RECYCLING BICG*

KAPIL AHUJA†, ERIC DE STURLER†, EUN R. CHANG†, AND SERKAN GUGERCIN†

Abstract. Engineering problems frequently require solving a sequence of dual linear systems. This paper introduces *Recycling BiCG*, which recycles two Krylov subspaces from one pair of linear systems to the next pair. An augmented bi-Lanczos algorithm and a modified two-term recurrence are developed to include recycling in the iteration. The recycle spaces are built from the approximate invariant subspaces corresponding to eigenvalues close to the origin. Experiments on convection-diffusion and model reduction problems give promising results.

Key words. Krylov subspace recycling, bi-Lanczos method, Petrov-Galerkin formulation.

AMS subject classifications. 65F10, 65N22, 93C05

1. Introduction.

We focus on solving the sequence of dual linear systems,

$$A^{(j)}x^{(j)} = b^{(j)}, \quad A^{(j)*}\tilde{x}^{(j)} = \tilde{b}^{(j)}, \tag{1.1}$$

where $A^{(j)} \in \mathbb{C}^{n\times n}$ and $b^{(j)}, \tilde{b}^{(j)} \in \mathbb{C}^n$ vary with j, and the matrices $A^{(j)}$ are large and sparse. In many practical applications, the change from one pair of systems to the next is small, such as, for example, in the Iterative Rational Krylov Algorithm (IRKA) for model reduction [5]. While solving one pair of systems, we select subspaces of the Krylov subspaces corresponding to $A^{(j)}$ and $A^{(j)*}$ and use these to accelerate solving the next pair. This process is called *Krylov subspace recycling*, and leads to faster convergence for the next pair of systems. The convergence of Krylov subspace methods for solving a linear system, to a great extent, depends on the spectrum of the matrix. Moreover, the deflation of eigenvalues close to the origin improves the convergence rate [7]. Eigenvalues can be deflated by including the corresponding eigenvectors in the Krylov subspace. Therefore, we use the approximate left and right invariant subspaces corresponding to small eigenvalues (in absolute value) for recycling.

For solving a single linear system, 'recycling' has been used in the GCROT [2] and the GMRES-DR [7] algorithms. For solving a sequence of linear systems, this idea was first proposed in [8] where it is applied to the GCROT and the GCRO-DR algorithms. The idea is further adapted in the RMINRES [13] algorithm. GCROT as in [8], GCRO-DR, and RMINRES all focus on solving a sequence of single systems rather than a sequence of two dual systems as is the focus here. For a comprehensive discussion of recycling algorithms see [8].

Since the most relevant Krylov subspace method for solving systems of the type (1.1) is the BiConjugate Gradient (BiCG) algorithm [3], we focus on Krylov subspace recycling for BiCG. We refer to our recycling BiCG method as RBiCG.

To simplify notation, we drop the superscript j in (1.1). At any particular point in the sequence of systems, we refer to $Ax = b$ as the primary system and $A^*\tilde{x} = \tilde{b}$ as the dual system. Throughout the paper, $||\cdot||$ refers to the two-norm, $(\cdot,\cdot)$ refers to the standard inner product, and e_i is the i-th canonical basis vector.

The remainder of this paper consists of five sections. In Section 2, we discuss the BiCG algorithm. The RBiCG iteration using an existing recycle space is derived in

*This material is based upon work supported by the National Science Foundation under Grant No. NSF-EAR 0530643.

†Department of Mathematics, Virginia Tech, Blacksburg, VA 24061.

Section 3. Section 4 shows how to compute a recycle space cheaply. We give results in Section 5, and provide conclusions in Section 6.

2. Background.

We briefly describe the BiCG algorithm for future reference. For the primary system, let x_0 be the initial guess with residual $r_0 = b - Ax_0$. Krylov subspace methods find approximate solutions by projection onto the Krylov subspace associated with A and r_0 [12]. The i-th solution iterate is given by

$$x_i = x_0 + \varrho_i, \tag{2.1}$$

where $\varrho_i \in K^i(A, r_0) \equiv span\{r_0,\ Ar_0,\ A^2r_0,\ \cdots,\ A^{i-1}r_0\}$ is defined by some projection. The BiCG method defines this projection using the Krylov subspace associated with the dual system, leading to two bi-orthogonal bases and a pair of cheap three-term or coupled two-term recurrences. This method is called the bi-Lanczos method [6, 3]. First we initialize the Lanczos vectors as follows:

$$v_1 = r_0/||r_0||, \quad \tilde{v}_1 = \tilde{r}_0/||\tilde{r}_0||. \tag{2.2}$$

