

ABLE: AN ADAPTIVE BLOCK LANCZOS METHOD FOR NON-HERMITIAN EIGENVALUE PROBLEMS*

ZHAOJUN BAI[†], DAVID DAY[‡], AND QIANG YE[§]

Abstract. This work presents an adaptive block Lanczos method for large-scale non-Hermitian Eigenvalue problems (henceforth the ABLE method). The ABLE method is a block version of the non-Hermitian Lanczos algorithm. There are three innovations. First, an adaptive blocksize scheme cures (near) breakdown and adapts the blocksize to the order of multiple or clustered eigenvalues. Second, stopping criteria are developed that exploit the semiquadratic convergence property of the method. Third, a well-known technique from the Hermitian Lanczos algorithm is generalized to monitor the loss of biorthogonality and maintain semibiorthogonality among the computed Lanczos vectors. Each innovation is theoretically justified. Academic model problems and real application problems are solved to demonstrate the numerical behaviors of the method.

Key words. non-Hermitian matrices, eigenvalue problem, spectral transformation, Lanczos method

AMS subject classifications. 65F15, 65F10

PII. S0895479897317806

1. Introduction. A number of efficient numerical algorithms for solving large-scale matrix computation problems are built upon the Lanczos procedure, a procedure for successive reduction of a general matrix to a tridiagonal form [28]. In the 1970s and '80s, great progress in mathematical and numerical analysis was made on applying the Lanczos algorithm for solving large sparse Hermitian eigenvalue problems. Today, a Lanczos-based algorithm has been accepted as the method of choice to large sparse Hermitian eigenvalue problems arising in many computational science and engineering areas.

Over the last decade there has been considerable interest in Lanczos-based algorithms for solving *non*-Hermitian eigenvalue problems. The Lanczos algorithm without rebiorthogonalization is implemented and applied to a number of application problems in [12]. Different schemes to overcome possible failure in the non-Hermitian Lanczos algorithm are studied in [38, 17, 53]. A Lanczos procedure with look-ahead scheme is available in QMRPACK [18]. Theoretical studies of breakdown and instability can be found in [21, 36, 23, 6]. Error analyses of the non-Hermitian Lanczos procedure implemented in finite precision arithmetic are presented in [2, 14].

Despite all this progress, a number of unresolved issues, some of which are related to the use of nonorthogonal basis and hence its conditional stability property, obstruct a robust and efficient implementation of the non-Hermitian Lanczos procedure. These issues include

- how to distinguish *copies* of converged Rayleigh–Ritz values from multiple or clustered eigenvalues,

*Received by the editors March 4, 1997; accepted for publication (in revised form) by Z. Strakos February 27, 1998; published electronically July 9, 1999. This work was supported in part by NSF grant ASC-9313958, DOE grant DE-FG03-94ER25219, and a research grant from NSERC of Canada. <http://www.siam.org/journals/simax/20-4/31780.html>

[†]Department of Mathematics, University of Kentucky, Lexington, KY 40506 (bai@ms.uky.edu).

[‡]MS 1110, Sandia National Laboratories, PO Box 5800, Albuquerque, NM 87185 (dday@cs.sandia.gov).

[§]Department of Applied Mathematics, University of Manitoba, Winnipeg, MB R3T 2N2, Canada (ye@gauss.amath.umanitoba.ca).

- how to treat a (near) breakdown to preserve the stringent numerical stability requirements on the Lanczos procedure for eigenvalue problems in finite precision arithmetic,
- how to explain and take advantage of the observed semiquadratic convergence rate of the Lanczos procedure, and
- how to extend the understanding of the Hermitian Lanczos algorithm with semiorthogonality [46] and the non-Hermitian Lanczos algorithm with semibiorthogonality [14] to the *block* non-Hermitian Lanczos algorithm.

In the adaptive block Lanczos method for large-scale non-Hermitian Eigenvalue problems (ABLE method) proposed in this work, we address each of these issues as follows:

- A block version of the Lanczos procedure is implemented. Several nontrivial implementation issues are addressed. The blocksize adapts to be at least the order of multiple or clustered eigenvalues, and the linear independence of the Lanczos vectors is maintained. This accelerates convergence in the presence of multiple or clustered eigenvalues.
- The blocksize also adapts to cure (near) breakdowns. The adaptive blocking scheme proposed here enjoys the theoretical advantage that any exact breakdown can be cured with fixed augmentation vectors. In contrast, the prevalent look-ahead techniques require an arbitrary number of augmentation vectors to cure a breakdown and may not be able to cure all breakdowns [38, 17, 36].
- An asymptotic analysis of the second-order convergence of the Lanczos procedure is presented and utilized in the stopping criteria.
- A scheme to monitor the loss of the biorthogonality and maintain semibiorthogonality is developed in the adaptive block Lanczos procedure.

The ABLE method is a generalization of the block Hermitian Lanczos algorithm [10, 19, 39, 22] to the non-Hermitian case. For general application codes that represent their matrices as out-of-core, block algorithms substitute matrix block multiplies and block solvers for matrix vector products and simple solvers [22]. In other words, higher level BLAS are used in the inner loop of block algorithms. This decreases the I/O costs essentially by a factor of the blocksize.

We will demonstrate numerical behaviors of the ABLE method using several numerical examples from academic model problems and real application problems. There are many competitive methods for computing large sparse non-Hermitian eigenvalue problems, namely, the simultaneous iteration method [5, 50, 15], Arnoldi's method [1, 42], the implicitly restarted Arnoldi method [48, 29], block Arnoldi [43, 45], the rational Krylov subspace method [40], Davidson's method [13, 44], and the Jacobi-Davidson method [47, 8]. In particular, ARPACK [31], an implementation of the implicitly restarted Arnoldi method, is gaining acceptance as a standard piece of mathematical software for solving large-scale eigenvalue problems. A comparative study of simultaneous iteration-based methods and Arnoldi-based methods is presented in [30]. It is beyond the scope of this paper to compare our ABLE method with the rest of the methods. However, a comprehensive comparison study is certainly a part of our future work.

The rest of this paper is organized as follows. In section 2, we present a basic block non-Hermitian Lanczos algorithm, discuss its convergence properties, and review how to maintain biorthogonality among Lanczos vectors computed in finite precision arithmetic. An adaptive block scheme to cure (near) breakdown and adapt the blocksize to the order of multiple or clustered eigenvalues is described in section 3. In section 4, we

1. Choose starting $n \times p$ vectors P_1 and Q_1 so that $P_1^T Q_1 = I$
2. $R_1 = (P_1^T A)^T$ and $S_1 = A Q_1$
3. For $j = 1, 2, \dots$
 - 3.1. $A_j = P_j^T S_j$
 - 3.2. $R_j := R_j - P_j A_j^T$ and $S_j := S_j - Q_j A_j$
 - 3.3. Compute the QR decompositions: $R_j = P_{j+1} B_{j+1}^T$ and $S_j = Q_{j+1} C_{j+1}$
 - 3.4. Compute the singular value decomposition: $P_{j+1}^T Q_{j+1} = U_j \Sigma_j V_j^H$
 - 3.5. $B_{j+1} := B_{j+1} U_j \Sigma_j^{1/2}$ and $C_{j+1} := \Sigma_j^{1/2} V_j^H C_{j+1}$
 - 3.6. $P_{j+1} := P_{j+1} \bar{U}_j \Sigma_j^{-1/2}$ and $Q_{j+1} := Q_{j+1} V_j \Sigma_j^{-1/2}$
 - 3.7. $R_{j+1} = (P_{j+1}^T A - C_{j+1} P_j^T)^T$ and $S_{j+1} = A Q_{j+1} - Q_j B_{j+1}$

FIG. 2.1. Basic block non-Hermitian Lanczos algorithm.

model the loss of biorthogonality among the Lanczos vectors in finite precision arithmetic and present an efficient algorithm for maintaining semibiorthogonality among the computed Lanczos vectors. The complete ABLE method is presented in section 5. In section 6, we briefly discuss how a spectral transformation is used to solve a generalized eigenvalue problem using the ABLE method. Numerical experiments are reported in section 7.

2. A block non-Hermitian Lanczos algorithm. In this section we present a block implementation of the non-Hermitian Lanczos algorithm and discuss its convergence properties for solving non-Hermitian eigenvalue problems. An adaptive block non-Hermitian Lanczos algorithm (see section 5) builds into this algorithm features presented in the intervening sections.

2.1. A basic block Lanczos algorithm. The basic block non-Hermitian Lanczos procedure presented in Figure 2.1 is a variation of the original Lanczos procedure as proposed by Lanczos [28]. Given an n by n matrix A and initial n by p block vectors P_1 and Q_1 , two sequences of n by p block vectors $\{P_j\}$ and $\{Q_j\}$, called Lanczos vectors, are generated such that for $j = 1, 2, \dots$,

$$\begin{aligned} \text{span}\{P_1^T, P_2^T, \dots, P_j^T\} &= \mathcal{K}_j(P_1^T, A) := \text{span}\{P_1^T, P_1^T A, P_1^T A^2, \dots, P_1^T A^{j-1}\}, \\ \text{span}\{Q_1, Q_2, \dots, Q_j\} &= \mathcal{K}_j(Q_1, A) := \text{span}\{Q_1, A Q_1, A^2 Q_1, \dots, A^{j-1} Q_1\}, \end{aligned}$$

where $\mathcal{K}_j(Q_1, A)$ and $\mathcal{K}_j(P_1^T, A)$ are right and left Krylov subspaces. The block vectors $\{P_j\}$ and $\{Q_j\}$ are constructed so that they are biorthonormal. Together these properties determine the computed quantities up to a scaling. Several nontrivial practical issues are resolved in the implementation presented in Figure 2.1.

