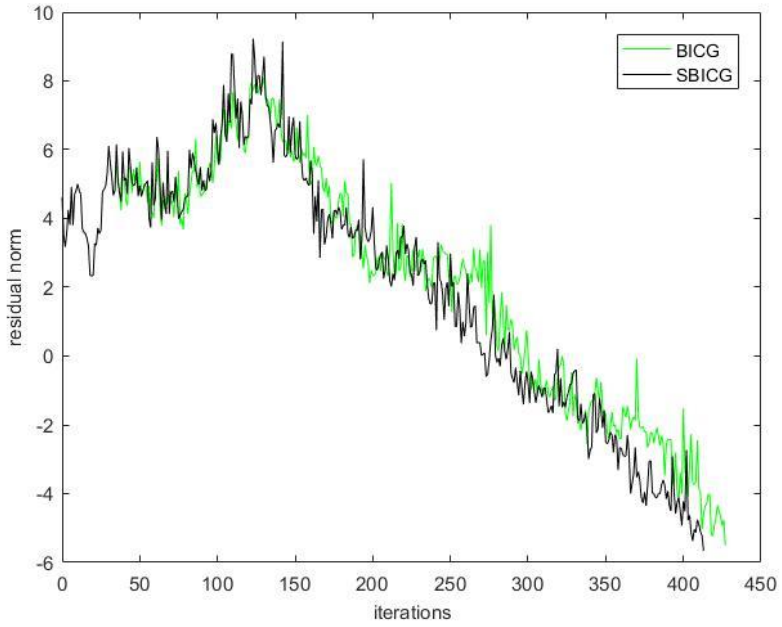
$$A = \begin{pmatrix} D_A & 0 & 0 & \cdots & 0 \\ I_u & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & I_u & D_A \end{pmatrix}_{n\times n} \in R^{u^2\times u^2},$$

$$D_A = \begin{pmatrix} 50 & 11 & 0 & \cdots & 0 \\ 5.9 & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 50 & 11 \\ 0 & 0 & \cdots & 5.9 & -3.9 \end{pmatrix}_{u\times u} \in R^{u\times u},$$

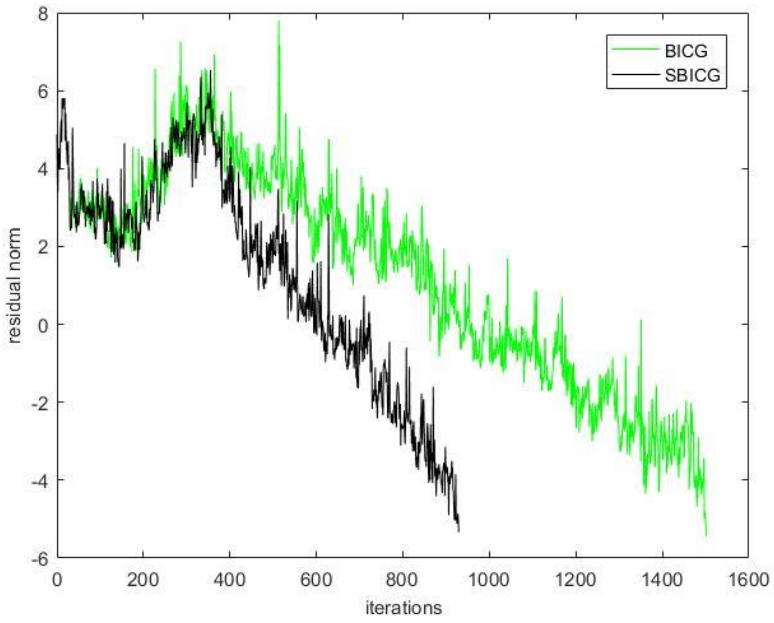
$$B = \begin{pmatrix} 4 & 3 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 3 \\ 0 & 0 & \cdots & 0 & 4 \end{pmatrix}_{s\times s} \in R^{s\times s}.$$

