SIAM J. MATRIX ANAL. APPL.
Vol. 17, No. 4, pp. 789–821, October 1996

006

DEFLATION TECHNIQUES FOR AN IMPLICITLY RESTARTED ARNOLDI ITERATION*

R. B. LEHOUCQ† AND D. C. SORENSEN‡

Abstract. A deflation procedure is introduced that is designed to improve the convergence of an implicitly restarted Arnoldi iteration for computing a few eigenvalues of a large matrix. As the iteration progresses, the Ritz value approximations of the eigenvalues converge at different rates. A numerically stable scheme is introduced that implicitly deflates the converged approximations from the iteration. We present two forms of implicit deflation. The first, a *locking* operation, decouples converged Ritz values and associated vectors from the active part of the iteration. The second, a *purging* operation, removes unwanted but converged Ritz pairs. Convergence of the iteration is improved and a reduction in computational effort is also achieved. The deflation strategies make it possible to compute multiple or clustered eigenvalues with a single vector restart method. A block method is not required. These schemes are analyzed with respect to numerical stability, and computational results are presented.

Key words. Arnoldi method, Lanczos method, eigenvalues, deflation, implicit restarting

AMS subject classifications. 65F15, 65G05

1. Introduction.

The Arnoldi method is an efficient procedure for approximating a subset of the eigensystem of a large sparse $n \times n$ matrix A. The Arnoldi method is a generalization of the Lanczos process and reduces to that method when the matrix A is symmetric. After k steps, the algorithm produces an upper Hessenberg matrix H_k of order k. The eigenvalues of this small matrix H_k are used to approximate a subset of the eigenvalues of the large matrix A. The matrix H_k is an orthogonal projection of A onto a particular *Krylov* subspace, and the eigenvalues of H_k are usually called *Ritz values* or *Ritz approximations*.

There are a number of numerical difficulties with Arnoldi/Lanczos methods. In [34] a variant of this method was developed to overcome these difficulties. This technique, the implicitly restarted Arnoldi iteration (IRA iteration), may be viewed as a truncation of the standard implicitly shifted QR iteration. This connection will be reviewed during the course of the paper. Because of this connection, an IRA iteration shares a number of the QR iteration's desirable properties. These include the well-understood deflation rules of the QR iteration. These deflation techniques are extremely important with respect to the convergence and stability of the QR iteration. Deflation rules have contributed greatly to the emergence of the practical QR algorithm as the method of choice for computing the eigensystem of dense matrices. In particular, the deflation rules allow the QR iteration to compute multiple and clustered eigenvalues.

This paper introduces deflation schemes that may be used within an IRA iteration. This iteration is designed to compute a selected subset of the spectrum of A such as

* Received by the editors February 10, 1995; accepted for publication (in revised form) by A. Greenbaum November 8, 1995. This work was supported in part by ARPA (U.S. Army ORA4466.01), by U.S. Department of Energy contract DE-FG0f-91ER25103, and by National Science Foundation cooperative agreement CCR-9120008.

† Computational and Applied Mathematics Department, Rice University, Houston, TX 77251 (lehoucq@rice.edu). Current address: Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (lehoucq@mcs.anl.gov) (http://www.mcs.anl.gov/home/lehoucq/index.html).

‡ Computational and Applied Mathematics Department, Rice University, Houston, TX 77251 (sorensen@rice.edu).

the k eigenvalues of largest real part. We refer to this selected subset as *wanted* and the remainder of the spectrum as *unwanted*. As the iteration progresses some of the Ritz approximations to eigenvalues of A may converge long before the entire set of wanted eigenvalues has been computed. These converged Ritz values may be part of the wanted or the unwanted portion of the spectrum. In either case, it is desirable to *deflate* the converged Ritz values and corresponding Ritz vectors from the unconverged portion of the factorization. If the converged Ritz value is wanted, it is necessary to keep it in the subsequent Arnoldi factorizations. This is called *locking*. If the converged Ritz value is unwanted then it must be removed from the current and subsequent Arnoldi factorizations. This is called *purging*. These notions will be made precise during the course of the paper. For the moment we note that the advantages of a numerically stable deflation strategy include

- reduction of the *working* size of the desired invariant subspace,
- prevention of the effects of the forward instability of the Lanczos and QR algorithms [27, 39],
- the ability to determine clusters of nearby eigenvalues without need for a block Arnoldi method [18, 32, 33].

The fundamentals of the Arnoldi algorithm are introduced in §2 as well as the determination of Ritz value convergence. The IRA iteration is reviewed in §3. Deflating within the IRA iteration is examined in §4. The deflation scheme for converged Ritz values is presented in §5. The practical issues associated with our deflation scheme are examined in §6. These include block generalizations of the ideas examined in §5 for dealing with a number of Ritz values simultaneously and avoiding the use of complex arithmetic when a complex conjugate pair of Ritz values converges. An error analysis of the deflated process is presented in §7. A brief survey of and comparisons with other deflation strategies is given in §8. An interesting connection with the various algorithms used to reorder a Schur form of matrix is presented in §9. Numerical results are presented in §10.

Capital and lowercase letters denote matrices and vectors whereas lowercase Greek letters denote scalars. The jth canonical basis vector is denoted by e_j. The norms used are the Euclidean and Frobenius ones denoted by $\|\cdot\|$ and $\|\cdot\|_F$, respectively. The range of a matrix A is denoted by $\mathcal{R}(A)$.

2. The Arnoldi factorization.

Arnoldi's method [1] is an orthogonal projection method for approximating a subset of the eigensystem of a general square matrix. The method builds, step by step, an orthogonal basis for the *Krylov* space

$$\mathcal{K}_k(A, v_1) \equiv \operatorname{span}\{v_1, Av_1, \ldots, A^{k-1}v_1\}$$

for A generated by the vector v_1. The original algorithm in [1] was designed to reduce a dense matrix to upper Hessenberg form. However, the method only requires knowledge of A through matrix–vector products, and its ultimate value as a technique for approximating a few eigenvalues of a large sparse matrix was soon realized. When the matrix A is symmetric, the procedure reduces to the Lanczos method [22].

Over a decade of research was devoted to understanding and overcoming the numerical difficulties of the Lanczos method [26]. Development of the Arnoldi method lagged behind due to the inordinate computational and storage requirements associated with the original method when a large number of steps are required for convergence. Not only is more storage required for V_k and H_k when A is nonsymmetric, but in general more steps are required to compute the desired Ritz value approximations. An explicitly restarted Arnoldi iteration (ERA iteration) was introduced by Saad [30] to overcome these difficulties. The idea is based on similar ones developed for the Lanczos process by Paige [25], Cullum and Donath [10], and Golub and Underwood [17]. Karush proposed the first example of a restarted iteration in [21].

After k steps, the Arnoldi algorithm computes a truncated factorization,

$$AV_k = V_k H_k + f_k e_k^T, \tag{2.1}$$

of $A \in \mathbf{R}^{n \times n}$ into upper Hessenberg form where $V_k^T V_k = I_k$. The vector f_k is the residual and is orthogonal to the columns of V_k. The matrix $H_k \in \mathbf{R}^{k \times k}$ is an upper Hessenberg matrix that is the orthogonal projection of A onto $\mathcal{R}(V_k) \equiv \mathcal{K}_k(A, v_1)$.

The following procedure shows how the factorization is extended from length k to $k+p$.

Algorithm 2.1.

```
function $[V_{k+p}, H_{k+p}, f_{k+p}]$ = Arnoldi $(A, V_k, H_k, f_k, k, p)$
    Input: $AV_k - V_k H_k = f_k e_k^T$ with $V_k^T V_k = I_k$, & $V_k^T f_k = 0$.
    Output: $AV_{k+p} - V_{k+p} H_{k+p} = f_{k+p} e_{k+p}^T$ with $V_{k+p}^T V_{k+p} = I_{k+p}$, & $V_{k+p}^T f_{k+p} = 0$.
  1. For $j = 1, 2 \dots p$
        2. $\beta_{k+j} \leftarrow \|f_{k+j-1}\|$;   if $\beta_{k+j} = 0$ then stop;
        3. $v_{k+j} \leftarrow f_{k+j-1}\beta_{k+j}^{-1}$; $V_{k+j} \leftarrow \begin{bmatrix} V_{k+j-1} & v_{k+j} \end{bmatrix}$;
        4. $w \leftarrow Av_{k+j}$;
        5. $h_{k+j} \leftarrow V_{k+j-1}^T w$; $\alpha_{k+j} \leftarrow v_{k+j}^T w$;
        6. $H_{k+j} \leftarrow \begin{bmatrix} H_{k+j-1} & h_{k+j} \\ \beta_{k+j} e_{k+j-1}^T & \alpha_{k+j} \end{bmatrix}$
        7. $f_{k+j} \leftarrow w - V_{k+j-1} h_{k+j} - v_{k+j}\alpha_{k+j}$;
```

If $k = 0$ then $V_1 = v_1$ represents the initial vector. In order to ensure that $V_k^T f_k \approx 0$ in finite precision arithmetic, the above algorithm requires some form of reorthogonalization at step 7; see Chapter 7 of [23].

In exact arithmetic, the algorithm continues until $f_k = 0$ for some $k \leq n$. All of the intermediate Hessenberg matrices H_j are *unreduced* for $j \leq k$. A Hessenberg matrix is said to be unreduced if all of its main subdiagonal elements are nonzero. The residual vanishes at the first step k such that $\dim \mathcal{K}_{k+1}(A, v_1) = k$ and hence is guaranteed to vanish for some $k \leq n$. The following result indicates when an exact truncated factorization occurs. This is desirable since the columns of V_k form a basis for an invariant subspace and the eigenvalues of H_k are a subset of those of A.

Theorem 2.2. *Let equation* (2.1) *define a k step Arnoldi factorization of A, with H_k unreduced. Then $f_k = 0$ if and only if $v_1 = Q_k y$ where $AQ_k = Q_k R_k$ with $Q_k^T Q_k = I_k$, and R_k is an upper quasi-triangular matrix of order k.*

Proof. See Chapter 2 of [23] or [34] for a proof based on the Jordan canonical form. □

In Theorem 2.2, the span of the k columns of Q_k represents an invariant subspace for A. The matrix equation $AQ_k = Q_k R_k$ is a partial real Schur decomposition of order k for A. The diagonal blocks of R_k contain the eigenvalues of A. The complex conjugate pairs are in blocks of order-2 and the real eigenvalues are on the diagonal of R_k, respectively. In particular, the theorem gives that if the initial vector is a linear combination of k linearly independent eigenvectors then the kth residual vector vanishes. It is therefore desirable to devise a method that forces the starting vector v_1 to lie in the invariant subspace associated with the wanted eigenvalues.

The algorithms of this paper are appropriate when the order of A is so large that storage and computational requirements prohibit completion of the algorithm that produces V_n and H_n. We also remark that working in finite precision arithmetic generally removes the possibility of the computed residual ever vanishing exactly.

As the norm of f_k decreases, the eigenvalues of H_k become better approximations to those of A. Experience indicates that $\|f_k\|$ rarely becomes small, let alone zero. However, as the order of H_k increases, certain eigenvalues of H_k may emerge as excellent estimates to eigenvalues of A. When an eigenvalue H_k is sufficiently near one of A, we will say that convergence occurred. Since the interest is in a small subset of the eigensystem of A, alternate criteria that allow termination for $k \ll n$ are needed. Let $H_k y = y\theta$ where $\|y\| = 1$. Define the vector $x = V_k y$ to be a *Ritz vector* and θ to be *Ritz value*. Then

$$\begin{aligned} \|AV_k y - V_k H_k y\| &= \|Ax - x\theta\| \\ &= \|f_k\| \, |e_k^T y| \end{aligned} \tag{2.2}$$

indicates that if the last component of an eigenvector for H_k is small the Ritz pair (x, θ) is an approximation to an eigenpair of A. This pair is exact for a nearby problem: it is easily shown that $(A+E)x = x\theta$ with $E = -(e_k^T y) f_k x^H$. The advantage of using the *Ritz estimate* (2.2) is to avoid explicit formation of the quantity $AV_k y - V_k y\theta$ when accessing the numerical accuracy of an approximate eigenpair. Recent work by Chatelin [8], Chatelin and Fraysée [9], and Godet-Thobie [14] suggests that when A is highly non-normal, the size of $e_k^T y$ is not an appropriate guide for detecting convergence. If the relative *departure from normality* defined by the Henrici number $\|AA^T - A^T A\|_F / \|A^2\|_F$ is large, the matrix A is considered highly non-normal. Assuming that A is diagonalizable, a large Henrici number implies that the basis of eigenvectors is ill conditioned [8]. Bennani and Braconnier compare the use of the Ritz estimate and direct residual $\|Ax - x\theta\|$ in Arnoldi algorithms [4]. They suggest normalizing the Ritz estimate by the norm of A, resulting in a stopping criteria based on the *backward* error. The backward error is defined as the smallest, in norm, perturbation ΔA such that the Ritz pair is an eigenpair for $A + \Delta A$. Scott [33] presents a lucid account of the many issues involved in determining stopping criteria for the unsymmetric problem.

3. The implicitly restarted Arnoldi iteration.

Theorem 2.2 motivates the selection of a starting vector that will lead to the construction of an approximate basis for the desired invariant subspace of A. The best possible starting vector would be a linear combination of a Schur basis for the desired invariant subspace. The IRA iteration iteratively restarts the Arnoldi factorization with the goal of forcing the starting vector closer and closer to the desired invariant subspace. The scheme is called *implicit* because the updating of the starting vector is accomplished with an implicitly shifted QR mechanism on H_k. This will allow us to update the starting vector by working with orthogonal matrices that live in $\mathbf{R}^{k \times k}$ rather than in $\mathbf{R}^{n \times n}$.

