Deflation for Implicitly Restarted Arnoldi Methods

D.C. Sorensen

CRPC-TR98775
May 1998

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Submitted September 1998; Available as Rice CAAM TR98-12

DEFLATION FOR IMPLICITLY RESTARTED ARNOLDI METHODS [*]

D. C. SORENSEN[†]

Abstract. The implicitly restarted Arnoldi method (IRAM) is an effective technique for computing a selected subset of the eigenvalues and corresponding eigenvectors of a large matrix **A**. However, the performance of this method can be improved considerably with the introduction of appropriate deflation schemes to isolate approximate invariant subspaces associated with converged Ritz values. These deflation strategies make it possible to compute multiple or clustered eigenvalues with a single vector implicit restart method.

It is of particular interest to provide schemes that can deflate with user specified relative error tolerances ϵ_D that are considerably greater than working precision ϵ_M. The primary contribution of this paper is to develop efficient and numerically stable schemes for this purpose. Two forms of deflation are presented. The first, a *locking* operation, decouples converged Ritz values and the associated invariant subspace from the active part of the IRAM iteration. The second, a *purging* operation, removes unwanted but converged Ritz pairs. Convergence of the IRAM iteration is improved and a reduction in computational effort is also achieved.

Key words. eigenvalues, deflation, implicit restarting, Krylov projection methods, Arnoldi method, Lanczos method

AMS subject classifications. Primary 65F15, Secondary 65G05

1. Introduction.

The implicitly restarted Arnoldi method IRAM is an efficient procedure for approximating a selected subset of the eigenvalues and corresponding eigenvectors of a large sparse or structured $n \times n$ matrix **A**. Implicitly restarting [7] enables the Arnoldi process to compute this selected subset within a pre-determined and relatively small amount of storage. This is the underlying algorithm in the large scale eigenvalue package ARPACK [3]. The method may be viewed as a truncation of the standard implicitly shifted QR-iteration. Through this connection, the IRAM shares a number of desirable properties with the QR-iteration. These include some well understood deflation rules that are extremely important with respect to convergence and stability. These deflation rules are essential for the QR-iteration to efficiently compute multiple or clustered eigenvalues. While these existing QR deflation rules are applicable to IRAM, they are not the most effective schemes possible. The purpose of this paper is to develop new deflation schemes that are better suited to implicit restarting.

In the large scale setting, it is highly desirable to provide schemes that can deflate with user a specified relative error tolerances ϵ_D that are considerably greater than working precision ϵ_M. Without this capability, excessive and unnecessary computational effort is often required to detect and deflate converged approximate eigenvalues. The ability to deflate at relaxed tolerances provides an effective way to compute multiple or clustered eigenvalues with a single-vector implicitly restarted Arnoldi method. The primary contribution of this paper is to develop efficient and numerically stable methods for these purposes.

Two forms of deflation are presented. The first, a *locking* operation, decouples converged approximate eigenvalues and associated invariant subspaces from the active

[*] This work was supported in part by NSF cooperative agreement CCR-9120008, NSF contract ASC-9408795 and by ARPA contract number DAAL03-91-C-0047 (administered by the U.S. Army Research Office).

[†] Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005-1892, (sorensen@caam.rice.edu).

part of the IRAM iteration. The second, a *purging* operation, removes unwanted but converged eigenpairs. With the aid of these deflation schemes, convergence of the IRAM iteration is improved and a reduction in computational effort is also achieved. These notions and appropriate methods were developed previously in [1, 2]. The new techniques developed here improve upon those deflation schemes. These new schemes provide the means to deflate at very relaxed tolerances ϵ_D. This new capability is achieved through new orthogonal transformations that structure the way deflation error can influence the remaining Arnoldi process.

This paper has the following organization. The fundamentals of the Arnoldi method and implicit restarting are briefly reviewed in § 2. We introduce an apparently new family of orthogonal transformations in § 4 that greatly improve the efficiency and stability of the deflation schemes we shall develop. Deflating a single converged Ritz value is examined in § 5. A real-arithmetic scheme for deflating a converged complex conjugate pair of approximate eigenvalues of a real matrix is presented in § 6. Brief error analyses of the numerical behavior of the new orthogonal transformations and of the deflation process in the context of the full IRAM iteration are presented in § 7. Numerical results and conclusions are presented in § 8.

Bold face capital and lower case letters denote matrices and vectors while lower case Greek letters denote scalars. The j-th canonical basis vector is denoted by $\mathbf{e}_j$. The Euclidean norm is used exclusively and this is denoted by $\|\cdot\|$.