The basic block Lanczos iteration implements the three-term recurrences

$$(2.1) \quad B_{j+1} P_{j+1}^T = P_j^T A - A_j P_j^T - C_j P_{j-1}^T,$$

$$(2.2) \quad Q_{j+1} C_{j+1} = A Q_j - Q_j A_j - Q_{j-1} B_j.$$

The procedure can be also viewed as a successive reduction of an $n \times n$ non-Hermitian matrix A to a block tridiagonal form. If we let

$$\mathcal{P}_j = [P_1, P_2, \dots, P_j], \quad \mathcal{Q}_j = [Q_1, Q_2, \dots, Q_j],$$

and

$$(2.3) \quad T_j = \begin{bmatrix} A_1 & B_2 & & & \\ C_2 & A_2 & \ddots & & \\ & \ddots & \ddots & B_j & \\ & & C_j & A_j & \end{bmatrix},$$

then the three-term recurrences (2.1) and (2.2) have the matrix form

$$(2.4) \quad \mathcal{P}_j^T A = T_j \mathcal{P}_j^T + E_j B_{j+1} P_{j+1}^T,$$

$$(2.5) \quad A \mathcal{Q}_j = \mathcal{Q}_j T_j + Q_{j+1} C_{j+1} E_j^T,$$

where E_j is a tall thin matrix whose bottom square is an identity matrix and which vanishes otherwise. Furthermore, the computed Lanczos vectors P_j and Q_j satisfy the *biorthonormality*

$$(2.6) \quad \mathcal{P}_j^T \mathcal{Q}_j = I.$$

When the blocksize $p = 1$, this is just the unblocked non-Hermitian Lanczos algorithm discussed in [20, p. 503].

Remark 1. For a complex matrix A we still use the transpose \cdot^T instead of the conjugate transpose \cdot^H . If A is complex symmetric and $P_1 = Q_1$, then (2.4) is the transpose of (2.5), and it is necessary to compute only one of these two recurrences provided that a complex symmetric scaling scheme is used at step 3.4 in Figure 2.1.

Remark 2. The above block Lanczos algorithm can breakdown prematurely if $R_j^T S_j$ is singular (see step 3.6 in Figure 2.1). We will discuss this issue in section 3.

Remark 3. Many choices of the $p \times p$ nonsingular scaling matrices B_{j+1} and C_{j+1} satisfy $R_j^T S_j = B_{j+1} C_{j+1}$. The one presented here involves a little more work (computing singular value decomposition (SVD) of $P_{j+1}^T Q_{j+1}$), but it maintains that the local basis vectors in P_{j+1} and Q_{j+1} are orthogonal and at the same time biorthogonal to each other. Furthermore, the singular values provide principal angles between the subspaces spanned by P_{j+1} and Q_{j+1} , which is a measure of the quality of the bases constructed (see Remark 4 below).

An alternative scaling maintains the unit length of all Lanczos vectors. This scaling scheme for the unblocked Lanczos algorithm is used in [17, 36, 14]. In this case the Lanczos algorithm determines a pencil $(\widehat{T}_j, \Omega_j)$, where \widehat{T}_j is tridiagonal and Ω_j is diagonal. It can be shown that the tridiagonal T_j determined by the above unblocked ($p = 1$) Lanczos algorithm and this pencil are related by

$$T_j = \pm |\Omega_j|^{1/2} \widehat{T}_j |\Omega_j|^{1/2}.$$

The Lanczos vectors are also related, up to sign, by a similar scaling.

Remark 4. The condition numbers of the Lanczos vectors \mathcal{P}_j and \mathcal{Q}_j can be monitored by the diagonal matrices $\Sigma_1, \Sigma_2, \dots, \Sigma_j$. Recall that the condition number of the rectangular matrix \mathcal{Q}_j is defined by

$$\text{cond}(\mathcal{Q}_j) \stackrel{\text{def}}{=} \|\mathcal{Q}_j\|_2 \|\mathcal{Q}_j^\dagger\|_2 = \frac{\|\mathcal{Q}_j\|_2}{\sigma_{\min}(\mathcal{Q}_j)},$$

where $\sigma_{\min}(\mathcal{Q}_j) = \min_{\|\mathbf{x}\|_2=1} \|\mathcal{Q}_j \mathbf{x}\|_2$. To derive the bound for $\text{cond}(\mathcal{Q}_j)$ observe from step 3.6 in Figure 2.1 that $\|P_i^T\|_2 = \|Q_i\|_2 = \|\Sigma_i^{-1/2}\|_2$. Then for a unit vector v such that $\|\mathcal{P}_j^T\|_2 = \|\mathcal{P}_j^T v\|_2$,

$$\|\mathcal{P}_j^T\|_2^2 = \|\mathcal{P}_j^T v\|_2^2 = \sum_{i=1}^j \|P_i^T v\|_2^2 \leq \sum_{i=1}^j \frac{1}{\min(\Sigma_i)},$$

where $\min(\Sigma_i)$ denotes the smallest diagonal element of Σ_i . The latter bound also applies to \mathcal{Q}_j . Furthermore, we note that the biorthonormality condition (2.6) implies that $\|\mathcal{P}_j\|_2 \sigma_{\min}(\mathcal{Q}_j) \geq 1$. Therefore,

$$\text{cond}(\mathcal{Q}_j) \leq \|\mathcal{P}_j\|_2 \|\mathcal{Q}_j\|_2 \leq \sum_{i=1}^j \frac{1}{\min(\Sigma_i)}.$$

The bound applies to $\text{cond}(\mathcal{P}_j)$ by the similar argument. This generalizes and slightly improves a result from [36].

Remark 5. This implementation is a generalization of the block Hermitian Lanczos algorithms of Golub and Underwood [19] and Grimes, Lewis, and Simon [22] to the non-Hermitian case. A simple version of the block non-Hermitian Lanczos procedure has been studied in [3]. Other implementations of the basic block non-Hermitian Lanczos procedure have been proposed for different applications in [7].

2.2. Eigenvalue approximation. To extract approximate the eigenvalues and eigenvectors of A , we solve the eigenvalue problem of the $jp \times jp$ block tridiagonal matrix T_j after step 3.3 in Figure 2.1. Each eigentriplet (θ, w^H, z) of T_j ,

$$w^H T_j = \theta w^H \quad \text{and} \quad T_j z = z \theta,$$

determines a *Rayleigh–Ritz triplet*, (θ, y^H, x) , where $y^H = w^H \mathcal{P}_j^T$ and $x = \mathcal{Q}_j z$. Rayleigh–Ritz triplets approximate eigentriplets of A .

To assess the approximation, (θ, y^H, x) , of an eigentriplet of the matrix A , let s and r denote the corresponding left and right residual vectors. Then by (2.4) and (2.5), we have

$$(2.7) \quad s^H = y^H A - \theta y^H = (w^H E_j) B_{j+1} P_{j+1}^T,$$

$$(2.8) \quad r = Ax - x\theta = Q_{j+1} C_{j+1} (E_j^T z).$$

Note that a remarkable feature of the Lanczos algorithm is that the residual norms $\|s^H\|_2$ and $\|r\|_2$ are available without explicitly computing y^H and x . There is no need to form y^H and x until their accuracy is satisfactory.

The residuals determine a backward error bound for the triplet. The biorthogonality condition, (2.6), applied to the definition of x and y^H yields

$$(2.9) \quad P_{j+1}^T x = 0 \quad \text{and} \quad y^H Q_{j+1} = 0.$$

From (2.8) and (2.7), we have the following measure of the backward error for the Rayleigh–Ritz triplet (θ, y^H, x) :

$$y^H (A - F) = \theta y^H \quad \text{and} \quad (A - F)x = x\theta,$$

where the backward error matrix F is

$$(2.10) \quad F = \frac{rx^H}{\|x\|_2^2} + \frac{ys^H}{\|y\|_2^2}$$

and $\|F\|_F^2 = \|r\|_2^2/\|x\|_2^2 + \|s^H\|_2^2/\|y\|_2^2$. That is, the left and right residual norms bound the distance to the nearest matrix to A with eigentriplet (θ, y^H, x) . In fact it has been shown that F is the smallest perturbation of A such that (θ, y^H, x) is an eigentriplet of $A - F$ [25]. The computed Rayleigh–Ritz value θ is a $\|F\|_F$ -pseudo eigenvalue of the matrix A [51].

If we write $A = B + F$, where $B = A - F$, then a first-order perturbation analysis indicates that there is an eigenvalue λ of A such that

$$|\lambda - \theta| \leq \text{cond}(\theta)\|F\|_2,$$

where $\text{cond}(\theta) = \|y^H\|_2 \|x\|_2/|y^Hx|$ is the condition number of the Rayleigh–Ritz value θ [20]. This first-order estimate is very often pessimistic because θ is a *two-sided* or generalized Rayleigh quotient [34]. A second-order perturbation analysis yields a more realistic error estimate, which should be used as a stopping criterion. Global second-order bounds for the accuracy of the generalized Rayleigh quotient may be found in [49] and [9]. Here we derive an asymptotic bound.

