SIAM J. SCI. COMPUT.
Vol. 33, No. 5, pp. 3037–3056

RESTARTING THE NONSYMMETRIC LANCZOS ALGORITHM FOR EIGENVALUES AND LINEAR EQUATIONS INCLUDING MULTIPLE RIGHT-HAND SIDES*

RONALD B. MORGAN† AND DYWAYNE A. NICELY‡

Abstract. A restarted nonsymmetric Lanczos algorithm is given for computing eigenvalues and both right and left eigenvectors. The restarting limits the storage so that finding eigenvectors is practical. Restarting also makes it possible to deal with roundoff error in new ways. We give a scheme for avoiding near-breakdown and discuss maintaining biorthogonality. A system of linear equations can be solved simultaneously with the eigenvalue computations. Deflation from the presence of the eigenvectors allows the linear equations to generally have good convergence in spite of the restarting. The right and left eigenvectors generated while solving the linear equations can be used to help solve systems with multiple right-hand sides.

Key words. eigenvalues, eigenvectors, left eigenvectors, linear equations, deflation, nonsymmetric Lanczos, multiple right-hand sides

AMS subject classifications. 65F10, 65F15, 15A06, 15A18

DOI. 10.1137/100799265

1. Introduction.

The nonsymmetric Lanczos algorithm [20] is a way to find eigenvalues of large sparse matrices. It is also the basis of many of the leading methods for solving large nonsymmetric systems of linear equations. Nonsymmetric Lanczos is very efficient because of three-term recurrences for generating biorthogonal bases for its Krylov subspaces. However, it is not popular for computing eigenvalues and eigenvectors because of roundoff error concerns. Also, since it is a nonrestarted method, a large amount of storage is needed for the eigenvector computation. So the implicitly restarted Arnoldi method (IRAM) [46] is generally used for finding eigenvalues and eigenvectors of a nonsymmetric matrix. However, the orthogonalization expense in IRAM can be significant as a proportion of total computations, especially for fairly sparse matrices.

We give an approach for restarting nonsymmetric Lanczos, and we use it to compute both right and left eigenvectors. Right and left approximate eigenvectors are saved at the restart and used for the next cycle. Linear equations can be solved simultaneously, and the presence of approximate eigenvectors keeps the convergence from being slowed as much by the restarting. This approach is called NLan-DR, for nonsymmetric Lanczos with deflated restarting. It gives an alternative to IRAM when both right and left eigenvectors are desired. The restarting limits the storage, but it also allows for new approaches for dealing with roundoff error. We consider both the roundoff effects of the occurrence of near-breakdown and the loss of biorthogonality.

Our application here is the solution of linear equations with multiple right-hand sides. Right and left eigenvectors are used to deflate eigenvalues. The eigenvectors are

*Received by the editors June 17, 2010; accepted for publication (in revised form) August 2, 2011; published electronically October 27, 2011. This work was partially supported by National Science Foundation Computational Mathematics Program grant 0310573.

http://www.siam.org/journals/sisc/33-5/79926.html

†Department of Mathematics, Baylor University, Waco, TX 76798-7328 (Ronald_Morgan@baylor.edu). This author's research was supported by the Baylor University Sabbatical Program.

‡Division of Math and Science, Ohio University-Chillicothe, Chillicothe, OH 45601-2178 (nicely@ohio.edu).

found during solution of the first right-hand side and then are available for the other right-hand sides. We give a method called deflated BiCGStab or D-BiCGStab that has first a projection over approximate eigenvectors and then solution with BiCGStab.

We mention quickly that there are other applications that use both right and left eigenvectors that we will not consider. These include estimating the conditioning of an eigenvalue, control theory [54], Perron theory [18], dynamical systems [9], and graph theory [47]. Another possible application which we do not pursue here is to dual linear systems.

Section 2 has a review of nonsymmetric Lanczos and of some restarted methods for both nonsymmetric and symmetric problems. Section 3 gives the new method NLan-DR and has some tests. Discussion of controlling near-breakdown is in section 4. Then section 5 addresses linear equations with multiple right-hand sides.

2. Review.

2.1. Nonsymmetric Lanczos for eigenvalues. Lanczos gave his algorithm in 1950 [20] as a way to tridiagonalize a matrix, but he also realized that it could be used iteratively and that some eigenvalues could be found with few iterations. The nonsymmetric Lanczos algorithm generates bases for two Krylov subspaces, one with A and one with A^T. Three-term recurrences make these bases orthogonal to each other (biorthogonal). If we let $\{v_1, \ldots v_j\}$ be the basis for the Krylov subspace with A and $\{w_1, \ldots w_j\}$ be the basis for the Krylov subspace with A^T, then $w_i^* v_p = 0$ (or 1 if $i = p$). The recurrences [38] for generating the bases are

$$\delta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1}$$

and

$$\beta_{j+1} w_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}.$$

These can be put together for all iterations as

$$A V_m = V_{m+1} T_{m+1,m} \tag{2.1}$$

and

$$A^T W_m = W_{m+1} T_{m,m+1}^T, \tag{2.2}$$

where $T_{m+1,m+1}$ is an $m+1$ by $m+1$ tridiagonal matrix and $T_{m+1,m}$ and $T_{m,m+1}$ are portions of it.

The method breaks down when the v_{j+1} and w_{j+1} vectors are orthogonal. If they are nearly orthogonal, this is called near-breakdown, and there can be significant error. Parlett, Taylor, and Liu [36] gave a look-ahead method to prevent near-breakdown in nonsymmetric Lanczos. See also [3, 17]. Cullum and Willoughby [10] implemented nonsymmetric Lanczos without look-ahead but used complex arithmetic partly to lessen the chance of near-breakdown. There have been several works on implicitly restarting the nonsymmetric Lanczos algorithm; see, for example, [16, 41]. The second of these considers restarting to prevent exact breakdown. The method that will be presented here is different in that it does not use implicit restarting. The implementation is thus simpler and avoids some numerical problems that can lead to complicated locking and purging. The method here also considers near-breakdown instead of just exact breakdown. There has also been work on restarted Lanczos bidiagonalization (see [6] and its references), though this is perhaps more related to symmetric Lanczos.

Another roundoff effect is the loss of biorthogonality that generally occurs. Paige around 1970 studied error in the symmetric version of the Lanczos algorithm [30, 31, 34]. He explained that the loss of orthogonality starts when Ritz vectors begin to converge and that the loss of orthogonality is in the direction of those Ritz vectors. Paige also showed that in spite of large error, the symmetric Lanczos algorithm is useful for computing eigenvalues (multiple copies of converged eigenvalues eventually appear). It can be observed that the nonsymmetric Lanczos algorithm behaves similarly in that orthogonality is lost in the direction of converged Ritz vectors; see [32] for recent analysis by Paige.

Several different approaches have been suggested for maintaining orthogonality in symmetric Lanczos. The first is full reorthogonalization, which takes every Lanczos vector formed by the three-term recurrence and reorthogonalizes it against every previous vector. Parlett and Scott proposed selective reorthogonalization (SO) [35] only against converged or converging Ritz vectors (using Paige's analysis that orthogonality is only lost in the direction of such Ritz vectors). A simpler way is periodic reorthogonalization [15]: at regular intervals reorthogonalize two consecutive vectors (see [53, 51]). Partial reorthogonalization (PRO) [42, 53, 51] monitors loss of orthogonality and thus determines when to reorthogonalize. See also [51, 23] for more on these methods.

2.2. Nonsymmetric Lanczos for linear equations. Many methods based on nonsymmetric Lanczos have been developed for solving systems of linear equations. The most basic is BiCG [21, 39]. The transpose-free methods BiCGStab [52] and TFQMR [13] are both well-established. See [45] for an extensive review. We will have a BiCG-like approach for solving linear equations in our new method NLan-DR. Also, we will use BiCGStab when solving multiple right-hand side systems.