2. Arnoldi's Method and Implicit Restarting. The Arnoldi process underlies practical schemes for implementing Krylov subspace projection methods for both eigenvalue problems and linear systems. Technically, the method amounts to nothing more than reducing a square matrix $\mathbf{A}$ to condensed form through an orthogonal similarity transformation. Unlike Householder or Given's method, the Arnoldi reduction proceeds one column at a time from left to right. After k-steps, one has

$$\mathbf{AV} = \mathbf{VH} + \mathbf{f}\mathbf{e}_k^T$$

where $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$, $\mathbf{V}^T\mathbf{f} = 0$ and $\mathbf{H}$ is a $k \times k$ upper Hessenberg matrix. The columns of $\mathbf{V}$ form an ortho-normal basis for the Krylov subspace

$$\mathcal{K}(\mathbf{A}, \mathbf{v}_1) \equiv Span\{\mathbf{v}_1, \mathbf{A}\mathbf{v}_1, \cdots \mathbf{A}^{k-1}\mathbf{v}_1\}.$$

The matrix $\widehat{\mathbf{A}} \equiv \mathbf{VHV}^T$ is the orthogonal projection of $\mathbf{A}$ onto this space. Ritz values and vectors are defined by a Galerkin condition: A vector $\mathbf{x} \in \mathcal{K}_k(\mathbf{A}, \mathbf{v}_1)$ is called a *Ritz vector* with corresponding *Ritz value* θ if the Galerkin condition

$$\langle \mathbf{w}, \mathbf{Ax} - \mathbf{x}\theta \rangle = 0 \ , \quad \text{for all} \quad \mathbf{w} \in \mathcal{K}_k(\mathbf{A}, \mathbf{v}_1)$$

is satisfied. It is easy to check that $\mathbf{x}, \theta$ is a Ritz pair if and only if $\mathbf{Hy} = \mathbf{y}\theta$ and $\mathbf{x} = \mathbf{Vy}$. The corresponding residual is given by

$$\mathbf{Ax} - \mathbf{x}\theta = \mathbf{f}\mathbf{e}_k^T\mathbf{y}$$

indicating that every Ritz residual is in the same direction and that

$$\|\mathbf{Ax} - \mathbf{x}\theta\| = \|\mathbf{f}\||\mathbf{e}_k^T\mathbf{y}|.$$

Thus the norm of the residual error is available without an additional reference to the matrix $\mathbf{A}$.

```
**Algorithm 1**: (IRAM) Implicitly Restarted Arnoldi Method

  **Input**:  $(\mathbf{A}, \mathbf{V}, \mathbf{H}, \mathbf{f})$ with $\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$,
              an $m$-Step Arnoldi Factorization;
  **Output**: $(\mathbf{V}_k, \mathbf{H}_k)$ such that $\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k$, $\mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$,
              and $\mathbf{H}_k$ is upper triangular.
  1. for $\ell = 1, 2, 3, ...$ until convergence
     1.1. Compute $\sigma(\mathbf{H}_m)$ and select set of $p$ shifts $\mu_1, \mu_2, ...\mu_p$
          based upon $\sigma(\mathbf{H}_m)$ or perhaps other information;
     1.2. $\mathbf{Q} = \mathbf{I}_m$
     1.3. for $j = 1, 2, ..., p$,
              1.3.1. Factor $[\mathbf{Q}_j, \mathbf{R}_j] = \mathrm{q}r(\mathbf{H}_m - \mu_j\mathbf{I})$;
              1.3.2. $\mathbf{H}_m \leftarrow \mathbf{Q}_j^H\mathbf{H}_m\mathbf{Q}_j$;   $\mathbf{Q} \leftarrow \mathbf{Q}\mathbf{Q}_j$;
          end_for

     1.4. $\hat{\beta}_k = \mathbf{H}_m(k+1, k)$;  $\sigma_k = \mathbf{Q}(m, k)$;
     1.5. $\mathbf{f}_k \leftarrow \mathbf{v}_{k+1}\hat{\beta}_k + \mathbf{f}_m\sigma_k$;

     1.6. $\mathbf{V}_k \leftarrow \mathbf{V}_m\mathbf{Q}(:, 1:k)$;   $\mathbf{H}_k \leftarrow \mathbf{H}_m(1:k, 1:k)$;

     1.7. Beginning with the $k$-step Arnoldi factorization
             $\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$,
             apply $p$ additional steps of the Arnoldi process
             to obtain a new $m$-step Arnoldi factorization
             $\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$ .
     end_for
```

FIG. 2.1. *An Implicitly Restarted Arnoldi Method (IRAM).*

Implicit Restarting is a technique for updating a sequence of k-step Arnoldi factorizations in a way that forces desired eigenvalues to appear in the spectrum of the leading $k \times k$ Hessenberg matrix $\mathbf{H}_k$. The basic iteration is described in Fig. 2.1. In that algorithm, the leading k columns $\mathbf{V}_k$ of the Arnoldi basis vectors and the leading $k \times k$ Hessenberg matrix $\mathbf{H}_k$ are precisely the same quantities that would appear in the leading k columns of $\mathbf{V}$ and the leading $k \times k$ submatrix of the Hessenberg matrix $\mathbf{H}$ if the same shifts were selected and applied to update the relation $\mathbf{AV} = \mathbf{VH}$ with steps of the full implicitly shifted QR iteration.