Recall that (θ, y^H, x) is an eigentriplet of $B = A - F$ and that $y^HF = s^H$ and $Fx = r$. Assume that B has distinct eigenvalues $\{\theta_i\}$ and the corresponding normalized left and right eigenvectors $\{y_i^H, x_i\}$ ($\|y_i^H\|_2 = \|x_i\|_2 = 1$). Let us perturb $\theta = \theta(0)$ toward an eigenvalue λ of A using the implicit function $\theta(t) = \theta(B + tE)$ for $E = F/\|F\|_2$. Under classical results from function theory [26], it can be shown that in a neighborhood of the origin there exist differentiable $\theta(t)$, $y^H(t)$, and $x(t)$ ($\|y^H(t)\|_2 = \|x(t)\|_2 = 1$) such that

$$(2.11) \quad y^H(t)(B + tE) = \theta(t)y^H(t) \quad \text{and} \quad (B + tE)x(t) = x(t)\theta(t).$$

Next expand $\theta(t)$ about $t = 0$:

$$\lambda = \theta(\|F\|_2) = \theta(0) + \theta'(0)\|F\|_2 + \frac{1}{2}\theta''(0)\|F\|_2^2 + O(\|F\|_2^3).$$

By differentiating (2.11) with respect to t , and setting $t = 0$, we obtain

$$\theta'(0) = \frac{1}{\|F\|_2} \frac{y^HFx}{y^Hx}.$$

Note that from (2.10), $y^HFx = y^Hr + s^Hx$. Substitute (2.7), (2.8), and (2.9) to find $y^HFx = 0$. This implies the stationarity property $\theta'(0) = 0$. Differentiate (2.11) with respect to t twice, and set $t = 0$, and there appears

$$\theta''(0) = \frac{2}{\|F\|_2} \frac{s^H}{y^Hx} x'(0).$$

Now the standard eigenvector sensitivity analysis gives

$$x'(0) = \sum_{\theta_i \neq \theta} \frac{y_i^HEx}{(\theta - \theta_i)y_i^Hx_i} x_i.$$

See, for example, Golub and Van Loan [20, p. 345]. From the above two formulas, up to the second order of $\|F\|_2$, we obtain

$$(2.12) \quad |\lambda - \theta| \leq \frac{\|s^H\|_2 \|r\|_2}{\text{gap}(\theta, B)} \left(\frac{1}{|y^H x|} \sum_{\theta_i \neq \theta} \frac{1}{|y_i^H x_i|} \right).$$

Here $\text{gap}(\theta, B) = \min_{\theta_i \neq \theta} |\theta - \theta_i|$. Note that the term in the parentheses involves the condition numbers of the eigenvalues $\{\theta_i\}$ of B .

The bound (2.12) shows that the accuracy of the Rayleigh–Ritz value θ is proportional to the product of the left and right residuals and the inverse of the gap in eigenvalues of B . We call this the *semiquadratic convergence*. Since $\text{gap}(\theta, B)$ is not computable, we use the $\text{gap}(\theta, T_j)$ to approximate $\text{gap}(\theta, B)$ when $\|F\|_2$ is small. From (2.10) and (2.12), we advocate accepting θ as an approximate eigenvalue of A if

$$(2.13) \quad \min \left\{ \|s^H\|_2, \|r\|_2, \frac{\|s^H\|_2 \|r\|_2}{\text{gap}(\theta, T_j)} \right\} \leq \tau_c,$$

where τ_c is a given accuracy threshold. Note that for ill-posed problems, small residuals (backward errors) do not imply high eigenvalue accuracy (small forward error). In this case, the estimate is optimistic. In any case, since both the left and right approximate eigenvectors are available, the approximate eigenvalue condition numbers are readily computable. This detects ill-posedness in an eigenvalue problem. See numerical example 5 in section 7.

It is well known that for Hermitian matrices, the Lanczos algorithm reveals first the outer and well-separated eigenvalues [35]. In the block Hermitian Lanczos algorithm with blocksize p , the outer eigenvalues and the eigenvalue clusters of order up to p that are well separated from the remaining spectra converge first [19, 41]. This qualitative understanding of convergence has been extended to the block Arnoldi algorithm for non-Hermitian eigenproblems in [42, 24].

2.3. Maintaining the biorthogonality of the Lanczos vectors. The quantities computed in the block Lanczos algorithm in the presence of finite precision arithmetic have different properties than the corresponding exact quantities. The biorthogonality property, (2.6), fails to hold, and the columns of the matrices \mathcal{P}_j and \mathcal{Q}_j are spanning sets but not bases. The loss of linear independence in the matrices \mathcal{P}_j and \mathcal{Q}_j computed by the three-term recurrence is coherent; as a Rayleigh–Ritz triplet converges to an eigentriplet of A , copies of the Rayleigh–Ritz values appear. At this iteration, \mathcal{Q}_j is singular because it maps a group of right eigenvectors of T_j to an eigenvector of A .

For example, in a Lanczos run of 100 iterations, one may observe 5 copies of the dominant eigenvalue of A among the Rayleigh–Ritz values. This increases the number of iterations required to complete a given task. As a partial remedy, we advocate maintaining *local biorthogonality* to ensure the biorthogonality among consecutive Lanczos vectors in the three-term recurrences [14]. Local biorthogonality is maintained as follows. After step 3.2 in Figure 2.1,

$$\begin{aligned} R_j &:= R_j - P_j(Q_j^T R_j), \\ S_j &:= S_j - Q_j(P_j^T S_j). \end{aligned}$$

Repeating this inner loop increases the number of floating point operations in a Lanczos iteration. However, no new data transfer is required, and without repetition

the local biorthogonality would normally be swamped. The cost effectiveness seems indisputable.

Another limitation of simple implementations of the three-term recurrences is that the multiplicity of an eigenvalue of A is not related in any practical way to the multiplicity of a Rayleigh–Ritz value. To reveal the multiplicity or clustering of an eigenvalue it typically suffices to explicitly enforce (2.6). This variation has been called a Lanczos algorithm with *full rebiorthogonalization* [38]. It is maintained by incorporating a variant of the Gram–Schmidt process called the two-sided modified Gram–Schmidt biorthogonalization (TSMGS) [36]. After step 3.6 in Figure 2.1, we biorthogonalize P_{j+1} and Q_{j+1} in place against all previous Lanczos vectors $\mathcal{P}_j = [P_1, P_2, \dots, P_j]$ and $\mathcal{Q}_j = [Q_1, Q_2, \dots, Q_j]$:

```

for  $i = 1, 2, \dots, j$ 
     $P_{j+1} := P_{j+1} - P_i(Q_i^T P_{j+1})$ 
     $Q_{j+1} := Q_{j+1} - Q_i(P_i^T Q_{j+1})$ 
end
    
```

Maintaining full biorthogonality substantially increases the cost per iteration of the Lanczos algorithm. To be precise, at Lanczos iteration j , an additional $8p^2jn$ flops is required. More importantly all the computed Lanczos vectors are accessed at each iteration. This is very often the most costly part of a Lanczos run, although there are cases where the matrix-vector multiplications may be the dominating factor. A less-costly alternative to full biorthogonality is presented in section 4.

3. An adaptive block Lanczos algorithm. In this section, we present an adaptive block scheme. This algorithm has the flexibility to adjust the blocksize to the multiplicity or the order of a cluster of desired eigenvalues. In addition, the algorithm can be used to cure (near) breakdowns.

3.1. Augmenting the Lanczos vectors. In a variable block Lanczos algorithm, at the j th iteration, P_j and Q_j have p_j columns, respectively. At the next $(j + 1)$ th iteration, the number of columns of the Lanczos vectors P_{j+1} and Q_{j+1} can be increased by k as follows.

First note that for any n by k matrices \widehat{P}_{j+1} and \widehat{Q}_{j+1} , the basic three-term recurrences (2.1) and (2.2) also hold with augmented $(j + 1)$ th Lanczos vectors $[P_{j+1} \widehat{P}_{j+1}]$ and $[Q_{j+1} \widehat{Q}_{j+1}]$:

$$\begin{bmatrix} B_{j+1} & 0 \end{bmatrix} \begin{bmatrix} P_{j+1}^T \\ \widehat{P}_{j+1}^T \end{bmatrix} = P_j^T A - A_j P_j^T - C_j P_{j-1}^T$$

and

$$\begin{bmatrix} Q_{j+1} & \widehat{Q}_{j+1} \end{bmatrix} \begin{bmatrix} C_{j+1} \\ 0 \end{bmatrix} = A Q_j - Q_j A_j - Q_{j-1} B_j.$$

Provided that

$$(3.1) \quad \begin{bmatrix} P_{j+1} & \widehat{P}_{j+1} \end{bmatrix}^T \begin{bmatrix} Q_{j+1} & \widehat{Q}_{j+1} \end{bmatrix}$$

is *nonsingular*, the Lanczos procedure continues as before under the substitutions

$$P_{j+1} \leftarrow \begin{bmatrix} P_{j+1} & \widehat{P}_{j+1} \end{bmatrix}, \quad Q_{j+1} \leftarrow \begin{bmatrix} Q_{j+1} & \widehat{Q}_{j+1} \end{bmatrix}$$

with proper normalization and

$$B_{j+1} \leftarrow \begin{bmatrix} B_{j+1} & 0 \end{bmatrix}, \quad C_{j+1} \leftarrow \begin{bmatrix} C_{j+1} \\ 0 \end{bmatrix}.$$

The only other constraint on \widehat{P}_{j+1} and \widehat{Q}_{j+1} is that they satisfy the biorthogonality condition among the Lanczos vectors; i.e., it is required that

$$\widehat{P}_{j+1}^T \mathcal{Q}_j = 0 \quad \text{and} \quad \mathcal{P}_j^T \widehat{Q}_{j+1} = 0.$$

As a consequence, the adaptive block scheme has the same governing equations and the same resulting Rayleigh–Ritz approximation properties as the basic block Lanczos method described in section 2.

