Recycling BiCG for families of shifted linear systems*

Jing Meng*, Pei-Yong Zhu, Hou-Biao Li

School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 611731, P. R. China

Abstract

Many problems in science and engineering fields require the solution of shifted linear systems. To solve such systems efficiently, the recycling BiCG (RBiCG) algorithm in [SIAM J. SCI. COMPUT, 34 (2012) 1925-1949] is extended in this paper. However, the shift-invariant property could no longer hold over the augmented Krylov subspace due to adding the recycling spaces. To remedy this situation, a strategy to enforce the collinearity condition on the shifted system is adopted and then a short term recurrence for the solution update of the shifted system is derived when the seed system is solving. The new method not only improves the convergence but also has a potential to simultaneously compute approximate solutions for shifted linear systems using only as many matrix-vector multiplications as the solution of a single system requires. In addition, some numerical experiments also confirm the efficiency of our method.

Key words: Shifted linear systems, Krylov subspace, RBiCG, RBiCG-sh, recycling subspace

* Supported by the National Natural Science Foundation of China (11026085, 11101071, 10671134, 11271001, 51175443) and the Fundamental Research Funds for China Scholarship Council.

* Corresponding author.

Email addresses: mengmeng-erni@163.com (Jing Meng), lihoubiao0189@163.com (Hou-Biao Li).

Preprint submitted to Elsevier 18 March 2014
1 Introduction

Let $A \in \mathbb{C}^{n \times n}$ be a non-singular matrix and $b \in \mathbb{C}^n$ be a given right-hand side. We now consider the solutions of two linear systems

$$
Ax = b,
(A + \sigma I)x^\sigma = b,
$$

where $I$ is an $n$ by $n$ identity matrix and $\sigma \in \mathbb{C}$ is a shift. The first equation is referred as the seed system while the second one is termed as the shifted system. Since two equations are only differ by a multiple of the identity, equations (1) often be called as shifted linear systems. Such problems arise in many scientific and engineering applications. For example, control theory [1], time-dependent differential equations [2] and lattice quantum chromodynamics (lattice QCD) [3].

For solving linear systems (1), Krylov subspace methods are appealing. It is because these approaches exploit a shift-invariant property that for any shift $\sigma$, $m$-th Krylov subspace generated by $A$ and $b$ is invariant under the shift, i.e.,

$$
\mathcal{K}_m(A, b) = \mathcal{K}_m(A + \sigma I, b)
$$

as long as the starting vectors are collinear. It means that equations (1) can be solved simultaneously by constructing only one sequence of Krylov subspace. This observation has led to many efficient implementations of known Krylov subspace solvers that can handle multiple shifts ($\sigma = \{-\sigma_i \in \mathbb{C} | 1 \leq i \leq s\}$) simultaneously using only as many matrix-vector multiplications as the solution of a single system requires, saving much more storage and time.

Methods that are based on the nonsymmetric Lanczos process have been straightforwardly derived to solve (1), such as CG [7], QMR, TFQMR [8], BiCGstab [9] and a recently proposed method IDR [10]. However, restarted Krylov solvers are not so straightforward because the residuals of the seed and the shifted system are not collinear any more. To handle this situation, Frommer and Glässner enforce the residuals to be collinear and propose shifted GMRES [12]. Note that this loss of collinearity does not occur for restarted FOM [13] as the residuals produced by FOM are all inherently collinear with the newest Arnoldi vector.

Furthermore, there are other algorithms in the literature to deal with multiple shifts along with multiple right-hand sides, like GMRES(m)-Proj(k)-Sh [14].

Recently, a number of studies [17,19,15,16,18,20,22] have established the benefits of using recycling technique when solving shifted linear system. It has been observed that significant improvements in convergence rates can be achieved from Krylov subspace methods by adding recycling spaces. For example, re-
cycled GCRO-DR [20] is extended for the solutions of shifted linear systems by Soodhalter, Szyld and Xue [21]. Due to restarting, it is not possible to approximate solutions for the shifted system over the same augmented Krylov subspace. As an alternative, they construct approximate corrections to the solutions of the shifted systems at each cycle while only minimizing the based system residual.

Inspired by [21], we extend the recycling BiCG (RBiCG) approach [19] to address shifted linear systems (1). However, the situation changes radically. Due to recycling spaces, the shift-invariance no longer holds over an augmented Krylov subspace. As a remedy to this situation, we use a strategy to enforce the collinearity condition on the shifted system. We then derive the short recurrence for the solution update of the shifted system. We refer to this approach as RBiCG for shifted linear systems (RBiCG-sh) which not only improves the convergence but also has a potential to simultaneously compute approximate solution for systems (1) at expense of only two matrix-vector multiplications per iteration.

Throughout this paper, $A^*$ is referred to the transpose conjugate operation of matrix $A$, $\bar{a}$ the conjugate operation of $a$. $U_\perp V$ denote that $U$ and $V$ are bi-orthogonality which implies that $U^*V$ is a diagonal matrix. $0_{i \times j}$ is defined as the zero rectangular matrix with $i$ rows and $j$ columns, $e_j$ the $k$th unit vectors. The notations of MATLAB style are also used, for example, $U(i,j)$ is the $U_{ij}$ entry of matrix $U$, $U(1:i,1:j)$ denotes the submatrix of the first $i$ rows and the first $j$ columns of $U$ and $U(:,j)$ refers to its $j$th column.

The structure of the paper is as follows. In Section 2, we describe the recycling BiCG method aimed at accelerating convergence. We introduce our new method to address shifted linear systems in Section 3. The effectiveness of the proposed method is also demonstrated in Section 4. Finally, some conclusions is summarized in Section 5.

2 Recycling BiCG algorithm

In this section, we briefly describe the RBiCG method which is adapted subspace recycling technology to accelerate convergence rate. The approach is derived from an augmented bi-Lanczos algorithm and develops a modified two-term recurrence including recycling in the iteration. First, we sketch the BiCG algorithm for solving a seed system. $\tilde{r}_0 = b - A^* \tilde{x}_0$
2.1 BiCG method

Let $x_0$ denote an initial vector with residual $r_0 = b - A x_0$ and $\tilde{r}_0$ be an initial residual for dual system ($A^* \tilde{x} = \tilde{b}$) such that $<\tilde{r}_0, r_0> \neq 0$. Given vectors $v_1 = r_0/||r_0||$, $\tilde{v}_1 = \tilde{r}_0/||\tilde{r}_0||$ with $<\tilde{v}_1, v_1> = 1$, $m$ iterations of the Bi-Lanczos algorithm [24] builds biorthogonal bases $V_m = [v_1, \ldots, v_m]$ and $\tilde{V}_m = [\tilde{v}_1, \ldots, \tilde{v}_m]$ of the Krylov subspaces

$$K_m(A, v_1) = \text{span} \{v_1, Av_1, A^2 v_1, \ldots, A^{m-1} v_1\},$$

$$K_m(A^*, \tilde{v}_1) = \text{span} \{\tilde{v}_1, A^* \tilde{v}_1, A^{*2} \tilde{v}_1, \ldots, (A^*)^{m-1} \tilde{v}_1\},$$

such that $\tilde{V}_m \perp V_m$. Then we have the bi-Lanczos relations

$$A V_j = V_{j+1} T_j = V_j T_j + t_{j+1,j} v_{j+1} e_j^T,$$

$$A^* \tilde{V}_j = \tilde{V}_{j+1} \tilde{T}_j = \tilde{V}_j \tilde{T}_j + \tilde{t}_{j+1,j} \tilde{v}_{j+1} e_j^T,$$