The iteration starts by extending a length k Arnoldi factorization by p steps. Next, p shifted QR steps are performed on H_{k+p}. The last p columns of the factorization are discarded resulting in a length k factorization. The iteration is defined by repeating the above process until convergence.

As an example, suppose that $p = 1$ and that k represents the dimension of the desired invariant subspace. Let μ be a real shift and let $H_{k+1} - \mu I = QR$ with Q orthogonal and R upper triangular matrices, respectively. Then from (2.1)

$$\begin{aligned} (A - \mu I)V_{k+1} - V_{k+1}(H_{k+1} - \mu I) &= f_{k+1} e_{k+1}^T, \\ (A - \mu I)V_{k+1} - V_{k+1} QR &= f_{k+1} e_{k+1}^T, \end{aligned} \tag{3.1}$$

$$\begin{aligned} (A-\mu I)(V_{k+1}Q)-(V_{k+1}Q)(RQ) &= f_{k+1}e_{k+1}^TQ, \\ A(V_{k+1}Q)-(V_{k+1}Q)(RQ+\mu I) &= f_{k+1}e_{k+1}^TQ. \end{aligned} \tag{3.2}$$

The matrices are updated via $V_{k+1}^+ \leftarrow V_{k+1}Q$ and $H_{k+1}^+ \leftarrow RQ+\mu I$ and the latter matrix remains upper Hessenberg since R is upper triangular and Q is upper Hessenberg. However, equation (3.2) is not quite a legitimate Arnoldi factorization. The relation of equation (3.2) fails to be an Arnoldi factorization since the matrix $f_{k+1}e_{k+1}^TQ$ has a nonzero kth column. Partitioning the matrices in the updated equation results in

$$\begin{aligned} A\begin{bmatrix} V_k^+ & v_{k+1}^+ \end{bmatrix} &= \begin{bmatrix} V_k^+ & v_{k+1}^+ \end{bmatrix}\begin{bmatrix} H_k^+ & h_{k+1}^+ \\ \beta_{k+1}^+e_k^T & \alpha_{k+1}^+ \end{bmatrix} \\ &\quad + f_{k+1}\begin{bmatrix} \sigma_k e_k^T & \gamma_k \end{bmatrix}, \end{aligned} \tag{3.3}$$

where $\sigma_k = e_{k+1}^TQe_k$ and $\gamma_k = e_{k+1}^TQe_{k+1}$. Equating the first k columns of (3.3) gives

$$AV_k^+ = V_k^+H_k^+ + (\beta_{k+1}^+v_{k+1}^+ + \sigma_k f_{k+1})e_k^T. \tag{3.4}$$

Performing the update $f_k^+ \leftarrow \beta_{k+1}^+v_{k+1}^+ + \sigma_k f_{k+1}$ and noting that $(V_k^+)^Tf_k^+ = 0$, it follows that equation (3.4) is a length k Arnoldi factorization.

We now show that the IRA iteration is equivalent to forming the leading portion of an implicitly shifted QR iteration. Note that equations (3.1) and (3.2) are valid for $1 \le k \le n$. In particular, extending the factorization of equation (3.1) by $n-k$ steps gives $f_n = 0$, and $AV_n - V_nH_n = 0$ defines a decomposition of A into upper Hessenberg form. Let $Q_nR_n = H_n - \mu I$ where Q_n and R_n are orthogonal and upper triangular matrices of order n, respectively. Since Q and R are the leading principal submatrices of order $k+1$ for Q_n and R_n, respectively, $V_nQ_nR_ne_1 = V_{k+1}QRe_1$ and $e_1^TR_ne_1 = e_1^TRe_1$ follow. Postmultiplication of equation (3.2) with e_1 exposes the relationship

$$(A-\mu I)v_1 = V_{k+1}Qe_1\rho_{11} = V_nQ_ne_1\rho_{11} = v_1^+,$$

where $\rho_{11} = e_1^TRe_1$, $v_1 = V_{k+1}e_1$, and $V_{k+1}^+e_1 = v_1^+$. In other words, the first column of the updated k step factorization matrix is the *same* as the first column of the orthogonal matrix obtained after a complete QR step on A with shift μ. Thus, the IRA iteration may be viewed as a truncated version of the standard implicitly shifted QR iteration. This idea may be extended for up to $p > 1$ shifts [34]. One cycle of the iteration is pictured in Figures 3.1–3.3. Application of the shifts may be performed implicitly as in the QR algorithm. If the shifts are in complex conjugate pairs then the implicit double shift can be used to avoid complex arithmetic.

Numerous choices are possible for the selection of the p shifts. One immediate choice is to use the p unwanted eigenvalues of H_{k+p}. In exact arithmetic, the last p subdiagonal elements of H_{k+p} are zero and the Arnoldi factorization decouples. For example, in equation (3.4), $\beta_{k+1}^+ = 0$ when μ is an eigenvalue of H_k. The reader is referred to [7, 23, 34] for further information.

The number of shifts to apply at each cycle of the above iteration is problem dependent. At present there is no a priori analysis to guide the selection of p relative to k. The only formal requirement is that $1 \le p \le n-k$. However, computational experience suggests that $p \ge k$ is preferable. If many problems of the same type are to be solved, experimentation with p for a fixed k should be undertaken. This usually decreases the required number of matrix–vector operations but increases the work and

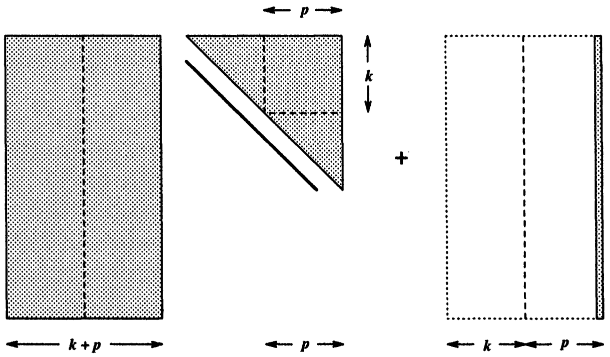

FIG. 3.1. *The set of rectangles represents the matrix equation $V_{k+p}H_{k+p} + f_{k+p}e_{k+p}^T$ of an Arnoldi factorization. The unshaded region on the right is a zero matrix of $k + p - 1$ columns.*

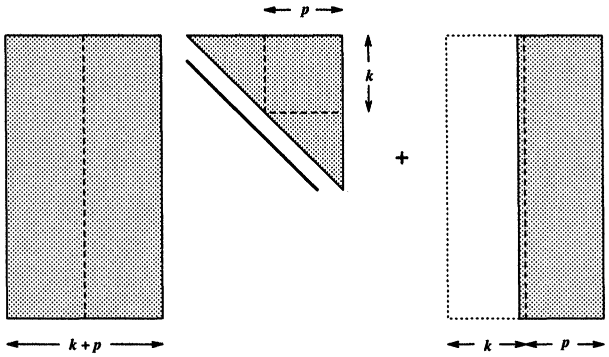

FIG. 3.2. *After performing p implicitly shifted QR steps on H_{k+p}, the middle set of pictures illustrates $V_{k+p}QQ^T H_{k+p}Q + f_{k+p}e_{k+p}^T Q$. The last $p+1$ columns of $f_{k+p}e_{k+p}^T Q$ are nonzero because of the QR iteration.*

storage required to maintain the orthogonal basis vectors. The optimal *crossover* with respect to CPU time varies and must be determined empirically. Lehoucq makes a connection with subspace iteration, in Chapter 8 of [23]. There has been considerable experience with subspace iteration, and this connection may eventually shed light on how to select p relative to k. For example, it is well known that performing subspace iteration on a subspace of dimension larger than the number of eigenvalues required typically leads to improved convergence rates; see the paper of Duff and Scott [12] for a discussion and further references.

Among the several advantages an implicit updating scheme possesses are

- fixed storage requirements,
- the ability to maintain a prescribed level of orthogonality for the columns of V since k is of modest size,
- application of the matrix polynomial $v_1^+ \leftarrow \Psi(A)v_1$ without needing to apply matrix–vector products with A,
- the incorporation of the well-understood numerical and theoretical behavior of the QR algorithm.

These last two points warrant further discussion. Quite often, the dominant cost during Arnoldi iterations is the matrix–vector products with A. Thus, the IRA iteration

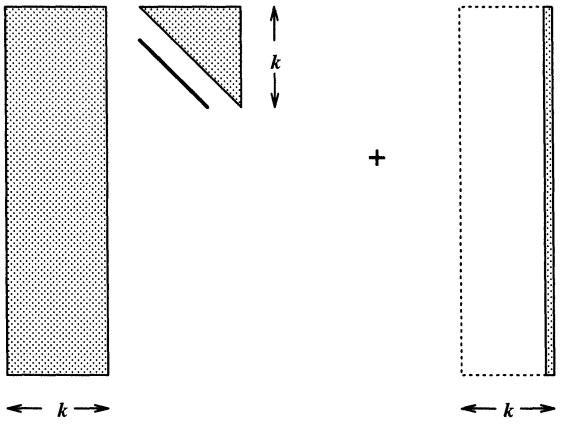

FIG. 3.3. *After discarding the last p columns, the final set represents $V_kH_k + f_ke_k^T$ of a length k Arnoldi factorization.*

may result in a substantial reduction in time when building a length $k+p$ Arnoldi factorization. The last point is important since it allows the possibility of constructing general purpose and reliable software for the large-scale eigenvalue problem.

4. Deflation within an IRA iteration.

As the iteration progresses, the Ritz estimates (2.2) decrease at different rates. When a Ritz estimate is small enough, the corresponding Ritz value is said to have converged. The converged Ritz value may be wanted or unwanted. In either case, a mechanism to deflate the converged Ritz value from the current factorization is desired. Depending on whether the converged Ritz value is wanted or not, it is useful to define two types of deflation. Before we do this, it will prove helpful to illustrate how deflation is achieved. Suppose that after m steps of the Arnoldi algorithm we have

$$A\begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} H_1 & G \\ \epsilon e_1 e_j^T & H_2 \end{bmatrix} + fe_m^T, \tag{4.1}$$

where $V_1 \in \mathbf{R}^{n\times j}$, $H_1 \in \mathbf{R}^{j\times j}$ for $1 \le j < m$. If ϵ is suitably small then the factorization *decouples* in the sense that a Ritz pair (y, θ) for H_1 provides an approximate eigenpair $(x = V_1 y, \theta)$ with a Ritz estimate of $|\epsilon e_j^T y|$. Setting ϵ to zero splits a nearby problem exactly and setting $\epsilon = 0$ is called *deflation*. If ϵ is suitably small, then all the eigenvalues of H_1 may be regarded as converged Ritz values.

4.1. Locking.

If deflation has taken place, the column vectors in V_1 are considered *locked*. This means that subsequent implicit restarting is done on the basis V_2. The submatrices affected during implicit restarting are G, H_2, and V_2. However, during the phase of the iteration that extends the Arnoldi factorization from k to $k+p$ steps, all of the columns of $\begin{bmatrix} V_1 & V_2 \end{bmatrix}$ participate just as if no deflation had occurred. This ensures that all of the new Arnoldi basis vectors are orthogonalized against converged Ritz vectors and prevents the introduction of spurious eigenvalues.

After deflation, equating the last $m-j$ columns of (4.1) results in $(I - V_1V_1^T)AV_2 = V_2H_2 + fe_{m-j}^T$. Thus, deflating V_1 and H_1 from the factorization defines a new Arnoldi factorization with the matrix $(I - V_1V_1^T)A$ and starting vector V_2e_1. This equivalence was noted by Saad [31, p. 182]. Moreover, this provides a means to safely compute multiple eigenvalues when they are present. A block method is not required if deflation and locking are used. The concept of locking was introduced by Stewart and Jennings [37] as a deflation technique for simultaneous iteration.

4.2. Purging. If deflation has occurred but some of the deflated Ritz values are unwanted then a further mechanism, purging, must be introduced to remove the unwanted Ritz values and corresponding vectors from the factorization. The basic idea of purging is perhaps best explained with the case of a single deflated Ritz value.

Let $j = 1$ in (4.1) and equate the first columns of both sides to obtain

$$Av_1 = v_1\alpha_1 + \epsilon V_2 e_1, \tag{4.2}$$

where $v_1 = V_1 e_1$ and $H_1 = \alpha_1$. Equation (4.2) is an Arnoldi factorization of length one. The Ritz value α_1 has Ritz estimate $|\epsilon|$.

Equating the last $m-1$ columns of (4.1) results in

$$AV_2 = V_1 G + V_2 H_2 + f e_{m-1}^T. \tag{4.3}$$

Suppose that α_1 represents an unwanted Ritz value. If A were symmetric then $G = \epsilon e_1^T$ and equation (4.3) would become

$$(A+E)V_2 = V_2 H_2 + f e_{m-1}^T,$$

where $E = -\epsilon v_1 (V_2 e_1)^T - \epsilon (V_2 e_1) v_1^T$. Since $\|E\| = \epsilon$ equation (4.3) defines a length $m-1$ Arnoldi factorization for a nearby problem. The unwanted Ritz pair (v_1, α_1) may be *purged* from the factorization simply by taking $V = V_2$ and $H = H_2$ and setting $G = 0$ in (4.3). If A is not symmetric, the $1 \times (m-1)$ matrix G couples v_1 to the rest of the basis vectors V_2. This vector may be decoupled using the standard Sylvester equation approach [15, pp. 386–387]. Purging then takes place as in the symmetric case. However, the new set of basis vectors must be reorthogonalized in order to return to an Arnoldi factorization. This procedure is developed in §§5 and 6 including the case of purging several vectors.