Before we turn to the usage of the adaptive block scheme, we discuss the choice of the increment vectors \widehat{P}_{j+1}^T and \widehat{Q}_{j+1} . Ideally we would like to choose augmentations so that the resulting matrix $P_{j+1}^T Q_{j+1}$ is well conditioned. To be precise we want the smallest singular value of $P_{j+1}^T Q_{j+1}$ to be larger than the given threshold τ_b , say $\tau_b = 10^{-8}$ in double precision. However, there may not exist \widehat{P}_{j+1}^T and \widehat{Q}_{j+1} such that the given threshold τ_b is satisfied. A natural choice to choose \widehat{P}_{j+1} and \widehat{Q}_{j+1} in practice is to biorthogonalize a pair of random n by k vectors to the previous Lanczos vectors. In other words, the vectors \widehat{P}_{j+1} and \widehat{Q}_{j+1} are computed by applying TSMGS (see section 2.3) to a pair of random n by k vectors. The construction is repeated a few times (say, 3 at most) if necessary to ensure that the smallest singular value of (3.1) is larger than a threshold. We observe that this works well in practice.

3.2. Adaptive blocking for clustered eigenvalues. If A has an eigenvalue of multiplicity greater than the blocksize, then the Rayleigh–Ritz values converge slowly to this group of eigenvalues [12, 19, 22, 3]. In some applications, information about multiplicity is available a priori and then the blocksize can be chosen accordingly. But when this information is not available, it is desirable to adjust the blocksize using the information obtained during the iteration.

In any variable block implementation of the Lanczos algorithm in which the biorthogonality of the computed Lanczos vectors is maintained, it is advantageous to increase the blocksize to the order of the largest cluster of Rayleigh–Ritz values, $\{\theta_i\}$. The adaptive block scheme proposed in section 3.1 offers such flexibility.

The cluster of Rayleigh–Ritz values about θ_i is the set of all θ_k such that

$$(3.2) \quad |\theta_i - \theta_k| \leq \eta \max(|\theta_i|, |\theta_k|),$$

where η is a user-specified clustering threshold. The order of the largest cluster of Rayleigh–Ritz values is computed whenever we test for convergence, and the blocksize is increased to the order of the largest cluster.

3.3. Adapting the blocksize to treat breakdown. A second reason to increase the blocksize is to overcome a breakdown in the block Lanczos algorithm. Recall from section 2.1 that breakdown occurs when $R_j^T S_j$ is singular. There are two cases:

- I. Either R_j or S_j is rank deficient.
- II. Both R_j and S_j are not rank deficient but $R_j^T S_j$ is.

Exact breakdowns are rare, but near breakdowns (i.e., $R_j^T S_j$ has singular values close to 0) do occur. In finite precision arithmetic this can cause numerical instability.

In case I, if S_j vanishes in step 3.2 of Figure 2.1 of the basic block Lanczos algorithm, an invariant subspace is detected. To restart the Lanczos procedure choose

Q_{j+1} to be any vector such that $\mathcal{P}_j^T Q_{j+1} = 0$. If S_j is just (nearly) rank deficient, then after the QR decomposition of S_j , $S_j = Q_{j+1}C_{j+1}$, we biorthogonalize Q_{j+1} to the previous left Lanczos vectors \mathcal{P}_j . This also effectively expands the Krylov subspace and continues the procedure. Rank deficiency of R_j is treated similarly. Note that in this case, the blocksize is not changed. This generalizes the treatment suggested by Wilkinson for the unblocked Lanczos procedure [52, p. 389].

Case II is called a *serious breakdown* [52]. Let us first examine the case of exact breakdown. Let $R_j = P_{j+1}B_{j+1}^T$ and $S_j = Q_{j+1}C_{j+1}$ be the QR decompositions of R_j and S_j . In this case, $P_{j+1}^T Q_{j+1}$ is singular. Suppose that $P_{j+1}^T Q_{j+1}$ has the SVD

$$P_{j+1}^T Q_{j+1} = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^H,$$

where Σ is nonsingular if it exists (Σ may be 0 by 0). Let us see how to augment P_{j+1} and Q_{j+1} so that $P_{j+1}^T Q_{j+1}$ is nonsingular. For clarity, drop the subscript $j + 1$ and partition $P\bar{U}$ and QV into

$$P\bar{U} = [P_{(1)} \quad P_{(2)}] \quad \text{and} \quad QV = [Q_{(1)} \quad Q_{(2)}].$$

Here the number of columns of $P_{(2)}$ and $Q_{(2)}$ is the number of zero singular values of $P^T Q$. Let the augmented Lanczos vectors be

$$P := [P_{(1)} \quad P_{(2)} \quad \hat{P}] \quad \text{and} \quad Q := [Q_{(1)} \quad Q_{(2)} \quad \hat{Q}],$$

where

$$\hat{P} = (I - \Pi_j)^T \bar{Q}_{(2)} \quad \text{and} \quad \hat{Q} = (I - \Pi_j) \bar{P}_{(2)}.$$

$\Pi_j = \mathcal{Q}_j \mathcal{P}_j^T$ is the oblique projector. The biorthogonality condition (2.6) and then the orthonormality of the columns of $[P_{(1)} \quad P_{(2)}]$ yield

$$P_{(1)}^T \hat{Q} = P_{(1)}^T (I - \Pi_j) \bar{P}_{(2)} = P_{(1)}^T \bar{P}_{(2)} = 0$$

and

$$P_{(2)}^T \hat{Q} = P_{(2)}^T (I - \Pi_j) \bar{P}_{(2)} = P_{(2)}^T \bar{P}_{(2)} = I.$$

Similarly, $\hat{P}^T Q_{(1)} = 0$ and $\hat{P}^T Q_{(2)} = I$. Therefore, we have

$$P^T Q = \begin{bmatrix} \Sigma & 0 & 0 \\ 0 & 0 & I \\ 0 & I & \hat{P}^T \hat{Q} \end{bmatrix},$$

which is nonsingular for any $\hat{P}^T \hat{Q}$. Therefore, we conclude that exact breakdowns are always curable by the adaptive blocksize technique.

However, for the near breakdown case, the situation is more complicated. The above choice may not succeed in increasing the smallest singular value of $P_{j+1}^T Q_{j+1}$ above a specific given threshold, τ_b . The difficulty involves the fact that the norms of \hat{P} and \hat{Q} can be large because of the use of oblique projector Π_j . In our implementation, we have chosen \hat{P} and \hat{Q} by dualizing a pair of random n by k vectors to the previous

Lanczos vectors as described in section 3.1. The increment to the blocksize is the number of singular values of $P_{j+1}^T Q_{j+1}$ below a specified threshold.

Another scheme for adjusting the block size to cure (near) breakdown is the *look-ahead* strategy [38, 17]. In the look-ahead Lanczos algorithm, the spans of the columns of \mathcal{P}_j and \mathcal{Q}_j remain within $\mathcal{K}(P_1^T, A)$ and $\mathcal{K}(Q_1, A)$, respectively. Specifically, P_{j+1}^T and Q_{j+1} are augmented by

$$\widehat{P} = (I - \Pi_j)^T A^T P_{j+1} = A^T P_{j+1} - P_j C_{j+1}^T$$

and

$$\widehat{Q} = (I - \Pi_j) A Q_{j+1} = A Q_{j+1} - Q_j B_{j+1}.$$

If $[P_{j+1} \widehat{P}^T] [Q_{j+1} \widehat{Q}]$ is not (nearly) singular, then one step of look-ahead is successful and P_{j+1} and Q_{j+1} are obtained from P and Q , respectively, after normalization. Since

$$\text{span}(\mathcal{Q}_{j+1}) = \text{span}(\mathcal{Q}_j, [Q_{j+1}, \widehat{Q}])$$

and

$$\begin{aligned} \text{span}(\mathcal{Q}_{j+2}) &= \text{span}(\mathcal{Q}_j, [Q_{j+1}, \widehat{Q}], A[Q_{j+1}, \widehat{Q}]) \\ &= \text{span}(\mathcal{Q}_{j+1}, A^2 Q_{j+1}), \end{aligned}$$

Q_{j+2} has no more columns than Q_{j+1} prior to augmentation. That is, the block size doubles at step $j + 1$ only and then returns to the ambient block size at the following step $j + 2$. It may be necessary to repeatedly augment the $(j + 1)$ th Lanczos block-vectors [36]. In contrast, we have shown that the adaptive strategy has the property that an exact breakdown is cured in using a fixed number of augmentation vectors. Moreover, to reveal clustered eigenvalues and to eliminate a potential source of slow convergence, we store \mathcal{P}_j and \mathcal{Q}_j and maintain biorthogonality (see section 4). We have found the adaptive block scheme to be a viable alternative to look-ahead strategies here.

4. Maintaining semibiorthogonality. In this section we present a form of *limited* rebiorthogonalization that is more efficient than the full rebiorthogonalization described in section 2.3. This method extends the block Hermitian Lanczos algorithm with *partial* reorthogonalization to the non-Hermitian case [22]. Instead of maintaining full biorthogonality (section 2.3), only *semibiorthogonality* is maintained at each iteration; i.e., for $j \geq 1$,

$$(4.1) \quad d_{j+1} = \max \left(\frac{\|\mathcal{P}_j^T Q_{j+1}\|_1}{\|\mathcal{P}_j\|_1 \|Q_{j+1}\|_1}, \frac{\|\mathcal{Q}_j^T P_{j+1}\|_1}{\|\mathcal{Q}_j\|_1 \|P_{j+1}\|_1} \right) \leq \sqrt{\epsilon},$$

where ϵ is the roundoff error unit. This generalizes the definition of semibiorthogonality for the unblocked Lanczos algorithm [14]. We will show that semibiorthogonality requires less computation and data transfer to maintain than full biorthogonality. In particular, \mathcal{P}_j and \mathcal{Q}_j are accessed only at certain iterations.

In section 4.1 we show how to monitor the loss of numerical biorthogonality without significantly increasing the number of floating point operations in the Lanczos recurrences. In section 4.2 we show how best to correct the loss of biorthogonality.