where $T_j$ and $\tilde{T}_j$ are $j \times j$ tridiagonal matrices, $t_{j+1,j}, \tilde{t}_{j+1,j}$ are the last element of the last row of $T_j \in \mathbb{C}^{(j+1) \times j}$ and $\tilde{T}_j \in \mathbb{C}^{(j+1) \times j}$ respectively. $V_j \perp b V_j$. For solving the linear system, the BiCG method [24] defines a projection process, i.e.,

$$r_j = b - A(x_0 + V_j \eta_j) \perp V_j,$$

$$\tilde{r}_j = \tilde{b} - A^*(\tilde{x}_0 + \tilde{V}_j \tilde{\eta}_j) \perp V_j,$$

where $\eta_j, \tilde{\eta}_j$ are determined by the orthogonality condition. Exploiting the efficiency of the bi-Lanczos relations, we obtain a coupled two-term recurrence for the solution update. In addition, the corresponding residuals $r_j, \tilde{r}_j$ are parallel to the bi-orthogonal basis vectors $v_{j+1}$ and $\tilde{v}_{j+1}$, respectively.

2.2 RBiCG algorithm

For solving a sequence of dual linear systems with multiple right-hand sides, Ahuja [19] proposed RBiCG by adapting recycling technique to BiCG. We briefly review this idea as follow.

Let the matrices $U, \tilde{U} \in \mathbb{C}^{n \times k}$ ($k << n$), derived from approximating right and left invariant subspaces of $A$, be used to define the primary ($Ax = b$) and dual system recycle space respectively. The matrices

$$C = AU \quad \text{and} \quad \tilde{C} = A^* \tilde{U} \quad \text{such that} \quad \tilde{C} \perp b C.$$  \hspace{1cm} (2)

In the following cycle, an augmented bi-Lanczos algorithm is developed to compute bi-orthogonal bases for the primary and dual Krylov subspaces while maintaining the relation $\tilde{C} \perp b C$. 

4
Since it is need to recycle spaces \( U \) and \( \tilde{U} \), the bi-Lanczos algorithm is modified to compute the columns of \( V_j \) and \( \tilde{V}_j \) such that

\[
[C \ V_j] \perp_b [\tilde{C} \ \tilde{V}_j].
\]  

(3)

Similar to the standard BiCG method, the \((j + 1)\)-th Lanczos vector for the primary system is modified by

\[
\gamma v_{j+1} = Av_j - V_j \tau - C \rho \perp [\tilde{C} \ \tilde{V}_j],
\]  

(4)

where \( \tau \) and \( \rho \) are determined by the following equations

\[
D_j \tau = \tilde{V}_j^* Av_j,
\]

\[
D_c \rho = \tilde{C}^* Av_j,
\]  

(5)

with \( D_i = \tilde{V}_j^* V_j \) and \( D_c = \tilde{C}^* C \) being diagonal matrices. \( \gamma \) is chosen such that \( <\tilde{v}_{j+1}, v_{j+1}> = 1 \). Observe that it requires a \((3 + k)\)-term recurrence for computing the Lanczos vector \( v_{j+1} \), where \( k \) is the number of column of \( C \).

Using (4) and (5), we get the augmented bi-Lanczos relation for the primary system

\[
(I - C\hat{C}^*)AV_j = V_{j+1}T_j,
\]  

(6)

where \( \hat{C} = \tilde{C}D_c^{-1} \). Then we can rewrite (6) together with (2) as the augmented bi-Lanczos-like relation,

\[
A \begin{bmatrix} U & V_j \end{bmatrix} = \begin{bmatrix} C \ V_{j+1} \end{bmatrix} \begin{bmatrix} I \hat{C}^* AV_j \\ 0 \ T_j \end{bmatrix}.
\]  

(7)

Similarly, we obtain

\[
(I - \hat{C}\tilde{C}^*)A^*\tilde{V}_j = \tilde{V}_{j+1}\tilde{T}_j,
\]  

(8)

\[
A^* \begin{bmatrix} \tilde{U} & \tilde{V}_j \end{bmatrix} = \begin{bmatrix} \tilde{C} \ \tilde{V}_{j+1} \end{bmatrix} \begin{bmatrix} I \hat{C}^* A^* \tilde{V}_j \\ 0 \ \tilde{T}_j \end{bmatrix}
\]  

(9)

with \( \hat{C} = CD_c^{-1} \) for the dual system. Next, we derive the solution update based on the augmented bi-Lanczos-like relation. With recycling, the \( j \)-th solution update can be written as

\[
x_j = x_0 + Uz_j + V_jy_j,
\]  

(10)

\[
\tilde{x}_j = \tilde{x}_0 + \tilde{U}\tilde{z}_j + \tilde{V}_j\tilde{y}_j,
\]  

(11)

and the corresponding residuals satisfy the following Petrov-Galerkin condition

\[
r_j = r_0 - AUz_j - AV_jy_j \perp [\hat{C}, \tilde{V}_j],
\]  

(12)

\[
\tilde{r}_j = \tilde{r}_0 - A^*\tilde{U}\tilde{z}_j - A^*\tilde{V}_j\tilde{y}_j \perp [C, V_j],
\]  

(13)
Let
\[ r_0 = C\hat{C}^* r_0 + (I - C\hat{C}^*) r_0 = [C, V_j] \begin{bmatrix} \hat{C}^* r_0 \\ \xi e_1 \end{bmatrix}, \] (14)
where \( \xi = \| (I - C\hat{C}^*) r_0 \| \), and then define \( v_1 = \xi^{-1} (I - C\hat{C}^*) r_0 \) such that \( v_1 \perp C \).

Using the augmented bi-Lanczos relation (7), the Petrov-Galerkin condition (12) and the bi-orthogonality (3), we get
\[
\begin{bmatrix} \hat{C}^* r_0 \\ \xi e_1 \end{bmatrix} [C, V_j] \begin{bmatrix} \hat{C}^* \\ -0 \\ T_j \\ y_j \end{bmatrix} = 0.
\] (15)

Therefore, we obtain
\[
T_j y_j = \xi e_1, \quad z_j = \hat{C}^* r_0 + \hat{C}^* A V_j y_j.
\] (16)

Substituting (16) in (10) gives the solution update
\[ x_j = x_0 + U\hat{C}^* r_0 + (I - U\hat{C}^* A) V_j y_j. \] (17)

From the equations (12) and (16), one can see that \( r_j = c_j v_{j+1}, \tilde{r}_j = \tilde{c}_j \tilde{v}_{j+1} \) for some scalars \( c_j, \tilde{c}_j \in \mathbb{C} \). As mentioned above, we seek an approximate solution \( x_j = x_0 + \mu_j \) such that \( \mu_j \in S \), where \( S = \mathcal{R}(U) + \mathcal{K}_{m-k}((I - C\hat{C}^*) A, (I - C\hat{C}^*) r_0) \) is an augmented Krylov subspace.