4.3. Complications. An immediate question follows: do any subdiagonal elements in the Hessenberg matrix of the factorization (4.1) become negligible as an IRA iteration progresses? Since a cycle of the Arnoldi iteration involves performing a sequence of QR steps, the question is answered by considering the behavior of the QR iteration upon upper Hessenberg matrices. In exact arithmetic, under the assumption that the Hessenberg matrix is unreduced, only the last subdiagonal element may become zero when shifting. But the other subdiagonal elements may become arbitrarily small.

In addition, in exact arithmetic, the purging technique would not be necessary as the implicit shift technique would accomplish the removal of the unwanted Ritz pairs from the leading portion of the iteration. For example, using the unwanted Ritz values as shifts accomplishes this removal.

Computing in finite precision arithmetic complicates the situation. A robust implementation of the QR algorithm sets a subdiagonal element to zero if it is in magnitude less than some prescribed threshold and this technique is also adopted for deflation. This deflation overcomes the technical difficulty associated with tiny subdiagonals and improves the convergence of the IRA iteration. In addition, it may be impossible to accomplish the removal of the unwanted Ritz values from the leading portion of the iteration due to the forward instability [27, 39] of the QR algorithm.

The phenomena of the forward instability of the tridiagonal QR iteration [27] was initially explored by Parlett and Le. They observe that while the implicitly shifted QR iteration is always backward stable, there are cases where severe forward instability can occur. It is possible for a QR iteration to result in a computed Hessenberg matrix with entries that have no significant digits in common with the corresponding entries of the Hessenberg matrix that would have been determined in exact arithmetic. The implication is that the computed subdiagonal entries may not be reliable indicators for decoupling the Arnoldi factorization. Parlett and Le's analysis formally implies that the computed Hessenberg matrix may lose significant digits when the shift used is nearly an eigenvalue of H and the last component of the normalized eigenvector is small. We also mention the work of Watkins [39], which investigates the transmission of the shift during a QR step through H.

Since convergence of a Ritz value is predicated upon the associated Ritz estimate being small, using shifts that are near these converged values may force the IRA iteration to undergo forward instability. This indicates that it may be impossible to filter out unwanted eigenvalues with the implicit restarting technique, and this is the motivation for developing both the locking and the purging techniques. Further details may be found in Chapter 5 of [23].

5. Deflating converged Ritz values.

During an Arnoldi iteration, a Ritz value may be near an eigenvalue of A with no small elements appearing on the subdiagonal of H_k. However, when a Ritz value converges, it is always possible to make an orthogonal change of basis in which the appropriate subdiagonal of H_k is zero. The following result indicates how to exploit the convergence information available in the last row of the eigenvector matrix for H_k. For notational convenience, all subscripts are dropped on the Arnoldi matrices V, H, and f for the remainder of this section.

LEMMA 5.1. *Let $Hy = y\theta$ where $H \in \mathbf{R}^{k\times k}$ is an unreduced upper Hessenberg matrix and $\theta \in \mathbf{R}$ with $\|y\| = 1$. Let W be a Householder matrix such that $Wy = e_1\tau$ where $\tau = -\text{sign}(e_1^T y)$. Then*

$$e_k^T W = e_k^T + w^T, \tag{5.1}$$

where $\|w\| \le \sqrt{2}|e_k^T y|$ and

$$W^T H W e_1 = e_1 \theta. \tag{5.2}$$

Proof. The required Householder matrix has the form

$$W = I - \gamma(y - \tau e_1)(y - \tau e_1)^T,$$

where $\gamma = (1 + |e_1^T y|)^{-1}$ and $\tau = -\text{sign}(e_1^T y)$. A direct computation reveals that

$$e_k^T W = e_k^T + w^T, \tag{5.3}$$

where $w^T = \gamma e_k^T y(\tau e_1^T - y^T)$. Estimating

$$\begin{aligned} \|w\| &= \frac{|e_k^T y|}{1 + |e_1^T y|}\|y - \tau e_1\| \\ &= \frac{|e_k^T y|}{1 + |e_1^T y|}\sqrt{2(1 + |e_1^T y|)} \\ &\le \sqrt{2}|e_k^T y| \end{aligned}$$

establishes the bound on $\|w\|$. The final assertion (5.2) follows from

$$\begin{aligned} W^T H W e_1 &= \tau^{-1} W^T H y \\ &= \tau^{-1} \theta W^T y \\ &= \tau^{-1} \theta W y \quad (W^T = W) \\ &= \theta e_1. \quad \square \end{aligned}$$

Lemma 5.1 indicates that the last row and column of W differ from the last row and column of I_k by terms of order $|e_k^T y|$. The Ritz estimate (2.2) will indicate when it is safe to deflate the corresponding Ritz value θ. Rewriting (2.1) as

$$AVW = VWW^T HW + fe_k^T W$$

and using both (5.1) and (5.2) and partitioning we obtain

$$AVW = VW \begin{bmatrix} \theta & \bar{h}^T \\ 0 & \bar{H} \end{bmatrix} + fe_k^T + fw^T. \tag{5.4}$$

Equation (5.4) is not an Arnoldi factorization. In order to return to an Arnoldi factorization, the matrix $\bar{H}$ of order $k-1$ needs to be returned to upper Hessenberg form and the term fw^T dropped. Care must be taken not to disturb the matrix fe_k^T and the first column of $W^T HW$. To start the process we compute a Householder matrix Y_1 such that

$$Y_1^T \bar{H} Y_1 = \begin{bmatrix} \bar{G} & \bar{g} \\ \bar{\beta}_k e_{k-2}^T & \gamma \end{bmatrix},$$

with $e_{k-1}^T Y_1 = e_{k-1}^T$. The above idea is repeated resulting in Householder matrices $Y_1, Y_2, \ldots, Y_{k-3}$ that return $\bar{H}$ to upper Hessenberg form. Defining

$$Y = \begin{bmatrix} 1 & 0 \\ 0 & Y_1 Y_2 \cdots Y_{k-3} \end{bmatrix},$$

it follows by the construction of the Y_j that $e_k^T Y = e_k^T$ and

$$Y^T W^T HWY e_1 = \theta e_1. \tag{5.5}$$

The process of computing a similarity transformation as in equation (5.5) is not new. Wilkinson discusses similar techniques in [40, pp. 587–596]. Wilkinson references the work of Feller and Forsythe [13], who appear to be the first to use elementary Householder transformations for deflation. Problem 7.4.8 of [15, p. 371] addresses the case when working with upper Hessenberg matrices. What appears to be new is the application to the Arnoldi factorization for converged Ritz values.

Since $\|fw^T Y\| = \|f\| \, \|Y^T w\| = \|f\| \, \|w\|$, the size of $\|fw^T\|$ remains unchanged. Making the updates

$$V \leftarrow VWY, \quad H \leftarrow Y^T W^T HWY, \quad w^T \leftarrow w^T Y,$$

we obtain the relation

$$AV = VH + fe_k^T + fw^T. \tag{5.6}$$

A deflated Arnoldi factorization is obtained from equation (5.6) by discarding the term fw^T.

The following theorem shows that the deflated Arnoldi factorization resulting from this scheme is an exact k step factorization of a nearby matrix.

THEOREM 5.2. *Let an Arnoldi factorization of length k be given by* (5.6), *where* $Hy = y\theta$ *and* $\sqrt{2}|e_k^Ty|\,\|f\| \leq \epsilon\|A\|$ *for some* $\epsilon > 0$. *Then there exists a matrix* $E \in \mathbf{R}^{n\times n}$ *such that*

$$(A+E)V = VH + fe_k^T, \tag{5.7}$$

where

$$\|E\| \leq \epsilon\|A\|.$$

Proof. Subtract fw^T from both sides of equation (5.6). Set $E = -f(Vw)^T$ and then

$$EV = -f(Vw)^TV = -fw^T$$

and equation (5.7) follows. Using Lemma 5.1 gives

$$\|E\| = \|f\|\,\|w\| = \sqrt{2}|e_k^Ty|\,\|f\| \leq \epsilon\|A\|. \qquad \square$$

If A is symmetric then the choice $E = -f(Vw)^T - (Vw)f^T$ results in a symmetric perturbation. If ϵ is on the order of unit round-off then the deflation scheme introduces a perturbation of the same order to those already present from computing the Arnoldi factorization in floating point arithmetic.

Once a converged Ritz value θ is deflated, the Arnoldi vector corresponding to θ is locked or purged as described in the previous section. The only difficulty that remains is purging when A is nonsymmetric.

If A is not symmetric, then the Ritz pair may not be purged immediately because of the presence of $\bar{h}$. A standard reduction of H to block diagonal form is used. If θ is not an eigenvalue of $\bar{H}$, then we may construct a vector $z \in \mathbf{R}^{k-1}$ so that

$$\begin{bmatrix} \theta & \bar{h}^T \\ & \bar{H} \end{bmatrix}\begin{bmatrix} 1 & z^T \\ & I_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & z^T \\ & I_{k-1} \end{bmatrix}\begin{bmatrix} \theta & \\ & \bar{H} \end{bmatrix}. \tag{5.8}$$

Solving the linear system

$$(\bar{H}^T - \theta I_{k-1})z = \bar{h} \tag{5.9}$$

determines z. Define

$$Z \equiv \begin{bmatrix} 1 & z^T \\ & I_{k-1} \end{bmatrix}.$$

Postmultiplication of equation (5.6) by Z results in

$$AVZ = VZ\begin{bmatrix} \theta & \\ & \bar{H} \end{bmatrix} + fe_k^T + fw^TZ$$

since $e_k^TZ = e_k^T$. Equating the last $k-1$ columns of the previous expression results in

$$AV\begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix} = V\begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix}\bar{H} + fe_{k-1}^T + fw^T\begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix}. \tag{5.10}$$

Compute the factorization (using $k-1$ Givens rotations)

$$QR = \begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix}, \tag{5.11}$$

where $Q \in \mathbf{R}^{k \times k-1}$ with $Q^TQ = I_{k-1}$ and R is an upper triangular matrix of order $k-1$. Since the last $k-1$ columns of Z are linearly independent, R is nonsingular. Postmultiplying equation (5.10) by R^{-1} gives

$$AVQ = VQR\bar{H}R^{-1} + \rho_{k-1}^{-1} f e_{k-1}^T + f w^T Q, \tag{5.12}$$

where $\rho_{k-1} = e_{k-1}^T R e_{k-1}$. The last term fw^TQ in (5.12) is discarded by the deflation scheme, and this relation shows that the discarded term is not magnified in norm by the purging procedure. The matrix $R\bar{H}R^{-1}$ remains upper Hessenberg since R is upper triangular.

Partitioning Q conformally with the right side of equation (5.11) results in

$$\begin{bmatrix} q_{11}^T \\ Q_{21} \end{bmatrix} R = \begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix},$$

and it follows that $R^{-1} = Q_{21}$. Using the Cauchy–Schwarz inequality, it follows that $|\rho_{k-1}^{-1}| = |e_{k-1}^T Q_{21} e_{k-1}| \le 1$ and hence the Arnoldi residual is not amplified by the purging. The final purged Arnoldi factorization is

$$AVQ = VQR\bar{H}Q_{21} + \rho_{k-1}^{-1} f e_{k-1}^T. \tag{5.13}$$

Performing the set of updates

$$V \leftarrow VQ, \quad H \leftarrow R\bar{H}Q_{21}, \quad f \leftarrow \rho_{k-1}^{-1} f$$

defines equation (5.13) as an Arnoldi factorization of length $k-1$. Theorem 5.2 implies this is an Arnoldi factorization for a nearby matrix. It is easily verified that $V^T f(e_{k-1}^T + w^T) = 0$ and that H is an upper Hessenberg matrix of order $k-1$. Since the term fw^T is discarded, the Ritz estimates given by the updated Arnoldi factorization for the remaining Ritz values will be slightly inaccurate. Lemma 5.1 and the fact that $\|R^{-1}\| \le 1$ may be used to show that the errors in these estimates are bounded above by $\|f\|(\sqrt{2}|e_k^T y|)$. If $w = 0$ then the Ritz estimates for the updated factorization would be exactly the same as the Ritz residuals and estimates for the original one.

6. A practical deflating procedure for the Arnoldi factorization.

The practical issues associated with a numerically stable deflating procedure are addressed in this section. These include

1. performing the deflation in real arithmetic when a converged Ritz value has a nonzero imaginary component,
2. deflation with more than one converged Ritz value,
3. error analysis.

Section 6.2 presents two algorithms that implement the deflation schemes. The error analysis of the two deflation schemes is presented in the next section.

6.1. Deflation with real arithmetic.

Suppose $H(y+iz) = (\theta + i\mu)(y+iz)$ where y and z are unit vectors in $\mathbf{R}^k$, $H \in \mathbf{R}^{k\times k}$, and $\mu \neq 0$. It then follows that

$$H \left[\begin{array}{cc} y & z \end{array}\right] = \left[\begin{array}{cc} y & z \end{array}\right] \left[\begin{array}{cc} \theta & \mu \\ -\mu & \theta \end{array}\right] \equiv \left[\begin{array}{cc} y & z \end{array}\right] C.$$

Thus, we may deflate a complex Ritz value in real arithmetic if $|e_k^T y|$ and $|e_k^T z|$ are small enough.