Deflation is an important concept in the practical implementation of the QR iteration. However, in this large scale setting, the usual QR deflation techniques are not always appropriate. There are situations that call for additional deflation capabilities specific to implicit restarting. In particular, it is highly desirable to have the ability to deflate at an accuracy level ϵ_D with $1 > \epsilon_D > \epsilon_M$ where ϵ_M is machine precision. There some important reasons for this. In theory (i.e. in exact arithmetic), when $\mathbf{A}$ has multiple eigenvalues it would be impossible for IRAM to compute more than one instance of this multiplicity. This is because it is a "single vector" rather than a block method. However, in practice, there is usually little difficulty in computing multiple eigenvalues because the method deflates itself as convergence takes place and round-off usually introduces components in new eigenvector directions in the subsequent starting vectors. Nevertheless, this can be unreliable and miss a multiple instance. And, in any case, this approach requires a stringent convergence tolerance to succeed in finding all of the multiplicities. It is far more efficient to deflate (i.e. lock) an approximate eigenvalue once it has converged to a certain level of accuracy and then force subsequent Arnoldi vectors to be orthogonal to the converged subspace. With this capability, additional instances of a multiple eigenvalue can be computed to the same specified accuracy without the expense of converging them to unnecessarily high accuracy.

3. Deflation for Implicit Restarting.

In the standard implicitly shifted QR iteration, it is common to associate convergence of eigenvalues with small or zero subdiagonal elements of $\mathbf{H}$. Deflation rules are associated with setting small subdiagonal elements to zero in a numerically stable manner. They are designed to assure that the computed eigenvalues of the problem that results from setting a small subdiagonal to zero will be exact eigenvalues of a problem that is an acceptably small perturbation to the original problem.

In the Arnoldi process, subdiagonal elements β_j of $\mathbf{H}$ are norms of residual vectors $||\mathbf{f}_j||$. As with a QR iteration, it is possible for some of the leading k subdiagonals to become small during the course of implicit restarting. However, it is usually the case that there are converged Ritz values appearing in the spectrum of $\mathbf{H}$ long before small subdiagonal elements appear. This convergence is usually detected through observation of a small last component in an eigenvector $\mathbf{y}$ of $\mathbf{H}$.

It turns out that in the case of a small last component of $\mathbf{y}$, there is an orthogonal similarity transformation of $\mathbf{H}$ that will give an equivalent Arnoldi factorization with a slightly perturbed $\mathbf{H}$ that does indeed have a zero subdiagonal and this is the basis of our deflation schemes. A technique for doing this was developed in [1, 2], but this scheme requires the Ritz error estimates $||\mathbf{f}|||\mathbf{e}_k^T\mathbf{y}| < ||\mathbf{H}||{\epsilon_M}^{3/2}$ before the deflation can take place without unacceptable perturbations to the deflated $\mathbf{H}$. Thus, while those methods are effective, they can still require considerably more work than necessary due to the stringent tolerance.

This work introduces a new, but related, technique that does allow for stable and efficient deflation (or locking) of Ritz values that have converged with a specified relative accuracy of ϵ_D which may be considerably larger than machine precision ϵ_M. This is particularly important when a shift-invert spectral transformation is not available to accelerate convergence. Typically, in this setting the number of matrix-vector products will be large and it will be highly desirable to lock converged Ritz values at low tolerances to avoid the expense of the matrix-vector products that would be required to achieve accuracy that would allow normal QR type deflation. Also, it is very important to be able to purge converged but unwanted Ritz values. As Lehoucq pointed out [1], the forward instability of the QR bulge-chase process discovered by Parlett and Le [5] will prevent implicit restarting to be used for purging converged unwanted Ritz values.

4. Some Useful Orthogonal Transformations.

In this section we develop a family of orthogonal transformations that will be useful for implementation of our deflation schemes. As in [1, 2] the deflation is related to an eigenvector associated with a Ritz value that is to be deflated. The following lemma explains how to construct an orthogonal matrix that can be used to accomplish the deflation.