Since \( \hat{C}^* V_j = 0 \) and \( C^* \tilde{V}_j = 0 \), the equations (6) and (8) can be rewritten as
\[
\begin{align*}
A_1 V_j &= V_{j+1} \tilde{T}_j, & \text{where} & & A_1 &= (I - C\hat{C}^*) A (I - C\hat{C}^*) \\
A_1^* \tilde{V}_j &= \tilde{V}_{j+1} \tilde{T}_j, & \text{where} & & A_1^* &= (I - \hat{C}^* \hat{C}) A^* (I - \hat{C}^* \hat{C}).
\end{align*}
\] (18)

Note that the operators in (18) are each other’s conjugate transpose. Therefore, following steps similar to the ones used in the derivation of BiCG, we can derive a simpler recurrence for the RBiCG solution update instead of the form (17). Moreover, it is never to compute any explicit matrix inverse. Some further simplification and improvements to make the code faster are given in [19]. RBiCG for solving sequence of linear systems described in Algorithm 1.

The recycle spaces \( U \) and \( \tilde{U} \) used in solving a pair of linear systems are fixed throughout the RBiCG iteration (in Algorithm 1); however, the bases of the recycle spaces for the next pair of linear systems are updated periodically. Ahuja [19] took harmonic Ritz vectors [26] to extract approximate left- and right invariant subspaces for the update of \( U \) and \( \tilde{U} \). This approach yielded the best results in finding eigenvalues nearest zero.
Algorithm 1 RBiCG algorithm [19].

1: Given $U$ and $\tilde{U}$ compute $C$ and $\tilde{C}$ using (2). If $U$ and $\tilde{U}$ are not available; then initialize $U$, $\tilde{U}$, $C$ and $\tilde{C}$ to empty matrices.

2: Choose $x_{-1}$, $\tilde{x}_{-1}$ and compute $x_0$, $\tilde{x}_0$ such that $r_0 \perp \tilde{C}$, $\tilde{r}_0 \perp C$.

3: if $(r_0, \tilde{r}_0) = 0$ then $\tilde{x}_{-1}$ to a random vector.

4: Set $p_0 = 0$, $\tilde{p}_0 = 0$, and $\beta_0 = 0$. Choose tol and maxit

5: for $l = 1, 2, \ldots, \text{maxit}$ do

6: $p_j = \beta_{j-1} p_{j-1} + r_j$; $\tilde{p}_j = \tilde{\beta}_{j-1} \tilde{p}_{j-1} + \tilde{r}_j$

7: $z_j = A p_j$; $\tilde{z}_j = A^{*} \tilde{p}_j$

8: $\mu_j = C^{*} z_j$; $\tilde{\mu}_j = C^{*} \tilde{z}_j$

9: $q_j = z_j - C \mu_j$; $\tilde{q}_j = \tilde{z}_j - \tilde{C} \tilde{\mu}_j$

10: $\alpha_j = (\tilde{r}_{j-1}, r_{j-1}) / (\tilde{p}_{j-1}, A p_{j-1})$; $\tilde{\alpha}_j = \tilde{\alpha}_j$

11: $\mu_c = \mu_c + \alpha_j \mu_j$; $\tilde{\mu}_c = \tilde{\mu}_c + \tilde{\alpha}_j \tilde{\mu}_j$

12: $x_j = x_{j-1} + \alpha_j p_j$; $\tilde{x}_j = \tilde{x}_{j-1} + \tilde{\alpha}_j \tilde{p}_j$

13: $r_j = r_{j-1} - \alpha_j q_j$; $\tilde{r}_j = \tilde{r}_{j-1} - \tilde{\alpha}_j A^{*} \tilde{q}_j$

14: if $||r_j|| \leq \text{tol}$ and $||\tilde{r}_j|| \leq \text{tol}$ then break;

15: end if

16: end for

17: $\beta_{j-1} = (\tilde{r}_j, r_j) / (\tilde{r}_{j-1}, r_{j-1})$

18: end for

19: $x_j = x_j - U \mu_c$; $\tilde{x}_j = \tilde{x}_j - \tilde{U} \tilde{\mu}_c$

The harmonic Ritz vectors, associated to the harmonic Ritz values of $k$ smallest magnitude, are computed by solving a small generalized eigenproblem which is constructed by a small window of BiCG residuals ($m$ linearly independent Lanczos vectors in every cycle) and the previous recycling spaces. Although it is cheap to solve the generalized eigenvalue problem, it would be expensive to set up the corresponding matrices in a straightforward manner [19]. Ahuja [19] then set up the problem efficiently by using recurrences to lead to cheaper computation, see [19] for more details.

3 Recycling BiCG for shifted linear systems

A number of recent articles have established the benefits of using recycling technique when solving multiple right-hand sides. It has been observed that significant improvements in convergence rates can be achieved from Krylov subspace methods by adding these recycling spaces. Therefore, we consider an extension of RBiCG to solve shifted linear systems in this section. However, the shift invariance no longer hold over the augmented Krylov subspace. To handle this situation, we exploit a strategy to enforce the residuals for the seed and the shifted systems are collinear. Moreover, the $(3+k)$-term recurrence for the solution update of the shifted system is derived. Thus, we will have an efficient
method which not only improves the convergence but also has a potential to simultaneously compute approximate solution for the shifted system together with those for the seed system without investing any additional matrix-vector multiplications for the iterates of the shift system.

3.1 Collinear Residuals

An initial vector \( x_0^\sigma \) for the shifted linear system is chosen such that the initial residuals are collinear, i.e., \( r_0^\sigma = \pi_0^\sigma r_0, \pi_0^\sigma \in \mathbb{C} \). Since

\[
(A + \sigma I) \begin{bmatrix} U \\ V_j \end{bmatrix} = \begin{bmatrix} C \\ \hat{C}^* \end{bmatrix} \begin{bmatrix} I \\ \sigma \hat{C}^* \end{bmatrix} V_j + \sigma U,
\]

we observed that the shift-invariant property is no long satisfied under the augmented Krylov subspace. Moreover, as mentioned in [21], the initial projection of the residual of the seed system (14) and resulting update of the solution rely on the relation (2) between \( U \) and \( C \). Even though we can bi-orthogonally project \( r_0^\sigma \) onto \( R(\hat{C})^\perp \), it is not easily update the approximation associated with the shifted system. In order to project the residual for the shifted system, Soodhalter et al. [21] presented an ideal method. The approach constructs a recycling space \( U^\sigma \) for each shift\(^1\) such that

\[
C = AU = (A + \sigma I) U^\sigma. \tag{19}
\]

Following the similar idea as in [21], we assume the relation (19) holds. With this condition (19), we will demonstrate that the residuals of the seed and the shifted system will be collinear.