Suppose that H corresponds to an Arnoldi factorization of length k and that $|e_k^T y| = O(\epsilon) = |e_k^T z|$. Factor

$$\left[\begin{array}{cc} y & z \end{array}\right] = U \left[\begin{array}{c} T \\ 0 \end{array}\right], \tag{6.1}$$

where $U^T U = I_k$ and T is an upper triangular matrix. It is easily shown that y and z are linearly independent as vectors in $\mathbf{R}^k$ since $\mu \neq 0$ and the nonsingularity of T follows. Performing a similarity transformation on H with U gives

$$U^T H U \left[\begin{array}{cc} e_1 & e_2 \end{array}\right] = \left[\begin{array}{c} TCT^{-1} \\ 0 \end{array}\right].$$

In order to deflate the complex conjugate pair of eigenvalues from the factorization in an implicit manner, we require that $e_k^T U = e_k^T + u^T$ where $\|u\| = O(\epsilon)$.

We now show that the magnitudes of the last components of y and z are not sufficient to guarantee the required form for U. Suppose that $z = y\cos\phi + r\sin\phi$ where r is a unit vector orthogonal to y and ϕ measures the positive angle between y and z. Lemma 5.1 implies that a Householder W matrix may be constructed such that

$$W^T \left[\begin{array}{cc} y & z \end{array}\right] = \left[\begin{array}{cc} \tau_1 e_1 & \tau_1 e_1 \cos\phi + W^T r \sin\phi \end{array}\right] \equiv \left[\begin{array}{cc} \tau_1 & \zeta \\ 0 & \hat{z} \end{array}\right],$$

where $\tau_1 = \pm 1$ and the last column and row of W and I_k are the same up to order $e_k^T y$. To compute the required orthogonal factorization in equation (6.1) another Householder matrix $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \hat{Q} \end{smallmatrix}\right]$ is needed so that $\hat{Q}^T \hat{z} = \pm \|\hat{z}\| e_1$. But Lemma 5.1 only results in $e_{k-1}^T \hat{Q} = e_{k-1}^T + \hat{q}^T$ with $\|\hat{q}\| = O(\epsilon)$ if $e_{k-1}^T \hat{z}$ is small relative to $\|\hat{z}\|$. Unfortunately, if ϕ is small, $W^T z \approx \tau_1 e_1$ and $\|\hat{z}\| \approx \phi$. Hence we cannot obtain the required form for $U = WQ$.

Fortunately, when y and z are nearly aligned, μ may be neglected, as the following result demonstrates.

LEMMA 6.1. *Let $H(y+iz) = (\theta + i\mu)(y+iz)$ where y and z are unit vectors in $\mathbf{R}^k$, $H \in \mathbf{R}^{k\times k}$, and $\mu \neq 0$. Suppose that ϕ measures the positive angle between y and z. Then*

$$|\mu| \le \sin\phi \|H\|. \tag{6.2}$$

Proof. Let $z = y\cos\phi + r\sin\phi$ where r is a unit vector orthogonal to y and ϕ measures the positive angle between y and z. Equating real and imaginary parts of $H(y+iz) = (\theta + i\mu)(y+iz)$ results in $Hy = y\theta - z\mu$ and $Hz = y\mu + z\theta$. The desired estimate follows since

$$2\mu = y^T H z - z^T H y = \sin\phi(y^T H r - r^T H y)$$

results in $|\mu| \leq \sin\phi \|H\|$. □

For small ϕ, y and z are almost parallel eigenvectors of H corresponding to a nearly multiple eigenvalue. Numerically, we set μ to zero and deflate one copy of θ from the Arnoldi factorization.

A computable bound on the size of the angle ϕ is now determined using only the real and imaginary parts of the eigenvector. The second Householder matrix Q should not be computed if

$$|e_{k-1}^T \hat{z}| > \|\hat{z}\| |e_k^T z|. \tag{6.3}$$

Recall that Lemma 5.1 gives $e_k^T W = e_k^T + w^T$ where $w^T = \gamma e_k^T y(\tau_1 e_1^T - y^T)$ and $\gamma = (1 + |e_1^T y|)^{-1}$. Thus

$$e_{k-1}^T \hat{z} = e_k^T W^T z = e_k^T W z = e_k^T z + w^T z$$

where the symmetry of W is used. The estimate

$$\|\hat{z}\| = \| \, [\, 0 \;\; \hat{z}^T \,]^T \, \| = \|W^T r\| \sin\phi = \sin\phi$$

follows since W is orthogonal and r is a unit vector. Rewriting equation (6.3), we obtain

$$\begin{aligned} \sin\phi &< \left| \frac{e_k^T z + w^T z}{e_k^T z} \right| \\ &= \left| 1 + \frac{w^T z}{e_k^T z} \right| \\ &= \left| 1 + \gamma(\tau_1 e_1^T z - y^T z) \frac{e_k^T y}{e_k^T z} \right| \end{aligned} \tag{6.4}$$

as our computable bound.

Suppose that $HX = XD$ where $X \in \mathbf{R}^{k \times j}$ and D is a quasi-diagonal matrix. The eigenvalues of H are on the diagonal of D if they have zero imaginary component and in blocks of two for the complex conjugate pairs. The columns of X span the eigenspace corresponding to diagonal values of D. For the blocks of order-2 on the diagonal the corresponding complex eigenvector is stored in two consecutive columns of X, the first holding the real part, and the second the imaginary part. If we want to block deflate X, where the last row is small, from H we could proceed as follows. Compute the orthogonal factorization $X = Q[\begin{smallmatrix} R \\ 0 \end{smallmatrix}]$ via Householder reflectors where $Q^T Q = I_k$ and $R \in \mathbf{R}^{k \times k}$ is upper triangular. Then the last row and column of Q differ from that of I_k with terms on the same order of the entries in the last row of X if the condition number of R is modest. Thus, if the columns of X are not almost linearly dependent, an appropriate Q may be determined. Finally, we note that when H is a symmetric tridiagonal matrix, an appropriate Q may always be determined.

6.2. Algorithms for deflating converged Ritz values.

The two procedures presented in this section extend the ideas of §4 to provide deflation of more than one converged Ritz value at a time. The first purges the factorization of the unwanted converged Ritz values. The second locks the Arnoldi vectors corresponding to the desired converged Ritz values. When both deflation algorithms are incorporated within an IRA iteration, the locked vectors form a basis for an approximate invariant subspace of A. This truncated factorization is an approximate partial Schur decomposition.

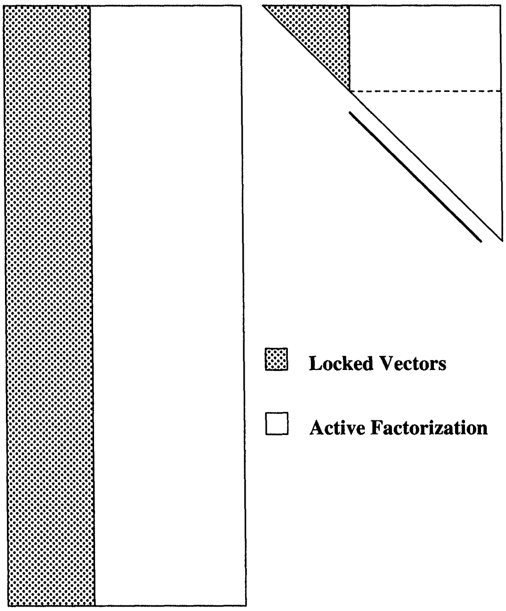

FIG. 6.1. *The matrix product V_mH_m of the factorization upon entering Algorithm* 6.2 *or* 6.3. *The shaded region corresponds to the converged portion of the factorization.*

When A is symmetric, the approximate Schur vectors are Ritz vectors and the upper quasi-triangular matrix is the diagonal matrix of Ritz values.

Partition a length m Arnoldi factorization as

$$A\begin{bmatrix} V_j & \bar{V}_{m-j} \end{bmatrix} = \begin{bmatrix} V_j & \bar{V}_{m-j} \end{bmatrix} \begin{bmatrix} H_j & G_j \\ 0 & \bar{H}_{m-j} \end{bmatrix} + f_m e_m^T + f w^T, \tag{6.5}$$

where H_j and $\bar{H}_{m-j}$ are upper quasi-triangular and unreduced upper Hessenberg matrices, respectively. The matrix $H_j \in \mathbf{R}^{j\times j}$ contains the wanted converged Ritz values of the matrix H_m. The columns of $V_j \in \mathbf{R}^{n\times j}$ are the locked Arnoldi vectors that represent an approximate Schur basis for the invariant subspace of interest. The matrix $\bar{H}_{m-j}$ designates the trailing submatrix of order $m-j$. Analogously, the last $m-j$ columns of V_m are denoted by $\bar{V}_{m-j}$. We shall refer to the last $m-j$ columns of (6.5) as the *active* part of the factorization. Finally, $G_j \in \mathbf{R}^{j\times(m-j)}$ denotes the submatrix in the northeast corner of H_m. Figure 6.1 illustrates the matrix product V_mH_m of equation (6.5).

If A is symmetric, the two deflation procedures simplify considerably. In fact, purging is only used when A is nonsymmetric for otherwise $G_j = 0_{j\times(m-j)}$ and both H_j and $\bar{H}_{m-j}$ are symmetric tridiagonal matrices. Both algorithms are followed by remarks concerning some of the specific details.

ALGORITHM 6.2.

function $[V_m, H_m, f_m] = \texttt{Lock}\ (V_m, H_m, f_m, X_i, j)$

INPUT: A length m Arnoldi factorization $AV_m = V_mH_m + f_me_m^T$. The first j columns of V_m represent an approximate invariant subspace for A. The leading principal submatrix H_j of order j of H_m is upper quasi-triangular and contains the converged Ritz values of interest. The columns of $X_i \in \mathbf{R}^{(m-j)\times i}$ are the eigenvectors corresponding to the eigenvalues that are to be locked.

OUTPUT: A length m Arnoldi factorization defined by V_m, H_m, and f_m where the first $j+i$ columns of V_m are an approximate invariant subspace for A.

```
1. Compute the orthogonal factorization
       $Q \begin{bmatrix} R_i \\ 0_{m-j-i} \end{bmatrix} = X_i,$
   where $Q \in \mathbf{R}^{(m-j)\times(m-j)}$ using Householder matrices;
2. Update the factorization
   $\bar{H}_{m-j} \leftarrow Q^T\bar{H}_{m-j}Q$; $\bar{V}_{m-j} \leftarrow \bar{V}_{m-j}Q$; $G_j \leftarrow G_jQ$;
3. Compute an orthogonal matrix $P \in \mathbf{R}^{(m-j-i)\times(m-j-i)}$ using Householder matrices that restore $\bar{H}_{m-j-i}$ to upper Hessenberg form;
4. Update the factorization
   $\bar{H}_{m-j-i} \leftarrow P^T\bar{H}_{m-j-i}P$; $\bar{V}_{m-j-i} \leftarrow \bar{V}_{m-j-i}P$; $G_{j+i} \leftarrow G_{j+i}P$;
```

Line 1 computes an orthogonal basis for the eigenvectors of $\bar{H}_{m-j}$ that correspond to the Ritz estimates that are converged. The matrix of eigenvectors in line 1 satisfies the equation $\bar{H}_{m-j}X_i = X_iD_i$ where D_i is a quasi-diagonal matrix containing the eigenvalues to be locked. From §6.1, we see that the leading submatrix of $Q^T\bar{H}_{m-j}Q$ of order i is upper quasi-triangular. The required relation $e_m^TQ = e_m^T + q^T$, with $\|q\|$ small, is guaranteed if the condition number of R_i is modest. Since i is typically a small number, we compute the condition number of R_i. The number of vectors to be locked is assumed to be such that the condition number of R_i is small. In particular, if H_m is a symmetric tridiagonal matrix, Q always has the required form. Lines 3–4 return the updated $\bar{H}_{m-j}$ to upper Hessenberg form.

Before entering Purge, the unwanted converged Ritz pairs are placed at the front of the factorization. A prior call to Lock places the unwanted values and vectors to the beginning of the factorization. Unlike Lock, the procedure Purge requires accessing and updating the entire factorization when A is nonsymmetric. Thus, for large-scale nonsymmetric eigenvalue computations, the amount of purging performed should be kept to a minimum.

ALGORITHM 6.3.

function $[V_{m-i}, H_{m-i}, f_{m-i}] =$ Purge (V_m, H_m, f_m, j, i)

INPUT: A length m Arnoldi factorization $AV_m = V_mH_m + f_me_m^T$. The first $i+j$ columns of V_m represent an approximate invariant subspace for A. The leading principal submatrix H_{i+j} of order $i+j$ of H_m is upper quasi-triangular and contains the converged Ritz values. The i unwanted converged eigenvalues are in the leading portion of H_{i+j}. The converged complex conjugate Ritz pairs are stored in 2×2 blocks on the diagonal of H_{i+j}.