Defining $V_i = [v_1\ v_2\ \ldots\ v_i]$ and $\tilde{V}_i = [\tilde{v}_1\ \tilde{v}_2\ \ldots\ \tilde{v}_i]$, the $(i+1)$-th Lanczos vectors are given by

$$\begin{aligned} v_{i+1} &= Av_i - V_i\tau \perp \tilde{V}_i, \\ \tilde{v}_{i+1} &= A^*\tilde{v}_i - \tilde{V}_i\tilde{\tau} \perp V_i, \end{aligned} \tag{2.3}$$

The bi-orthogonality condition leads to a pair of 3-term recurrences (see [9]) such that computation of the $(i+1)$-th Lanczos vector requires only the i-th and the $(i-1)$-th Lanczos vector. These 3-term recurrences are called the bi-Lanczos relations, and they are defined as follows:

$$\begin{aligned} AV_i &= V_{i+1}\underline{T}_i = V_iT_i + t_{i+1,i}v_{i+1}e_i^T, \\ A^*\tilde{V}_i &= \tilde{V}_{i+1}\underline{\tilde{T}}_i = \tilde{V}_i\tilde{T}_i + \tilde{t}_{i+1,i}\tilde{v}_{i+1}e_i^T, \end{aligned} \tag{2.4}$$

where $t_{i+1,i}$ is the last element of the last row of the tridiagonal $\underline{T}_i$, and similarly, $\tilde{t}_{i+1,i}$ is the last element of the last row of the tridiagonal $\underline{\tilde{T}}_i$.

The next step is to find approximate solutions by projections. Two standard ways of finding an approximate solution over the Krylov subspace include the Ritz-Galerkin approach (the CG algorithm) and the minimum norm residual approach (the GMRES algorithm) [12]. However, to exploit the efficiency of the short-term recurrences in the bi-Lanczos algorithm, we utilize the bi-orthogonality condition to define a cheap projection. This leads to a Petrov-Galerkin approach. Since the columns of V_i form a basis for $K^i(A, r_0)$, we can define ϱ_i in (2.1) as $\varrho_i = V_iy_i$, and the Petrov-Galerkin condition then implies

$$r_i = b - A(x_0 + \varrho_i) = r_0 - AV_iy_i \perp \tilde{V}_i. \tag{2.5}$$

The vector y_i is defined by this orthogonality condition, and in turn defines ϱ_i and the solution iterate x_i (2.1). Further simplifications lead to the standard BiCG algorithm.

3. Recycling BiCG: Using a Recycle Space.

In this section we modify the BiCG algorithm to use a given recycle space. First, we briefly describe the recycling idea used in the GCRO-DR algorithm. After solving the j-th primary system in (1.1), GCRO-DR retains k approximate eigenvectors of $A^{(j)}$, which are used to compute the matrices $U,\ C \in \mathbb{C}^{n\times k}$, such that range$(U)$ is the approximate invariant subspace of

$A^{(j)}$ (and hopefully also of $A^{(j+1)}$), $A^{(j+1)}U = C$ and $C^*C = I$. It then adapts the Arnoldi process to compute the orthogonal basis for the Krylov subspace such that each new Krylov vector v is also orthogonal to range(C). This produces the Arnoldi relation

$$(I - CC^*)AV_i = V_{i+1}\underline{H}_i,$$

where $\underline{H}_i$ is an $(i+1) \times i$ upper Hessenburg matrix. GCRO-DR finds the optimal solution over the (direct) sum of the recycle space, range(U), and the new Krylov space generated, range(V_i).

For our RBiCG too, the columns of the matrix U define the basis of the primary system recycle space, and we define $C = A^{(j+1)}U$, where U is derived from the approximate right invariant subspace of $A^{(j)}$. Similarly, the columns of the matrix $\tilde{U}$ define the basis of the dual system recycle space, and we define $\tilde{C} = A^*\tilde{U}$. Analogously, $\tilde{U}$ is derived from the approximate left invariant subspace of $A^{(j)}$. As in GCRO-DR, the number of vectors selected for recycling is denoted by k, so that U, $\tilde{U}$, C, and $\tilde{C}$ are $n \times k$ matrices. Here, C is no longer an orthogonal matrix, instead the columns of C and $\tilde{C}$ are bi-orthogonality. An alternative choice is discussed later in the section.

The rest of this section is divided into two subsections. In Section 3.1, we derive augmented bi-Lanczos algorithm that computes basis for our two Krylov subspaces that satisfy an augmented bi-orthogonality condition. The two-term recurrence for the solution update in RBiCG is derived in Section 3.2.