Suppose that we have matrices \( U_\sigma \) and \( U \) satisfying the relations (19) and \( C \perp \hat{C} \). Then we can bi-orthogonally project \( r_0^\sigma \) onto \( R(\hat{C})^\perp \) and update \( x_1^\sigma = x_0^\sigma + U^\sigma \hat{C}^* r_0^\sigma \). Since \( \hat{C}^* V_j = 0 \) and \( \hat{C}^* V_j = 0 \), the following equation holds

\[
(I - \hat{C}^*) (A + \sigma I) V_j = V_{j+1} (T_j + \begin{bmatrix} \sigma I_{j \times j} \\ 0_{1 \times j} \end{bmatrix}). \tag{20}
\]

Combining the conditions (19) and (20) yields an augmented bi-Lanczos-like relation for the shifted system,

\[
(A + \sigma I) \begin{bmatrix} U^\sigma \\ V_j \end{bmatrix} = \begin{bmatrix} C \\ V_{j+1} \end{bmatrix} \begin{bmatrix} I \\ B \end{bmatrix} \begin{bmatrix} T_j^\sigma \\ 0 \end{bmatrix}. \tag{21}
\]

\(^1\) How to construct the matrices \( U^\sigma \) for each shift is given in the Appendix A.
where \( B = \hat{C}^* AV \) and \( T_j^\sigma = T_j + \begin{bmatrix} \sigma I_{j \times j} \\ 0_{1 \times j} \end{bmatrix} \). Using this relation, we can follow the similar steps as in \[19\] and construct the \( j \)-th solutions for shift system,

\[
x_j^\sigma = x_0^\sigma + U^\sigma z_j^\sigma + V_j y_j^\sigma.
\]

Then the corresponding residual is

\[
r_j^\sigma = r_0^\sigma - (A + \sigma I) \begin{bmatrix} U^\sigma & V_j \end{bmatrix} \begin{bmatrix} z_j^\sigma \\ y_j^\sigma \end{bmatrix} = C\hat{C}^* r_0^\sigma + (I - C\hat{C}^*) r_0^\sigma - \begin{bmatrix} C & V_{j+1} \\ 0 & T_j^\sigma \end{bmatrix} \begin{bmatrix} I & B \\ 0 & T_j^\sigma \end{bmatrix} \begin{bmatrix} z_j^\sigma \\ y_j^\sigma \end{bmatrix}.
\]

(23)

Since \([C, V_j] \perp_b [\hat{C}, \hat{V}_j]\), we seek an approximation solution \( x_j^\sigma \) by enforcing \( r_j^\sigma \perp [\hat{C}, \hat{V}_j] \). Therefore, we obtain

\[
y_j^\sigma = (T_j^\sigma)^{-1} e_1,
\]

\[
z_j^\sigma = C\hat{C}^* r_0^\sigma - B y_j^\sigma.
\]

(24)

Substituting (24) in (22) yields

\[
x_j^\sigma = x_0^\sigma + \frac{1}{c_0} U^\sigma \hat{C}^* r_0^\sigma + (I - U^\sigma \hat{C}^* A) V_j y_j^\sigma.
\]

(25)

The following theorem shows that the residuals for the seed and the shifted system are collinear.

**Theorem 3.1** Let \( x_j \in \mathcal{R}(U) + \mathcal{K}_{m-k}((I-C\hat{C}^*)A, (I-C\hat{C}^*)r_0) \) be an approximation to the solution of \( Ax = b \) defined via the following Petrov-Galerkon condition for the residual \( r_j = b-Ax_j \) such that \( r_j \perp [\hat{C}, \hat{V}_j], \ j = 1, \ldots, k \). Similarly, let \( x_i^\sigma \in \mathcal{R}(U^\sigma) + \mathcal{K}_{m-k}((I-C\hat{C}^*)(A+\sigma I), (I-C\hat{C}^*)r_0^\sigma) \) be the approximation to the solution of \((A+\sigma I)x^\sigma = b\) with the residual \( r_i^\sigma = b-(A+\sigma I)x_i^\sigma \), again satisfying \( r_j^\sigma \perp [\hat{C}, \hat{V}_j], \ j = 1, \ldots, k \). Assume that residuals for the shifted and based system satisfied \( r_0^\sigma = (1/\pi_0^\sigma) r_0 \) at the beginning of a cycle and \( C = AU = (A+\sigma I)U^\sigma \), then \( r_j \) and \( r_j^\sigma \) are collinear, i.e.

\[
r_j^\sigma = (1/\pi_j^\sigma) r_j, \ \pi_j^\sigma \in \mathbb{C}.
\]
Proof. Using (25), we get
\begin{equation}
\begin{aligned}
r_{j}^{\sigma} &= b - (A + \sigma I)x_{j}^{\sigma}, \\
&= r_{0}^{\sigma} - C(\frac{1}{\pi_{0}^{\sigma}}\hat{C}^{*}r_{0} - By_{j}^{\sigma}) - (A + \sigma I)V_{j}y_{j}^{\sigma}
\end{aligned}
\end{equation}

Since $\tilde{C}^{*}V_{j} = 0$, we have
\begin{equation}
\begin{aligned}
r_{j}^{\sigma} &= \frac{1}{\pi_{0}^{\sigma}}(I - C\hat{C}^{*})r_{0} - (I - C\hat{C}^{*})(A + \sigma I)V_{j}y_{j}^{\sigma} \\
&= \frac{1}{\pi_{0}^{\sigma}}(I - C\hat{C}^{*})r_{0} - V_{j+1}T_{j}^{\sigma}y_{j}^{\sigma}
\end{aligned}
\end{equation}

Substituting (24) yields
\begin{equation}
\begin{aligned}
r_{j}^{\sigma} &= -\frac{1}{\pi_{0}^{\sigma}}\xi_{j+1,j}^{\sigma}e_{j}^{T}(T_{j}^{\sigma})^{-1}e_{1}v_{j+1}.
\end{aligned}
\end{equation}

This implies $r_{j}^{\sigma} = (1/\pi_{j}^{\sigma})r_{j}$ for some scalar $\pi_{j}^{\sigma} \in \mathbb{C}$. □

3.2 Shifted RBiCG

As we previously demonstrated, the residuals for the seed and shift systems are collinear. With this condition, we can derive the short-term recurrence for the solution update of the shift system while solving the seed system.

Let $P = I - C\hat{C}^{*}$ be a projection operator. Using (17), we observed that a recycling Krylov subspace method produces iterative solutions $x_{j}$ for which the residuals $r_{j} = b - Ax_{j}$ are in the Krylov space $K_{m-k}(PA, Pr_{0})$. As a consequence, the residual can be written as
\begin{equation}
\begin{aligned}
r_{j} &= \Phi_{j}(PA)r_{0} = \Phi_{j}(PA)Pr_{0},
\end{aligned}
\end{equation}

where $\Phi_{j}$ is a polynomial of degree $\leq j - 1$ with $\Phi_{j}(0) = 1$.

Similarly, for the shifted system, any vector $x_{j}^{\sigma}$ from the recycling Krylov subspace $R(U^{\sigma}) + K_{m-k}(P(A + \sigma I), Pr_{0}^{\sigma})$ can be represented as the form (25). Therefore, the corresponding residual $r_{j}^{\sigma} = b - (A + \sigma I)x_{j}^{\sigma}$ satisfies
\begin{equation}
\begin{aligned}
r_{j}^{\sigma} &= \Phi_{j}^{\sigma}(PA + \sigma P)Pr_{0}^{\sigma},
\end{aligned}
\end{equation}

with $\Phi_{j}^{\sigma}(0) = 1$.