4.1. Monitoring the loss of biorthogonality. When the Lanczos algorithm is implemented in finite precision arithmetic, the computed quantities can be modeled by perturbed three-term recurrences:

$$(4.2) \quad B_{j+1}P_{j+1}^T = P_j^T A - A_j P_j^T - C_j P_{j-1}^T - F_j^T,$$

$$(4.3) \quad Q_{j+1}C_{j+1} = A Q_j - Q_j A_j - Q_{j-1} B_j - G_j,$$

where F_j and G_j represent the roundoff error introduced at iteration j . By applying the standard model of the rounding errors committed in floating point arithmetic [52], it can be shown that to first order in roundoff errors there holds

$$\begin{aligned} \|F_j\|_F &\leq \mathbf{u}(\|A\|_1 \|P_j\|_1 + \|A_j\|_1 \|P_j\|_1 + \|C_j\|_1 \|P_{j-1}\|_1), \\ \|G_j\|_F &\leq \mathbf{u}(\|A\|_1 \|Q_j\|_1 + \|A_j\|_1 \|Q_j\|_1 + \|B_j\|_1 \|Q_{j-1}\|_1), \end{aligned}$$

where \mathbf{u} is a constant multiple of the roundoff error unit ϵ . The governing equations for the computed quantities are

$$(4.4) \quad \mathcal{P}_j^T A = T_j \mathcal{P}_j^T + E_j B_{j+1} P_{j+1}^T + \mathcal{F}_j^T,$$

$$(4.5) \quad A \mathcal{Q}_j = \mathcal{Q}_j T_j + Q_{j+1} C_{j+1} E_j^T + \mathcal{G}_j,$$

where the matrices $\mathcal{F}_j = [F_1, F_2, \dots, F_j]$ and $\mathcal{G}_j = [G_1, G_2, \dots, G_j]$ are such that

$$(4.6) \quad \max(\|\mathcal{F}_j\|_F, \|\mathcal{G}_j\|_F) \leq \mathbf{u}(\|A\|_1 + \|T_j\|_1) \max(\|\mathcal{P}_j\|_F, \|\mathcal{Q}_j\|_F).$$

A detailed analysis for the unblocked case can be found in [2, 14].

Now we use this model of rounding errors in the Lanczos process to quantify the propagation of the loss of biorthogonality from iteration to iteration. The biorthogonality of the $(j+1)$ th Lanczos vectors to the previous Lanczos vectors can be measured using the short vectors

$$X_j = \mathcal{P}_j^T Q_{j+1} \quad \text{and} \quad Y_j = P_{j+1}^T \mathcal{Q}_j.$$

In the following, we show that these vectors satisfy perturbed three-term recurrences which we can use to efficiently monitor the biorthogonality loss.

The recurrence for X_j is derived as follows. Note that

$$(4.7) \quad \mathcal{P}_j^T Q_j = \begin{bmatrix} X_{j-1} \\ 0 \end{bmatrix} + E_j.$$

Let $W_{ij} = P_i^T Q_j$. Multiply (4.3) by \mathcal{P}_j^T on the left, substitute in (4.4) $\times Q_j$, and there appears

$$(4.8) \quad \begin{aligned} X_j C_{j+1} &= T_j \mathcal{P}_j^T Q_j - \mathcal{P}_j^T Q_j A_j - \mathcal{P}_j^T Q_{j-1} B_j \\ &\quad + E_j B_{j+1} W_{j+1,j} + \mathcal{F}_j^T Q_j - \mathcal{P}_j^T G_j. \end{aligned}$$

Substitute (4.7) above and (2.3), the definition of T_j , and simplify to find

$$(4.9) \quad T_j \mathcal{P}_j^T Q_j - \mathcal{P}_j^T Q_j A_j = T_j \begin{bmatrix} X_{j-1} \\ 0 \end{bmatrix} - \begin{bmatrix} X_{j-1} \\ 0 \end{bmatrix} A_j + E_{j-1} B_j.$$

In addition, we have the identity

$$(4.10) \quad \mathcal{P}_j^T Q_{j-1} = \begin{bmatrix} X_{j-2} \\ 0 \\ 0 \end{bmatrix} + E_{j-1} + W_{j,j-1} E_j.$$

Substituting (4.9) and (4.10) into (4.8) finally yields

$$(4.11) \quad X_j C_{j+1} = T_j \begin{bmatrix} X_{j-1} \\ 0 \end{bmatrix} - \begin{bmatrix} X_{j-1} \\ 0 \end{bmatrix} A_j - \begin{bmatrix} X_{j-2} \\ 0 \\ W_{j,j-1} \end{bmatrix} B_j + E_j B_{j+1} W_{j+1,j} + O(\mathbf{u}_j),$$

where $O(\mathbf{u}_j)$ represents the local rounding error term $\mathcal{F}_j^T Q_j - \mathcal{P}_j^T G_j$ and

$$\mathbf{u}_j = \mathbf{u}(\|T_j\|_1 + \|A\|_1) \max(\|\mathcal{P}_j\|_F, \|\mathcal{Q}_j\|_F).$$

The similar analysis of the left Lanczos vectors yields

$$(4.12) \quad B_{j+1} Y_j = [Y_{j-1}, 0] T_j - A_j [Y_{j-1}, 0] - C_j [Y_{j-2}, 0, W_{j-1,j}] + W_{j,j+1} C_{j+1} E_j + O(\mathbf{u}_j).$$

Equations (4.11) and (4.12) model the propagation of the loss of the numerical biorthogonality among Lanczos vectors from iteration to iteration. The following algorithm implements these recurrence relations to monitor the biorthogonality loss. Note that the scalar parameter \hat{d}_{j+1} is our measure of the biorthogonality. When $\hat{d}_{j+1} > \sqrt{\epsilon}$, then TSMGS is invoked to recover biorthogonality as described in the next section.¹

ALGORITHM FOR MONITORING THE LOSS OF BIORTHOGONALITY.

Initially, when $j = 1$, we set $X_1 = 0, Y_1 = 0, d_1 = \mathbf{u}$, compute $X_2 = P_1^T Q_2, Y_2 = P_2^T Q_1$, and let $W_1^{(l)} = Y_2, W_1^{(r)} = X_2$. When $j > 1$,

1. $W_2^{(l)} = P_{j+1}^T Q_j$
2. $X_3 = T_j \begin{bmatrix} X_2 \\ 0 \end{bmatrix} - \begin{bmatrix} X_2 \\ 0 \end{bmatrix} A_j - \begin{bmatrix} X_1 \\ W_1^{(l)} \end{bmatrix} B_j + \begin{bmatrix} 0 \\ B_{j+1} W_2^{(l)} \end{bmatrix}$
3. $X_3 := (X_3 + F_j) C_{j+1}^{-1}$
4. $X_1 = \begin{bmatrix} X_2 \\ 0 \end{bmatrix}; \quad X_2 = X_3$
5. $W_2^{(r)} = P_j^T Q_{j+1}$
6. $Y_3 = [Y_2 \ 0] T_j - A_j [Y_2 \ 0] - C_j \begin{bmatrix} Y_1 & W_1^{(r)} \end{bmatrix} + \begin{bmatrix} 0 & W_2^{(r)} C_{j+1} \end{bmatrix}$
7. $Y_3 := B_{j+1}^{-1} (Y_3 + F_j^T)$
8. $Y_1 = [Y_2 \ 0]; \quad Y_2 = Y_3$
9. $W_1^{(l)} = W_2^{(l)}; \quad W_1^{(r)} = W_2^{(r)}$
10. $\hat{d}_{j+1} = \max(\|X_2\|_1 / (\|\mathcal{P}_j\|_1 \|Q_{j+1}\|_1), \|Y_2\|_\infty / (\|\mathcal{Q}_j\|_1 \|P_{j+1}\|_1))$

The matrix F_j is a random matrix scaled to have norm \mathbf{u}_j to simulate the roundoff errors in the three-term recurrences. The number of floating point operations per iteration of the monitoring algorithm is $2j^2 + O(n)$, where the $2j^2$ is for the multiplications by T_j in steps 2 and 6 above and the n comes from the ‘‘inner products’’ of block Lanczos vectors in steps 1 and 5 above. If the block tridiagonal structure of T_j is taken in account, then the cost is just $O(n)$. Therefore the cost of the monitoring algorithm is not significant, as promised.

¹To economize on storage there is a subtle change of notation in the following monitoring algorithm. At Lanczos iteration j , the vectors X_{j-1}, X_j , and X_{j+1} are denoted X_1, X_2 , and X_3 , and the previous X_k are not stored. Similar conventions apply to $\{Y_i\}$ and $\{W_{i,k}\}$.

4.2. Correcting the loss of biorthogonality. When action is required to maintain semibiorthogonality (4.1), TSMGS (see section 2.3) is invoked to rebiorthonormalize or *correct* the candidate Lanczos vectors P_{j+1} and Q_{j+1} . Recall from (4.11) that the sequence $\{X_j\}$ satisfies a perturbed three-term recurrence. Correcting Q_{j+1} annihilates the $O(\sqrt{\epsilon})$ matrix X_j , but at the next Lanczos iteration X_{j+1} will be a multiple of the nearly $O(\sqrt{\epsilon})$ matrix X_{j-1} . Instead, as Q_{j+1} is corrected to maintain semibiorthogonality, we also correct Q_j ; in this way the biorthogonality of the following Lanczos vectors can deteriorate gradually. The similar comments hold for the left Lanczos vectors. There is a much better way to do this than to apply TSMGS at consecutive iterations to the pairs of P_j and Q_j and P_{j+1} and Q_{j+1} , respectively. Instead, as the columns of \mathcal{P}_j and \mathcal{Q}_j are transferred from slow storage to the computational unit to correct P_{j+1} and Q_{j+1} , the previous Lanczos vectors P_j and Q_j also can be *retroactively* corrected. This halves the amount of data transfer required.