LEMMA 4.1. *Let $\mathbf{y}^T = (\eta_1, \eta_2, \ldots, \eta_k)$ be a k dimensional vector of unit norm ($\|\mathbf{y}\| = 1$) and define $\mathbf{y}_j^T = (\eta_1, \eta_2, \ldots, \eta_j)$. Let $\tau_j = \|\mathbf{y}_j\|$ for $2 \le j \le n$, and define*

$$\mathbf{q}_j = \begin{bmatrix} \mathbf{y}_{j-1}\gamma_j \\ \rho_j \\ 0 \end{bmatrix} \quad \text{where} \ \ \gamma_j \equiv \frac{-\eta_j}{\tau_{j-1}\tau_j}, \quad \rho_j \equiv \frac{\tau_{j-1}}{\tau_j}.$$

If $\mathbf{Q} \equiv [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]$ with $\mathbf{q}_1 = \mathbf{y}$ then $\mathbf{Q}$ is orthogonal ($\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$) with $\mathbf{Q}\mathbf{e}_1 = \mathbf{y}$ and with $\mathbf{e}_k^T\mathbf{Q} = [\eta_k, \tau_{k-1}\mathbf{e}_{k-1}^T]$.

Proof. First, observe that $\mathbf{q}_i^T\mathbf{q}_j = \mathbf{q}_i^T\mathbf{y}\gamma_j$ if $i < j$. Thus, it is sufficient to show that $\mathbf{q}_i^T\mathbf{y} = 0$ for $i = 2, 3, \ldots, k$. This follows easily from the definitions of γ_i and ρ_i, since

$$\begin{aligned} \mathbf{q}_i^T\mathbf{y} &= \mathbf{y}_{i-1}^T\mathbf{y}_{i-1}\gamma_i + \eta_i\rho_i \\ &= \tau_{i-1}^2\left(\frac{-\eta_i}{\tau_{i-1}\tau_i}\right) + \eta_i\frac{\tau_{i-1}}{\tau_i} \\ &= 0, \quad \text{for } 2 \le i \le n. \end{aligned}$$

Moreover, $\mathbf{q}_i^T\mathbf{q}_i = 1$, since

$$\begin{aligned} \mathbf{q}_i^T\mathbf{q}_i &= \gamma_i^2\mathbf{y}_{i-1}^T\mathbf{y}_{i-1} + \rho_i^2 \\ &= \left(\frac{-\eta_i}{\tau_{i-1}\tau_i}\right)^2\tau_{i-1}^2 + \left(\frac{\tau_{i-1}}{\tau_i}\right)^2 \\ &= \frac{\eta_i^2 + \tau_{i-1}^2}{\tau_i^2} \\ &= 1, \quad \text{for } 2 \le i \le n. \end{aligned}$$

and the lemma is proved □

The orthogonal matrix $\mathbf{Q}$ constructed as prescribed in Lemma(4.1) may be written as

$$\mathbf{Q} = \mathbf{R} + \mathbf{y}\mathbf{e}_1^T, \quad \text{with} \ \ \mathbf{R}\mathbf{e}_1 = 0, \ \mathbf{R}^T\mathbf{y} = 0, \tag{4.1}$$

where $\mathbf{R}$ is upper triangular. It may also be written as

$$\mathbf{Q} = \mathbf{L} + \mathbf{y}\mathbf{g}^T, \quad \text{with} \ \ \mathbf{L}\mathbf{e}_1 = 0, \ \mathbf{L}^T\mathbf{y} = \mathbf{e}_1 - \mathbf{g}, \tag{4.2}$$

where $\mathbf{L}$ is lower triangular, and $\mathbf{g}^T \equiv \mathbf{e}_1^T + \mathbf{e}_1^T\mathbf{R}/\eta_1$. Moreover, if $\mathbf{S}_L$ is the left-circular-shift operator (i.e., $\mathbf{x}^T\mathbf{S}_L = (\xi_2, \xi_3, \ldots, \xi_k, \xi_1)$ for any $\mathbf{x}^T = (\xi_1, \xi_2, \ldots, \xi_k)$), then

$$\mathbf{U} \equiv \mathbf{Q}\mathbf{S}_L \ \ \text{satisfies} \ \ \mathbf{U}\mathbf{e}_k = \mathbf{y}, \ \ \text{and} \ \ \mathbf{U}^T\mathbf{U} = \mathbf{I}. \tag{4.3}$$

It is easily seen that $\mathbf{U}$ is upper Hessenberg and orthogonal.