3.1. Augmented Bi-Lanczos Algorithm. The standard bi-Lanczos algorithm computes columns of V_i and $\tilde{V}_i$ such that, in exact arithmetic, $V_i \perp_b \tilde{V}_i$ where $\perp_b$ denotes bi-orthogonality and implies that $\tilde{V}_i^* V_i$ is a diagonal matrix. Since we recycle spaces U and $\tilde{U}$, the bi-Lanczos algorithm can be modified to compute the columns of V_i and $\tilde{V}_i$ such that either

$$[U\ V_i] \perp_b \left[\tilde{U}\ \tilde{V}_i\right] \tag{3.1}$$

or

$$[C\ V_i] \perp_b \left[\tilde{C}\ \tilde{V}_i\right]. \tag{3.2}$$

We decided to build bi-orthogonality given by (3.2) because this leads to simpler algebra. It also has the advantage that the RBiCG algorithm has a form similar to the standard BiCG algorithm. Next, we derive the recurrences that build the above bi-orthogonality (3.2). We assume

$$C \perp_b \tilde{C}.$$

This bi-orthogonality is cheaply enforced while building the recycle space. As in the standard BiCG algorithm, we assume v_1 and $\tilde{v}_1$ are available from the initial residuals r_0 and $\tilde{r}_0$. We make this statement more precise later in the paper. The $(i+1)$-th Lanczos vector for the primary system is computed by $\hat{v}_{i+1} = Av_i - V_i\tau - C\rho \perp \left[\tilde{C}\ \tilde{V}_i\right]$. This implies

$$\begin{bmatrix} \tilde{C}^* \\ \tilde{V}_i^* \end{bmatrix} (Av_i - V_i\tau - C\rho) = 0. \tag{3.3}$$

Using (3.2), we get the following equations:

$$\begin{aligned} \mathcal{D}_c\rho &= \tilde{C}^*Av_i, \\ \mathcal{D}_i\tau &= \tilde{V}_i^*Av_i, \end{aligned} \tag{3.4}$$

where $\mathcal{D}_c = \tilde{C}^*C$ and $\mathcal{D}_i = \tilde{V}_i^*V_i$. As in the standard BiCG algorithm we assume $\mathcal{D}_i$ is nonsingular ("serious breakdown" does not occur). We ensure $\mathcal{D}_c$ is nonsingular while building the recycle space. Substituting τ and ρ from (3.4) into (3.3) gives the $(i+1)$-Lanczos vector. As in the standard bi-Lanczos algorithm, the bi-orthogonality condition (3.2) converts the full recurrence of $\hat{v}_{i+1}$ to a $(3+k)$-term recurrences where k is the number of columns of C. This implies the computation of the $(i+1)$-th Lanczos vector require the i-th and $(i-1)$-th Lanczos vectors and k columns of C. Similarly, we get a $(3+k)$ term recurrence for computing the Lanczos vectors for the dual system. These pair of $(3+k)$-term recurrences in the matrix form are termed as augmented bi-Lanczos relations. The augmented bi-Lanczos relation for the primary system is given as follows:

$$(I - C\hat{C}^*)AV_i = V_{i+1}\underline{T}_i, \tag{3.5}$$

where $\hat{C} = \begin{bmatrix} \frac{\tilde{c}_1}{c_1^*\tilde{c}_1} & \frac{\tilde{c}_2}{c_2^*\tilde{c}_2} & \cdots & \frac{\tilde{c}_k}{c_k^*\tilde{c}_k} \end{bmatrix}$ and $\underline{T}_i$ is the $(i+1)\times i$ tridiagonal matrix. Similarly, we get the augmented bi-Lanczos relation for the dual system as follows:

$$(I - \tilde{C}\check{C}^*)A^*\tilde{V}_i = \tilde{V}_{i+1}\underline{\tilde{T}}_i, \tag{3.6}$$

where $\check{C} = \begin{bmatrix} \frac{c_1}{\tilde{c}_1^*c_1} & \frac{c_2}{\tilde{c}_2^*c_2} & \cdots & \frac{c_k}{\tilde{c}_k^*c_k} \end{bmatrix}$.