OUTPUT: A length $m-i$ Arnoldi factorization defined by V_{m-i}, H_{m-i}, and f_{m-i} purged of the unwanted converged Ritz values and corresponding Schur vectors.

```
Lines 1–3 purge the factorization of the unwanted converged Ritz values contained in the leading portion of $H_m$;
1. Solve the Sylvester set of equations,
       $Z\bar{H}_{m-i} - H_iZ = G_i,$
   for $Z \in \mathbf{R}^{i\times(m-i)}$ that arise from block diagonalizing $H_m$
       $H_m \begin{bmatrix} I_i & Z \\ & I_{m-i} \end{bmatrix} = \begin{bmatrix} I_i & Z \\ & I_{m-i} \end{bmatrix} \begin{bmatrix} H_i & \\ & \bar{H}_{m-i} \end{bmatrix};$
```

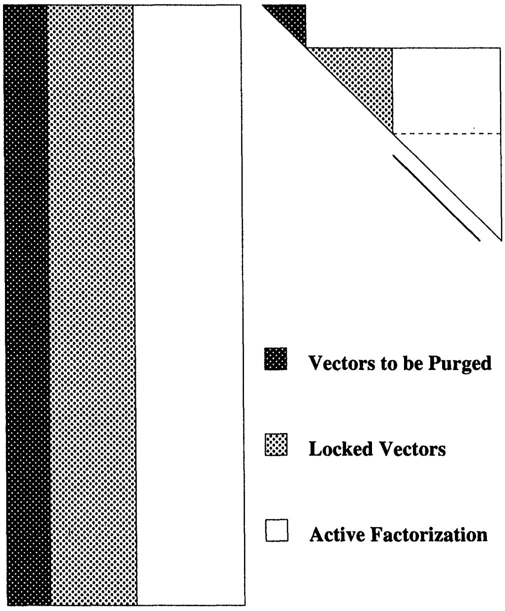

FIG. 6.2. *The matrix product V_mH_m of the factorization just prior to discarding in Algorithm 6.3. The darkly shaded regions may now be dropped from the factorization.*

2. *Compute the orthogonal factorization*

$$QR_{m-i} = \begin{bmatrix} Q_i \\ Q_{m-i} \end{bmatrix} R_{m-i} = \begin{bmatrix} Z \\ I_{m-i} \end{bmatrix},$$

where $Q \in \mathbf{R}^{m \times (m-i)}$ using Householder matrices;

3. *Update the factorization and obtain a length $m-i$ factorization* $H_{m-i} \leftarrow R_{m-i}\bar{H}_{m-i}Q_{m-i}$; $V_{m-i} \leftarrow V_mQ$; $f_{m-i} \leftarrow \rho^{-1}_{m-i,m-i}f_m$; *where* $\rho_{m-i,m-i} = e^T_{m-i}R_{m-i}e_{m-i}$;

At the completion of Algorithm 6.3 the factorization is of length $m-i$ and the leading submatrix of order j will be upper quasi-triangular. The wanted converged Ritz values will be either on the diagonal if real or in blocks of two for the complex conjugate pairs. Figure 6.2 shows the structure of the updated V_mH_m just prior to discarding the unwanted portions.

The solution of the Sylvester equation at line 1 determines the matrix Z that block diagonalizes the spectrum of H_m into two submatrices. The unwanted portion is in the leading corner and the remaining eigenvalues of H_m are in the other block. A solution Z exists when the H_i and $\bar{H}_{m-i}$ do not have a common eigenvalue. If there is an eigenvalue that is shared by H_i and $\bar{H}_{m-i}$, then H_m has an eigenvalue of multiplicity greater than one. The remedy is a criterion that determines whether to increase or decrease i, the number of Ritz values that require purging. Analysis similar to that in §5 demonstrates that after line 3 the Ritz estimates for the eigenvalues of H_{m-i} are not altered. We also remark that R_{m-i} is nonsingular since the matrix

$$\begin{bmatrix} Z \\ I_{m-i} \end{bmatrix}$$

is of full column rank and $|\rho^{-1}_{m-i,m-i}| \leq 1$.

7. Error analysis.

This section examines the numerical stability of the two deflation algorithms when computing in finite precision arithmetic. A stable algorithm computes the exact solution of a nearby problem. It will be shown that Algorithms 6.3 and 6.2 deflate slightly perturbed matrices.

For ease of notation

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

replaces $H_m \in \mathbf{R}^{m \times m}$ used by procedures `Lock` and `Purge` of §6.2. The submatrix H_{11} is of order i and H_{21} is zero except for the subdiagonal entry of H located in the northeast corner. Analogously, $\hat{H}$ represents H after the similarity transformation performed by `Lock` or `Purge` partitioned conformally.

7.1. Locking.

The locking scheme is considered successful if the desired eigenvalues end up in $\hat{H}_{11}$ and $\hat{H}_{21}$ is small in norm. The largest source of error is from computing an orthogonal factorization from the approximate eigenvector matrix containing the vectors to be locked.

The matrix pair (X, D) represents an approximate quasi-diagonal form for H. The computed eigenvalues of H are on the diagonal of D if they have zero imaginary component and in blocks of two for the complex conjugate pairs. The computed columns of X span the right eigenspace corresponding to diagonal values of D. For the blocks of order-2 on the diagonal, the corresponding complex eigenvector is stored in two consecutive columns of X, the first holding the real part and the second the imaginary part. We assume that X is a nonsingular matrix and that each column is a unit vector.

Standard results give $\|XD - HX\| \leq \epsilon_1 \|H\|$ where ϵ_1 is a small multiple of machine precision for a stable algorithm. Defining the matrix $E = (XD - HX)Y^T$ where $X^{-1} = Y^T$ it follows that $(H + E)X = XD$. If $\sigma_m^{-1}(X)$ is the smallest singular value of X then $\|X^{-1}\| = \sigma_m^{-1}(X)$. Since each column of X is a unit vector, $\|X\| \leq \sqrt{m}$. If $\kappa(X) = \|X\| \|X^{-1}\|$ is the condition number for the matrix of approximate eigenvectors, $\|E\| \leq \epsilon_1 \kappa(X) \|H\|$. If X is a well-conditioned matrix, then the approximate quasi-diagonal form for H is exact for a nearby matrix. In particular, if H is symmetric then E is always a small perturbation. As the columns of X become linearly dependent, $\sigma_m(X)$ decreases and E may represent a large perturbation.

The following result informs us that locking is a conditionally stable process.

THEOREM 7.1. *Let $H \in \mathbf{R}^{m \times m}$ be an unreduced upper Hessenberg matrix with distinct eigenvalues. Suppose that*

$$X = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \text{ and } D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

is an approximate quasi-diagonal form for H that satisfies $(H + E)X = XD$ where $\|E\| \leq \epsilon_1 \kappa(X) \|H\|$. Let $Q_1 R_1 = X_1 \in \mathbf{R}^{m \times j}$ where $Q_1^T Q_1 = I_j$. Suppose a QR factorization of X_1 is computed so that $\hat{Q}\hat{R} = X_1 + \hat{E}$ where $\hat{Q}^T \hat{Q} = I_m$ and $\|\hat{E}\| \leq \epsilon_2 \|X_1\|$. Both ϵ_1 and ϵ_2 are small multiples of the machine precision ϵ_M. Let $\epsilon = \max(\epsilon_1, 2\epsilon_2)$ and let $\kappa(R_1) = \|R_1\| \|R_1^{-1}\|$ be the condition number for R_1 where

$$\mu \equiv \frac{\kappa(R_1)}{1 - \epsilon_2 \kappa(R_1)}.$$

If $\theta \equiv \epsilon(\kappa(X)+\epsilon\mu(1+\epsilon\mu\kappa(R_1))) < 1$, *then there exists a matrix* $C \in \mathbf{R}^{m\times m}$ *such that*

$$\hat{Q}^T(H-C)\hat{Q} = \hat{H} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ 0 & \hat{H}_{22} \end{bmatrix},$$

where $\hat{H}_{11}$ *is an upper quasi-triangular matrix similar to* D_1 *and*

$$\|C\| \le \epsilon(\kappa(X)+\mu)\|H\| + O(\epsilon^2). \tag{7.1}$$

A few remarks are in order.

1. If H is symmetric, $\hat{H}_{12} = 0$ and $\hat{H}_{11}$ is diagonal. Procedure Lock is stable, since noted previously $\kappa(X) = 1$ and $\mu \approx 1$. Parlett [26, pp. 85–86] proves Theorem 7.1 for symmetric matrices when locking one approximate eigenvector.
2. If only one column is locked, then $\mu = 1 + O(\epsilon)$ and $\|C\|$ is small relative to $\kappa(X)\|H\|$.
3. If $\kappa(R_1)$ is large, the columns of X_1 are nearly dependent. In this case, $\kappa(X)$ will also be large and locking will likely introduce no more error into the computation than already present from computing the quasi-diagonal pair (X, D). The factor of μ may be minimized by decreasing j, the number of columns locked.
4. A conservative strategy locks only one vector at a time. The only real concern is when locking two vectors corresponding to a complex conjugate pair. If the real and imaginary parts of the complex eigenvector are nearly aligned, μ will be large and locking may be unstable. But as §6.1 explains, the complex conjugate pair may be numerically regarded as a double eigenvalue with zero imaginary part. Only one copy is deflated and $\mu \approx 1$.

Proof. Partition

$$X = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \text{ and } D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}.$$

The i columns of X_1 are a basis for the right eigenspace to be locked, and D_1 contains the corresponding eigenvalues. We assume that the eigenvalues of D_1 and D_2 are distinct and that X is nonsingular. Let

$$Y^T = \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix}$$

denote the inverse of X. The rows of Y_1^T span the left eigenspace associated with the computed eigenvalues of D_1.

Let the product $\hat{Q}\hat{R}$ be an exact QR factorization of a matrix near X_1:

$$\hat{Q}\hat{R} = \begin{bmatrix} \hat{Q}_1 & \hat{Q}_2 \end{bmatrix} \begin{bmatrix} \hat{R}_1 \\ 0 \end{bmatrix} = X_1 + \hat{E}$$

where $\|\hat{E}\| \le \epsilon_2\|X_1\|$. Using Theorem 1.1 of Stewart [36], since $\|R_1^{-1}\|\|\hat{E}\| < \theta < 1$ there exist matrices $W_1 \in \mathbf{R}^{m\times j}$ and $F_1 \in \mathbf{R}^{j\times j}$ such that $(Q_1+W_1)(R_1+F_1) = \hat{Q}_1\hat{R}_1$ where

$$QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = X_1$$

and $(Q_1 + W_1)^T(Q_1 + W_1) = I_j$. Define

$$F = \begin{bmatrix} F_1 \\ 0 \end{bmatrix} \text{ and } W = \begin{bmatrix} W_1 & 0 \end{bmatrix}.$$

The matrices W and F are the perturbations that account for the backward error $\hat{E}$ produced by computation.

Partitioning W conformally with Q gives

$$\begin{aligned}
\hat{Q}^T H \hat{Q} &= \hat{Q}^T X D Y^T \hat{Q} - \hat{Q}^T E \hat{Q} \\
&= \hat{Q}^T (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) \hat{Q} - \hat{Q}^T E \hat{Q} \\
&\approx \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \\
&\quad + W^T (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \\
&\quad + \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) W - \hat{Q}^T E \hat{Q},
\end{aligned} \tag{7.2}$$

where the second-order terms involving W are ignored. From the decomposition $X_1 = Q_1 R_1$ it follows that $Q_1 = X_1 R_1^{-1}$ which gives $Q_2^T X_1 = 0$. The equality $Y^T = X^{-1}$ implies that $Y_l^T X_l = I$ for $l = 1, 2$ and $Y_2^T X_1 = 0 = Y_1^T X_2$ and hence $Y_2^T Q_1 = 0$.

Using these relationships, equation (7.2) becomes

$$\hat{Q}^T H \hat{Q} = \begin{bmatrix} R_1 D_1 R_1^{-1} & Q_1^T X D Y^T Q_2 \\ 0 & Q_2^T X_2 D_2 Y_2^T Q_2 \end{bmatrix} + \hat{C} \tag{7.3}$$

$$\equiv \hat{H} + \hat{C}, \tag{7.4}$$

where the matrix $\hat{C}$ absorbs the three matrix products involving W or E on the right-hand side of equation (7.2). We note that if H is symmetric, $Q_1^T X_2 = 0 = Y_1^T Q_2$, R_1 is a diagonal matrix, and hence $R_1 D_1 R_1^{-1} = D_1$. Thus $\hat{H}$ is also a symmetric matrix. Defining $C = \hat{Q} \hat{C} \hat{Q}^T$ equation (7.4) is rewritten as $\hat{Q}^T (H - C) \hat{Q} = \hat{H}$. Since $Q\hat{H} = (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) Q$ and using the definition of $\hat{C}$ from equation (7.2),

$$\hat{C} = W^T Q \hat{H} + Q^T W \hat{H} - \hat{Q}^T E \hat{Q}, \tag{7.5}$$

it follows that $\|C\| \le 2\|W^T Q\| \|\hat{H}\| + \|E\|$. The result of Theorem 1.1 of Stewart [36] also provides the estimate

$$\|W^T Q\| \le \|W\| \le \epsilon_2 \mu (1 + \epsilon_2 \mu \kappa(R_1)),$$

where $O(\epsilon^3)$ terms are ignored. For modest values of μ, W is numerically orthogonal to Q. From equation (7.5)

$$\begin{aligned}
\|C\| &= \|\hat{C}\| \\
&\le 2\epsilon_2 \mu (1 + \epsilon_2 \mu \kappa(R_1)) \|\hat{H}\| + \epsilon_1 \kappa(X) \|H\| \\
&\le 2\epsilon_2 \mu (1 + \epsilon_2 \mu \kappa(R_1)) (\|H\| + \|C\|) + \epsilon_1 \kappa(X) \|H\| \\
&\le \epsilon(\kappa(X) + \mu(1 + \epsilon \mu \kappa(R_1))) \|H\| + \epsilon \mu (1 + \epsilon \mu \kappa(R_1)) \|C\| \\
&\equiv \theta \|H\| + \hat{\theta} \|C\|,
\end{aligned}$$

where the second inequality uses equation (7.4). Since $\hat{\theta} < \theta$, rearranging the last inequality gives $\|C\|(1 - \theta) \le \theta \|H\|$. Ignoring $O(\theta^2)$ terms $\|C\| \le \theta \|H\|$. The estimate on the size of C in equation (7.1) now follows since $\theta = \epsilon(\kappa(X) + \mu(1 + \epsilon \mu \kappa(X))) \le \epsilon(\kappa(X) + \mu) + O(\epsilon^2)$. □

7.2. Purging. The success of the purging scheme depends upon the solution of the Sylvester set of equations required by Algorithm 6.3. We rewrite the Sylvester set of equations in Algorithm 6.3 as $ZH_{22} - H_{11}Z = H_{12}$. The job is to examine the effect of performing the similarity transformation $RH_{22}R^{-1}$, where

$$QR \equiv \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} R = \begin{bmatrix} Z \\ I \end{bmatrix} \equiv S.$$

The last relation implies that $R^{-1} = Q_2^T$. In actual computation, this equality obviates the need to solve linear systems with R necessary for the similarity transformation. For the error analysis that follows, R^{-1} is used in a formal sense.