**Theorem 3.2** Let $P = I - C\hat{C}^{*}$ be a projection operator. The following equation holds
\begin{equation}
\begin{aligned}
\Phi_{j}^{\sigma}(PA + \sigma P)Pr_{0}^{\sigma} = \Phi_{j}^{\sigma}(PA + \sigma I)Pr_{0}^{\sigma}.
\end{aligned}
\end{equation}
Proof. Since $P^2 = P$, then

$$(PA + \sigma P)^j = (PA + \sigma P)(PA + \sigma P)(PA + \sigma P)^{j-2}$$

$$= (PA + \sigma I)(PA + \sigma P)(PA + \sigma P)^{j-2}$$

$$= (PA + \sigma I)^2(PA + \sigma P)(PA + \sigma P)^{j-3}$$

$$= \ldots$$

$$= (PA + \sigma I)^{j-1}(PA + \sigma P),$$

Combining $(PA + \sigma P)Pr_0^\sigma = (PA + \sigma I)Pr_0^\sigma$ yields

$$r_j^\sigma = \Phi_j^\sigma(FA + \sigma P)Pr_0^\sigma$$

$$= \Phi_j^\sigma(FA + \sigma I)Pr_0^\sigma,$$

which completes the proof. □

For solving the shifted system, we take the collinear residual approach as follows,

$$r_j^\sigma = (1/\pi_j^\sigma)r_j, \quad \pi_j^\sigma \in C.$$  \hspace{1cm} (31)

Combining (29), (30) and (31) yields $\Phi_j^\sigma(FA + \sigma I)Pr_0^\sigma = (1/\pi_j^\sigma)\Phi_j(FA)Pr_0^\sigma$. Since $Pr_0^\sigma, PAPr_0^\sigma, (PA)^2Pr_0^\sigma, \ldots, (PA)^mPr_0^\sigma$ form a basis of $K_{m-k}(FA, Pr_0^\sigma)$, we can compared coefficients and obtain the following identity

$$\Phi_j^\sigma(t + \sigma) = (1/\pi_j^\sigma)\Phi_j(t).$$

Since $\Phi_j^\sigma(0) = 1$, the equation $\pi_j^\sigma = \Phi_j(-\sigma)$ is finally obtained.

From the (6-9)th steps and the 13th one in the Algorithm 1, the following recurrence relation holds,

$$r_{j+1} = (1 + \frac{\beta_{j-1}}{\alpha_{j-1}} \alpha_j - \alpha_j FA)r_j - \frac{\beta_{j-1}}{\alpha_{j-1}} \alpha_j r_{j-1}$$

$$= -\alpha_j FA r_j + (1 + \frac{\beta_{j-1}}{\alpha_{j-1}} \alpha_j)r_j - \frac{\beta_{j-1}}{\alpha_{j-1}} \alpha_j r_{j-1}.$$  \hspace{1cm} (32)

Since $r_j = \Phi_j(FA)Pr_0^\sigma$, $\Phi_j$ is the RBiCG polynomials, (32) becomes

$$\Phi_{j+1}(t) = (1 + \frac{\beta_{j-1}}{\alpha_{j-1}} \alpha_j - \alpha_j t)\Phi_j(t) - \frac{\beta_{j-1}}{\alpha_{j-1}} \alpha_j \Phi_{j-1}(t).$$  \hspace{1cm} (33)

Evaluating at $t = -\sigma$ and using $\pi_j^\sigma = \pi_j(-\sigma)$ we get

$$\pi_{j+1}^\sigma = (1 + \alpha_j \sigma)\pi_j^\sigma + \frac{\alpha_j \beta_j}{\alpha_{j-1}} (\pi_j^\sigma - \pi_j^{\sigma-1}).$$  \hspace{1cm} (34)

Similarly, applying the RBiCG method to the shifted system $(A + \sigma I)x^\sigma = b$, we consider updating the residual $r_{j+1}^\sigma$ of the shifted system with the following
three-term recurrence relation

\[ r_{j+1}^\sigma = (1 + \frac{\beta_{j-1}^\sigma}{\alpha_{j-1}^\sigma}) \alpha_j^\sigma P(A + \sigma I) r_j^\sigma - \frac{\beta_{j-1}^\sigma}{\alpha_{j-1}^\sigma} \alpha_j^\sigma r_{j-1}^\sigma, \]  \quad (35)

where \( \alpha_j^\sigma, \beta_j^\sigma \) are analogous to the coefficients \( \alpha_j, \beta_j \) in RBiCG algorithm. To obtain the computational formula for \( r_{j+1}^\sigma \), three parameters \( \alpha_j^\sigma, \beta_j^\sigma \) and \( \pi_{j+1}^\sigma \) are completely fixed. Hence, we give computational formulas for the three values. Substituting (31) into (35) we obtain

\[ r_j = -\alpha_j^\sigma \left( \frac{\pi_j^\sigma}{\pi_{j+1}^\sigma} \right) P A r_j + (1 + \frac{\beta_{j-1}^\sigma}{\alpha_{j-1}^\sigma} \alpha_j^\sigma - \alpha_{j-1}^\sigma I) \frac{\pi_{j+1}^\sigma}{\pi_j^\sigma} r_j \]

\[ = -\frac{\beta_{j-1}^\sigma \alpha_j^\sigma}{\alpha_{j-1}^\sigma \pi_{j+1}^\sigma} r_{j-1}. \]  \quad (36)

Then comparing the coefficients with (32), the computational formulas for the other parameters are obtained

\[ \alpha_j^\sigma = \alpha_j \left( \frac{\pi_j^\sigma}{\pi_{j+1}^\sigma} \right), \quad \beta_j^\sigma = \left( \frac{\pi_{j+1}^\sigma}{\pi_j^\sigma} \right)^2 \beta_j. \]

(37)

If we initialize \( \pi_{-1}^\sigma = 1 \), the relations (31) and (37) remain hold for \( j = 0 \).

By (35), we are now able to formulate the recurrence for the solution update of the shift system

\[ p_j = \beta_{j-1} p_{j-1} + r_j^\sigma \]

\[ = \frac{1}{\pi_{j-1}^\sigma} r_{j-1} - \beta_{j-1} p_{j-1}^\sigma \]  \quad (38)

and

\[ x_j^\sigma = x_{j-1}^\sigma + \alpha_j (I - U^\sigma \tilde{C}^*) (A + \sigma I) p_j^\sigma. \]  \quad (39)

From (39), it is noted that we will need an additional matrix-vector multiplication, i.e., \( Ap_j^\sigma \). But we can use a strategy to eliminate this multiplication at expense of one auxiliary vector to store. To make this idea more concrete, we present outline of this process in Algorithm 2.

**Algorithm 2** RBiCG algorithm.

1. \( p_j = \beta_{j-1} p_{j-1} + r_j \); \quad \( p_j^\sigma = \frac{1}{\pi_{j-1}^\sigma} r_{j-1} - \beta_{j-1} p_{j-1}^\sigma \)
2. \( u_j = A r_j \);  
3. \( z_j = u_j - \beta_{j-1} z_{j-1} \); \quad \( q_j = \frac{1}{\pi_{j-1}^\sigma} u_{j-1} - \beta_{j-1} q_{j-1} \)
4. \( \mu_j = \tilde{C}^* z_j \); \quad \( \mu_j^\sigma = \tilde{C}^* (q_{j-1}^\sigma + \alpha p_{j-1}^\sigma) \)
5. \( q_j = z_j - C \mu_j \);  
6. \( \mu = \mu + \alpha_j \mu_j \); \quad \( \mu_\sigma = \mu_\sigma + \alpha_j^\sigma \mu_j^\sigma \)

12
Finally, the resulting algorithm, RBiCG-sh, is described in Algorithm 3.

Algorithm 3  Shift RBiCG algorithm.