RETROACTIVE TSMGS. Biorthogonalize P_j , P_{j+1} , Q_j , and Q_{j+1} against the previous Lanczos vectors *in place*.

```

for  $i = 1, 2, \dots, j - 1$ 
     $P_j := P_j - P_i(Q_i^T P_j)$ 
     $P_{j+1} := P_{j+1} - P_i(Q_i^T P_{j+1})$ 
     $Q_j := Q_j - Q_i(P_i^T Q_j)$ 
     $Q_{j+1} := Q_{j+1} - Q_i(P_i^T Q_{j+1})$ 
end
 $P_{j+1} := P_{j+1} - P_j(Q_j^T P_{j+1})$ 
 $Q_{j+1} := Q_{j+1} - Q_j(P_j^T Q_{j+1})$ 

```

We do *not* update the QR decompositions and SVDs computed in the basic Lanczos algorithm after retroactive TSMGS for the same technical reasons discussed in section 6.3 of [14] for the unblocked Lanczos algorithm.

5. The ABLE method. In summary, the ABLE method presented in Figure 5.1 incorporates an adaptive blocking scheme (section 3) into the basic block Lanczos algorithm (section 2) and maintains the local and semibiorthogonality of Lanczos vectors (section 4). Specifically, we have the following:

- At step 3.3, we suggest the use of (2.13) in section 2.2 as the stopping criterion. Then, at the end of a Lanczos run, we compute the residual norms $\|s^H\|_2$ and $\|r\|_2$ corresponding to the converged Rayleigh–Ritz triplets (θ, y^H, x) . See (2.7) and (2.8) in section 2.2. Note that the theory in section 2.2 is based on the exact biorthogonality. When only semibiorthogonality is maintained, $\theta'(0)$ is no longer zero. However, using (4.4), (4.5), and semibiorthogonality (4.1), it is easy to see that $\theta'(0)$ is still in the magnitude of $\sqrt{\epsilon}$. Thus, as far as $\|F\|_2$ is not too small (not less than $O(\sqrt{\epsilon})$), the second term in the expansion for λ still dominates the first term $\theta'(0)\|F\|_2$, and therefore, (2.13) would be valid. (Specifically, $y^H r \sim \sqrt{\epsilon}\|r\|_2$ and the first term in the expansion satisfies $\theta'(0)\|F\|_2 \sim \frac{1}{|y^H x|} \sqrt{\epsilon}\|F\|_2$.)
- At step 3.4, (3.2) is used to compute the order of the largest cluster as described in section 3.2.
- For step 3.7, see section 3.3 for an explanation.
- At step 3.9, τ_b is a threshold for breakdown. $\min(\Sigma)$ is the smallest singular value of the matrix $P_{j+1}^T Q_{j+1}$. If there is (near) breakdown and/or the order of the largest cluster of the converged Rayleigh–Ritz values is larger than the blocksize, then the blocks are augmented as described in section 3.1.

1. Choose starting vectors P_1 and Q_1 so that $P_1^T Q_1 = I$
2. $R = (P_1^T A)^T$ and $S = A Q_1$
3. For $j = 1, 2, \dots$ until convergence
 - 3.1. $A_j = P_j^T S$
 - 3.2. $R := R - P_j A_j^T$ and $S := S - Q_j A_j$
 - 3.3. Compute the eigen-decomposition of T_j , and test for convergence
 - 3.4. Find the largest order δ of the clustering of converged Rayleigh–Ritz values
 - 3.5. Local biorthogonality: $R := R - P_j (Q_j^T R)$ and $S := S - Q_j (P_j^T S)$
 - 3.6. Compute the QR decompositions: $R = P_{j+1} B_{j+1}^T$ and $S = Q_{j+1} C_{j+1}$
 - 3.7. If R or S (or both) is rank deficient, apply TSMGS to biorthogonalize P_{j+1} and Q_{j+1} against the previous Lanczos vectors
 - 3.8. Compute the SVD: $P_{j+1}^T Q_{j+1} = U \Sigma V^H$
 - 3.9. Increase blocksize if $\min(\Sigma) < \tau_b$ and/or $\delta > p_j$
 - 3.10. $B_{j+1} := B_{j+1} U \Sigma^{1/2}$ and $C_{j+1} := \Sigma^{1/2} V^H C_{j+1}$
 - 3.11. $P_{j+1} := P_{j+1} \bar{U} \Sigma^{-1/2}$ and $Q_{j+1} := Q_{j+1} V \Sigma^{-1/2}$
 - 3.12. Monitor the loss of biorthogonality, and correct if necessary
 - 3.13. $R = (P_{j+1}^T A - C_{j+1} P_j^T)^T$ and $S = A Q_{j+1} - Q_j B_{j+1}$

FIG. 5.1. *ABLE method.*

- Algorithms for monitoring the loss of biorthogonality and maintaining semibiorthogonality at step 3.12 are described in sections 4.1 and 4.2.

The non-Hermitian Lanczos algorithm is also called the two-sided Lanczos algorithm because both the operations

$$X^T A \quad \text{and} \quad A X$$

are required at each iteration. A is referenced only as a rule to compute these matrix-vector products. Because of this feature, the algorithm is well suited for large sparse matrices or large structured dense matrices for which matrix-vector products can be computed cheaply. The efficient implementation of these products depends on the data structure and storage format for the A matrix and the Lanczos vectors.

If no Lanczos vectors are saved, the three-term recurrences can be implemented using only six block vectors of length n . To maintain the semibiorthogonality of the computed Lanczos vectors \mathcal{P}_j and \mathcal{Q}_j , it is necessary to store these vectors in core or out-of-core memory. This consumes a significant amount of memory. The user must be conscious of how much memory is needed for each application. For very large matrices it may be best to store the Lanczos vectors out-of-core. After each Lanczos iteration, save the current Lanczos vectors to an auxiliary storage device. The Lanczos vectors are recalled in the procedure TSMGS for rebiorthogonalization and when the converged Rayleigh–Ritz vectors are computed at the end of a Lanczos run.

A block Lanczos algorithm is ideal for application codes that represent A out-of-core. The main cost of a Lanczos iteration, with or without blocks, is accessing A . Block algorithms compute the matrix block vectors product with only one pass over the data structure defining A , with a corresponding savings of work.

The most time-consuming steps in a Lanczos run are to

1. apply the matrix A (from the left and the right),
2. apply retroactive TSMGS to maintain semibiorthogonality, and

3. solve the eigenproblem for the block tridiagonal matrix T_j when j increases. Items 1 and 2 have been addressed already (see the above and section 4.2). For item 3, we presently use the QR algorithm for T_j . We note that it is not necessary to solve the eigenproblem for T_j at each Lanczos iteration. A way to reduce such cost is to solve the eigenvalue problem for T_j only after a correction iteration has been made to maintain semibiorthogonality. This technique utilizes the connection between the loss of the biorthogonality and convergence [33, 35, 2].

6. A spectral transformation ABLE method. In this section we briefly discuss how to use the ABLE method to compute some eigenvalues of the generalized eigenvalue problem

$$(6.1) \quad Kx = \lambda Mx$$

nearest an arbitrary complex number, σ . We assume that $K - \sigma M$ is nonsingular and that it is feasible to solve the linear system of equations with coefficient matrix $K - \sigma M$. The reward for solving this linear system of equations is the rapid convergence of the Lanczos algorithm. In section 7 we apply the ABLE method to such a generalized eigenvalue problem arising in magneto-hydro-dynamics (MHD).

We apply a popular shift-and-invert strategy to the pair (K, M) with shift σ [16]. In this approach, the ABLE method is applied with

$$(6.2) \quad A = (K - \sigma M)^{-1}M.$$

The eigenvalues, μ , of A are $\mu = 1/(\lambda - \sigma)$. The outer eigenvalues of A are now the eigenvalues of (K, M) nearest to σ . This spectral transformation also generally improves the separation of the eigenvalues of interest from the remaining eigenvalues of (K, M) , a very desirable property.

When we apply the ABLE method to the matrix $A = (K - \sigma M)^{-1}M$, the governing equations become

$$(6.3) \quad \mathcal{P}_j^T (K - \sigma M)^{-1}M = T_j \mathcal{P}_j^T + E_j B_{j+1} P_{j+1}^T,$$

$$(6.4) \quad (K - \sigma M)^{-1}M \mathcal{Q}_j = \mathcal{Q}_j T_j + Q_{j+1} C_{j+1} E_j^T.$$

If (θ, w^H, z) is an eigentriplet of T_j , then from the above governing equations (6.3) and (6.4), the triplet

$$\left(\tilde{\lambda}, \tilde{y}^H, \tilde{x} \right) := \left(\sigma + \frac{1}{\theta}, w^H \mathcal{P}_j^T (K - \sigma M)^{-1}, \mathcal{Q}_j z \right)$$

is an approximate eigentriplet of the matrix pair (K, M) . The corresponding left and right residuals are

$$s^H = \tilde{y}^H K - \tilde{\lambda} \tilde{y}^H M = -\frac{1}{\theta} w^H E_j B_{j+1} P_{j+1}^T,$$

$$r = K \tilde{x} - \tilde{\lambda} M \tilde{x} = -\frac{1}{\theta} (K - \sigma M) Q_{j+1} C_{j+1} E_j^T z.$$

The matrix-vector products $Y = [(K - \sigma M)^{-1}M]X$ and $Z^T = X^T[(K - \sigma M)^{-1}M]$ required in the inner loop of the algorithm can be performed by first computing the LU factorization of $K - \sigma M = LU$ and then solving the linear systems of equations $LUY = MX$ and $Z^T = X^T(LU)^{-1}M$ for Y and Z^T , respectively.