An algorithm based upon Lemma 4.1 for computing the orthogonal matrix $\mathbf{Q} = \mathbf{Q}(\mathbf{y})$ is is presented in Figure 4.1. A simple indexing modification to Algorithm 4.1

```
function [Q] = orthQ(y);

Input: y of dimension $k$ with $||\mathbf{y}|| = 1$ and $\eta = \mathbf{e}_k^T\mathbf{y}$.
Output: Q such that $\mathbf{Q}\mathbf{e}_1 = \mathbf{y}$, $\mathbf{e}_k^T\mathbf{Q} = (\eta, \tau\mathbf{e}_{k-1}^T)$,
        with $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$.

    1. $\mathbf{Q} = \mathbf{0}$;  $\mathbf{Q}(:,1) = \mathbf{y}$;
    2. $\sigma = \mathbf{y}(1)^2$;  $\tau_o = |\mathbf{y}(1)|$;
    3. for $j = 2 : n$,
         3.1. $\sigma \leftarrow \sigma + \mathbf{y}(j) \cdot \mathbf{y}(j)$;
         3.2. $\tau = \sqrt{\sigma}$;
         3.4. if ( $\tau_o \neq 0$),
                   3.4.1. $\gamma = (\mathbf{y}(j)/\tau)/\tau_o$;
                   3.4.2. $\mathbf{Q}(1:j-1,j) = -\mathbf{y}(1:j-1)\gamma$;
                   3.4.3. $\mathbf{Q}(j,j) = \tau_o/\tau$;
              else
                   3.4.4. $\mathbf{Q}(j-1,j) = 1$;
              end_if
         3.5. $\tau_o = \tau$;
    4. end_for;
```

FIG. 4.1. *Computation of a Special Orthogonal Matrix* $\mathbf{Q}$

will compute the orthogonal upper Hessenberg matrix $\mathbf{U} = \mathbf{U}(\mathbf{y})$ with $\mathbf{U}\mathbf{e}_k = \mathbf{y}$. We express the algorithm in Figure 4.2 with explcit reference to $\mathbf{S}_L$ for convenience and also to emphasize the connection between the two transformations. In the following discussion we shall refer to the procedure described in Lemma 4.1 and shown in Figure 4.1 for computing $\mathbf{Q}$ as $orthQ(\mathbf{y})$. We shall refer to the procedure for computing an upper Hessenberg orthogonal matrix $\mathbf{U}$ described in Figure 4.2 as $orthU(\mathbf{y})$.

It is worth noting that while underflow (flush to zero) might occur during the computation of γ at step 3.4, this is not catastrophic. Since the magnitude of each component of $\mathbf{y}$ is bounded by one, if γ underflows at step 3.4 then every component of $\mathbf{y}(1:j-1)\gamma$ must also underflow at step 3.5 regardless of how it is computed.

Finally, it should be noted that, as computed by Algorithm 4.1, $\mathbf{Q}$ will have componentwise relative errors on the order of machine precision ϵ_M with no element growth. Moreover, extension to complex arithmetic is completely straightforward (unlike Given's or Householder transformations).

5. Locking or Purging a Single Eigenvalue. The orthogonal transformations developed in the previous section will provide stable and efficient transformations needed to implement locking and purging. The simplest case to consider is the treatment of a single eigenvalue. When working in complex arithmetic, this will suffice. Handling complex conjugate eigenvalues of a real non-symmetric matrix in real arithmetic is a bit more complicated and this will be discussed in the next section.

Locking θ: The first instance to discuss is the locking of a single converged Ritz value. Assume that

$$\mathbf{H}\mathbf{y} = \mathbf{y}\theta \ , \quad ||\mathbf{y}|| = 1,$$

```
function [U] = orthU(y);

Input: y of dimension $k$ with $||\mathbf{y}|| = 1$ and $\eta = \mathbf{e}_k^T\mathbf{y}$.
Output: U such that $\mathbf{U}\mathbf{e}_k = \mathbf{y}$, $\mathbf{e}_k^T\mathbf{Q} = (\tau\mathbf{e}_{k-1}^T, \eta)$,
        with $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and U upper Hessenberg.

    $\mathbf{Q} = orthQ(\mathbf{y})$;
    $\mathbf{U} = \mathbf{Q}\mathbf{S}_L$;    % Left circular shift.
    end;
```

FIG. 4.2. *Computation of an Orthogonal Hessenberg Matrix* **U**

with $\mathbf{e}_k^T\mathbf{y} = \eta$, where $|\eta| \le \epsilon_D||\mathbf{H}||$. Here, it is understood that $\epsilon_M \le \epsilon_D < 1$ is a specified relative accuracy tolerance between ϵ_M and 1.