2.3. Restarted methods for eigenvalue problems. Methods that compute eigenvectors may need restarting to control storage and orthogonalization expense. Sorensen in 1992 gave the implicitly restarted Arnoldi method (IRAM) [46], which made it possible to efficiently restart the Arnoldi iteration [4, 37] and keep several approximate eigenvectors at the restart. At the time of a restart, let the Ritz vectors be $\{y_1, y_2, \ldots, y_k\}$ and let r be a multiple of the residual vectors for these Ritz vectors (the residuals are parallel). Then the next cycle of IRAM builds the subspace

$$Span\{y_1, y_2, \ldots y_k, r, Ar, A^2r, A^3r, \ldots, A^{m-k-1}r\}. \tag{2.3}$$

This subspace contains a Krylov subspace with each approximate eigenvector as starting vector [25]. For more on restarted Arnoldi, see [22, 25, 49, 51, 29, 8, 5, 12]. For symmetric problems, Wu and Simon gave thick restarted Lanczos (TRLAN) [53] with a simpler implementation than IRAM. They also use partial reorthogonalization. In [1], the restarted Lanczos method Lan-DR is proposed for both eigenvalues and linear equations. The eigenvalue portion is the same as TRLAN, but several different reorthogonalizations are discussed, including a combination of selective reorthogonalization and periodic reorthogonalization.

2.4. Deflated restarted methods for linear equations. Related to the restarted eigenvalue methods just mentioned are deflated methods for solving linear equations. In 1995 [24], a restarted version of GMRES [40] that both solves linear equations and computes eigenvalues and eigenvectors was given. Approximate eigenvectors in the subspace deflate eigenvalues and thus improve convergence for the linear equations. In [26], FOM-DR solves linear equations and has an eigenvalue portion

that is equivalent to IRAM. The same subspace (2.3) is used (with r being the current linear equations residual). The method GMRES-DR is similar but uses harmonic Ritz vectors. Lan-DR [1] is a symmetric version of FOM-DR. There are other deflated versions of GMRES; see, for example, [11, 33, 7] and the references in [26, 1].

2.5. Multiple right-hand sides. Nonsymmetric systems with multiple right-hand sides occur in many applications (see [14] for some examples). Block methods are often used (see, for example, [39, 14, 27]). Other approaches for multiple right-hand sides use information from the solution of the first right-hand side (and possibly others) to assist subsequent right-hand sides. Seed methods [19] project over entire subspaces generated while solving previous right-hand sides. Simoncini and Gallopoulos [43, 44] suggest hybrid methods. In [28] GMRES-DR is applied to the first right-hand side and generates a small subspace that contains important information about approximate eigenvectors, and this is used to improve convergence for the subsequent right-hand sides. See [33] for a method for multiple right-hand sides that can also handle a changing matrix.

3. Nonsymmetric Lanczos with deflated restarting. We propose a restarted nonsymmetric Lanczos method, NLan-DR, that both computes eigenvalues and right and left eigenvectors and solves linear equations. The maximum size of the subspaces is m and the number of vectors saved at the restart for both right and left subspaces is k. In our examples, we will target the eigenvalues nearest the origin, but other choices can be made.

At the time of a restart, let v_{m+1} and w_{m+1} be the last right and left vectors that have been generated. These become the new v_{k+1} and w_{k+1} vectors for the new cycle. Let the right Ritz vectors that we want to keep from the just finished cycle be $\{y_1, y_2, \ldots, y_k\}$ and the left Ritz vectors be $\{u_1, u_2, \ldots, u_k\}$. Then the new cycle of NLan-DR builds the right subspace

$$Span\{y_1, y_2, \ldots y_k, v_{k+1}, Av_{k+1}, A^2v_{k+1}, A^3v_{k+1} \ldots, A^{m-k-1}v_{k+1}\} \tag{3.1}$$

and the left subspace

$$Span\{u_1, u_2, \ldots u_k, w_{k+1}, A^Tw_{k+1}, (A^T)^2w_{k+1}, (A^T)^3w_{k+1} \ldots, (A^T)^{m-k-1}w_{k+1}\}. \tag{3.2}$$

NLan-DR generates the recurrence formulas

$$AV_m = V_{m+1}T_{m+1,m}, \tag{3.3}$$

where V_m is an n-by-m matrix whose columns span the subspace (3.1) and V_{m+1} is the same except for an extra column, and

$$A^TW_m = W_{m+1}T^T_{m,m+1}, \tag{3.4}$$

where W_m is an n-by-m matrix whose columns span the subspace (3.2) and W_{m+1} is the same except for an extra column. Also $T_{m+1,m}$ and $T_{m,m+1}$ are portions of $T_{m+1,m+1}$, which is an $(m+1)$-by-$(m+1)$ matrix that is tridiagonal except for the $(k+1)$-by-$(k+1)$ leading portion. This portion has Ritz values on the main diagonal if they are real and 2-by-2 blocks on the diagonal for each complex Ritz pair and has the rest zero except for the $k+1$ row and $k+1$ column. A part of recurrence (3.3) can be separated out to give

$$AV_k = V_{k+1}T_{k+1,k}, \tag{3.5}$$

where V_k is an n-by-k matrix whose columns span the subspace of Ritz vectors, V_{k+1} is the same except for an extra column, and $T_{k+1,k}$ is the leading $(k+1)$-by-k portion of $T_{m+1,m+1}$. This recurrence allows access to both the approximate eigenvectors (the Ritz vectors) and their products with A while requiring storage of only $k+1$ vectors of length n. Similarly, for the left side there is

$$A^T W_k = W_{k+1} T_{k,k+1}^T. \tag{3.6}$$

We next discuss why NLan-DR is worth considering and then give theorems indicating why NLan-DR can be both efficient and effective.

Compared to regular nonrestarted Lanczos, NLan-DR makes computation of eigenvectors more practical, because the dimensions of the subspaces are limited. Compared to restarted Arnoldi, NLan-DR has less orthogonalization expense. This is particularly significant when the matrix is fairly sparse, so that the matrix-vector products do not dominate the expense. The lesser orthogonalization expense of NLan-DR also may make it possible to use larger subspaces than would be practical for Arnoldi, thus possibly giving faster convergence. The solution of tough problems particularly benefits from having large subspaces.

The subspace (3.1) has a Krylov portion that is augmented with approximate eigenvectors. However, the following theorem shows that the entire subspace is also a Krylov subspace, although with a different starting vector. A similar theorem can be given for subspace (3.2). These theorems are important because together they show that the regular nonsymmetric Lanczos three-term recurrences can be used once the method has been restarted.

THEOREM 3.1. *The right subspace* (3.1) *generated by NLan-DR is a Krylov subspace.*

Proof. As mentioned above, the vector v_{k+1} is the v_{m+1} vector from the previous cycle. For the right Ritz pairs (θ_i, y_i) generated at the end of the previous cycle, it is fairly standard using (3.3) to derive that the residual vectors are all multiples of the old v_{m+1}. We express this as

$$r_i = Ay_i - \theta_i y_i = \gamma_i v_{k+1}. \tag{3.7}$$

We will show that subspace (3.1) is equal to $Span\{s, As, A^2s, A^3s \ldots, A^{m-1}s\}$ for some starting vector s that is a combination of right Ritz vectors. Let

$$s = \sum_{i=1}^{k} \beta_i y_i.$$

Then, using (3.7),

$$As = \sum_{i=1}^{k} \beta_i A y_i = \sum_{i=1}^{k} \beta_i \theta_i y_i + \left(\sum_{i=1}^{k} \beta_i \gamma_i \right) v_{k+1}.$$

We eliminate the v_{k+1} term by setting $\sum_{i=1}^{k} \beta_i \gamma_i = 0$. This gives us a first equation that must be satisfied. Next, again multiply by A and again use (3.7):

$$A^2 s = \sum_{i=1}^{k} \beta_i \theta_i A y_i = \sum_{i=1}^{k} \beta_i \theta_i^2 y_i + \left(\sum_{i=1}^{k} \beta_i \theta_i \gamma_i \right) v_{k+1}.$$

Again eliminate the v_{k+1} term and get a second equation that must be satisfied: $\sum_{i=1}^{k} \beta_i \theta_i \gamma_i = 0$. We continue this until we have

$$A^{k-1}s = \sum_{i=1}^{k} \beta_i \theta_i^{k-1} y_i$$

and the system of equations with unknowns $\beta_1, \ldots, \beta_k$ and with matrix having ith row $[\theta_1^{i-1}\gamma_1, \ldots, \theta_k^{i-1}\gamma_k]$. This $k-1$ by k homogeneous system has a least a one-dimensional subspace of solutions. Therefore there is an s so that

$$Span\{s, As, A^2s, \ldots, A^{k-1}s\} = Span\{y_1, y_2, \ldots y_k\}.$$

Next, we continue multiplying by A and using (3.7) but no longer eliminate the term with v_{k+1}. Thus

$$A^k s = \sum_{i=1}^{k} \beta_i \theta_i^{k-1} A y_i = \sum_{i=1}^{k} \beta_i \theta_i^k y_i + \left(\sum_{i=1}^{k} \beta_i \theta_i^{k-1} \gamma_i \right) v_{k+1}$$

and

$$Span\{s, As, A^2s, \ldots, A^k s\} = Span\{y_1, y_2, \ldots y_k, v_{k+1}\}.$$

Continuing this process gives the desired result. □

The next theorem says that the subspace (3.1) contains Krylov subspaces with each Ritz vector as starting vector. This shows why NLan-DR is effective at computing eigenvalues and why the results are similar to those from restarted Arnoldi which has the same property.