Note that we can write (3.5) and (3.6) as

$$A_1V_i = V_{i+1}\underline{T}_i, \quad \text{where} \quad A_1 = (I - C\mathcal{D}_c^{-1}\tilde{C}^*)A(I - C\mathcal{D}_c^{-1}\tilde{C}^*), \tag{3.7}$$

$$A_1^*\tilde{V}_i = \tilde{V}_{i+1}\underline{\tilde{T}}_i, \quad \text{where} \quad A_1^* = (I - \tilde{C}\mathcal{D}_c^{-*}C^*)A^*(I - \tilde{C}\mathcal{D}_c^{-*}C^*), \tag{3.8}$$

since $A_1V_i = (I - C\hat{C}^*)AV_i$ and $A_1^*\tilde{V}_i = (I - \tilde{C}\check{C}^*)A^*\tilde{V}_i$. Here $\tilde{C}^*C = \mathcal{D}_c$ is a diagonal matrix. This new form of the augmented bi-Lanczos relations simplifies the algebra while deriving the RBiCG solution update recurrence because the operators in (3.7) and (3.8) are conjugate transpose of each other. The next step is to obtain the solution update recurrence that utilizes the recycle space. This is described in the next subsection.

3.2. Updating the Solution Using the Recycle Space. We replace the i-th solution iterate of the standard BiCG algorithm [12]

$$x_i = x_0 + V_iy_i, \quad \tilde{x}_i = \tilde{x}_0 + \tilde{V}_i\tilde{y}_i.$$

by

$$x_i = x_0 + Uz_i + V_iy_i, \quad \tilde{x}_i = \tilde{x}_0 + \tilde{U}\tilde{z}_i + \tilde{V}_i\tilde{y}_i, \tag{3.9}$$

in the RBiCG algorithm (as in the GCRO-DR and the RMINRES algorithms). In the standard BiCG algorithm, y_i and $\tilde{y}_i$ are computed by enforcing the following orthogonality conditions (Petrov-Galerkin approximation):

$$r_i = r_0 - AV_iy_i \perp \tilde{V}_i, \quad \tilde{r}_i = \tilde{r}_0 - A^*\tilde{V}_i\tilde{y}_i \perp V_i.$$

With recycling, the Petrov-Galerkin approximation gives

$$r_i = r_0 - AUz_i - AV_iy_i \perp \left[\tilde{C}\ \tilde{V}_i\right], \quad \tilde{r}_i = \tilde{r}_0 - A^*\tilde{U}\tilde{z}_i - A^*\tilde{V}_i\tilde{y}_i \perp [C\ V_i]. \tag{3.10}$$

For the primary system, the above orthogonality condition leads to the following:

$$\begin{bmatrix} \tilde{C}^* \\ \tilde{V}_i^* \end{bmatrix} \left[r_0 \; - \; A[U\ V_i] \begin{bmatrix} z_i \\ y_i \end{bmatrix} \right] = 0. \tag{3.11}$$

The steps are similar for the dual system. The actual computation is somewhat more efficient than the above equation. Lets first look at r_0. Defining $\zeta = ||(I - C\hat{C}^*)r_0||$ we get

$$\begin{aligned} r_0 &= C\hat{C}^*r_0 + r_0 - C\hat{C}^*r_0, \\ &= C\hat{C}^*r_0 + \zeta v_1, \qquad \text{taking } v_1 = \tfrac{(I-C\hat{C}^*)r_0}{\zeta}, \\ &= C\hat{C}^*r_0 + \zeta V_{i+1}e_1, \\ &= [C\ V_{i+1}] \begin{bmatrix} \hat{C}^*r_0 \\ \zeta e_1 \end{bmatrix}. \end{aligned} \tag{3.12}$$

It is here that we define our first Lanczos vector v_1. Note that v_1 is obtained from the initial residual r_0. Now let's look at the second term of (3.11). For the following derivation, augmented bi-Lanczos relation (3.5) is used.

$$\begin{aligned} A\,[U \quad V_i] \begin{bmatrix} z_i \\ y_i \end{bmatrix} &= [AU \quad AV_i] \begin{bmatrix} z_i \\ y_i \end{bmatrix}, \\ &= \left[C \quad C\hat{C}^*AV_i + V_{i+1}\underline{T}_i\right] \begin{bmatrix} z_i \\ y_i \end{bmatrix}, \\ &= [C \quad V_{i+1}] \begin{bmatrix} I & \hat{C}^*AV_i \\ 0 & \underline{T}_i \end{bmatrix} \begin{bmatrix} z_i \\ y_i \end{bmatrix}. \end{aligned} \tag{3.13}$$

Substituting (3.12) and (3.13) in (3.11) gives

$$\begin{bmatrix} \tilde{C}^* \\ \tilde{V}_i^* \end{bmatrix} [C\ V_{i+1}] \left[\begin{bmatrix} \hat{C}^*r_0 \\ \zeta e_1 \end{bmatrix} - \begin{bmatrix} I & \hat{C}^*AV_i \\ 0 & \underline{T}_i \end{bmatrix} \begin{bmatrix} z_i \\ y_i \end{bmatrix} \right] = 0. \tag{3.14}$$