If θ is "wanted" then it is desirable to lock θ. However, in order to accomplish this it will be necessary to arrange a transformation of the current Arnoldi factorization to one with a small subdiagonal to isolate θ. This may be accomplish by constructing a $k \times k$ orthogonal matrix $\mathbf{Q} = \mathbf{Q}(\mathbf{y})$ using the algorithm shown in Figure 4.1

$$\mathbf{Q}\mathbf{e}_1 = \mathbf{y} \quad \text{and} \quad \mathbf{e}_k^T\mathbf{Q} = (\eta, \tau\mathbf{e}_{k-1}^T),$$

with $\eta^2 + \tau^2 = 1$. The following lemma exhibits the form of the matrix $\mathbf{H}_+ = \mathbf{Q}^T\mathbf{H}\mathbf{Q}$.

LEMMA 5.1. *Suppose* $\mathbf{H}$ *is upper Hessenberg,* $\mathbf{H}\mathbf{y} = \mathbf{y}\theta$*, with* $||\mathbf{y}|| = 1$*. Let* $\mathbf{Q} \equiv \mathbf{Q}(\mathbf{y})$*, as described in Lemma 4.1. Then* $\mathbf{H}_+ \equiv \mathbf{Q}^T\mathbf{H}\mathbf{Q}$ *is of the form*

$$\mathbf{H}_+ = \begin{bmatrix} \theta & \mathbf{h}^T \\ 0 & \widehat{\mathbf{H}} \end{bmatrix}.$$

Moreover, if $\mathbf{H} = \mathbf{H}^T$ *is symmetric and tridiagonal then* $\mathbf{h} = 0$ *and* $\widehat{\mathbf{H}}$ *is also symmetric and tridiagonal.*

Proof. Consider the quantity $\mathbf{Q}^T\mathbf{H}\mathbf{Q}$. The substitutions $\mathbf{Q}^T = (\mathbf{L} + \mathbf{y}\mathbf{g}^T)^T$, $\mathbf{Q} = (\mathbf{R} + \mathbf{y}\mathbf{e}_1^T)$ from Equations 4.2 and 4.1 and the facts $\mathbf{Q}^T\mathbf{H}\mathbf{y} = \theta\mathbf{e}_1$ and $1 = \mathbf{y}^T\mathbf{y}$ will give

$$\begin{aligned}\mathbf{Q}^T\mathbf{H}\mathbf{Q} &= \mathbf{Q}^T\mathbf{H}(\mathbf{R} + \mathbf{y}\mathbf{e}_1^T) \\ &= (\mathbf{L}^T + \mathbf{g}\mathbf{y}^T)\mathbf{H}\mathbf{R} + \theta\mathbf{e}_1\mathbf{e}_1^T \\ &= \mathbf{L}^T\mathbf{H}\mathbf{R} + \mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{R} + \theta\mathbf{e}_1\mathbf{e}_1^T.\end{aligned}$$

Since both $\mathbf{L}^T$ and $\mathbf{R}$ are upper triangular, it follows that $\mathbf{L}^T\mathbf{H}\mathbf{R}$ is upper Hessenberg with the first row and the first column each being zero due to $\mathbf{L}\mathbf{e}_1 = \mathbf{R}\mathbf{e}_1 = 0$. Also, $\mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{R}$ is a rank-one matrix with zero first column. Therefore $\mathbf{H}_+$ is of the form

$$\mathbf{H}_+ = \begin{bmatrix} \theta & \mathbf{h}^T \\ 0 & \widehat{\mathbf{H}} \end{bmatrix}$$

as claimed with $(0, \mathbf{h}^T) = \mathbf{y}^T\mathbf{H}\mathbf{R}$.

This result could have easily been arrived at through the facts that $\mathbf{Q}\mathbf{e}_1 = \mathbf{y}$ and $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, but this proof exposes the additional structure that $\widehat{\mathbf{H}}$ is an upper

Hessenberg matrix plus a rank one matrix. Moreover, if $\mathbf{H} = \mathbf{H}^T$ is symmetric and tridiagonal, then $\mathbf{y}^T\mathbf{H}\mathbf{R} = \theta\mathbf{y}^T\mathbf{R} = 0$, and this implies

$$\mathbf{L}^T\mathbf{H}\mathbf{R} = \mathbf{Q}^T\mathbf{H}\mathbf{Q} - \theta \mathrm{e}_1\mathrm{e}_1^T$$

is symmetric. Since, $\mathbf{L}^T\mathbf{H}\mathbf{R}$ is both symmetric and Hessenberg, it must be tridiagonal, and this concludes the proof. □