THEOREM 3.2. *The right subspace* (3.1) *generated by NLan-DR contains a Krylov subspace with each Ritz vector as starting vector. These subspaces are* $Span\{y_i, Ay_i, A^2y_i, \ldots, A^{m-k}y_i\}$.

We omit the proof. It involves starting with y_i and multiplying repeatedly by A and using (3.7) (see the proof in [25] for restarted Arnoldi). We can also give a corresponding theorem for left Ritz vectors and the left space.

Next is a sketch of the algorithm.

NLAN-DR(m, k).

1. *Start.* Choose m, the maximum size of the subspace, and k, the desired number of approximate eigenvectors. If there is an initial guess, x_0, then the linear equations problem becomes $A(x - x_0) = r_0$, with $r_0 = b - Ax_0$. Let the right starting vector be $v_1 = r_0/\|r_0\|$ and the left starting vector be $w_1 = v_1$.
2. *First cycle.* Apply m iterations of the standard nonsymmetric Lanczos algorithm. This computes the matrix V_{m+1} that has the right Lanczos vectors as columns, and W_{m+1} with left Lanczos vectors as columns, and the $(m+1)$-by-$(m+1)$ tridiagonal matrix $T_{m+1,m+1}$. Use some form of rebiorthogonalization during the iteration (such as full rebiorthogonalization or periodic rebiorthogonalization).
3. *Eigenvector computation.* Compute the k smallest magnitude (or others, if desired) eigenvalues θ_i along with right eigenvectors g_i and left eigenvectors h_i of $T_{m,m}$, the m-by-m portion of $T_{m+1,m+1}$.
4. *Linear equations.* Find $c_m = W_k^T r_0$. For the first cycle, this is computed as $c_m = \|r_0\| e_1$, and for other cycles $c_m = \|r_0\| e_{k+1}$, where e_1 and e_{k+1} are coordinate vectors. Solve $T_{m,m}d = c_m$, and set $\tilde{x} = x_0 + V_m d$. Then $r = r_0 - A\tilde{x} = b - V_{m+1}T_{m+1,m}d$. Let the new $x_0 = \tilde{x}$ and $r_0 = r$.

5. *Restart.* Binormalize all real right Ritz vectors against the corresponding left Ritz vectors. Replace right and left complex Ritz vectors with their separated real and imaginary parts (this will keep the Lanczos vectors real). Both real and imaginary parts of complex vectors need to be included, so temporarily adjust k if necessary. For each set of separated vectors, biorthonormalize the two right vectors and against the two left vectors. Let G and H be the m-by-k matrices made up respectively of the resulting biorthonormalized right and left vectors. Extend G and H by appending another row of zeros and then appending the coordinate vector e_{m+1} as an extra column. So now G and H are $(m+1)$-by-$(k+1)$. Let $T^{new} = H^T T_{m+1,m+1} G$, but to compute it, let all elements be zero except that the diagonal elements corresponding to real Ritz values are set equal to the Ritz values; then two-by-two diagonal blocks corresponding to complex eigenvalues can be found with $T^{new}(i,j) = h_i^T T_{m+1,m+1} g_j$ for $j = i, i+1$ (h_i and g_j are current columns of H and G). Next the $k+1$ rows and columns are $T^{new}(k+1,i) = T(m+1,m)G(m,i)$ and $T^{new}(i,k+1) = T(m,m+1)H(m,i)$. The $k+1,k+1$ element of T^{new} will be computed in the next step. Form $V^{new} = V_m G(1:m, 1:k)$ and $W^{new} = W_m H(1:m, 1:k)$; then append a $k+1$ column to each: let $v_{k+1}^{new} = v_{m+1}$ and $w_{k+1}^{new} = w_{m+1}$. Next let $V_{k+1} = V^{new}$ and $W_{k+1} = W^{new}$.
6. *Main iteration.* First we find the v_{k+2} and w_{k+2} vectors. Compute $\hat{v} = Av_{k+1} - \sum_{i=1}^{k} T(i,k+1)v_i$ and $\hat{w} = A^T w_{k+1} - \sum_{i=1}^{k} T(k+1,i)w_i$. Then $T^{new}(k+1,k+1) = w_{k+1}^T \hat{v}$. Let $T_{k+1,k+1} = T^{new}$. Next $\hat{v} = \hat{v} - T(k+1,k+1)v_{k+1}$ and $\hat{w} = \hat{w} - T(k+1,k+1)w_{k+1}$. Binormalize $\hat{v}$ and $\hat{w}$ to form v_{k+2} and w_{k+2}. We choose to rebiorthonormalize v_{k+1}, w_{k+1} and v_{k+2}, w_{k+2} against preceding vectors. Next, apply the regular nonsymmetric Lanczos algorithm (with some rebiorthogonalization; see below for more detail) out to iteration m. Go to step 3.

We always set the left starting vector equal to the right, because this seems to reduce roundoff error. If desired, converged Ritz vectors can be locked in by placing them first in V_{new} and W_{new} and setting corresponding T^{new} entries in the $k+1$ row and column equal to zero. Sometimes accuracy of converged Ritz vectors begins to decrease if locking is not done.

As mentioned in the algorithm, some rebiorthogonalization is needed. In fact, unlike for nonrestarted Lanczos, here it is essential that the Lanczos vectors be kept fairly biorthonormal. All of the various reorthogonalization techniques discussed in section 2.1 can be applied. Selective orthogonalization is more practical here than for nonrestarted Lanczos [35] because the converged eigenvectors are readily available, and also the restarting often prevents unwanted eigenvectors from converging. We will use full rebiorthogonalization or use periodic rebiorthogonalization. Periodic can be done against all previous Lanczos vectors or only against the Ritz vectors (then it is a combination of periodic and selective rebiorthogonalization). See [1] for further discussion and examples in the symmetric case.

Example 1. We use a bidiagonal test matrix of size $n = 2500$. The diagonal entries are $0.1, 0.2, 0.3, 0.4, 1, 2, 3, \ldots, 2495, 2496$ and the superdiagonal has all 0.1's. The goal is to compute 12 eigenvalues along with right and left eigenvectors. We use NLan-DR(60,15), so each cycle generates right and left subspaces of dimension 60; then 15 right and left Ritz vectors are saved at the restart. It is well known that it is best to save more vectors than the number of desired eigenvalues. The right starting vector has entries generated random normal, and as mentioned, the left starting vector

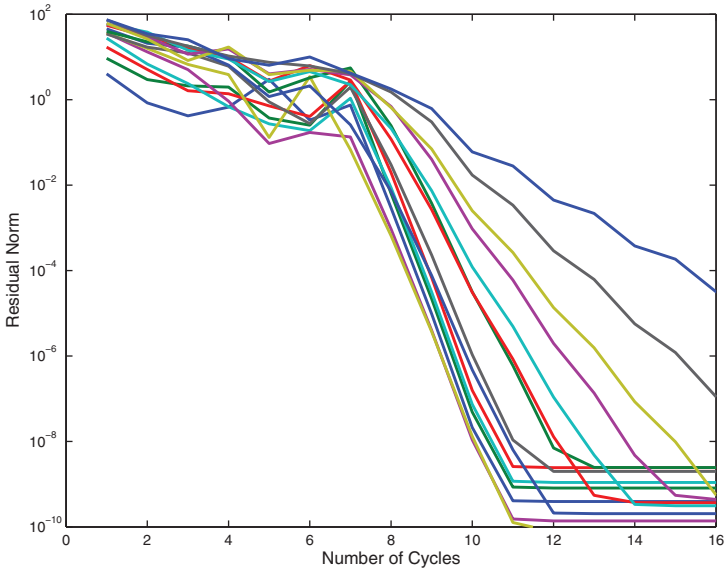

FIG. 3.1. *Convergence for eigenvalues of nearly symmetric text matrix.*

is set equal to the right. Periodic rebiorthogonalization of two consecutive vectors is done every 15 iterations.