If K and M are real, and the shift σ is complex, one can still keep the Lanczos procedure in real arithmetic using a strategy proposed by Parlett and Saad [37].

In many applications, M is symmetric positive definite. In this case, one can avoid factoring M explicitly by preserving M -biorthogonality among the Lanczos vectors [16, 35, 11, 22]. Numerical methods for the case in which M is symmetric indefinite are discussed in [22, 32].

7. Summary of numerical examples. This section summarizes our numerical experience with the ABLE method. We have selected test eigenvalue problems from real applications to demonstrate the major features of the ABLE method. Each numerical example illustrates a property of the ABLE method. All test matrices presented here can be found in the test matrix collection for non-Hermitian eigenvalue problems [4].

The ABLE method has been implemented in Matlab 4.2 with sparse matrix computation functions. All numerical experiments are performed on a SUN Sparc 10 workstation with IEEE double precision floating point arithmetic. The tolerance value τ_c for the stopping criterion (2.13) is set to be 10^{-8} . The clustering threshold (3.2) is $\eta = 10^{-6}$. The breakdown threshold is $\tau_b = 10^{-8}$.

Example 1. The block algorithm accelerates convergence in the presence of multiple and clustered eigenvalues. When the desired eigenvalues are known in advance to be multiple or clustered, we should initially choose the blocksize as the expected multiplicity or the cluster order. For example, the largest eigenvalue of the 656×656 Chuck matrix has multiplicity 2. If we use the unblocked ABLE method, then at iteration 20 the converged Rayleigh–Ritz values,

$$\begin{aligned} &5.502378378875370e + 00, \\ &1.593971696766128e + 00, \end{aligned}$$

approximate the two largest *distinct* eigenvalues. But the multiplicity is not yet revealed. However, if we use the ABLE method with initial blocksize 2, then at iteration 7 the converged Rayleigh–Ritz values are

$$\begin{aligned} &5.502378378347202e + 00, \\ &5.502378378869873e + 00. \end{aligned}$$

Each computed Rayleigh–Ritz value agrees to 10 to 12 decimal digits compared with the one computed by the dense QR algorithm.

Example 2. Full biorthogonality is very expensive to maintain in terms of floating point operations and memory access. Based on our experience, maintaining semi-biorthogonality is a reliable and much less expensive alternative. Our example is a 2500×2500 block tridiagonal coefficient matrix obtained by discretizing the two-dimensional model convection-diffusion differential equation

$$\begin{aligned} -\Delta u + 2p_1 u_x + 2p_2 u_y - p_3 u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

using finite differences, where Ω is the unit square $\{(x, y) \in \mathbf{R}^2, 0 \leq x, y \leq 1\}$. The eigenvalues of the coefficient matrix can be expressed analytically in terms of the parameters p_1 , p_2 , and p_3 . In our test run, we choose $p_1 = 0.5$, $p_2 = 2$, and $p_3 = 1$. For this set of parameters, all eigenvalues of the resulting matrix A are positive real and distinct. With full biorthogonality, at iteration 132, the two largest

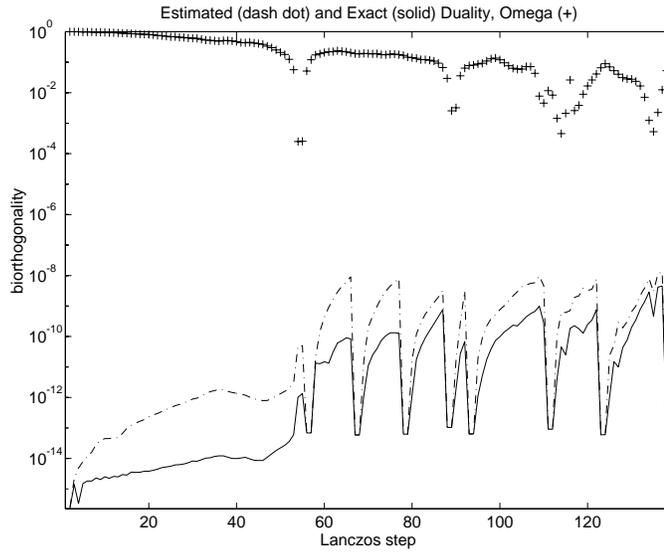


FIG. 7.1. The exact (solid line) and estimated (dash-dot line) biorthogonality of the Lanczos vectors and the smallest singular values (+) of $P_{j+1}^T Q_{j+1}$.

eigenvalues are converged. If we use the ABLE method with semibiorthogonality, at iteration 139, the same two largest eigenvalues are converged to the same accuracy. The difference is that only 8 corrections of biorthogonality loss are invoked to maintain semibiorthogonality, compared to 132 corrections for full biorthogonality.

In Figure 7.1 the solid and dotted lines display the exact and estimated biorthogonality of the computed Lanczos vectors, and the “+”-points concern breakdown and are the smallest singular values of $P_j^T Q_j$. The solid line plots

$$d_{j+1} = \max \left(\frac{\|P_j^T Q_{j+1}\|_1}{\|P_j\|_1 \|Q_{j+1}\|_1}, \frac{\|Q_j^T P_{j+1}\|_1}{\|Q_j\|_1 \|P_{j+1}\|_1} \right)$$

for $j = 1, 2, \dots, 132$. Each sharp decrease corresponds to a correction. The dotted line plots the estimate, \hat{d}_{j+1} , of this quantity computed by the monitoring algorithm of section 4.2. Correction iterations are taken when the dotted line increases to the threshold $\sqrt{\epsilon}$, where ϵ denotes the machine precision. The observation that the solid line is below the dotted line indicates that the monitoring algorithm is prudent. A near breakdown occurs if the smallest singular value of $P_{j+1}^T Q_{j+1}$ is less than the breakdown threshold, but this is not the case in this example.

Example 3. As mentioned before, when we know the multiplicity of the eigenvalues in advance, we should choose the appropriate blocksize, otherwise the adaptive scheme presented in section 3 can dynamically adjust the blocksize to accelerate convergence. This smooths the convergence behavior to clusters of eigenvalues. For example, we apply the ABLE method with initial blocksize 1 to the 656×656 Chuck matrix. At iteration 24, the double eigenvalue is detected and the blocksize is doubled.

Example 4. Exact breakdowns are rare but near breakdowns are not. In general, we can successfully cure the near breakdowns. For example, when the ABLE method is applied to the 882×882 Quebec Hydro matrix from the application of numerical

of the spectrum but an internal branch, known as the Alfvén spectrum. We need to use a spectral transformation technique to transfer the interesting spectrum to the outer part. In section 6, we have outlined a general approach. Now, we show the numerical results of this general approach for the MHD test problem. Both K and M are 416 by 416 block tridiagonal matrices with 16 by 16 blocks. To two significant digits, there holds

$$\|K\|_1 = 3100 \quad \text{and} \quad \|M\|_1 = 2.50,$$

but the estimated condition number of M is 5.05×10^9 ; M is quite ill conditioned. The computational task is to calculate the eigenvalues close to the shift $\sigma = -0.3 + 0.65i$ [8].

We ran the unblocked spectral transformation ABLE method. After only 30 iterations, 10 Rayleigh–Ritz values are converged; their accuracy ranges from 10^{-8} to 10^{-12} , compared with the eigenvalues computed by the QZ algorithm. The following table lists the 10 converged Rayleigh–Ritz values θ_i and the corresponding left and right residual norms, where

$$\text{Res-L}_i = \frac{\|y_i^H K - \theta_i y_i^H M\|_2}{\max(\|K\|_1, \|M\|_1)}, \quad \text{Res-R}_i = \frac{\|Kx_i - \theta_i Mx_i\|_2}{\max(\|K\|_1, \|M\|_1)},$$

and (y_i^H, x_i) are the normalized approximate left and right eigenvectors of (K, M) (i.e., $\|y_i^H\|_2 = \|x_i\|_2 = 1$):

i	θ_i	Res-L _{i}	Res-R _{i}
1	$-2.940037576164888e - 01 + 5.871546479737660e - 01i$	$3.82e - 12$	$6.59e - 11$
2	$-2.381814888866186e - 01 + 5.914958688660595e - 01i$	$2.66e - 11$	$4.46e - 11$
3	$-3.465530921874517e - 01 + 5.468970786348115e - 01i$	$1.23e - 11$	$2.76e - 10$
4	$-3.780991425908282e - 01 + 5.071655448857557e - 01i$	$6.18e - 11$	$3.98e - 10$
5	$-2.410301845692590e - 01 + 5.238090347100917e - 01i$	$9.81e - 11$	$4.32e - 10$
6	$-1.989292783177918e - 01 + 5.900118523050361e - 01i$	$5.34e - 11$	$8.55e - 11$
7	$-2.045328538082208e - 01 + 5.678048139549289e - 01i$	$5.97e - 11$	$1.12e - 10$
8	$-3.092857309948118e - 01 + 4.687528684165645e - 01i$	$5.23e - 09$	$2.59e - 08$
9	$-1.749780170739634e - 01 + 5.920044440850396e - 01i$	$5.62e - 10$	$9.58e - 10$
10	$-1.573456542107287e - 01 + 5.976613227972810e - 01i$	$5.98e - 09$	$9.63e - 09$

In addition, six other Rayleigh–Ritz values range in accuracy from 10^{-5} to 10^{-7} . Figure 7.3 shows Alfvén spectrum computed by the QZ algorithm (+) and the Rayleigh–Ritz values (o) computed by the spectral transformation ABLE method.