1: Given $U$ and $\tilde{U}$ compute $C$ and $\tilde{C}$ such that $\tilde{C} \perp b \perp C$. If $U$ and $\tilde{U}$ are not available; then initialize $U$, $\tilde{U}$, $C$ and $\tilde{C}$ to empty matrices.
2: Choose $x_{-1}$, $x^\sigma_{-1}$, $\bar{x}_{-1}$ and compute $x_0$, $x^\sigma_0$, $\bar{x}_0$ such that $r_0 \perp \tilde{C}$, $r^\sigma_0 \perp \tilde{C}$, and $\tilde{r}_0 \perp C$.
3: if $(r_0, \tilde{r}_0) = 0$ then $\tilde{x}_{-1}$ to a random vector.
4: Set $p_0 = 0$, $p^\sigma_0 = 0$, $\tilde{p}_0 = 0$, and $\beta_0 = 0$. Choose $\text{tol}$ and $\text{maxit}$
5: for $l = 1, 2, \ldots$, $\text{maxit}$ do
   6: $p_j = \beta_{j-1}p_{j-1} + r_j$; $\tilde{p}_j = \tilde{\beta}_{j-1}\tilde{p}_{j-1} + \tilde{r}_j$
   7: $u_{j-1} = Ar_{j-1}$;
   8: $z_j = u_{j-1} - \beta_{j-1}z_{j-1}$; $\tilde{z}_j = A^*\tilde{p}_j$
   9: $\mu_j = C^*z_j$; $\tilde{\mu}_j = C^*\tilde{z}_j$
10: $q_j = z_j - C\mu_j$; $\tilde{q}_j = \tilde{z}_j - \tilde{C}\tilde{\mu}_j$
11: $\alpha_j = (\tilde{r}_{j-1}, r_{j-1})/(\tilde{p}_{j-1}, Ap_{j-1})$; $\tilde{\alpha}_j = \tilde{\alpha}_j$
12: $\mu_c = \mu_c + \alpha_j\mu_j$; $\tilde{\mu}_c = \tilde{\mu}_c + \tilde{\alpha}_j\tilde{\mu}_j$
13: $x_j = x_{j-1} + \alpha_jp_j$; $\tilde{x}_j = \tilde{x}_{j-1} + \tilde{\alpha}_j\tilde{p}_j$
14: shift system
15: $\pi^\sigma_{j+1} = (1 + \alpha_j\sigma)\pi^\sigma_j + \frac{\alpha_j\beta_{j-1}}{\alpha_{j-1}}(\pi^\sigma_{j-1} - \pi^\sigma_j)$
16: $\alpha^\sigma_j = \alpha_j(\pi^\sigma_{j+1})$
17: $p^\sigma_j = \frac{1}{\pi^\sigma_{j+1}}r^\sigma_{j-1} - \beta_{j-1}p^\sigma_{j-1}$
18: $q^\sigma_j = \frac{1}{\pi^\sigma_{j+1}}u_{j-1} - \beta^\sigma_{j-1}q^\sigma_{j-1}$
19: $\mu^\sigma_j = \tilde{C}^*(q^\sigma_j + \sigma p^\sigma_{j-1})$
20: $\mu^\sigma_c = \mu^\sigma_c + \alpha^\sigma_j\mu^\sigma_j$
21: $x^\sigma_j = x^\sigma_{j-1} + \alpha^\sigma_jp^\sigma_j$
22: $r_j = r_{j-1} - \alpha_jq_j$; $\tilde{r}_j = \tilde{r}_{j-1} - \tilde{\alpha}_jA^*\tilde{q}_j$
23: if $||r_j|| \leq \text{tol}$ and $||r^\sigma_j|| \leq \text{tol}$ then
24: break;
25: end if
26: $\beta_{j-1} = (\tilde{r}_j, r_j)/(\tilde{r}_{j-1}, r_{j-1})$
27: $\beta^\sigma_j = (\pi^\sigma_{j-1}/\pi^\sigma_j)^2\beta_j$.
28: end for
29: $x_j = x_j - U\mu_c$; $\tilde{x}_j = \tilde{x}_j - \tilde{U}\tilde{\mu}_c$
30: $x^\sigma_j = x^\sigma_j - U^\sigma\mu^\sigma_c$

Note that it would require an additional $k$ vectors of storage for each shift. Nevertheless, this has to be balanced with its faster convergence speed as shown in Section 4.
4 Numerical results

In this section, some numerical experiments will be described. The goal of these experiments is to examine the effectiveness of RBiCG-sh (Algorithm 3).

To analyze RBiCG-sh, we solve shifted linear systems two times. The first time without available recycling spaces is referred as RBiCG-sh1. The numerical behavior of RBiCG-sh1 will be the same as the ones by applying BiCG to solve shift linear systems. The only difference is that the recycling spaces for next run are generated during the process of RBiCG-sh1. For the second time (RBiCG-sh2), we solve the same systems by RBiCG-sh with recycling spaces. This is a useful method for indicating how well subspaces recycling work without any effect from changing matrices or right-hand sides. Furthermore, the serial RBiCG method (with recycling spaces) is applied to the seed and shifted systems independently to illustrate how cheaply RBiCG-sh to solve shifted linear systems. The convergence curve, corresponding to the serial RBiCG method, is marked as RBiCG-2S.

In all of our runs we use a $0_{n \times 1}$ initial guess and choose different values for $\sigma$. The parameters are taken as $m = 30$, $k = 8$. The stopping criterion in all iterative methods is $\frac{||r_k||}{||r_0||} \leq 10^{-7}$.

All the numerical experiments were performed in MATLAB 2011b. The machine we have used is a PC-Pentium(R), CPU 2.80 GHz, 2.00 GB of RAM.

Example 4.1. Following [25], the first test matrix is a tridiagonal matrix with entries 0.01, 0.02, 0.03, 0.04, 0.05, 6, 7, ..., 1000 on the main diagonal, sub-diagonal entries all 1 and super-diagonal entries all $-1$. The shift is taken four different values 0.5, $-0.5$, 1, $-1$. The right-hand side $b$ is taken by $b = \hat{b}/\text{norm}(\hat{b})$ with $\hat{b} = A\hat{x}, \hat{x} = (1, \ldots , 1)^T$.

![Convergence history of serial RBiCG and RBiCG-sh for tridiagonal problem. Left: $\sigma = 0.5$, Right: $\sigma = -0.5$.](image)

The results in Figure 1 and 2 highlight the interest of performing recycling
Fig. 2. Convergence history of serial RBiCG and RBiCG-sh for tridiagonal problem. Left: $\sigma = 1$, Right: $\sigma = -1$.

technique which could improve convergence rate.

**Example 4.2.** We test RBiCG-sh on the reaction-diffusion Brusselator model matrices RDB1250 and RDB2048 from Matrix Market [34]. For the first example, we take two values for the shift, $\sigma = \frac{1}{2} + \frac{1}{5}i, -\frac{1}{2} - \frac{1}{5}i$. For the second one, we consider $1 + \frac{1}{2}i$ and $-1 - \frac{1}{2}i$ for the shift $\sigma$. The right-hand side is set as $b = \hat{b}/\text{norm}(\hat{b})$ with $\hat{b} = (1, \ldots ,1)^T$.

Fig. 3. Convergence history of serial RBiCG and RBiCG-sh for RDB1250. Left: $\sigma = \frac{1}{2} + \frac{1}{5}i$, Right: $\sigma = -\frac{1}{2} - \frac{1}{5}i$.

Fig. 4. Convergence history of serial RBiCG and RBiCG-sh for RDB2048. Left: $\sigma = 1 + \frac{1}{2}i$, Right: $\sigma = -1 - \frac{1}{2}i$.