Figure 3.1 shows the residual norms for the 15 right Ritz vectors, and we note that the left Ritz vector residuals are about the same. The residual norms for the desired smallest 12 eigenvalues all go to $2.5e-9$ or below. If the rebiorthogonalization of two vectors is done only at the end of the cycle, the residual norms go to $1.1e-8$ or less. With selective rebiorthogonalization only against the Ritz vectors done every 15 iterations, the residual norms reach $5.5e-8$. So for this example, not much reorthogonalizing is needed. There are no quickly converging eigenvalues that would necessitate frequent reorthogonalization.

Figure 3.2 has a comparison between NLan-DR and restarted Arnoldi for the first, sixth and twelfth eigenvalues. Arnoldi also restarts after hitting $m = 60$ and saves $k = 15$ Ritz vectors. The implementation used is from [29], which is mathematically equivalent to IRAM. This includes locking of eigenvectors and full reorthogonalization. For this nearly symmetric matrix, Lanczos and Arnoldi have almost identical convergence after the first few iterations.

We now compare the expense for the two methods assuming both right and left eigenvectors are computed (so Arnoldi is run twice to get both sets). We look at one cycle (not the first one) and count the major costs, which are matrix-vector products and length-n vector operations (such as dot products and axpy's). Two runs of restarted Arnoldi use $2(m-k)$ matrix-vector products per cycle and NLan-DR uses the same for its one run. For the Arnoldi orthogonalization and reorthogonalization, about $4((m+1)^2-(k+1)^2)$ length-n vector operations are required. Then forming the Ritz vectors takes $4km$ vector operations. NLan-DR also needs $4km$ for forming right and left Ritz vectors. It uses about $10m$ for the Lanczos iteration. This does not include computing residual norms of the Ritz vectors but does include checking for near-breakdown and rebiorthogonalizing the $k+1$ and $k+2$ vectors. Costs for further rebiorthogonalization depend on how frequently it is done. Storage for NLan-DR is

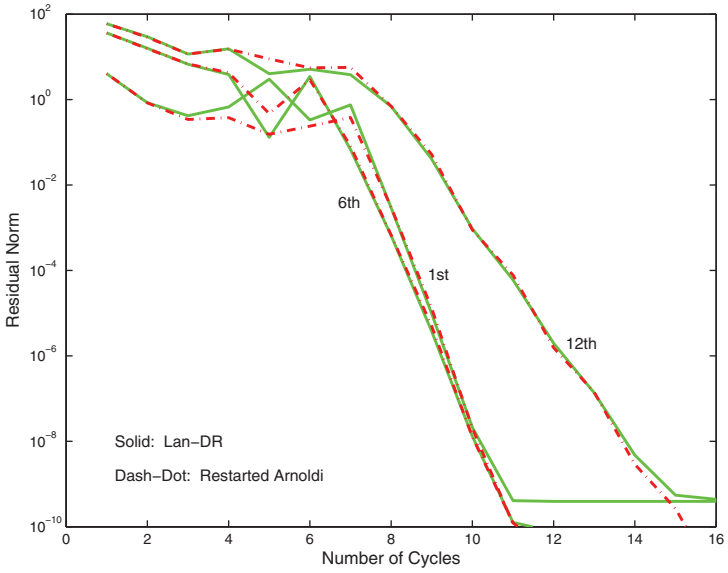

FIG. 3.2. *Comparison between NLan-DR and Arnoldi.*

TABLE 3.1
*Different values of m for NLan-DR(m,*15*).*

m	30	45	60	75	120
cycles	47	21	13	9	5
mat-vecs	1440	1290	1200	1110	1080

a little over $2(m+k)n$ which is about double of that for Arnoldi, since NLan-DR has both right and left spaces. For our specific example of NLan-DR(60,15) with rebiorthogonalization of two vectors every 15 iterations, a cycle of NLan-DR has 3600 vector operations for forming Ritz vectors, and about 600 for the Lanczos iteration and 1540 for reorthogonalization. The total of 5740 vector ops and 90 mat-vecs compares to 17,460 vector ops and 90 mat-vecs for Arnoldi. This shows that NLan-DR has potential to significantly lower expenses for some cases.

Table 3.1 has results for different subspace sizes while keeping the numbers of right and left Ritz vectors fixed at 15. The last row has the number of matrix-vector products needed to compute 12 eigenvalues with residual norms below $1.e-6$. Big subspaces help, but there is not a large decrease in the number of matrix-vector products as m increases. This is similar to what is observed next for linear equations, that large subspaces are not necessary because of the approximate eigenvectors saved at the restart.

Figure 3.3 has NLan-DR's solution of the linear equations compared with restarted Arnoldi in the form of FOM-DR [26]. The residual norm is plotted against iterations, but note that NLan-DR uses two matrix-vector products per iteration and FOM-DR only one (NLan-DR can also solve a dual linear system if that is desired). Also plotted is the convergence curve for nonrestarted Lanczos, which is equivalent to BiCG. Although it is not generally practical, full nonrestarted GMRES is included to show the best possible Krylov result. Not included is nonrestarted FOM, but the results are close to restarted FOM-DR most of the way and then end up very close to non-

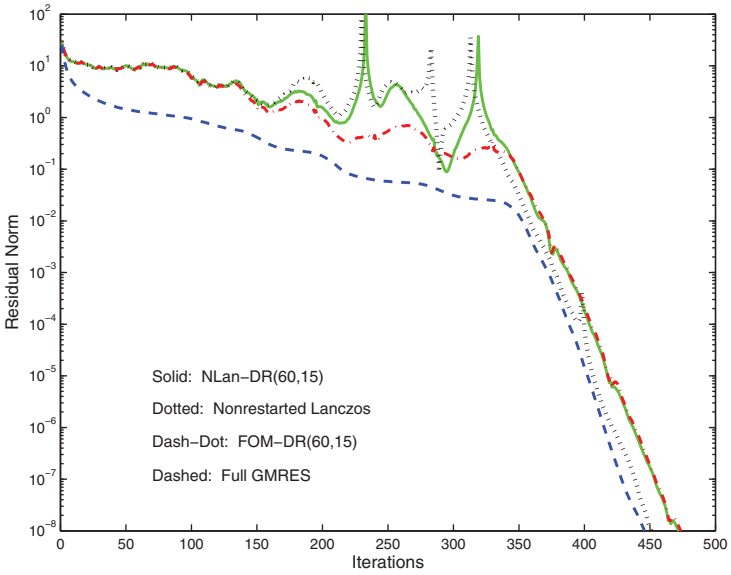

FIG. 3.3. *Solution of linear equations.*

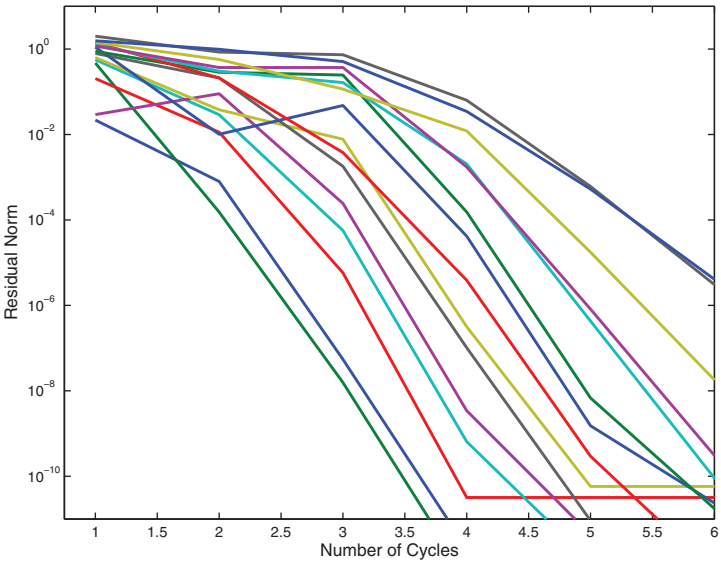

FIG. 3.4. *Eigenvalue residual norms for Sherman4.*

restarted Lanczos. The most important point is that the presence of approximate eigenvectors allows the restarted methods to come close to the convergence of the nonrestarted ones. Also, the Lanczos methods give final results that are very similar to methods that fully orthogonalize.