Using the bi-orthogonality condition (3.2) in the above equation we get[1]

$$\begin{bmatrix} \hat{C}^*r_0 \\ \zeta e_1 \end{bmatrix} - \begin{bmatrix} I & \hat{C}^*AV_i \\ 0 & T_i \end{bmatrix} \begin{bmatrix} z_i \\ y_i \end{bmatrix} = 0. \tag{3.15}$$

Therefore, the equations for finding y_i and z_i are given by

$$\begin{aligned} y_i &= \zeta T_i^{-1}e_1, \\ z_i &= \hat{C}^*r_0 - \hat{C}^*AV_iy_i. \end{aligned}$$

As in the standard BiCG algorithm, the algorithm here fails if a singular tridiagonal is encountered [4]. For clarity of exposition we assume this does not occur. Substituting the above expressions in (3.9) leads to the following solution update for the primary system:

$$x_i = \chi_0 + (I - U\hat{C}^*A)V_iy_i, \tag{3.16}$$

[1]Note that the dimension of e_1 in (3.15) is one less than that of e_1 in (3.14), although both denote the first canonical vector.

where $\chi_0 = x_0 + U\hat{C}^* r_0$ and $y_i = \zeta T_i^{-1} e_1$. Next, as in the standard BiCG algorithm, we compute LDU decomposition of the tridiagonal matrix T_i to get a two-term recurrence. If

$$\begin{aligned} T_i &= L_i D_i R_i, \\ G_i &= (I - U\hat{C}^* A) V_i R_i^{-1}, \text{ and} \\ \varphi_i &= \zeta D_i^{-1} L_i^{-1} e_1, \end{aligned} \tag{3.17}$$

then the two-term recurrence for the solution update of the primary system is given by

$$x_i = x_{i-1} + \varphi_{i,i} g_i, \tag{3.18}$$

where g_i is the last column of G_i and $\varphi_{i,i}$ is the last entry of the vector φ_i. It is similar for the dual system. We now do a slight change of notation to make the future derivations simpler. Let x_{-1} be the initial guess and $r_{-1} = b - Ax_{-1}$ the corresponding initial residual. We define

$$x_0 = x_{-1} + U\hat{C}^* r_{-1} \text{ and } r_0 = (I - C\hat{C}^*) r_{-1}. \tag{3.19}$$

Similarly, for the dual system, let $\tilde{x}_{-1}$ be the initial guess and $\tilde{r}_{-1} = \tilde{b} - A^*\tilde{x}_{-1}$ the corresponding initial residual. We define

$$\tilde{x}_0 = \tilde{x}_{-1} + \tilde{U}\check{C}^* \tilde{r}_{-1} \text{ and } \tilde{r}_0 = (I - \tilde{C}\check{C}^*) \tilde{r}_{-1}. \tag{3.20}$$

x_0 in (3.18) is defined by (3.19). Note that in (3.17) we never compute any explicit matrix inverse. The matrices under consideration are either diagonal, or lower triangular, or upper triangular. Moreover, we can obtain the required information even without generating the tridiagonal matrix explicitly. The solution (x) is now updated using standard iteration vectors p and r, and scalars α and β. We skip the derivation of this and state the final result in Algorithm 1[2]. A detailed derivation is done in [1].

4. Recycling BiCG: Computing a Recycle Space.

In this section we are concerned with how to build U and $\tilde{U}$. In GCRO-DR [8] and RMINRES [13], approximate invariant subspaces have been successfully used to build the recycle space. It has been demonstrated there (in [8] and [13]) and in [7] that using the approximate invariant subspace corresponding to small eigenvalues (in absolute value) is effective. For RBiCG we follow the same strategy.

Harmonic Ritz vectors [10] provide an approximate invariant subspace cheaply, and to compute them we need the Lanczos vectors. Instead of using all the Lanczos vectors to build the recycle space, we update the recycle space periodically to keep the memory requirements modest [13]. The iteration process between two updates of the recycle space is referred to as a "cycle". The length of the cycle "s" refers to the number of iterations between updating the recycle space. Let V_j and $\tilde{V}_j$ contain the Lanczos vectors generated during the j^{th} cycle, and let Υ_j and $\tilde{\Upsilon}_j$ be V_j and $\tilde{V}_j$ respectively with one previous and one subsequent Lanczos vector. The augmented bi-Lanczos relations for the j^{th} cycle are now given by

$$\begin{aligned} (I - C\hat{C}^*) A V_j &= \Upsilon_j \Gamma_j \quad \text{and} \\ (I - \tilde{C}\check{C}^*) A^* \tilde{V}_j &= \tilde{\Upsilon}_j \tilde{\Gamma}_j, \end{aligned} \tag{4.1}$$

[2]This version assumes that a recycle space exists. Also, the algorithmic improvements to make the code faster are not shown here.