Let $\hat{Z}$ be the computed solution to the Sylvester set of equations. In a similar analysis, Bai and Demmel [2] assume that the QR factorization of S is performed exactly and we do also. The major source of error is that arising from computing $\hat{Z}$.

Suppose that $\hat{Q}\hat{R} = [\begin{smallmatrix}\hat{Z}\\ I\end{smallmatrix}] \equiv \hat{S}$. Write $\hat{Z} = Z + E$ where E is the error in $\hat{Z}$. If $QR = S$ and $\|R^{-1}\|\|E\| < 1$, then Theorem 1.1 of Stewart [36] gives matrices W and F such that $(Q+W)(R+F) = \hat{Q}\hat{R}$ where $(Q+W)^T(Q+W) = I_m$. The result gives the bound $\|F\| \le \|R\|\|E\| + O(\|E\|^2)$. Up to first-order perturbation terms,

$$\hat{R}H_{22}\hat{R}^{-1} = (R+F)H_{22}(R+F)^{-1} = RH_{22}R^{-1} + RH_{22}R^{-1}FR^{-1} + FH_{22}R^{-1}.$$

Defining the error matrix $C = H_{22}R^{-1}F + R^{-1}FH_{22}$ it follows that

$$\hat{R}H_{22}\hat{R}^{-1} = R(H_{22} + C)R^{-1}.$$

Ignoring second-order terms, we obtain the estimate

$$\|C\| \le 2\|R^{-1}\|\|F\|\|H_{22}\| \le 2\kappa(S)\|E\|\|H_{22}\|.$$

The invariance of $\|\cdot\|$ under orthogonal transformations gives $\kappa(S) = \|R^{-1}\|\|R\|$. Since the singular values of S are the square roots of the eigenvalues of S^TS it follows that

$$\kappa(S) = \sqrt{\frac{1+\sigma_{\max}^2(Z)}{1+\sigma_{\min}^2(Z)}},$$

where $\sigma_{\max}(Z)$ and $\sigma_{\min}(Z)$ are the largest and smallest singular values of Z. Since Z^TZ is a symmetric positive semidefinite matrix, $\lambda_{\max}(Z^TZ) = \|Z\|^2$, and then $\kappa(S) \le \sqrt{1+\|Z\|^2}$, with equality if zero is an eigenvalue of Z^TZ.

The previous discussion is summarized in the following result.

THEOREM 7.2. *Let $\hat{Z}$ be the computed solution to the Sylvester set of equations, $ZH_{22} - H_{11}Z = H_{12}$, where the eigenvalues of H_{11} and H_{22} are distinct. Let $\hat{Z} = Z + E$ where E is the error in $\hat{Z}$ and suppose that $\|R^{-1}\|\|E\| < 1$ where $QR = [\begin{smallmatrix}Z\\ I\end{smallmatrix}]$.*

Then there exists a matrix C such that

$$\hat{R}H_{22}\hat{R}^{-1} = R(H_{22} + C)R^{-1},$$

where

$$\|C\| \le 2\sqrt{1+\|Z\|^2}\,\|E\|\,\|H\|. \tag{7.6}$$

If $\|E\|$ is a modest multiple of machine precision and the solution of Sylvester's equations is not large in norm, then purging is backward stable since $\|C\|$ is small relative to $\|H\|$.

The two standard approaches [3, 16] for solving Sylvester's equation show that $\|\hat{F}\|_F \leq \epsilon_3(\|H_{11}\|_F + \|H_{22}\|_F)\|\hat{Z}\|_F$ where $\hat{F} \equiv H_{12} - \hat{Z}H_{22} + H_{11}\hat{Z}$ and ϵ_3 is a modest multiple of machine precision. Standard bounds [8, 15] also give $\|Z\|_F \leq \text{sep}^{-1}(H_{11}, H_{22})\|H_{12}\|_F$ where

$$\text{sep}(H_{11}, H_{22}) \equiv \min_{X\neq 0} \frac{\|XH_{22} - H_{11}X\|_F}{\|X\|_F}$$

is the *separation* between H_{11} and H_{22}. Although

$$\text{sep}(H_{11}, H_{22}) \leq \min_{k,l} |\lambda_k(H_{11}) - \lambda_l(H_{22})|,$$

Varah [38] indicates that if the matrices involved are highly non-normal, the smallest difference between the spectrums of H_{11} and H_{22} may be an overestimate of the actual separation. Recently, Higham [19] gave a detailed error analysis for the solution of Sylvester's equation. The analysis takes into account the special structure of the equations involved. For example, Higham shows that $\|E\|_F \leq \text{sep}^{-1}(H_{11}, H_{22})\|\hat{F}\|_F$, but this may lead to an arbitrarily large estimate of the true forward error. For use in practical error estimation, LAPACK-style software is available.

A robust implementation of procedure Lock determines the backward stability by estimating both $\|Z\|$ and $\|E\|$.

8. Other deflation techniques.

Wilkinson [40, pp. 584–602] has given a comprehensive treatment of various deflation schemes associated with iterative methods. Recently, Saad [31, pp. 117–125, 180–182] discussed several deflation strategies used with both simultaneous iteration and Arnoldi's method. Algorithm 6.2 is an in-place version of one of these schemes [31, p. 181]. Saad's version explicitly orthonormalizes the newly converged Ritz vectors against the already computed approximate j Schur vectors. This is the form of locking used by Scott [33]. Instead, procedure Lock achieves the same task implicitly through the use of Householder matrices in $\mathbf{R}^{m\times m}$. Thus, we are able to orthogonalize vectors in $\mathbf{R}^n$ at a reduced expense since $m \ll n$.

Other deflation strategies include the various Wielandt deflation techniques [31, 40]. We briefly review those that do not require the approximate left eigenvectors of A or complex arithmetic. Denote by $\lambda_1, \ldots, \lambda_j$ the wanted eigenvalues of A. The Wielandt and Schur–Wielandt forms of deflation determine a rank j modification of A,

$$A_j = A - U_j S_j U_j^T, \tag{8.1}$$

where $S_j \in \mathbf{R}^{j\times j}$ and j represents the dimension of the approximate invariant subspace already computed. The idea is to choose S_j so that A_j will converge to the remainder of the invariant subspace desired. For example, S_j is selected to be a diagonal matrix of shifts $\sigma_1, \ldots, \sigma_j$ so that A_j has eigenvalues $\{\lambda_1 - \sigma_1, \ldots, \lambda_j - \sigma_j, \lambda_{j+1}, \ldots, \lambda_n\}$.

Both forms of deflation differ in the choice of U_j. The Wielandt variant uses converged Ritz vectors while the Schur–Wielandt uses approximate Schur vectors. With either form of deflation, the eigenvalues of A_j are $\lambda_i - \sigma_i$ for $i \leq j$ and λ_i otherwise, and both forms leave the Schur vectors unchanged. This motivates Saad to suggest that an approximate Schur basis should be incrementally built as Ritz vectors of A_j converge. Braconnier [6] employs the Wielandt variant and discusses the details of deflating a converged Ritz value that has a nonzero imaginary part in real arithmetic.

We now compare our locking scheme to the Schur–Wielandt deflation technique. We shall assume that $AU_j = U_jR_j$ is a real partial Schur form of order j for A, and we will put $S_j = R_j$ in the Schur–Wielandt deflation scheme. Suppose that

$$A \begin{bmatrix} U_j & V_m \end{bmatrix} = \begin{bmatrix} U_j & V_m \end{bmatrix} \begin{bmatrix} R_j & M_j \\ 0 & H_m \end{bmatrix} + f_{m+j}e_{m+j}^T \tag{8.2}$$

is a length $m + j$ Arnoldi factorization obtained after locking. Consider any associated round-off errors as being absorbed in A here. Equate the last m columns of equation (8.2) to obtain

$$AV_m = U_jM_j + V_mH_m + f_{m+j}e_m^T. \tag{8.3}$$

Since U_j is orthogonal to V_m, it follows that $(I - U_jU_j^T)A(I - U_jU_j^T)V_m = V_mH_m + f_{m+j}e_m^T$. This implies that the Arnoldi factorization (8.2) is equivalent to applying Arnoldi's method to the projected matrix $(I-U_jU_j^T)A(I-U_jU_j^T)$ with the first column of V_m as the starting vector. Keeping the locked vectors active in the construction and the IRA update of this Arnoldi factorization ensures that the Krylov space generated by V_m remains free of components corresponding to locked Ritz values. The appearance of spurious Ritz values in the subsequent factorization is automatically avoided. Note that when A is symmetric, this is equivalent to the selective orthogonalization scheme proposed by Parlett [26, pp. 275–284] and Scott.

In contrast to locking, consider the consequences of applying the Schur–Wielandt deflation scheme to construct a new Arnoldi factorization using V_me_1 as a starting vector. In the symmetric case with exact arithmetic, the two schemes would be mathematically equivalent. Without these assumptions, there may be considerable differences. From equation (8.3), it follows that (with A replaced with A_j of equation (8.1))

$$(A - U_jR_jU_j^T)V_m = A(I - U_jU_j^T)V_m = U_jM_j + V_mH_m + f_{m+j}e_m^T. \tag{8.4}$$

From equation (8.4) we can use an easy induction to derive the relations

$$(A - U_jR_jU_j^T)^iV_me_1 = (U_jM_j + V_mH_m)H_m^{i-1}e_1, \quad i \geq 1.$$

Thus, the Krylov subspace $\mathcal{K}_k(A - U_jR_jU_j^T, V_me_1)$ and hence the corresponding Arnoldi factorization of $A - U_jR_jU_j^T$ *must* be corrupted with components in $\mathcal{R}(U_j)$ when the starting vector is orthogonal to $\mathcal{R}(U_j)$. Within the context of Arnoldi iterations, the Schur–Wielandt technique does not deflate the invariant subspace information contained in the $\mathcal{R}(U_j)$ from the remainder of the iteration. In other words, Schur–Wielandt deflation is unstable.

This helps to explain why Saad suggests that Wielandt and Schur–Wielandt deflation techniques should not be used "to compute more than a few eigenvalues and eigenvectors" [31, p. 125]. We note that if $M_j \approx 0$, then the Wielandt forms of deflation may safely be used within an Arnoldi iteration. This will always be true when A is symmetric.

The cost of matrix–vector products with A_j increases due to the rank j modifications of A required. Moreover, every time an approximate Schur vector or a Ritz vector converges, the iteration needs to be explicitly restarted with A_j. The two deflation techniques introduced in this paper allow the iteration to be implicitly restarted—avoiding the need to build a new factorization from scratch.

Finally, we mention that the idea of deflating a converged Ritz value from a Lanczos iteration is also discussed by Parlett and Nour-Omid [28]. They present an explicit deflation technique by using the QR algorithm with converged Ritz values as shifts. Parlett indicates that this was a primary reason for undertaking the study concerning the forward instability of the QR algorithm [27].

9. Reordering the Schur form of a matrix. We now establish a connection between the IRA iteration with locking and the algorithms used to reorder the Schur form of a matrix. Suppose a matrix A is reduced to upper quasi-triangular form by the QR algorithm

$$Q^TAQ = T \equiv \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}, \tag{9.1}$$

where Q is the orthogonal matrix computed by the algorithm. Equation (9.1) is a Schur form for A of order $p+q$ where the submatrices T_{11} and T_{22} are of order p and q, respectively. Assume that the spectrums of T_{11} and T_{22} are distinct. In practice, the order in which the computed eigenvalues of A appear on the diagonal of T is somewhat random. The first p columns of Q are an orthogonal basis for the unique invariant subspace associated with the eigenvalues of T_{11}. If the eigenvalues of interest are located in T_{22} and an orthonormal basis for them is wanted, we must either increase the number of columns of Q used or somehow place them at the top of T. Algorithms for reordering a Schur form accomplish this task by using orthogonal matrices that move the wanted eigenvalues to the top of T. The recent work of Bai and Demmel [2] attempts to correct the occasional numerical problems encountered by Stewart's algorithm [35] EXCHNG. Their work was motivated by that of Ruhe [29] and that of Dongarra, Hammarling, and Wilkinson [11]. Both algorithms swap consecutive 1×1 and 2×2 blocks of a quasi-triangular matrix to attain the desired ordering.