Example 2. We now try the Sherman4 matrix from the Matrix Market collection (and Harwell-Boeing). It has dimension 1104 and has only mild nonnormality with $\frac{||A-A^T||}{||A||} = .20$ and Henrici number $\frac{||AA^T-A^TA||}{||A^2||} = .12$. There are nine eigenvalues less than 1 and then 558 eigenvalues at 1. We again run NLan-DR(60,15). Figure 3.4 shows

residual norms for the eigenvalues if periodic rebiorthogonalization of two vectors every 10 iterations is used. If we have rebiorthogonalization every 15 iterations, then some residual norms only reach $1.e-8$. However, actually it is only the first cycle that needs the greater frequency. This is because some Ritz vectors converge fairly fast in the first cycle. There is loss of biorthogonality if Ritz vectors converge significantly in between rebiorthogonalizations. The results for restarted Arnoldi with full reorthogonalization are almost identical to Figure 3.4, though some eigenvalues do converge to greater accuracy. With an average of 3.4 nonzeros per row, this matrix is quite sparse, so reducing the orthogonalization expense is worthwhile.

4. Near-breakdown. The next example shows that near-breakdown can cause loss of accuracy in computing the eigenvalues.

Example 3. The matrix is the same as in Example 1, except the superdiagonal elements are all 1's, so it is more nonnormal. The four small eigenvalues are fairly ill-conditioned, with cosines of the angles between right and left eigenvectors ranging from 0.001 to 0.004. The cosines for other eigenvalues are 0.1 or larger. We run NLan-DR(60,15) for 16 cycles and for this example use full rebiorthogonalization. We checked with starting vectors generated by the MATLAB *randn* function with seeds of 0 through 9 (and again left starting vector equal to the right). The best results are for $seed = 3$ and the worst for $seed = 8$, so we show these two cases. Figure 4.1 has the residual norms for both choices of starting vectors. With $seed = 3$, the 12 smallest eigenvalues reach residual norms of $2.6e-8$ or below. With $seed = 8$, some only get to $2.2e-2$. The loss of accuracy is caused by near-breakdown. Figure 4.2 has a plot of the cosines of the angles between the corresponding right and left Lanczos eigenvectors that are generated. Some degree of near-breakdown occurs several times with the worst case when the cosine of the angle goes to about $3.0e-6$ for iterations 297 and 298. For $seed = 3$, the lowest cosine is $3.7e-4$.

The fact that NLan-DR restarts gives the opportunity to take measures to reduce near-breakdown. This is an alternative to using look-ahead. It is not the goal here to

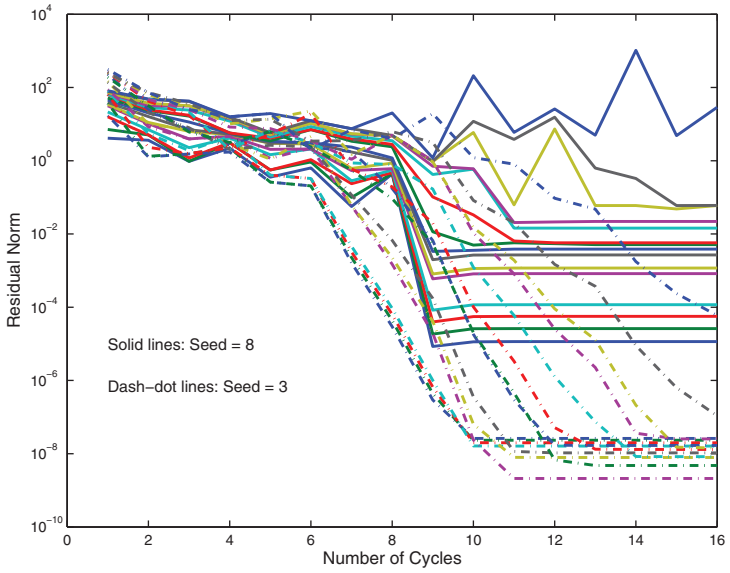

FIG. 4.1. *Accuracy of eigenvalues for two choices of starting vector.*

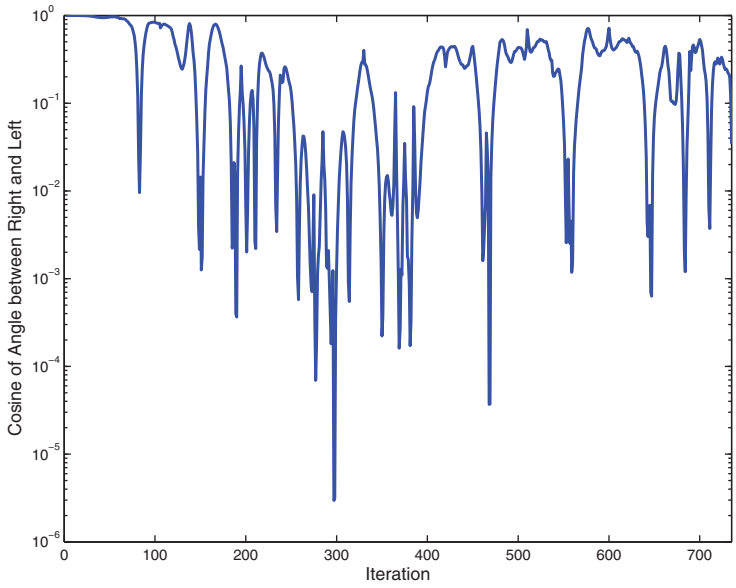

FIG. 4.2. *Cosines of the angles between right and left vectors.*

give a solution to all near-breakdowns. Instead we will give one approach that often improves the roundoff situation, and we will mention some other possibilities that should be further investigated.

Near-breakdown will be monitored by checking if the cosine of the angle between right and left Lanczos vectors goes below a specified threshold. When it does, we will go back two iterations and restart. Note that we only go back one iteration if we are at the $k+2$ iteration and cannot go back if it is the $k+1$ iteration. By both going back and restarting, the pattern of Lanczos vectors is changed.

Example 4. We test the go-back-and-restart method on the matrix from Example 3 with starting vector from $seed = 8$. Whenever the cosine of the angle between the newly formed right and left Lanczos vectors (vectors v_{j+1} and w_{j+1}) is less than the threshold of $1.e-4$, we go back two iterations and then restart. The results are much better. The 12 residual norms all reach $8.6e-7$ compared to the $2.2e-2$ in the previous example. There are two restarts due to near-breakdown. The first is at iteration 277 when the cosine hits $6.9e-5$ and then at iteration 632 with cosine of $1.2e-5$.