Algorithm 1. *RBiCG*

```
1. Assuming $U$ and $\tilde{U}$ are available compute $C = AU$ and $\tilde{C} = A^*\tilde{U}$.
2. Choose initial guesses $x_{-1}$ and $\tilde{x}_{-1}$ and compute $x_0$, $\tilde{x}_0$, $r_0$, and $\tilde{r}_0$ using (3.19) and (3.20) respectively.
3. if $(r_0, \tilde{r}_0) = 0$ then initialize $\tilde{r}_0$ to a random vector.
4. Set $p_0 = 0$, $\tilde{p}_0 = 0$, and $\beta_0 = 0$. Let $tol$ be the convergence tolerance, and $itn$ be the maximum iterative steps the algorithm can take.
5. for $i = 1 \ldots itn$ do
◇    $p_i = r_{i-1} + \beta_{i-1}p_{i-1}$.
◇    $\tilde{p}_i = \tilde{r}_{i-1} + \bar{\beta}_{i-1}\tilde{p}_{i-1}$.
◇    $q_i = (I - U\hat{C}^*A)p_i$.
◇    $\tilde{q}_i = (I - \tilde{U}\check{C}^*A^*)\tilde{p}_i$.
◇    $\alpha_i = (\tilde{r}_{i-1}, r_{i-1})/(\tilde{p}_i, Aq_i)$.
◇    $x_i = x_{i-1} + \alpha_i q_i$.
◇    $\tilde{x}_i = \tilde{x}_{i-1} + \bar{\alpha}_i\tilde{q}_i$.
◇    $r_i = r_{i-1} - \alpha_i Aq_i$.
◇    $\tilde{r}_i = \tilde{r}_{i-1} - \bar{\alpha}_i A^*\tilde{q}_i$.
◇    if $||r_i|| \leq tol$ and $||\tilde{r}_i|| \leq tol$ then break.
◇    $\beta_i = (\tilde{r}_i, r_i)/(\tilde{r}_{i-1}, r_{i-1})$.
5. end for.
```

where Γ_j and $\tilde{\Gamma}_j$ are the tridiagonals from (3.5) and (3.6) with an extra row at the top and at the bottom.

For computing the Harmonic Ritz pairs, we also need to decide which vector spaces to use for building the recycle space. The discussion in this paragraph, and the next, concerns only the primary system ($Ax = b$). Similar results hold for the dual system ($A^*\tilde{x} = \tilde{b}$). Let columns of U_{j-1} denote the recycle space generated at the end of the $(j-1)$ cycle for the current linear system, and columns of U be the recycle space available from the previous linear system. We want to obtain U_j from V_j, U_{j-1}, and U. There are many options for selecting U_j [13]. For RBiCG we build U_j from $range([U_{j-1}\ V_j])$ because it keeps the algebra simple.

Let

$$\Phi_j = [U_{j-1}\ V_j]\,,\ \tilde{\Phi}_j = \left[\tilde{U}_{j-1}\ \tilde{V}_j\right].$$

In RMINRES [13] harmonic Ritz pairs of A with respect to the subspace $range(A\Phi_j)$ have been successfully used to build the recycle space. Since here we work in a Petrov-Galerkin framework, using harmonic Ritz pairs with respect to the subspace $range(A^*\tilde{\Phi}_j)$ is more intuitive and leads to cheaper computations later. Let (λ, u) denote the harmonic Ritz pair of A we are interested in. Then, λ and u are defined by the condition [10]

$$(Au - \lambda u) \perp range\left(A^*\tilde{\Phi}_j\right), \tag{4.2}$$

where $u \in range(\Phi_j)$ (since we build U_j from $range([U_{j-1}\ V_j])$). Further simplifications convert (4.2) to a generalized eigenvalue problem. If the columns of W_j and $\tilde{W}_j$ denote the chosen right and left eigenvectors of this generalized eigenvalue problem (corresponding to the eigenvalues close to the origin), then $U_j = \Phi_j W_j$ and $\tilde{U}_j = \tilde{\Phi}_j\tilde{W}_j$.