Let both T_{11} and T_{22} of equation (9.1) be matrices of at most order-2. When swapping adjacent blocks of order-1, $p = 1 = q$, EXCHNG constructs a plane rotation that zeros the second component of the eigenvector corresponding to the eigenvalue $\lambda_2 = T_{22}$. A similarity transformation is performed on T with the plane rotation and the diagonal blocks are interchanged. We refer to a strategy that constructs an orthogonal matrix and performs a similarity transformation to interchange the eigenvalues as a *direct* swapping algorithm. Consider the following alternate *iterative* swapping algorithm: Perform a similarity transformation on T with an arbitrary orthogonal matrix followed by one step of the QR iteration with shift equal to λ_2. The arbitrary orthogonal similarity transformation introduces a nonzero off-diagonal element in the $2,1$ entry so that the transformed T is an unreduced upper Hessenberg matrix with the diagonal blocks now coupled. The standard convergence theory of the QR algorithm dictates that λ_1 and λ_2 are switched and the $2,1$ entry is zero. If the order of T_{22} is equal to two, EXCHNG uses the iterative swapping strategy using a standard double shift to reorder the diagonal blocks. The direct swapping algorithm instead computes an appropriate orthogonal matrix by computing the QR factorization of a basis of two vectors that span the desired invariant subspace. For example, the factorization used in equation (6.1) in §6.1 may be used. The reader is referred to [2, 11] for further details.

The iterative swapping algorithm is equivalent to the implicit restarting technique used by the IRA iteration, since both depend upon an implicitly shifted QR step applied to an unreduced upper Hessenberg matrix to interchange T_{11} and T_{22}. The direct swapping algorithm is equivalent to the locking technique. An orthogonal matrix is constructed from a basis for the invariant subspace corresponding to T_{22}. When this is applied as a similarity transformation, the diagonal blocks of T are swapped. In exact arithmetic, both swapping variants result in a matrix that is upper quasi-triangular with the blocks interchanged. Unfortunately, these existing reordering techniques do not preserve the leading portion of the Arnoldi factorization and thus explicit restarting would have to be used.

The following example demonstrates that the two variants may produce drastically different output matrices when computed in floating point arithmetic. The following experiment was carried out in MATLAB, Version 4.2a, on a SUN SPARC station IPX. The floating point arithmetic is IEEE standard double precision with machine precision of $\epsilon_M \equiv 2^{-52} \approx 2.2204 \cdot 10^{-16}$. Let

$$T = \begin{bmatrix} 1 + 10\epsilon_M & 1 \\ 0 & 1 \end{bmatrix}.$$

An eigenvector corresponding to $\lambda_2 = 1$ is

$$\begin{bmatrix} -1 \\ 10\epsilon_M \end{bmatrix}.$$

Denote by Z the plane rotation that transforms this eigenvector to a multiple of the first column of the identity matrix in $\mathbf{R}^{2\times 2}$. Let

$$U = \begin{bmatrix} 1 & -5\epsilon_M \\ 10\epsilon_M & 1 \end{bmatrix},$$

so that U is orthogonal up to a small multiple of machine precision. The matrix U acts as the arbitrary orthogonal transformation required by the iterative algorithm. Let $\hat{T}$ denote the matrix computed by performing one step of the QR iteration to the matrix U^TTU with shift equal to $\lambda_1 = 1 + 10\epsilon_M$. We remark that for matrices of order-2, the explicit and implicit formulations of the QR iteration are equivalent. The two computed matrices are

$$Z^TTZ = \begin{bmatrix} 1 & -1 \\ 0 & 1 + 10\epsilon_M \end{bmatrix},$$

$$\hat{T} = \begin{bmatrix} 1.400000000000003 & -7.999999999999996 \cdot 10^{-1} \\ 2.000000000000002 \cdot 10^{-1} & 6.000000000000001 \cdot 10^{-1} \end{bmatrix}.$$

The computed eigenvalues of $\hat{T}$ are 1.000000033320011 and $9.999999666799921 \cdot 10^{-1}$ which both lost eight digits of accuracy. If we perform another QR step on the matrix $\hat{T}$ with the same shift,

$$\begin{bmatrix} 1.000000000000003 & 1.000000000000001 \\ \approx 1.09 \cdot 10^{-15} & 1 \end{bmatrix}$$

is computed. Note that the off-diagonal element is slightly larger than machine precision so that a standard QR algorithm does not set it to zero. Moreover, even if the off-diagonal element is set to zero, the iterative swapping algorithm fails to interchange the eigenvalues. Continuing to apply QR steps with the shift equal to λ_1 does not result in a properly interchanged matrix.

The explanation of why the iterative algorithm fails to work is simple enough. The matrix T constructed is poorly conditioned with respect to the eigenvalue problem since the eigenvectors are nearly aligned. The eigenvalues of U^TTU are

$$1.000000033320011 \quad \text{and} \quad 9.999999666799921 \cdot 10^{-1}.$$

Thus, the small relative errors on the order of machine precision that occur when computing U^TTU produce a nearby matrix in which both eigenvalues differ by eight digits of accuracy. Performing a shifted QR step with λ_1 incurs forward instability since the last components of the eigenvectors for U^TTU are on the order of $\sqrt{\epsilon_M}$. This is the necessary and sufficient condition of Parlett and Le [27]. Another QR step with the same shift on $\hat{T}$ almost zeros out the subdiagonal element since the last components of the eigenvectors for $\hat{T}$ are of order 10^{-1} and the shift is almost the average of the eigenvalues of $\hat{T}$ and quite close to both. We emphasize that the loss of accuracy of the computed eigenvalues is one of the deleterious effects of forward instability.

Bai and Demmel [2] present an example which compares their direct swapping approach with Stewart's algorithm EXCHNG. The matrix considered is

$$A(\tau) = \begin{bmatrix} 7.001 & -87 & 39.4\tau & 22.2\tau \\ 5 & 7.001 & -12.2\tau & 36.0\tau \\ 0 & 0 & 7.01 & -11.7567 \\ 0 & 0 & 37 & 7.01 \end{bmatrix}.$$

When $\tau = 10$, ten QR iterations are required to interchange the two blocks. As before, the eigenvalues undergo a loss of accuracy. The iterative swapping algorithm fails for the matrix $A(100)$. No explanation is given for the failure of Stewart's algorithm. The explanation for the failure is the same as for the previous example. Using a direct algorithm, the eigenvalues of $A(10)$ and $A(100)$ are correctly swapped and the eigenvalues lose only a tiny amount of accuracy.

Bai and Demmel present a rigorous analysis of their direct swapping algorithm. Although backward stability is not guaranteed, it appears that only when both T_{11} and T_{22} are of order-2 and have almost indistinguishable eigenvalues [5] is stability lost. In this case, the interchange is not performed. Bojanczyk and Van Dooren [5] present an alternate swapping algorithm that appears to be backward stable.

10. Numerical results.

An IRA iteration using the two deflation procedures of §6.2 was written in MATLAB, Version 4.2a. An informal description given parameters k and p is given in Table 10.1. The codes are available from the first author upon request. A high-quality and robust implementation of the deflation procedures is planned for the Fortran software package ARPACK [24].

In the examples that follow, Q_k and R_k denote the approximate Schur factors for an invariant subspace of order-k computed by an IRA iteration. All the experiments used the starting vector equal to `randn(n, 1)`, where the seed is set with `randn('seed', 0)` and `n` is the order of the matrix. The shifting strategy uses the unwanted eigenvalues of H_{k+p} that have not converged. An eigenpair (θ, y) of H_{k+p} is accepted if its Ritz estimate (2.2) satisfies

$$|e_{k+p}^T y| \, \|f_{k+p}\| \leq \eta |\theta|. \tag{10.1}$$

The value of η is chosen according to the relative accuracy of the Ritz value desired.

10.1. Example 1.

The first example illustrates the use of the deflation techniques when the underlying matrix has several complex repeated eigenvalues. The example also demonstrates how the iteration locks and purges blocks of Ritz values in real arithmetic. A block diagonal matrix C was generated having n blocks of order-2. Each block was of the form

$$\begin{bmatrix} \xi_l & \eta_l \\ -\eta_l & \xi_l \end{bmatrix},$$

TABLE 10.1
Formal description of an IRA iteration.

```
1. Initialize an Arnoldi factorization of length $k$
2. Main Loop
   3. Extend an Arnoldi factorization to length $k+p$
   4. Check for convergence
      Exit if $k$ wanted Ritz values converge
      Let $i$ and $j$ denote the wanted and unwanted converged
      Ritz values, respectively
   5. Lock the $i+j$ converged Ritz values
   6. Implicit application of shifts resulting in an
      Arnoldi factorization of length $k+j$
   7. Purge the $j$ unwanted converged Ritz values.
```

where

$$\xi_{l=i+j-1} \equiv 4\sin^2\left(\frac{i\pi}{2(n+1)}\right) + 4\sin^2\left(\frac{j\pi}{2(n+1)}\right)$$

for $1 \le i, j \le n$, and $\eta_l \equiv \sqrt{\xi_l}$. The eigenvalues of C are $\xi_l \pm \eta_l i$ where $i = \sqrt{-1}$. Since the eigenvalues of a quasi-diagonal matrix are invariant under orthogonal similarity transformations, using an IRA iteration on C with a randomly generated starting vector is general. An IRA iteration was used to compute the $k = 12$ eigenvalues of C_{450} with smallest real part. The number of shifts used was $p = 16$ and the convergence tolerance η was set equal to 10^{-10}. With these choices of k and p, the iteration stores at most 28 Arnoldi vectors. There are four eigenvalues with multiplicity two. Table 10.2 shows the results attained. Let the diagonal matrix D_{12} denote the eigenvalues of the upper triangular matrix R_{12} computed by the iteration. The diagonal matrix Λ_{12} contains the wanted eigenvalues. After 24 iterations, 12 Ritz values converged. But the pair of Ritz values purged at iteration 21 was a previously locked value which the iteration discarded. This behavior is typical when there are clusters of eigenvalues.

10.2. Example 2. Consider the eigenvalue problem for the convection–diffusion operator

$$-\Delta u(x,y) + \rho(u_x(x,y) + u_y(x,y)) = \lambda u(x,y)$$

on the unit square $[0,1] \times [0,1]$ with zero boundary data. Using a standard five-point scheme with centered finite differences, the matrix L_{n^2} that arises from the discretization is of order n^2 where $h = 1/(n+1)$ is the cell size. The eigenvalues of L_{n^2} are

$$\lambda_{ij} = 2\sqrt{1-\gamma}\cos\left(\frac{i\pi}{n+1}\right) + 2\sqrt{1-\gamma}\cos\left(\frac{j\pi}{n+1}\right)$$

for $1 \le i, j \le n$ where $\gamma = \rho h/2$. An IRA iteration was used to compute the $k = 6$ smallest eigenvalues of L_{625} where $\rho = 25$. The number of shifts used was $p = 10$ and the convergence tolerance η was set equal to 10^{-8}. With these choices of k and p, the iteration stores at most 16 Lanczos vectors. Let the diagonal matrix D_6 denote the eigenvalues of the upper triangular matrix R_6 computed by the iteration. The diagonal matrix $\Lambda_6 \in \mathbf{R}^{6\times 6}$ contains the six smallest eigenvalues. We note that there are two eigenvalues with multiplicity two. Table 10.3 shows the results attained. The diagonal matrix D_6 approximates Λ_6. After 30 iterations six Ritz values converged.

TABLE 10.2
Convergence history for Example 1.

IRA iteration for C_{450}		
$k = 12$ and $p = 16$ with convergence tolerance $\eta = 10^{-10}$		
Iteration	Ritz values Locked	Ritz values Purged
9	2	0
10	2	0
12	2	0
13	2	0
17	2	0
21	0	2
24	2	0
28	0	2
31	2	0
Totals	14	4
Number of matrix–vector products		436
$\|C_{450}Q_{12} - Q_{12}R_{12}\| \approx 10^{-12}$		
$\|Q_{12}^T C_{450}Q_{12} - R_{12}\| \approx 10^{-11}$		
$\|Q_{12}^T Q_{12} - I_{12}\| \approx 10^{-14}$		
$\|D_{12} - \Lambda_{12}\|_\infty \approx 10^{-15}$		

But the Ritz value purged at iteration 24 was a previously locked value. The other purged Ritz values are approximations to the eigenvalues of L_{625} larger than λ_6.

Figure 10.1 gives a graphical interpretation of the expense of an IRA iteration in terms of matrix–vector products when the value of p is increased. For all values of p shown, the results of the iteration were similar to those of Table 10.3. The results presented in Table 10.3 correspond to the value of p that gave the minimum number of matrix–vector products. For the value of $p = 1$, the iteration converged to the five smallest eigenvalues after 999 matrix–vector products. But the iteration was not able to converge to the second copy of λ_5. For $p = 2$, the only form of deflation employed was locking. All other values of p shown demonstrated similar behavior to that of Table 10.3.

In order to determine the benefit of the two deflation techniques, experiments were repeated without the use of locking or purging. In addition, all the unwanted Ritz values were used as shifts, converged or not. The first run used the same parameters as given in Table 10.3. After 210 matrix–vector products, the iteration converged to six Ritz values. But the second copy of the fifth smallest eigenvalue was not among the final six. The value of p was increased to 23 with the same results.