This proof shows that the deflation will be complete if $\mathbf{H}$ is symmetric and tridiagnonal. However, the deflated matrix will not be in Hessenberg form when it is nonsymmetric. More work will have to be done in this case to return the matrix to Hessenberg form using orthogonal similarity transformations. Of course, we do not wish to destroy the structure of the last row of $\mathbf{Q}$ in this process. One convenient way to accomplish this is to apply a succession of orthogonal transformations of the form

$$\widehat{\mathbf{U}}_j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{U}_j & 0 \\ 0 & 0 & \mathbf{I}_{k-j} \end{bmatrix}$$

so that $\mathbf{H}_+ \leftarrow \widehat{\mathbf{U}}_j^T\mathbf{H}\widehat{\mathbf{U}}_j$ is constructed as in Figure 4.2 to introduce zeros in positions $2 : j-1$ of row $j+1$ for $j = k-1, k-2, \ldots, 3$. The Matlab style code shown in Figure 5.1 gives the simplest explanation of how this deflation proceeds. Of course, the orthogonal matrix is updated in the same way to give $\mathbf{Q} \leftarrow \mathbf{Q}\widehat{\mathbf{U}}_j,\ \ j = k-1, k-2, \ldots, 3$. On completion, the $k-th$ row of $\mathbf{Q}$ remains undisturbed from the original construction.

At step j in Fig. 5.1, The procedure *orthU*($\mathbf{z}$) produces an upper Hessenberg $\mathbf{U}_j$ as described in equation 4.3 such that $\mathbf{e}_j^T = \mathbf{z}^T\mathbf{U}_j$. The end result of these transformations is

$$\mathbf{A}\mathbf{v}_1 = \mathbf{v}_1\theta + \mathbf{f}\eta, \quad \text{where} \quad \mathbf{v}_1^T\mathbf{f} = 0$$
$$\mathbf{A}\mathbf{V}_2 = (\mathbf{v}_1, \mathbf{V}_2)\begin{pmatrix} \mathbf{h}_1^T \\ \mathbf{H}_2 \end{pmatrix} + \mathbf{f}\tau\mathbf{e}_{k-1}^T.$$

where $[\mathbf{v}_1, \mathbf{V}_2] = \mathbf{V}\mathbf{Q}$.
This means that subsequent implicit restarting takes place as if

$$\mathbf{A}\mathbf{V}_2 = \mathbf{V}_2\mathbf{H}_2 + \mathbf{f}\tau\mathbf{e}_{k-1}^T$$

with all the subsequent orthogonal matrices and column deletions associated with implicit restarting applied to $\mathbf{h}_1^T$ and never disturbing the relation $\mathbf{A}\mathbf{v}_1 = \mathbf{v}_1\theta + \mathbf{f}\eta$. Now, if $\widehat{\mathbf{Q}}$ represents a $(k-1)\times(k-1)$ orthogonal matrix associated with an implicit restart than

$$\mathbf{A}\mathbf{V}_2\widehat{\mathbf{Q}} = (\mathbf{v}_1, \mathbf{V}_2\widehat{\mathbf{Q}})\begin{pmatrix} \mathbf{h}_1^T\widehat{\mathbf{Q}} \\ \widehat{\mathbf{Q}}^T\mathbf{H}_2\widehat{\mathbf{Q}} \end{pmatrix} + \mathbf{f}\tau\mathbf{e}_{k-1}^T\widehat{\mathbf{Q}}.$$

In subsequent Arnoldi steps, $\mathbf{v}_1$ participates in the orthogonalization so that the selective orthogonalization recommended by Parlett and Scott [6, 4] is accomplished automatically.

```
function $[\mathbf{V},\mathbf{H},\mathbf{f}] = \mathbf{defN}(\mathbf{V},\mathbf{H},\mathbf{f},\mathbf{y},\theta)$;

  Input: $(\mathbf{V},\mathbf{H},\mathbf{y},\theta)$ with $\mathbf{H}$ upper Hessenberg, $\mathbf{H}\mathbf{y} = \mathbf{y}\theta$, $\|\mathbf{y}\| = 1$,

  Output: $(\mathbf{V},\mathbf{H},\mathbf{f})$ such that
        $\mathbf{V} \leftarrow \mathbf{V}\mathbf{Q}$,  $\mathbf{H} \leftarrow \mathbf{Q}'\mathbf{H}\mathbf{Q}$  $\mathbf{f} \leftarrow \mathbf{f}\mathbf{Q}(k,k)$
        with $\mathbf{Q}(:,1) = \mathbf{y}, \mathbf{Q}^T\mathbf{Q} = \mathbf{I}, \mathbf{H}(1,1) = \theta, \mathbf{H}(2:k,1) = 0$.

        1. $\mathbf{Q} = orthQ(\mathbf{y})$;
        2. $\mathbf{H} \leftarrow \mathbf{Q}'\mathbf{H}\mathbf{Q}$;
        3. for $j = k : -1 : 4$,
             3.1. $\mathbf{z} = \mathbf{H}(j, 2 : j-1)$;
             3.2. $\mathbf{z} = \mathbf{z}'/\|\mathbf{z}\|$;
             3.3. $\mathbf{U} = orthU(\mathbf{z})$;
             3.4. $\mathbf{H}(:, 2 : j-1) = \mathbf{H}(:, 2 : j-1)\mathbf{U}$;
             3.5. $\mathbf{H}(2 : j-1, :) = \mathbf{U}'\mathbf{H}(2 : j-1, :)$;
             3.6. $\mathbf{Q}(:, 2 : j-1) = \mathbf{Q}(:, 2 : j-1)\mathbf{U}$;
             3.7. $\tau_o = \tau$;
          end;
        4. $\mathbf{V} \leftarrow \mathbf{V}\mathbf{Q}$;
        5. $\mathbf{f} \leftarrow \mathbf{f} \cdot \mathbf{Q}(k,k)$;
```