Example 5. This example has a more highly non-normal matrix and there is more difficulty in controlling near-breakdown. The matrix is the same as in the previous example, except the off-diagonal elements are 5.0. The left and right eigenvectors are nearly orthogonal. The cosines of the angles between the pair of eigenvectors for the smallest four eigenvalues are between $1.e-7$ and $1.e-8$, and these cosines increase to 0.00036 by the eighth pair and stay there until the largest few eigenvalues. In our tests, it seems to be inevitable that near-breakdown eventually occurs. We describe a run using the check for cosine between right and left Lanczos vectors less than $1.e-5$ and then go back two steps and restart approach. NLan-DR(60,15) starts well, and after 8 cycles (375 iterations) most eigenvalues are converging and four have residual norms below $1.e-3$. There have been no restarts for near-breakdown at that point. But the near-breakdown threshold is hit in the next cycle, and the restarting approach

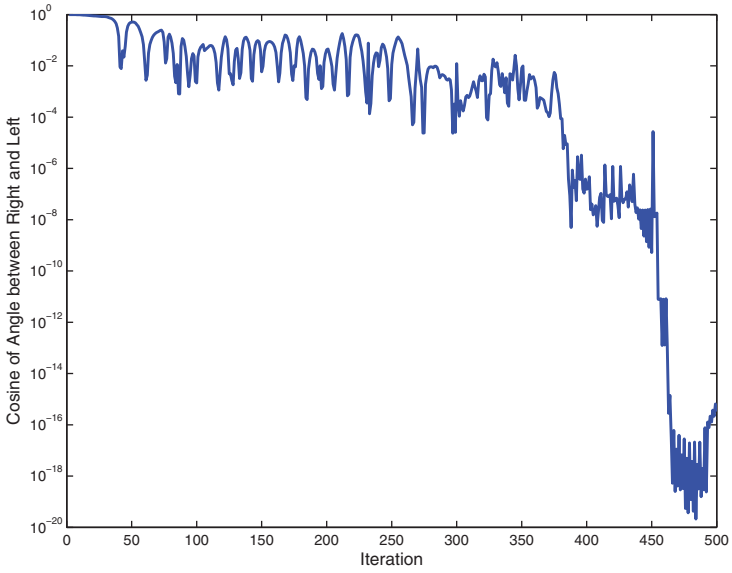

FIG. 4.3. *Cosines of the angles between right and left vectors for the highly non-normal matrix.*

does not work. The cosines between right and left Lanczos vectors stay small, and restarting happens with every near-breakdown check. The residual norms get a little worse and then stall out. Figure 4.3 shows the cosines that occur during the run.

Next we give a better test of when to restart due to near-breakdown and use it in an example that attempts to summarize results for the matrices of increasing non-normality. In Example 4, restarting when the cosine of the angle went below 10^{-4} worked well, but with some other random starting vectors a higher threshold like 10^{-3} is needed. However, there may be too much restarting with the higher threshold. Therefore we suggest beginning with the higher level but then cutting the threshold in half every time that the method does a restart to control near-breakdown. This avoids the too-frequent restarting.

DECREASING THRESHOLD TEST FOR CONTROLLING NEAR-BREAKDOWN.

1. Choose an initial threshold.
2. At every iteration monitor the cosine of the angle between the vectors v_{j+1} and w_{j+1}. When the cosine is below the threshold, go back two iterations and then restart. If at iteration $k+2$, only go back one iteration. If at $k+1$, don't restart.
3. For each restart due to near-breakdown, cut the threshold in half.

Example 6. We apply this test to the three matrices of increasing nonnormality from Examples 1, 3, 4 and 5. We run NLan-DR(60,15) using 10 different random starting vectors for each matrix. We use an initial threshold of 10^{-2} for the matrix with superdiagonal elements of 0.1, 10^{-3} with 1's on the superdiagonal and 10^{-4} for the highly nonnormal matrix with 5's. Table 4.1 gives the best, worst, and average (using logs) residual norms of the twelfth most accurate eigenvalue for the ten runs. Using no control is compared with the decreasing threshold test just suggested. Note that the near-breakdown control gives consistently good results for the second matrix. For the highly nonnormal matrix, it sometimes gives more accurate results. In fact, the control actually works fairly well except for the case of near-breakdown with the v_{k+1} and w_{k+1} vectors, when the method cannot restart again. Of the ten runs, four

TABLE 4.1
Controlling near-breakdown for increasingly non-normal matrices.

	matrix w/ 0.1's		with 1's		with 5's	
	w/o control	with	w/o	with	w/o	with
best	$1.7e-10$	$2.1e-10$	$2.6e-8$	$2.8e-9$	$3.4e-5$	$2.9e-7$
worst	$2.1e-6$	$5.1e-7$	$2.2e-2$	$5.4e-7$	0.52	.26
average	$8.0e-9$	$1.8e-9$	$1.1e-6$	$8.0e-8$	$1.0e-2$	$7.5e-4$

do not have this case, and they have average accuracy of $6.6e-6$ for the 12th best eigenvalue. The other 6 runs only average $1.8e-2$. Even for the first matrix with 0.1's on the superdiagonal, the worst result with accuracy $5.1e-7$ is due to this case of near-breakdown at the $k+1$ iteration. This points out the need for a better control method, which should be the focus of further research.

Other possible approaches for controlling near-breakdown include switching at the early restart to a method that saves some different Ritz vectors or that uses harmonic Rayleigh–Ritz or more general Krylov–Schur restarting [50]. Another option is to change to a different inner-product at the early restart. This is fairly complicated, but perhaps the new inner-product could be tailored to push the vectors away from near orthogonality.

5. Multiple right-hand sides. Next, the solution of systems with multiple right-hand sides is considered. The first right-hand side is solved with NLan-DR, and then the right and left eigenvectors thus generated are used to deflate eigenvalues for the other right-hand sides. After a projection with the right and left Ritz vectors from the end of NLan-DR, BiCGStab is applied. We call this approach deflated BiCGStab or D-BiCGStab. In [2] it was used for quantum chromodynamics problems for which left eigenvectors are automatically available from the right eigenvectors. The left-right projection is a Petrov–Galerkin projection [39] with W_k on the left and V_k on the right. So for the system $A(x-x_0)=r_0$, we solve $W_k^T A V_k d = W_k^T r_0$, then set $x = x_0 + V_k d$ and $r = r_0 - AV_k d$. This simplifies some, as given in the following algorithm for D-BiCGStab.

D-BICGSTAB FOR THE SECOND AND SUBSEQUENT RIGHT-HAND SIDES.

1. After applying the initial guess x_0, let the system of equations be $A(x-x_0) = r_0$.
2. If it is known that the right-hand sides are closely related, project over the previous computed solution vectors. For example, if we project over one solution at a time with a Galerkin projection, then for each system $Ax^i = b^i$ that has already been solved: compute $\delta = (x^i)^T r_0/((x^i)^T b^i)$; then form $x_0 = x_0 + \delta x^i$ and $r_0 = r_0 - \delta b^i$.
3. Apply a left-right projection for V_k. This uses the V_k, W_k and $T_{k,k}$ matrices that were developed while solving the first right-hand side with NLan-DR. Specifically, solve $T_{k,k} d = W_k^T r_0$, and let the new approximate solution be $x_0 = x_0 + V_k d$ and the new residual be $r_0 = r_0 - AV_k d$ (or one can use $r_0 - V_{k+1} T_{k+1,k} d$).
4. Apply BiCGStab to $A(x-x_0) = r_0$ until satisfied with convergence.

Example 7. We look again at the matrix Sherman4 that was used in Example 2 and solve systems with 20 random normal right-hand sides to relative accuracy 10^{-6}. For the first, NLan-DR(60,15) is applied as before, except only 4 cycles are needed to get optimal overall results. Then the other 19 systems use D-BiCGStab. The average number of matrix-vector products to solve these 19 systems is 67.6. This compares