Table 3：Numerical results of Example 4.1

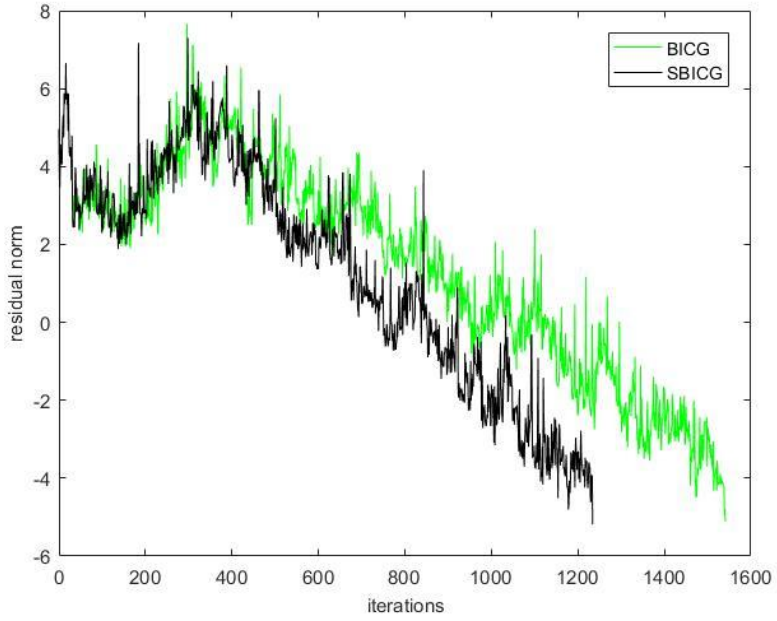
Matrix order	Algorithm	Its	CPU
$n=u^2=400$ $s=25$	BICG	427	0.081654
	SBICG	413	0.124118
$n=u^2=900$ $s=30$	BICG	1501	0.886542
	SBICG	929	0.883460
$n=u^2=1225$ $s=32$	BICG	1541	1.216110
	SBICG	1234	1.740855



$(a)\, n=u^2=400, s=55$



$(b)\, n=u^2=625, s=70$



$(c)\, n = u^2 = 1089, s = 74$

Figure 1：The convergence history of Example 4.1

The numerical results are given in Table 3 and Figure 1. From Table 3, it can be seen that, in most cases, the SGBiCG method works more efficiently than the GBiCG method in terms of iteration steps and CPU time. As can be seen from Figure 1, the relative residual norms of the two methods show the large oscillation phenomenon.

Example 4.2 in this experiment, we compare the SGBiCGStab method with the GBiCGStab method for the stein matrix equation $AXB + X = C$ where;

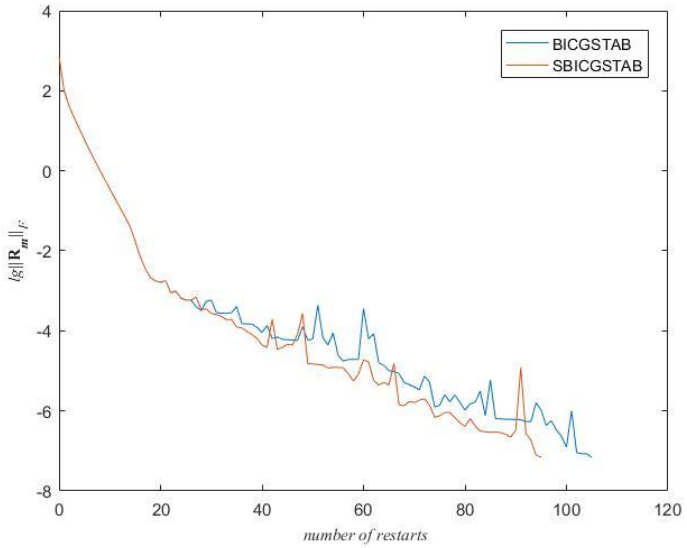
$$A = \begin{pmatrix} D_A & -I_u & 0 & \cdots & 0 \\ -I_u & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -I_u \\ 0 & 0 & \cdots & -I_u & D_A \end{pmatrix}_{n\times n} \in R^{u^2\times u^2},$$

$$D_A = \begin{pmatrix} 13 & 4 & 0 & \cdots & 0 \\ 1 & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 13 & 4 \\ 0 & 0 & \cdots & 1 & -3.9 \end{pmatrix}_{u\times u} \in R^{u\times u},$$

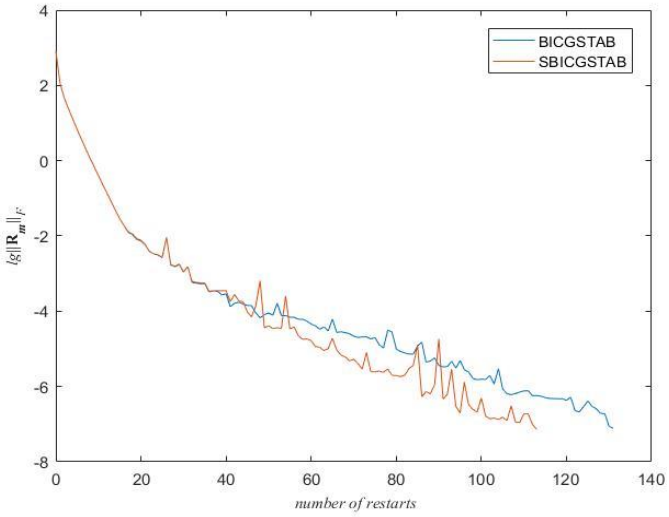
$$B = \begin{pmatrix} 8 & 3 & 0 & \cdots & 0 \\ 3 & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 3 \\ 0 & 0 & \cdots & 3 & 8 \end{pmatrix}_{s\times s} \in R^{s\times s},$$