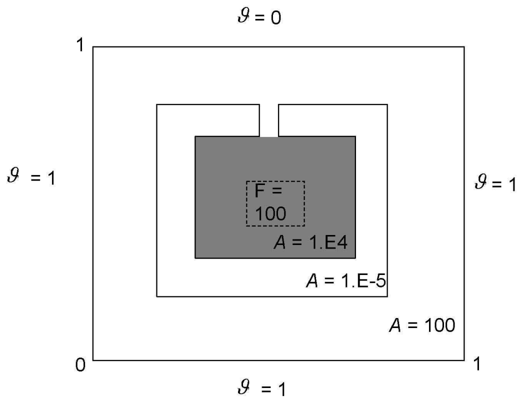

FIG. 5.1. *Coefficients for the PDE.*

The generalized eigenvalue problem obtained is of much smaller dimension $(k+s)\times(k+s)$ as compared to A, which is $n \times n$. Thus, computing its left and right eigenvectors is cheap. However, building this eigenvalue problem itself is expensive. We perform algebraic simplifications to build it cheaply. These simplifications, and some special cases, can be referred in [1].

5. Results.

For testing our algorithms, we use the linear system obtained by finite difference discretization of a partial differential equation. The partial differential equation used is [11]

$$-(\mathcal{A}\vartheta_x)_x - (\mathcal{A}\vartheta_y)_y + \mathcal{B}(x,y)\vartheta_x = \mathcal{F},$$

with $\mathcal{A}$ as shown in Figure 5.1, $\mathcal{B}(x,y) = 2e^{2(x^2+y^2)}$, and $\mathcal{F} = 0$ everywhere except in a small square in the center (see Figure 5.1) where $\mathcal{F} = 100$. The domain is $(0,1)\times(0,1)$ with Dirichlet boundary conditions

$$\begin{aligned}&\vartheta(0,y) = \vartheta(1,y) = \vartheta(x,0) = 1,\\ &\vartheta(x,1) = 0.\end{aligned}$$

The standard second order central difference scheme is used for the discretization, and the mesh width is taken as 1/128. This leads to a nonsymmetric linear system of 127^2 unknowns. The linear system is preconditioned by a Crout version of ILUT factorization [9] with a drop tolerance of 0.2.

Figure 5.2 shows the benefit of recycling in BiCG. The linear system is solved twice with RBiCG. The subspace is recycled from the first run to the second. This is the "ideal" case for recycling; "We do this to exclude the effects of right-hand sides having slightly different eigenvector decompositions" [8]. Note that the second solve uses 25% fewer iterations than the first solve. For this experiment we take $s = 80$ and $k = 20$. These are chosen based on experience with other recycling algorithms [8, 13]. We do a third run too (that uses the recycle space generated in the second run) to show that the recycle space gets accurate with more runs. This is evident in the figure (the algorithm converges fastest in the third run).

Next, we briefly show the application of RBiCG in IRKA [5] that is used for model reduction. IRKA requires linear solves for building matrices

$$\begin{aligned}\mathcal{V}_i &= [A(\sigma_{i_1})^{-1}b,\ \ldots,\ A(\sigma_{i_j})^{-1}b],\\ \mathcal{W}_i &= [A(\sigma_{i_1})^{-T}\tilde{b},\ \ldots,\ A(\sigma_{i_j})^{-T}\tilde{b}],\end{aligned}$$

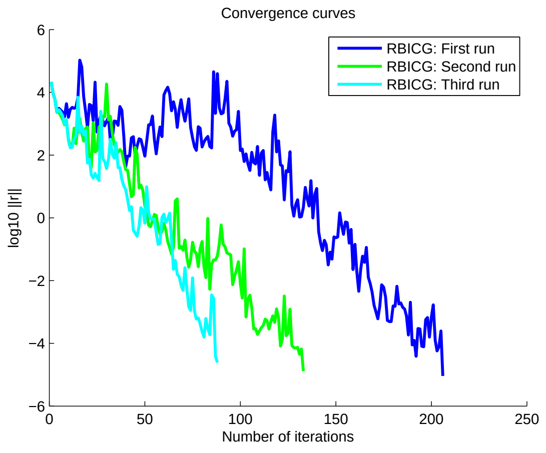

FIG. 5.2. *Recycling in BiCG.*

where A is a matrix that depends on the scalar σ. This is a good example to show the usefulness of RBiCG because of the following reasons. A column of $\mathcal{V}_i$ denotes the primary system and the corresponding column of $\mathcal{W}_i$ denotes the dual system. The change in σ, both horizontally (along the columns of $\mathcal{V}_i$ and $\mathcal{W}_i$) and vertically (with change in i), is not substantial.