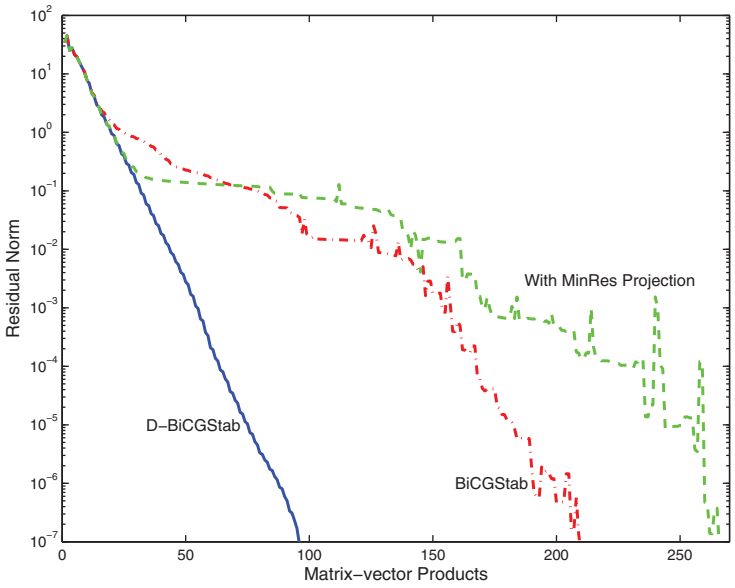

FIG. 5.1. *Comparison of left-right deflated BiCGStab versus MINRES deflated and regular BiCGStab for a second right-hand side.*

to an average of 188.7 for regular BiCGStab. Convergence is faster for D-BiCGStab, because the left-right projection effectively removes or deflates the components in the direction of small eigenvectors from the problem. The total for solving all 20 systems is 1675 matrix-vector products for NLan-DR/D-BiCGStab (390 for the first and 1285 for the other 19) and 3585 for BiCGStab. The left-right projection is fairly inexpensive, but NLan-DR does have the cost of forming the right and left Ritz vectors. Figure 5.1 has the residual norm curves for solving the second right-hand side with D-BiCGStab, regular BiCGStab and with a deflated BiCGStab that uses a minimum residual (MINRES) projection [39] over only the right Ritz vectors. D-BiCGStab does not have to deal with the small eigenvalues that would slow it down, so it converges from the beginning at about the rate that BiCGStab eventually achieves after it has generated in its subspaces approximations to the right and left eigenvectors corresponding to the small eigenvalues. Meanwhile, the method with the MINRES projection beats regular BiCGStab briefly, then is worse. It is not possible to effectively deflate the eigenvalues using only the right eigenvectors, unless one is willing to frequently restart the iterative method and redo the projection [28]. The next theorems address this need for left eigenvectors.

Even though in some sense MINRES is the best projection, it does not necessarily reduce small eigencomponents enough. To see this problem, we assume that we have one exact eigenvector. We would like to be able to eliminate the component of the residual vector in that direction, but the MINRES projection does not necessarily accomplish this.

THEOREM 5.1. *Suppose A has a full set of eigenvectors. Let $z_1, z_2, \ldots, z_n$ be the normalized eigenvectors. Let the current linear equations problem be $A(x - x_0) = r_0$, with $r_0 = \alpha_1 z_1 + \alpha_2 z_2 + \cdots + \alpha_n z_n$. Then after the MINRES projection over the subspace Span$\{z_1\}$, the new residual's component in the direction of z_1 is*

$$-\sum_{i=2}^{n} \alpha_i z_1^* z_i. \tag{5.1}$$

Proof. The MINRES projection over the subspace spanned by z_1 gives a residual orthogonal to $Span\{Az_1\}$,

$$0 = (Az_1)^* r = \bar{\lambda}_1 z_1^*(r_0 - Az_1 d) = \bar{\lambda}_1(z_1^* r_0 - \lambda_1 d).$$

Thus

$$d = \frac{z_1^* r_0}{\lambda_1},$$

and then

$$\begin{aligned} r &= r_0 - Az_1 d \\ &= r_0 - (z_1^* r_0) z_1 \\ &= (\alpha_1 - z_1^* r_0) z_1 + \sum_{i=2}^{n} \alpha_i z_i \\ &= \left(- \sum_{i=2}^{n} \alpha_i z_1^* z_i \right) z_1 + \sum_{i=2}^{n} \alpha_i z_i. \end{aligned}$$

This shows the z_1 component of r is as given in (5.1). ∎

Equation (5.1) shows that the MINRES projection over one exact eigenvector works best if the components of the residual in the directions other than z_1 are small and/or if the eigenvectors are nearly orthogonal. For nonsymmetric matrices, this projection may only reduce the size of the z_1 component to roughly the size of the other components. There is no reason to expect the z_1 component to be reduced far enough so that it will not need any further reduction from the BiCGStab iteration. Next we look at using the left-right projection with the left and right subspaces spanned by single arbitrary vectors. Then in the corollary, we consider again the case of having the exact eigenvector z_1.

THEOREM 5.2. *Suppose A has a full set of eigenvectors. Let $z_1, z_2, \ldots, z_n$ be the right eigenvectors of length one and $u_1, u_2, \ldots, u_n$ be the left eigenvectors binormalized so that $u_i^* z_i = 1$. Let the current linear equations problem be $A(x - x_0) = r_0$, with $r_0 = \sum_{i=1}^{n} \alpha_i z_i$. Let the left subspace be $Span\{w\}$, where $w = \sum_{i=1}^{n} \beta_i u_i$, and the right subspace be $Span\{v\}$, where $v = \sum_{i=1}^{n} \gamma_i z_i$. Then after the left-right projection, the new residual's component in the direction of z_1 is*

$$\alpha_1 - \left(\frac{\sum_{i=1}^{n} \alpha_i \beta_i}{\sum_{i=1}^{n} \lambda_i \beta_i \gamma_i} \right) \gamma_1 \lambda_1. \tag{5.2}$$

Proof. With the left-right projection, the residual is orthogonal to the left space, which is represented by the single vector w:

$$0 = w^* r = w^*(r_0 - Avd) = \sum_{i=1}^{n} \alpha_i \beta_i - \sum_{i=1}^{n} \lambda_i \beta_i \gamma_i d.$$

We can solve this for

$$d = \frac{\sum_{i=1}^{n} \alpha_i \beta_i}{\sum_{i=1}^{n} \lambda_i \beta_i \gamma_i},$$

and then

$$r = r_0 - Avd = \sum_{i=1}^{n} \alpha_i z_i - \left(\sum_{i=1}^{n} \gamma_i \lambda_i z_i \right) d.$$

Thus

$$r = \sum_{i=1}^{n} \left(\alpha_i - \left(\frac{\sum_{i=1}^{n} \alpha_i \beta_i}{\sum_{i=1}^{n} \lambda_i \beta_i \gamma_i} \right) \gamma_i \lambda_i \right) z_i,$$

and the result for the z_1 component of r follows. ∎

If we assume as in Theorem 5.1 that the right subspace is the exact eigenvector, then we get the following result.

COROLLARY 5.3. *With the same assumptions as the previous theorem except that the right space is Span$\{z_1\}$, the new residual's component in the direction of z_1 is*

$$-\sum_{i=2}^{n} \alpha_i \frac{\beta_i}{\beta_1}. \tag{5.3}$$

This corollary tells us that the left-right projection does a better job of reducing the size of the residual's component in the z_1 direction if the other components are small and/or if the left vector w is mostly in the direction of the left eigenvector u_1. There is no need for nearly orthogonal right eigenvectors as in the MINRES projection.

A stronger result can be given if instead of an exact eigenvector for the right space, an exact left eigenvector is used for the left space. Then Theorem 5.2 gives that the corresponding component of the residual vector goes to zero, regardless of the right vector. So having an accurate left space is probably even more important than an accurate right space. However, the result of Theorem 5.2 shows that if neither right or left vector is exact, then the accuracy of both comes into play.

The next theorem shows that for projections over more than one vector, we still have the property that an exact left eigenvector in the left space gives a zero component in the residual.

THEOREM 5.4. *Suppose A has a full set of eigenvectors. Assume we apply the left-right projection with the left eigenvector u_1 contained in the left subspace. Then the component of the residual vector in the direction of z_1 is zeroed out.*

Proof. Since the residual vector is orthogonal to the entire left space, it must be orthogonal to u_1, so

$$0 = u_1^* r.$$

Using the spectral projector $z_1 u_1^*$ and the biorthogonality of the left and right eigenvectors, the z_1 component of r is $z_1 u_1^* r = 0$. ∎

We next give an example showing even more improvement for NLan-DR/D-BiCGStab than in the previous example.

Example 8. Now the bidiagonal matrix from Example 3 with 1's on the superdiagonal is used with multiple right-hand sides. The starting vector is again from *seed* = 8, and again there is full rebiorthogonalization and the simple near-breakdown control from Example 4. There are a total of 20 right-hand sides solved to relative tolerance of 10^{-6}. NLan-DR(60,15) is run for 12 cycles, which uses 1094 matrix-vector products. For the subsequent right-hand sides, D-BiCGStab needs an average of 128.7 matrix-vector products. Regular BiCGStab applied to all 20 right-hand sides uses an

TABLE 5.1
Deflating increasing numbers of eigenvalues for BiCGStab.