Three corrections to maintain semibiorthogonality were taken at iterations 13, 20, and 26. The convergence history of the Rayleigh–Ritz values are shown in the following table, where j is the Lanczos iteration and k is the number of converged Rayleigh–Ritz values at the j th iteration:

j	≤ 14	15–18	19	20–22	23–24	25–26	27–28	29	30
k	0	1	2	3	4	7	8	9	10

Moreover, at Lanczos iteration 45, the entire Alfvén branch of spectra of the MHD test problem are revealed: 20 Rayleigh–Ritz values converged, and 12 other Rayleigh–Ritz values range in accuracy from 10^{-7} up to 10^{-5} . No copies of eigenvalues are observed.

Acknowledgments. The authors would like to thank the referees for their valuable comments on the manuscript.

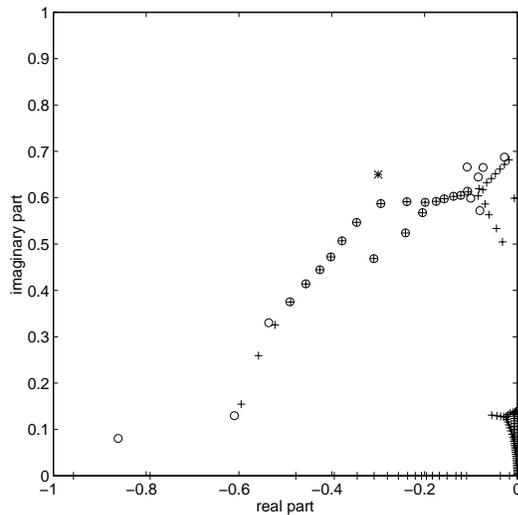


FIG. 7.3. *The Alfvén spectra of the MHD test problem.* “+” denotes the eigenvalues computed by the QZ algorithm. “o” are the Rayleigh–Ritz values computed by 30 steps of the spectral transformation ABLE method. “*” is the shift $\sigma = -0.3 + 0.65i$.

REFERENCES

- [1] W. E. ARNOLDI, *The principle of minimized iteration in the solution of the matrix eigenproblem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
- [2] Z. BAI, *Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem*, Math. Comp., 62 (1994), pp. 209–226.
- [3] Z. BAI, *A spectral transformation block nonsymmetric Lanczos algorithm for solving sparse non-Hermitian eigenproblems*, in Proc. Fifth SIAM Conference on Applied Linear Algebra, J. G. Lewis, ed., SIAM, Philadelphia, PA, 1994, pp. 307–311.
- [4] Z. BAI, D. DAY, D. DEMMEL, AND J. DONGARRA, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, available online from <http://math.nist.gov/MatrixMarket>.
- [5] Z. BAI AND G. W. STEWART, *SRRIT: A Fortran subroutine to calculate the dominant invariant subspace of a nonsymmetric matrix*, ACM Trans. Math. Software, 23 (1997), pp. 494–513.
- [6] D. BOLEY, S. ELHAY, G. H. GOLUB, AND M. H. GUTKNECHT, *Nonsymmetric Lanczos and Finding Orthogonal Polynomials Associated with Indefinite Weights*, Numerical Analysis report NA-90-09, Stanford University, Palo Alto, CA, 1990.
- [7] D. BOLEY AND G. GOLUB, *The nonsymmetric Lanczos algorithm and controllability*, Systems Control Lett., 16 (1991), pp. 97–105.
- [8] J. G. L. BOOTEN, H. A. VAN DER VORST, P. M. MEIJER, AND H. J. J. TE RIELE, *A Preconditioned Jacobi-Davidson Method for Solving Large Generalized Eigenvalue Problems*, Technical report NM-R9414, Dept. of Numerical Math, CWI, Amsterdam, the Netherlands, 1994.
- [9] F. CHATELIN, *Eigenvalues of Matrices*, John Wiley, Chichester, England, 1993.
- [10] J. CULLUM AND W. E. DONATH, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large sparse real symmetric matrices*, in Proc. 1974 IEEE Conference on Decision and Control, Phoenix, AZ, 1974, pp. 505–509.
- [11] J. CULLUM, W. KERNER, AND R. WILLOUGHBY, *A generalized nonsymmetric Lanczos procedure*, Comput. Phys. Comm., 53 (1989), pp. 19–48.
- [12] J. CULLUM AND R. WILLOUGHBY, *A practical procedure for computing eigenvalues of large sparse nonsymmetric matrices*, in Large Scale Eigenvalue Problems, J. Cullum and R. Wiloughby, eds., North-Holland, Amsterdam, 1986.
- [13] E. R. DAVIDSON, *The iteration calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, Comput. Phys., 17 (1975), pp. 87–94.

- [14] D. DAY, *Semi-Duality in the Two-Sided Lanczos Algorithm*, Ph.D. thesis, University of California, Berkeley, CA, 1993.
- [15] I. DUFF AND J. SCOTT, *Computing selected eigenvalues of sparse nonsymmetric matrices using subspace iteration*, ACM Trans. Math. Software, 19 (1993), pp. 137–159.
- [16] T. ERICSSON AND A. RUHE, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problem*, Math. Comp., 35 (1980), pp. 1251–1268.
- [17] R. W. FREUND, M. H. GUTKNECHT, AND N. M. NACHTIGAL, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.
- [18] R. W. FREUND, N. M. NACHTIGAL, AND J. C. REEB, *QMRPACK User's Guide*, Technical report ORNL/TM-12807, Oak Ridge National Laboratory, Oak Ridge, TN, 1994.
- [19] G. GOLUB AND R. UNDERWOOD, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. Rice, ed., Academic Press, New York, 1977, pp. 364–377.
- [20] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.
- [21] W. B. GRAGG, *Matrix interpretations and applications of the continued fraction algorithm*, Rocky Mountain J. Math., 5 (1974), pp. 213–225.
- [22] R. GRIMES, J. LEWIS, AND H. SIMON, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.
- [23] M. H. GUTKNECHT, *A completed theory of the unsymmetric Lanczos process and related algorithms, Parts I and II*, SIAM J. Matrix Anal. Appl., Part I, 13 (1992), pp. 594–639, Part II, 15 (1994), pp. 15–58.
- [24] Z. JIA, *Generalized block Lanczos methods for large unsymmetric eigenproblems*, Numer. Math., 80 (1998), pp. 239–266.
- [25] W. KAHAN, B. N. PARLETT, AND E. JIANG, *Residual bounds on approximate eigensystems of nonnormal matrices*, SIAM J. Numer. Anal., 19 (1982), pp. 470–484.
- [26] T. KATO, *Perturbation Theory for Linear Operators*, 2nd ed., Springer-Verlag, Berlin, 1980.
- [27] W. KERNER, *Large-scale complex eigenvalue problems*, J. Comput. Phys., 85 (1989), pp. 1–85.
- [28] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natl. Bur. Stand., 45 (1950), pp. 225–280.
- [29] R. LEHOUCQ, *Analysis and Implementation of an Implicitly Restarted Arnoldi Iterations*, Ph.D. thesis, Rice University, Houston, Texas, 1995.
- [30] R. LEHOUCQ AND J. A. SCOTT, *An Evaluation of Software for Computing Eigenvalues of Sparse Nonsymmetric Matrices*, Preprint MCS-P547-1195, Argonne National Laboratory, Argonne, IL, 1996.
- [31] R. LEHOUCQ, D. SORENSEN, AND C. YANG, *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, PA, 1998.
- [32] K. MEERBERGEN AND A. SPENCE, *Implicitly Restarted Arnoldi with Purification for the Shift-Invert Transformation*, report tw 225, Dept. of Comput. Sci., Katholieke Universiteit Leuven, Belgium, 1995.
- [33] C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, London University, London, UK, 1971.
- [34] B. PARLETT, *The Rayleigh quotient algorithm iteration and some generalizations for nonnormal matrices*, Math. Comp., 28 (1974), pp. 679–693.
- [35] B. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [36] B. PARLETT, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 567–593.
- [37] B. PARLETT AND Y. SAAD, *Complex shift and invert strategies for real matrices*, Linear Algebra Appl., 88/89 (1987), pp. 575–595.
- [38] 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.
- [39] A. RUHE, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Math. Comp., 33 (1979), pp. 680–687.
- [40] A. RUHE, *Rational Krylov, a Practical Algorithm for Large Sparse Nonsymmetric Matrix Pencils*, Computer Science Division UCB/CSD-95-871, University of California, Berkeley, CA, 1995.
- [41] Y. SAAD, *On the rates of convergence of the Lanczos and block Lanczos methods*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.
- [42] Y. SAAD, *Variations on Arnoldi's method for computing eigenlements of large unsymmetric*

- matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
- [43] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press (division of John Wiley), New York, 1992.
 - [44] M. SADKANE, *Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems*, Numer. Math., 64 (1993), pp. 195–211.
 - [45] M. SADKANE, *A block Arnoldi-Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numer. Math., 64 (1993), pp. 181–193.
 - [46] H. SIMON, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–142.
 - [47] G. L. G. SLEIJPEN AND H. A. VAN DER VORST, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.
 - [48] D. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
 - [49] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
 - [50] W. J. STEWART AND A. JENNINGS, *Algorithm 570 LOPSI: A simultaneous iteration algorithm for real matrices*, ACM Trans. Math. Software, 7 (1981), pp. 230–232.
 - [51] L. N. TREFETHEN, *Pseudospectra of matrices*, in Numerical Analysis 1991, Dundee, Scotland, Longman Sci. Tech., Harlow, 1992.
 - [52] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.
 - [53] Q. YE, *A breakdown-free variation of the nonsymmetric Lanczos algorithms*, Math. Comp., 62 (1994), pp. 179–207.