We apply the RBiCG algorithm to the above solves as follows. While solving the system $A(\sigma_{i+1_m})^{-1}b$ (also the dual system), with $m = 1, \ldots, j$, we pick the recycle space generated in the previously solved system that has σ closest to σ_{i+1_m}. The pool for picking σ currently is $\sigma_{i_1}, \ldots \sigma_{i_j}, \sigma_{i+1_1}, \ldots, \sigma_{i+1_{m-1}}$.

Our 1357×1357 test dynamical system comes from the rail models (courtesy: Dr. Peter Benner). We take $j = 6$ and initialize the σ's as $logspace(-5, 0.7, 6)$. We do the linear solves both with RBiCG and the standard BiCG algorithm. Figure 5.3 shows the convergence curves at $i = 3$. This is for the primary systems (the columns of $\mathcal{V}_i$). Similar graph exists for the dual systems (the columns of $\mathcal{W}_i$). The curves in bold are corresponding to the systems without recycling, and the dashed curves are corresponding to the systems that use recycling. It is evident that the systems that use a recycle space converge in fewer iterations as compared to the systems that do not use recycling. Note that only three systems use recycling. This is because the other three converge fast, and applying recycling to them is not useful. To summarize, RBiCG works well in IRKA for small models. For larger models, deterioration of the space $range\left(A^* \left[\tilde{U}_{j-1} \; \tilde{V}_j\right]\right)$ makes the algorithm converge poorly. Having a good basis for this space fixes the problem, and we are currently working to obtain such a basis cheaply.

6. Conclusion.

In this paper, we apply Krylov subspace recycling to the BiCG algorithm. We first assume the recycle space exists and modify the algorithm to utilize this space. Second, we show how to build the recycle space cheaply. The resulting algorithm is termed RBiCG.

We test our algorithms on a linear system arising from the finite difference discretization of a PDE. To simulate slowly changing linear systems, we solve our linear system twice where-in the subspace is recycled from the first run to the second. This example is an “ideal” case of recycling because it best exemplifies the usefulness of the

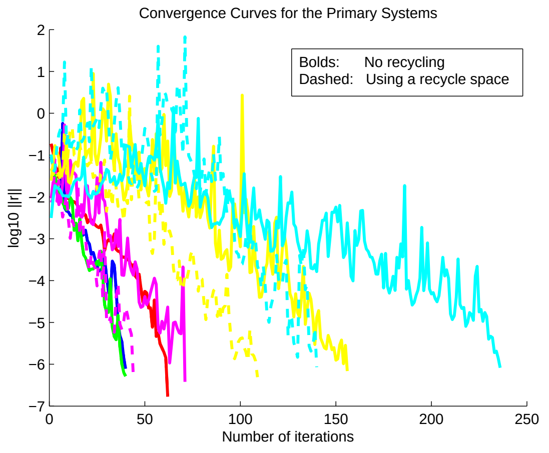

FIG. 5.3. *Using RBiCG in IRKA.*

recycling algorithms. The results indicate that the RBiCG algorithm can help solve slowly varying sequence of dual linear systems faster. A "real life" example arises while performing model reduction using IRKA [5]. Utilizing RBiCG inside IRKA is currently being researched upon.

REFERENCES

[1] K. Ahuja and E. de Sturler (advisor). Recycling bi-Lanczos algorithms: BiCG, CGS, and BiCGSTAB. MS Thesis, Department of Mathematics, Virginia Tech, http://scholar.lib.vt.edu/theses/available/etd-08252009-161256/, 2009.

[2] E. de Sturler. Truncation strategies for optimal Krylov subspace methods. *SIAM Journal on Numerical Analysis*, 36(3):864–889, 1999.

[3] R. Fletcher. Conjugate gradient methods for indefinite systems. *Lecture Notes in Mathematics, Springer Berlin-Heidelberg*, 506:73–89, 1976.

[4] A. Greebaum. *Iterative Methods for Solving Linear Systems.* Society for Industrial and Applied Mathematics, 1997.

[5] S. Gugercin, A. C. Antoulas, and C. A. Beattie. H2 model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30(2):609–638, 2008.

[6] C. Lanczos. Solution of systems of linear equations by minimized iterations. *J. Res. Natl. Bur. Stand.*, 49:33–53, 1952.

[7] R. B. Morgan. GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1):20–37, 2002.

[8] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM Journal on Scientific Computing*, 28(5):1651–1674, 2006.

[9] Y. Saad. *Iterative Methods for Sparse Linear Systems, 1st Edition.* PWS (and now ITP), 1996.

[10] 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*, 17:401–425, 1996.

[11] H. A. van der Vorst. 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, 1992.

[12] H. A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems.* Cambridge University Press, 2003.

[13] S. Wang, E. de Sturler, and G. H. Paulino. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69(12):2422–2468, 2006.