(m,k)	(50,5)	(55,10)	(60,15)	(75,30)	(90,45)	(105,60)
cycles, 1st rhs	20	15	15	17	19	21
mat-vecs, 1st rhs	1680	1198	1222	1396	1632	1806
mat-vecs, 2nd rhs	296	174	133	96	83	121

average of 860 matrix-vector products. The total for NLan-DR/D-BiCGStab is 3539 versus 17,200 for BiCGStab alone.

We continue with the same matrix and starting vector, and now compare deflating different numbers of eigenvalues for D-BiCGStab. The first right-hand side uses NLan-DR(m,k) with k going from 5 to 60. It is run until four-fifths of the Ritz vectors converge to $1.e-6$. We also now switch to the decreasing threshold near-breakdown test from Example 6 with starting threshold of $1.e-3$ and use locking when Ritz vectors reach accuracy $1.e-8$ (this gives better results for the tests with 45 and 60 Ritz vectors). The last row of Table 5.1 has the results for solving the second right-hand side with D-BiCGStab. Having more right and left approximate eigenvectors improves the convergence for BiCGStab except for the $k = 60$ case, where the eigenvectors are not quite as accurate. However, there is extra expense for computing more eigenvectors, both for additional matrix-vector products and for forming Ritz vectors. If there are many right-hand sides to be solved, then it is worthwhile to use more eigenvectors.

We mention that if higher accuracy solution of the linear equations is needed, the residual curve sometimes flattens out near the end and is very slow reaching the tolerance. For instance, after NLan-DR(60,15), running D-BiCGStab to relative tolerance of 10^{-8} takes 1040 mat-vecs. This happens due to the approximate right and left eigenvectors in the projection not being accurate enough to completely remove the corresponding components in the residual vector. This can be fixed by restarting at some point and applying another left-right projection before running BiCGStab again. Stathopoulos and Orginos restart a deflated conjugate gradient method for symmetric problems in [48].

6. Conclusion. NLan-DR is a method for computing eigenvalues and both right and left eigenvectors and simultaneously solving linear equations. It is a nonsymmetric version of the Lan-DR method, and generally requires some rebiorthogonalization. An approach for controlling near-breakdown is given, and some others are suggested for future investigation. However, for highly non-normal matrices, NLan-DR is likely to still have problems. NLan-DR can use less orthogonalization than restarted Arnoldi methods, and so can be considerably less expensive for finding right and left eigenvectors if the matrix-vector product is not the major expense and the A^T operation is easy.

NLan-DR can be used along with deflated BiCGStab to solve systems with multiple right-hand sides. It is important to have both right and left approximate eigenvectors to deflate out eigenvalues for BiCGStab. If the matrix has small eigenvalues, then this approach may use far fewer matrix-vector products than BiCGStab alone.

Acknowledgments. The authors would like to thank the referees for their careful reading and many suggestions that improved the paper.

REFERENCES

[1] A. M. Abdel-Rehim, R. B. Morgan, D. A. Nicely, and W. Wilcox, *Deflated and restarted symmetric Lanczos methods for eigenvalues and linear equations with multiple right-hand sides*, SIAM J. Sci. Comput., 32 (2010), pp. 129–149.

[2] A. M. Abdel-Rehim, W. Wilcox, and R. B. Morgan, *Deflated BiCGStab for linear equations in QCD problems*, in Proceedings of LAT2007, 2007, pp. 026/1–026/7.

[3] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, *A Lanczos-type method for multiple starting vectors*, Math. Comp., 69 (2000), pp. 1577–1601.

[4] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[5] J. Baglama, *Augmented block householder Arnoldi method*, Linear Algebra Appl., 429 (2008), pp. 2315–2334.

[6] J. Baglama and L. Reichel, *Restarted block Lanczos bidiagonalization methods*, Numer. Algorithms, 43 (2006), pp. 251–272.

[7] A. H. Baker, E. R. Jessup, and T. Manteuffel, *A technique for accelerating the convergence of restarted GMRES*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 962–984.

[8] C. Beattie, M. Embree, and J. Rossi, *Convergence of restarted Krylov subspaces to invariant subspaces*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1074–1109.

[9] L. Billings and I. B. Schwartz, *Identifying almost invariant sets in stochastic dynamical systems*, Chaos, 18 (2008), pp. 023122/1–8.

[10] J. Cullum and R. A. Willoughby, *A practical procedure for computing eigenvalues of large sparse nonsymmetric matrices*, in Large Scale Eigenvalue Problems, North-Holland Math. Stud. 127, J. Cullum and R. A. Willoughby, eds., North-Holland, Amsterdam, 1986, pp. 193–240.

[11] E. De Sturler, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.

[12] M. Embree, *The Arnoldi eigenvalue iteration with exact shifts can fail*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1–10.

[13] R. W. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.

[14] R. W. Freund and M. Malhotra, *A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl., 254 (1997), pp. 119–157.

[15] J. Grcar, *Analyses of the Lanczos Algorithm and of the Approximation Problem in Richardson's Method*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1981.

[16] E. J. Grimme, D. C. Sorensen, and P. Van Dooren, *Model reduction of state space systems via an implicitly restarted Lanczos method*, Numer. Algorithms, 12 (1996), pp. 1–31.

[17] M. H. Gutknecht and K. J. Ressel, *Look-ahead procedures for Lanczos-type product methods based on three-term Lanczos recurrences*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1051–1078.

[18] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.

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

[20] C. Lanczos, *An iterative method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.

[21] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 33–53.

[22] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[23] G. A. Meurant, *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, SIAM, Philadelphia, 2006.

[24] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.

[25] R. B. Morgan, *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.

[26] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[27] R. B. Morgan, *Restarted block-GMRES with deflation of eigenvalues*, Appl. Numer. Math., 54 (2005), pp. 222–236.

[28] R. B. Morgan and W. Wilcox, *Deflated Iterative Methods for Linear Equations with Multiple Right-Hand Sides*, arXiv:math-ph/0405053v2, 2004.

[29] R. B. Morgan and M. Zeng, *A harmonic restarted Arnoldi algorithm for calculating eigenvalues and determining multiplicity*, Linear Algebra Appl., 415 (2006), pp. 96–113.
[30] C. C. Paige, *The Computation of Eigenvectors and Eigenvalues of Very Large Sparse Matrices*, Ph.D. thesis, University of London, 1971.
[31] C. C. Paige, *Computational variants of the Lanczos method for the eigenproblem*, J. Inst. Math. Appl., 10 (1972), pp. 373–381.
[32] C. C. Paige, *A useful form of unitary matrix obtained from any sequence of unit* 2*-norm n-vectors*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 565–583.
[33] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.
[34] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[35] B. N. Parlett and D. S. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.
[36] B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44 (1985), pp. 105–124.
[37] Y. Saad, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
[38] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, New York, 1992.
[39] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, 1996.
[40] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[41] G. De Samblanx and A. Bultheel, *Nested Lanczos: Implicitly restarting an unsymmetric Lanczos algorithm*, Numer. Algorithms, 18 (1998), pp. 31–50.
[42] H. D. Simon, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–136.
[43] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.
[44] V. Simoncini and E. Gallopoulos, *A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides*, J. Comput. Appl. Math., 66 (1996), pp. 457–469.
[45] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.
[46] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[47] D. A. Spielman, *Spectral graph theory and its applications*, in 48th Annual IEEE Symposium on Foundations of Computer Science, 2007, pp. 29–38.
[48] A. Stathopoulos and K. 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), pp. 439–462.
[49] A. Stathopoulos, Y. Saad, and K. Wu, *Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods*, SIAM J. Sci. Comput., 19 (1998), pp. 227–245.
[50] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.
[51] G. W. Stewart, *Matrix Algorithms* II: *Eigensystems*, SIAM, Philadelphia, 2001.
[52] H. A. van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
[53] K. Wu and H. Simon, *Thick-restart Lanczos method for symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.
[54] T. Y. Wu and K. W. Wang, *Reduction of structural accoustic radiation via left and right eigenvector assignment approach*, J. Intell. Mat. Syst. Str., 20 (2009), pp. 2173–2186.