Table 4：Numerical results of Example 4.2

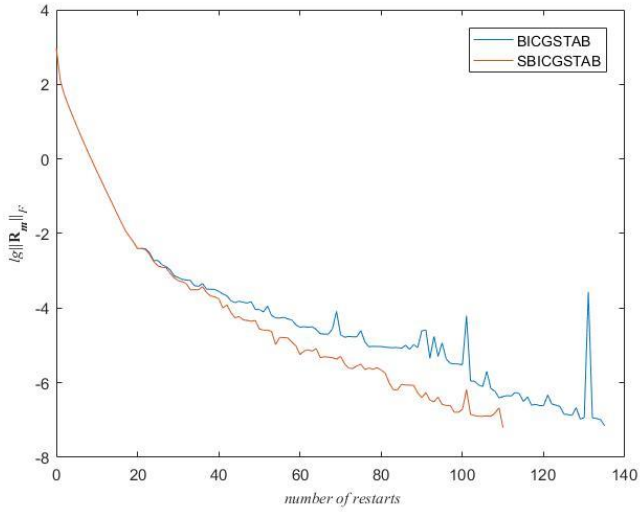
Matrix order	Algorithm	Its	CPU
$n = u^2 = 1225$ $s = 25$	BICGStab	105	0.234184
	SBICGStab	95	0.277105
$n = u^2 = 1521$ $s = 31$	BICGStab	131	0.410258
	SBICGStab	113	0.420095
$n = u^2 = 2401$ $s = 40$	BICGStab	135	0.932351
	SBICGStab	110	0.929066
$n = u^2 = 6889$ $s = 75$	BICGStab	178	10.482451
	SBICGStab	154	10.341231



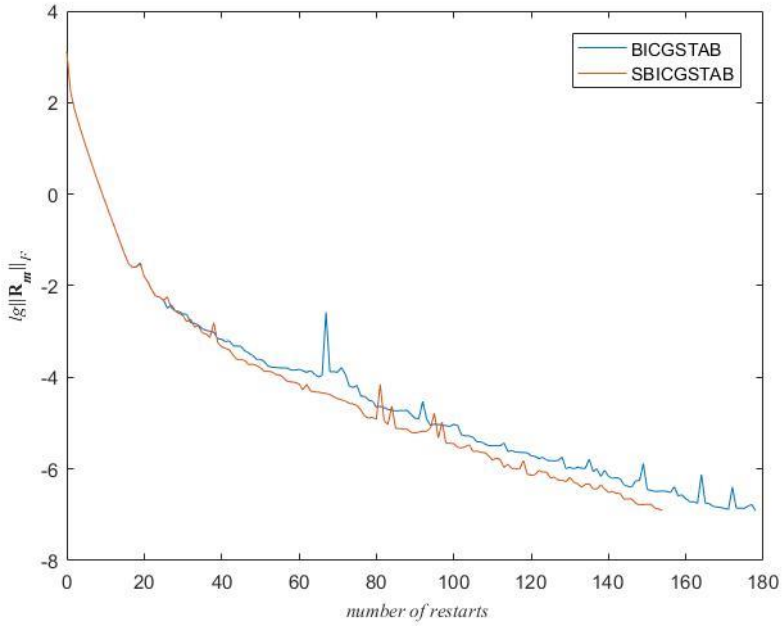
$(a)\, n = u^2 = 1225, s = 25$



$(b)\, n = u^2 = 1521, s = 31$



$(c)\, n = u^2 = 2401, s = 40$



$(d)\, n = u^2 = 6889, s = 75$

Figure 2：The convergence histories of Example 4.2

The numerical results are shown in Table 4 and Figure 2. According to Table 4, it can be seen that the GBiCGStab method needs more iteration steps and CPU time than the SGBiCGStab method. From Figure 2, we can see that the SGBiCGStab method and the GBiCGStab method show similar convergence behavior, and the oscillations of relative residual norms are relatively slight.

6 CONCLUSION

In the present paper, based on global BiCG-type methods, we proposed two shifted global methods, named as the SGBiCG method and the SGBiCGStab method, to solve the large Stein matrix equation. The proposed methods make full use of the shifted structure of the matrix equation. Numerical results show that the SGBiCG method and the SGBiCGStab method can effectively solve the Stein matrix equation.