Figures 3 and 4 depict the residual history of examples using RBiCG-sh and
the serial RBiCG method. Comparing BiCG-sh1 and RBiCG-sh2, we see a significant difference in convergence. This difference is due solely to subspace recycling. Comparing with the serial RBiCG method, RBiCG-sh2 can simultaneously compute solutions for the seed and shifted systems without any additional matrix-vector for iterates of the shifted system.

Example 4.3. In this part, the model is obtained from the heat problem by adding a convection term, which is part of the benchmark problems, namely

\[
\begin{aligned}
\frac{\partial u}{\partial t} &= \kappa \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + c \frac{\partial u}{\partial x}, \quad \text{for } (x, y) \in [0,1]^2 \text{ and } t > 0, \\
u(x, 0, t) &= u(x, 1, t) = 0, \\
u(0, y, t) &= u(1, y, t) = 0, \\
u(x, y, 0) &= x(1-x^2)y(1-y), \quad \kappa, \ c \in \mathbb{R}. 
\end{aligned}
\]

(40)

To solve the problem numerically, we start by discretizing the spatial domain into uniformly spaced grid points. Using the finite difference method approximates the derivatives of the function by the derivatives of the local interpolant on the grid [36, p. 2]. Here, we approximate the model problem (40) by the 9-point finite difference scheme,

\[
\begin{aligned}
\frac{l u(x_i, y_j)}{dt} &\approx \kappa \left( \frac{u_{i-1,j-1} + 4u_{i,j-1} + u_{i+1,j-1} + 4u_{i-1,j} - 20u_{i,j} + 4u_{i+1,j} + u_{i-1,j+1}}{6h^2} \\
&\quad + \frac{4u_{i,j+1} + u_{i+1,j+1}}{6h^2} \right) + c \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \quad 1 \leq i, j \leq N, \ h = \frac{1}{N+1}.
\end{aligned}
\]

This finite difference approximation with the natural ordering results in a system of ODEs as following form

\[
\frac{du}{dt} = Au(t), \quad u(0) = u_0,
\]

(41)

where \( A = \text{tridiag}(C, B, C) \) with

\[
B = \frac{1}{6h^2} \begin{bmatrix}
-20\kappa & 4\kappa + 3hc & 0 & \cdots & 0 \\
4\kappa - 3hc & -20\kappa & 4\kappa + 3hc & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 4\kappa - 3hc & -20\kappa
\end{bmatrix},
\]

\[
C = \left[ \begin{array}{cccc}
\kappa \\
\kappa \\
\ddots \\
\kappa
\end{array} \right].
\]
and

\[ C = \frac{\kappa}{6h^2} \begin{bmatrix} 4 & 1 & 0 & \cdots & 0 \\ 1 & 4 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 4 \\ 0 & \cdots & 1 & 4 \end{bmatrix} \]

Since spatial finite difference methods result in systems of ODEs of the form (41) where \( u \) is the vector containing the unknown solution. For the time integration of (41) we adopt a relatively new approach by using a contour approximation method as discussed in [38]. This time integration scheme involves solving the shifted linear systems. The major computational cost is solving a linear systems (42) for each \( z_k \), refer to [38]. The shifted linear system are defined as

\[(z_k I - A)x_k = u_0.\] (42)

We take the mesh grid size equal to \( h = \frac{1}{106} \), \( c = 5 \) and \( \kappa = \frac{1}{30} \). Then \( A \) is a 3136 by 3136 matrix. Here, four different values are chosen for the shift, \( z_k = \frac{1}{10} - \frac{4}{5}i, -\frac{1}{10} + \frac{4}{5}i, 10 - \frac{1}{2}i, -10 + \frac{1}{2}i \) respectively.

![Fig. 5. Convergence history of serial RBiCG and RBiCG-sh for heat problem. Left: \( \sigma = \frac{1}{10} - \frac{4}{5}i \), Right: \( \sigma = -\frac{1}{10} + \frac{4}{5}i \).](image)

![Fig. 6. Convergence history of serial RBiCG and RBiCG-sh for heat problem. Left: \( \sigma = 10 - \frac{1}{2}i \), Right: \( \sigma = -10 + \frac{1}{2}i \).](image)
Their convergence behaviors are plotted in Figure 5 and 6. RBiCG-sh2 enjoys a significant decrease in the number of matrix-vector products than RBiCG-sh1. In addition, it simultaneously solve all shifted linear systems in only one augmented Krylov subspace, leading to large saving in storage and time. Thus, it is demonstrated that the potential effectiveness of our algorithm for multiple shifted linear systems.

5 Conclusions

In this paper, we have derived the extension of RBiCG for shifted linear systems. We use a strategy to make the residuals of shift systems collinear. With this condition, we derive the recursion, as short as the seed system, for the shifted linear systems. Moreover, some examples illustrate that our method not only enjoys the benefits of using recycling technique, but also can exploit the shift invariance, without computing additional matrix-vector products for solving the shifted linear systems. Therefore, it is concluded that the RBiCCG-sh is a competitive method for solving the linear systems with multiple shifts.

Acknowledgements

The authors sincerely thank Dr. Kapil Ahuja for suggesting and providing the information of the recycling BiCG method.

References

[1] B. Datta, Y. Saad, Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment, Linear Algebra Appl. 154-156 (1991) 225-244.

[2] R. Sweet, A parallel and vector variant of the cyclic reduction algorithm, SIAM J. Sci. Statist. Comput. 9 (1988) 89-105.

[3] J.C.R. Bloch, T. Wettig, Domain-well and overlap fermions at nonzero quark chemical potential, Phys. Rev. D 76 (2007) 114-511.

[4] P. Soudais, Iterative solution methods of a 3-D scattering problem from arbitrary shaped multidielectric and multiconducting bodies, IEEE Trans. on Antennas and Propagation 42 (1994) 954-959.

[5] Jing Meng, Peiyong Zhu, Houbiao Li, A block GCROT(m,k) method for linear systems with multiple right-hand sides, Journal of Computational and Applied Mathematics 255 (2014) 544-554.
[6] Hou-biao Li, Ting-zhu Huang, Hong Li, Inclusion sets for singular values. Linear Algebra and its Applications 428 (2008) 2220-2235.

[7] J. Van den Eshof, G.L.G. Sleijpen, Accurate conjugate gradient methods for families of shifted systems, Appl. Numer.Math., 49 (2004) 17-37.

[8] R.W Freund, Solution of shifted linear systems by quasi-minimal residual iterations, Numerical Linear Algebra: Proceedings of the Conference in Numerical Linear and Scientific computation Kent (Ohio), New York (1993).

[9] Andreas Frommer, BiCGStab(l) for families of shifted linear systems, Computing 70 (2003) 87-109.

[10] Peter Sonneveld and Martin B. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations, SIAM J. Sci. Comput. 31 (2008) 1035-1062.

[11] H. Calandra, S. Gratton, R. Lago, X. Vasseur, L. M. Carvalho, A deflated minimal block residual method for the solution of non-hermitian linear systems with multiple right-hand sides. http://www.cerfacs.fr/algol/publications/, Accessed 6 July (2012).

[12] A. Frommer, Uwe Glässner, Restarted GMRES For Shifted Linear Systems, SIAM J. Sci. COMPUT. 19 (1998) 15-26.

[13] V. Simoncini, Restarted Full Orthogonalization Method for Shifted linear systems, BIT Numerical Mathematics 43 (2003) 459-466.