10.3. Example 3. The following example shows the behavior of the iteration on a matrix with a very ill-conditioned basis of eigenvectors. Define the Clement tridiagonal matrix [20] of order $n + 1$:

$$B_{n+1} = \begin{bmatrix} 0 & n & \cdots & 0 \\ 1 & 0 & n-1 & \\ \vdots & \ddots & \ddots & \\ 0 & & n & 0 \end{bmatrix}.$$

TABLE 10.3
Convergence history for Example 2.

IRA iteration on L_{625}		
$k = 6$ and $p = 10$ with convergence tolerance $\eta = 10^{-8}$		
Iteration	Ritz values Locked	Ritz values Purged
14	1	0
16	1	0
19	1	0
21	1	0
23	1	1
24	0	1
30	1	0
35	0	1
38	1	1
Totals	7	4
Number of matrix–vector products		325
$\|\|L_{625}Q_6 - Q_6R_6\|\| \approx 10^{-9}$		
$\|\|Q_6^TL_{625}Q_6 - R_6\|\| \approx 10^{-9}$		
$\|\|Q_6^TQ_6 - I_6\|\| \approx 10^{-14}$		
$\|\|D_6 - \Lambda_6\|\|_\infty \approx 10^{-7}$		

The eigenvalues are $\pm n, \pm n - 2, \ldots, \pm 1$, and zero if n is even. We note that $B_{n+1} = S_{n+1}A_{n+1}S_{n+1}^{-1}$ where $S_{n+1}^2 = \mathrm{diag}(1, \frac{n}{1}, \frac{n}{1}\frac{n-1}{2}, \ldots, \frac{n!}{n!})$ is a diagonal matrix. Thus the condition number of the basis of eigenvectors for B_{n+1} is $\|S_{n+1}\|\,\|S_{n+1}^{-1}\|$ which implies that the eigenvalue problem for B_{n+1} is quite ill conditioned. An IRA iteration was used to compute the $k = 4$ largest in magnitude eigenvalues of B_{1000}. The number of shifts used was $p = 16$, and the convergence tolerance η was set equal to 10^{-6}. With these choices of k and p, the iteration stores at most 20 Arnoldi vectors. Let the diagonal matrix D_4 denote the eigenvalues of the upper triangular matrix R_4 computed by the iteration. The diagonal matrix $\Lambda_4 \in \mathbf{R}^{4\times 4}$ contains the four largest in magnitude eigenvalues. Table 10.4 shows the results attained. Although the iteration needed a large number of matrix–vector products, the iteration was able to extract accurate Ritz values given the convergence tolerance.

10.4. Example 4. Finally, we present a dramatic example of how the convergence of an IRA iteration benefits from the two deflation procedures. A matrix T of order-10 had the values

$$\tau_1 = 10^{-6}, \tau_{i=2:8} = i \cdot 10^{-3}, \tau_{9:10} = 1$$

on the diagonal. Since the eigenvalues of a matrix are invariant under orthogonal similarity transformations, using an IRA iteration on T with a randomly generated starting vector is general. An IRA iteration was used to compute an approximation to the smallest eigenvalue. The number of shifts used was $p = 3$ and the convergence tolerance η was set equal to 10^{-3}. Table 10.5 shows the results attained. Another experiment was run with the locking and purging mechanisms turned off. Additionally, all unwanted Ritz values were used as shifts. The same parameters were used as in Table 10.5 but the iteration now consumed 41 matrix–vector products. As in

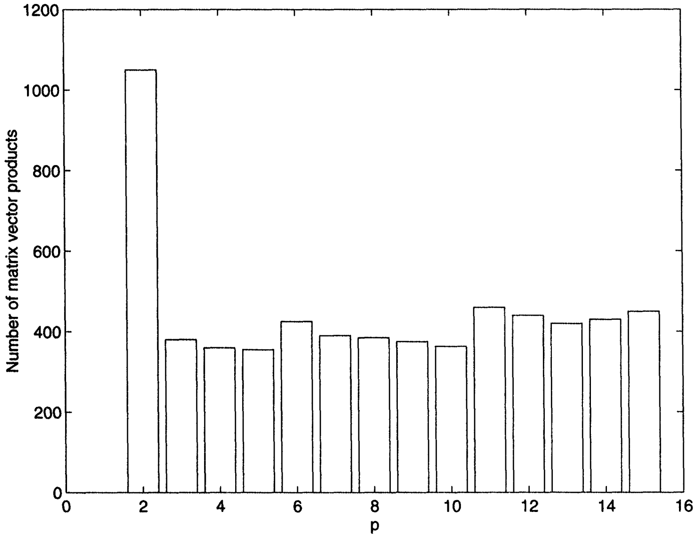

FIG. 10.1. *Bar graph of the number of matrix–vector products used by an IRA iteration for Example* 2 *as a function of p.*

TABLE 10.4
Convergence history for Example 3.

IRA iteration on B_{1000}		
$k = 4$ and $p = 16$ with convergence tolerance $\eta = 10^{-6}$		
Iteration	Ritz values Locked	Ritz values Purged
76	1	0
85	1	0
91	2	0
Totals	4	0
Number of matrix–vector products	1423	
$\|B_{1000}Q_4 - Q_4R_4\|/\|B_{1000}\| \approx 10^{-6}$		
$\|Q_4^T B_{1000}Q_4 - R_4\| \approx 10^{-6}$		
$\|Q_4^T Q_4 - I_4\| \approx 10^{-14}$		
$\|D_4 - \Lambda_4\|_\infty/\|B_{1000}\|_\infty \approx 10^{-6}$		

the results for Table 10.5, the modified iteration converged to one of the dominant eigenvalues after one iteration. After six iterations, the leading block of H_4 split off, having converged to the invariant subspace corresponding to $\tau_{9:10}$. But since purging was turned off, the modified iteration had to continue attempting to converge to τ_1 using only the lower block of order-2 in H_4. Incidentally, if the iteration instead simply discarded the leading portion of the factorization corresponding to $\tau_{9:10}$ after the sixth iteration, convergence to τ_1 never occurred. Crucial to the success of an IRA iteration is the ability to deflate converged Ritz values in a stable manner. Both purging and locking allow faster convergence.

11. Conclusions. In the paper, we developed deflation techniques for an implicitly restarted Arnoldi iteration. The first technique, locking, allows an orthogonal

TABLE 10.5
Convergence history for Example 4.

IRA iteration on T		
$k = 1$ and $p = 3$ with convergence tolerance $\eta = 10^{-3}$		
Iteration	Ritz values Locked	Ritz values Purged
1	0	1
15	1	1
Totals	1	2
Number of matrix–vector products		32
$\Vert TQ_1 - Q_1R_1\Vert/\tau_1 \approx 10^{-3}$		
$\Vert Q_1^TTQ_1 - R_1\Vert/\tau_1 \approx 10^{-3}$		
$\Vert Q_1^TQ_1 - I_1\Vert \approx 10^{-15}$		
$\Vert R_1 - \tau_1\Vert_\infty/\tau_1 \approx 10^{-3}$		

change of basis for an Arnoldi factorization which results in a partial Schur decomposition containing the converged Ritz values. The corresponding Ritz value is deflated in an implicit but direct manner. The second technique, purging, allows implicit removal of unwanted converged Ritz values from the Arnoldi iteration. Both deflation techniques are accomplished by working with matrices in the projected Krylov space which for large eigenvalue problems is a fraction of the order of the matrix from which estimates are sought. Since both deflation techniques are implicitly applied to the Arnoldi factorization the need for explicit restarting associated with all other deflation strategies is avoided. Both techniques were carefully examined with respect to numerical stability and computational results were presented. Convergence of the Arnoldi iteration is improved and a reduction in computational effort is realized. Although a direct comparison with block Arnoldi/Lanczos methods was not given, computational experience shows that if an IRA iteration builds the same size factorization used by the block methods and the convergence tolerance is small enough, multiple or clustered eigenvalues are correctly computed. The connection between an IRA and QR iteration explains the reason for the size of the convergence tolerance used.

REFERENCES

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

[2] Z. BAI AND J. W. DEMMEL, *On swapping diagonal blocks in real Schur form*, Linear Algebra Appl., 186 (1993), pp. 73–95.

[3] R. H. BARTELS AND G. W. STEWART, *Algorithm 432: Solution of the matrix equation $AX + XB = C$*, Comm. ACM, 15 (1972), pp. 820–826.

[4] M. BENNANI AND T. BRACONNIER, *Stopping Criteria for Eigensolvers*, Tech. report TR/PA/93/25, CERFACS, Toulouse, France, 1993.

[5] A. BOJANCZYK AND P. VAN DOOREN, *Reordering diagonal blocks in the Schur form*, in Linear Algebra for Large Scale and Real Time Applications, NATO ASI Series, Kluwer Academic Publishers, Norwell, MA, 1993, pp. 351–352.

[6] T. BRACONNIER, *The Arnoldi–Tchebycheff Algorithm for Solving Large Nonsymmetric Eigenproblems*, Tech. report TR/PA/93/25, CERFACS, Toulouse, France, 1993.

[7] D. CALVETTI, L. REICHEL, AND D. C. SORENSEN, *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, Electron. Trans. Numer. Anal., 2 (1994), pp. 1–21.

[8] F. CHATELIN, *Eigenvalues of Matrices*, John Wiley, New York, 1993.

[9] F. CHATELIN AND V. FRAYSÉE, *Qualitative Computing: Elements of a Theory for Finite-Precision Computation*, Tech. report, CERFACS and THOMSON–CSF, June 1993. Lecture Notes for the Commett European Course, June 8–10, Orsay, France.

[10] J. CULLUM AND W. E. DONATH, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace for large, sparse symmetric matrices*, in Proc. 1974 IEEE Conference on Decision and Control, New York, 1974, pp. 505–509.

[11] J. DONGARRA, S. HAMMARLING, AND J. WILKINSON, *Numerical considerations in computing invariant subspaces*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 145–161.

[12] I. S. DUFF AND J. A. SCOTT, *Computing selected eigenvalues of large sparse unsymmetric matrices using subspace iteration*, ACM Trans. Math. Software, 19 (1993), pp. 137–159.

[13] W. FELLER AND G. FORSYTHE, *New matrix transformations for obtaining characteristic vectors*, Quart. Appl. Math., 8 (1951), pp. 325–331.

[14] S. GODET-THOBIE, *Eigenvalues of Large Highly Nonnormal Matrices*, Ph.D. thesis, University Paris IX, Dauphine, Paris, France, 1993; Tech. report TH/PA/93/06, CERFACS, Toulouse, France, 1993.

[15] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[16] G. H. GOLUB, S. NASH, AND C. F. VAN LOAN, *A Hessenberg–Schur method for the problem* $AX + XB = C$, IEEE Trans. Automat. Control, AC-24 (1979), pp. 909–913.

[17] G. H. GOLUB AND R. UNDERWOOD, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, pp. 361–377.

[18] R. G. GRIMES, J. G. LEWIS, AND H. D. SIMON, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.

[19] N. J. HIGHAM, *Perturbation theory and backward error for* $AX - XB = C$, BIT, 33 (1993), pp. 124–136.

[20] ———, *The Test Matrix Toolbox for Matlab*, Numerical Analysis Report 237, University of Manchester, England, 1993.

[21] W. KARUSH, *An iterative method for finding characteristics vectors of a symmetric matrix*, Pacific J. Math., 1 (1951), pp. 233–248.

[22] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.

[23] R. B. LEHOUCQ, *Analysis and Implementation of an Implicitly Restarted Iteration*, Ph.D. thesis, Rice University, Houston, TX, May 1995; Tech. report TR95-13, Dept. of Computational and Applied Mathematics, Rice University, Houston, TX, 1993.

[24] R. B. LEHOUCQ, D. C. SORENSEN, AND P. VU, *ARPACK:* An Implementation of the Implicitly Re-started Arnoldi Iteration That Computes Some of the Eigenvalues and Eigenvectors of a Large Sparse Matrix, 1995. Available from netlib@ornl.gov under the directory scalapack.

[25] C. C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, University of London, London, England, 1971.

[26] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[27] B. N. PARLETT AND J. LE, *Forward instability of tridiagonal QR*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 279–316.

[28] B. N. PARLETT AND B. NOUR-OMID, *The use of a refined error bound when updating eigenvalues of tridiagonals*, Linear Algebra Appl., 68 (1984), pp. 179–219.

[29] A. RUHE, *An algorithm for numerical determination of the structure of a general matrix*, BIT, 10 (1970), pp. 196–216.

[30] Y. SAAD, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.

[31] ———, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, New York, 1992.

[32] M. SADKANE, *A block Arnoldi–Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numer. Math., 64 (1993), pp. 181–193.

[33] J. A. SCOTT, *An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices*, ACM Trans. Math. Software, 21 (1995), pp. 432–475.

[34] D. C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[35] G. W. STEWART, *ALGORITHM* 506: *HQR*3 *and EXCHANG: Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenberg matrix* [F2], ACM Trans. Math. Software, 2 (1976), pp. 275–280.

[36] ———, *Perturbation bounds for the QR factorization of a matrix*, SIAM J. Numer. Anal., 14

(1977), pp. 509–518.

[37] W. Stewart and A. Jennings, *A simultaneous iteration algorithm for real matrices*, ACM Trans. Math. Software, 7 (1981), pp. 184–198.

[38] J. M. Varah, *On the separation of two matrices*, SIAM J. Numer. Anal., 16 (1979), pp. 216–222.

[39] D. S. Watkins, *Forward stability and transmission of shifts in the QR algorithm*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 469–487.

[40] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.