ACKNOWLEDGEMENTS

This research is supported by CUIT (KYTD202243), Foundation of Science and Technology Department of Sichuan Province(2022JDR0043) and NSFC (11901065).

2010 AMS Subject Classifications: 65F10; 65F30.

REFERENCES

1. Jankowski, T. (1992). A NUMERICAL SOLUTION OF IMPLICIT ORDINARY DIFFERENTIAL EQUATIONS. *Demonstratio Mathematica*, *25*(1-2), 279-304.
2. Barbour, A. D., & Utev, S. (1998). Solving the Stein equation in compound Poisson approximation. *Advances in Applied Probability*, *30*(2), 449-475.
3. Van Dooren, P. M. (2000). Gramian based model reduction of large-scale dynamical systems. *Chapman and Hall CRC Research Notes in Mathematics*, 231-248.
4. Zhang, Y., Jiang, D., & Wang, J. (2002). A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Transactions on Neural Networks*, *13*(5), 1053-1063.
5. Bouhamidi, A., & Jbilou, K. (2007). Sylvester Tikhonov-regularization methods in image restoration. *Journal of Computational and Applied Mathematics*, *206*(1), 86-98.
6. Zhao, X. L., Wang, F., Huang, T. Z., Ng, M. K., & Plemmons, R. J. (2013). Deblurring and sparse unmixing for hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, *51*(7), 4045-4058.
7. Datta, B. (2004). Numerical methods for linear control systems. *Academic Press.*
8. Kong, H., Zhou, B., & Zhang, M. R. (2010, May). A Stein equation approach for solutions to the Diophantine equations. In *2010 Chinese Control and Decision Conference* (pp. 3024-3028). IEEE.
9. Golub, G. H., Nash, S., Van Loan, C., & Hessenberg, A. (1979). Schur method for the problem AX+ XB= C. *IEEE Trans. Automat. Control*, *24*(6), 909-913.
10. Bartels, R. H., & Stewart, G. W. (1972). Solution of the matrix equation AX+ XB= C [F4]. *Communications of the ACM*, *15*(9), 820-826.
11. Bao, L., Lin, Y., & Wei, Y. (2006). Krylov subspace

methods for the generalized Sylvester equation. *Applied mathematics and computation*, *175*(1), 557-573.

12. El Guennouni, A., Jbilou, K., & Riquet, A. J. (2002). Block Krylov subspace methods for solving large Sylvester equations. *Numerical Algorithms*, *29*(1), 75-96.
13. Jbilou, K. (2006). Low rank approximate solutions to large Sylvester matrix equations. *Applied mathematics and computation*, *177*(1), 365-376.
14. Bao, L., Lin, Y., & Wei, Y. (2007). A new projection method for solving large Sylvester equations. *Applied numerical mathematics*, *57*(5-7), 521-532.
15. Bouhamidi, A., Hached, M., Heyouni, M., & Jbilou, K. (2013). A preconditioned block Arnoldi method for large Sylvester matrix equations. *Numerical Linear Algebra with Applications*, *20*(2), 208-219.
16. Addam, M., Heyouni, M., & Sadok, H. (2017). The block Hessenberg process for matrix equations. *Electronic Transactions on Numerical Analysis*, *46*, 460-473.
17. Bentbib, A. H., Jbilou, K., & Sadek, E. M. (2017). On some extended block Krylov based methods for large scale nonsymmetric Stein matrix equations. *Mathematics*, *5*(2), 21. doi:10.3390/math5020021.
18. Bentbib, A., Jbilou, K., & Sadek, E. M. (2015). On some Krylov subspace based methods for large-scale nonsymmetric algebraic Riccati problems. *Computers & Mathematics with Applications*, *70*(10), 2555-2565.
19. Frommer, A., & Glässner, U. (1998). Restarted GMRES for shifted linear systems. *SIAM Journal on Scientific Computing*, *19*(1), 15-26.
20. Frommer, A. (2003). BiCGStab (ℓ) for families of shifted linear systems. *Computing*, *70*(2), 87-109.
21. Gu, X. M., Huang, T. Z., Meng, J., Sogabe, T., Li, H. B., & Li, L. (2014). BiCR-type methods for families of shifted linear systems. *Computers & Mathematics with Applications*, *68*(7), 746-758.
22. Fletcher, R. (1976). Conjugate gradient methods for indefinite systems. *Lecture Notes in Math,* 506, 73-89.
23. Van der Vorst, H. A. (1992). Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, *13*(2), 631-644.
24. Jbilou, K., Sadok, H., & Tinzefte, A. (2005). Oblique projection methods for linear systems with multiple right-hand sides. *Electronic Transactions on Numerical Analysis*, *20*, 119-138.