[14] Dean Darnell, Ronald B. Morgan, and Walter Wilcox, Deflated GMRES for systems with multiple shifts and multiple right-hand sides, Linear Algebra and its Applications 429 (2008) 2415-2434.

[15] Ronald B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM Journal on Matrix Analysis and Applications 21 (2000) 1112-1135.

[16] E. de Sturler, Truncation strategies for optimal Krylov subspace methods. SIAM J. Numer. Anal. 36 (1999) 864-889.

[17] Jason. E. Hicken, David W. Zingg, A simplified and flexible variant of GCROT for solving nonsymmetric linear systems. SIAM J. Sci. Comput. 32 (2010) 1672-1694.

[18] Y Saad, M Yeung, J Erhel, F Guyomarč, A deflated version of the conjugate gradient algorithm. SIAM J. Sci. Comput. 21 (2000) 1909-1926.

[19] K. Ahuja, E. de Sturler, S. Gugercin, E. R. Chang, Recycling BiCG with an application to model reduction. SIAM J. Sci. Comput., 34 (2012) 1925-1949.

[20] Michael L. Parks, Eric de Sturler, Greg Mackey, Duane D. Johnson, and Spandan Maiti, Recycling Krylov subspaces for sequences of linear systems, SIAM J. Sci. Comput. 28 (2006) 1651-1674.
[21] Kirk M. Soodhalter, Danidl B. Szyld, Fei Xue, Krylov subspace recycling for sequences of shifted linear systems, (2013). [http://arxiv.org/abs/1301.2650](http://arxiv.org/abs/1301.2650)

[22] Yousef Saad, Analysis of augmented Krylov subspace methods, SIAM Journal on Matrix Analysis and Applications 18 (1997) 435-449.

[23] Misha E. Kilmer and Eric de Sturler, Recycling subspace information for diffuse optical tomography, SIAM J. Sci. Comput. 27 (2006) 2140-2166.

[24] Y Saad, Iterative methods for sparse linear systems, SIAM 2003.

[25] R. B. Morgan, Restarted block GMRES with deflated restarting, Appl. Numer. Math. 54 (2005) 222-236.

[26] G. L. G. Sleijpen and H. A. van der Vorst, A Jacobi-Davidson iteration method for linear eigenvalue problems. SIAM Journal on Matrix Analysis and Applications 17 (1996) 401-425.

[27] Qiang Du, Max Gunzburger, Rich Lehoucq, Kun Zhou, Analysis and approximation of nonlocal diffusion problems with volume constraints, SIAM Review 54 (2012) 667-696.

[28] Max Gunzburger, Rich Lehoucq, A nonlocal vector calculus with application to nonlocal boundary value problems, Multiscale Modeling and Simulation, 8 (2010) 1581-1598.

[29] M. Kilmer, E. Miller, C. Rappaport, QMR-based projection techniques for the solution of non-Hermitian systems with multiple right-hand sides. SIAM J. Sci. Comput. 23 (2001) 761-780.

[30] Ronald B. Morgan, Dywayne A. Nicely, Restarting the nonsymmetric Lanczos algorithm for eigenvalues and linear equations including multiple right-hand sides. SIAM J. Sci. Comput. 33 (2011) 3037-3056.

[31] Andreas Stathopoulos, Konstantions Orginos, Computing and deflating eigenvalues while solving multiple right-hand side linear systems with an application to quantum chromodynamics. SIAM J. Sci. Comput. 32 (2010) 439-462.

[32] K. Meerbergen, Z. Bai, The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides. SIAM J. Sci. Comput. 4 (2011) 1642-1662.

[33] J. Langou, Iterative methods for solving linear systems with multiple right-hand sides. PhD thesis, CERFACS 2003.

[34] National Institute of Standards and Technology: Matrix Market, [http://math.nist.gov/Matrix-Market](http://math.nist.gov/Matrix-Market)

[35] H. Liu, B. J. Zhong, Simpler block GMRES for nonsymmetric systems with multiple right-hand sides. Elec. Trans. Numer. Anal. 30 (2008) 1-9.

[36] L.N. Trefethen, Spectral methods in MATLAB, Software, Environments, and Tools, vol.10, SIAM, Philadelphia, PA, 2000.
Appendix A. Constructing Multiple Deflation Space

In order to determine multiple recycling spaces for each shift $\sigma_i$ ($\sigma_0$ for the seed system), Soodhalter et al. [21] took the following scenario. Let $\{\sigma^i\} \subset \mathbb{C}$ be a sequence of shifts and $\bar{U} \in \mathbb{C}^{n \times k}$ be an approximate recycling space of the matrix $A$. Define the matrices

$$U^{(\sigma^i)} = \left[ \prod_{i=1, i \neq j}^{s+1} (A + \sigma^i I) \right] \bar{U} R^{-1} \in \mathbb{C}^{n \times k} \quad \text{and} \quad C = \left[ \prod_{i=1}^{s+1} (A + \sigma^i) \right] \bar{U} R^{-1}. \quad (43)$$

In addition, they proposed a stable implementation of this procedure required as few matrix-vector multiplication as possible (Algorithm 4); see [21] for more details.

In practice, the choice of multiple deflation spaces is a delicate task. The selection algorithm has a profound impact on the overall performance of the enrichment algorithms (Algorithm 3) on some problems. Furthermore, it would require an additional $k$ vectors of storage for each shift. Therefore, the problem, selecting the optimal recycling space for each shift, is worth considering in the future.
Algorithm 4 Framework of ensemble learning for our system [21].

Input: \( A \in (\mathbb{C}^{n \times n}, \{\sigma^i\}_{i=1}^{s+1} \subset \mathbb{C}, U \in \mathbb{C}^{n \times k} \); 

Output: \( C \in \mathbb{C}^{n \times k} \) such that \( C^* C = I_k \), \( \{U^{(\sigma^i)}\}_{i=1}^{s+1} \subset \mathbb{C} \), such that 

\[
(A + \sigma_i I)U^{(\sigma_i)} = C \quad \text{for all } i;
\]

1: for \( i = 1 \) to \( s+1 \) do
2: \( U^{(\sigma^i)} = \bar{U} \)
3: end for

4: for \( i = 1 \) to \( s \) do
5: \( U^{(\sigma^i)} = \bar{U} \)
6: Compute the QR-factorization \( U^{(\sigma^{s+1})} = QR \)
7: \( U^{(\sigma^{s+1})} \leftarrow Q \)
8: for \( i = 1 \) to \( s \) do
9: \( \hat{U} = U^{(\sigma^i)}R^{-1} \)
10: \( k_1 = (j - 1 \bmod s + 1) + (1 - \left\lfloor \frac{j - 1}{s + 1} \right\rfloor)(s + 1) \)
11: \( k_2 = (j + 1 \bmod s + 1) + (1 - \left\lfloor \frac{j}{s + 1} \right\rfloor)(s + 1) \)
12: \( U^{(\sigma^i)} = U^{(\sigma^{k_1})} + (\sigma^{k_2} - \sigma^i)\hat{U} \)
13: end for
14: end for
15: \( C = AU^{(\sigma_{s+1})} + \sigma_{s+1} U^{(\sigma_{s+1})} \)
16: Compute the QR-factorization \( C = QR \)
17: \( C \leftarrow Q \)
18: for \( i = 1 \) to \( s+1 \) do
19: \( U^{(\sigma^i)} = U^{(\sigma^i)}R^{-1} \)
20: end for