FIG. 5.1. *Nonsymmetric Locking*

Purging θ: If θ is "unwanted" then we may wish to remove θ from the spectrum of the projected matrix $\mathbf{H}$. However, the implicit restart strategy using exact shifts will sometimes fail to purge a converged unwanted Ritz value [2]. A mechanism called purging was developed in [2] to remove converged unwanted Ritz values from $\mathbf{H}$ in a manner that preserves an Arnoldi relation. As with locking, this mechanism was based upon using right eigenvectors.

However, there is an alternative based upon deflating with a left eigenvector that has some attractive properties. The purging process is quite analogous to the locking process just described. Let $\mathbf{y}$ be a left eigenvector of $\mathbf{H}$ corresponding to θ , i.e.

$$\mathbf{y}^T\mathbf{H} = \theta\mathbf{y}^T.$$

Then use *orthU* to construct a $k \times k$ orthogonal matrix $\mathbf{U}$ such that

$$\mathbf{U} = \mathbf{Q}\mathbf{S}_L, \quad \text{with} \quad \mathbf{y}^T\mathbf{U} = \mathbf{e}_k^T, \quad \text{and} \quad \mathbf{e}_k^T\mathbf{U} = (0,\cdots,0,\tau,\eta,),$$

where $\eta = \mathbf{e}_k^T\mathbf{y}$ and $\tau^2 + \eta^2 = 1$. Then

$$\mathbf{A}(\widehat{\mathbf{V}},\mathbf{v}_k) = (\widehat{\mathbf{V}},\mathbf{v}_k)\begin{pmatrix} \widehat{\mathbf{H}} & \mathbf{h} \\ 0 & \theta \end{pmatrix} + \mathbf{f}(\tau\mathbf{e}_{k-1}^T,\eta),$$

where $[\widehat{\mathbf{V}},\mathbf{v}_k] = \mathbf{V}\mathbf{U}$. Now, simply delete the last column on both sides to get

$$\mathbf{A}\widehat{\mathbf{V}} = \widehat{\mathbf{V}}\widehat{\mathbf{H}} + \mathbf{f}\tau\mathbf{e}_{k-1}^T.$$

It is easily seen that $\widehat{\mathbf{H}}$ is upper Hessenberg due to the following lemma.

Lemma 5.2. *Suppose* $\mathbf{H}$ *is upper Hessenberg,* $\mathbf{y}^T\mathbf{H} = \theta\mathbf{y}^T$*, with* $\|\mathbf{y}\| = 1$*. Let* $\mathbf{U} = \mathbf{Q}\mathbf{S}_L$ *, where* $\mathbf{Q} \equiv \mathbf{Q}(\mathbf{y})$ *as described in Lemma 4.1. Then*

$$\mathbf{H}_+ \equiv \mathbf{U}^T\mathbf{H}\mathbf{U}$$

is upper Hessenberg with $\mathbf{e}_k^T\mathbf{H}_+ = \theta\mathbf{e}_k^T$*.*

Proof. The proof is much like the proof of Lemma 5.1. Since $\mathbf{H}_+ = \mathbf{S}_L^T\mathbf{Q}^T\mathbf{H}\mathbf{Q}\mathbf{S}_L$, let us first consider the quantity $\mathbf{Q}^T\mathbf{H}\mathbf{Q}$. The substitutions $\mathbf{Q}^T = (\mathbf{L} + \mathbf{y}\mathbf{g}^T)^T$, $\mathbf{Q} = (\mathbf{R} + \mathbf{y}\mathbf{e}_1^T)$ from Equations 4.2 and 4.1, and the facts $0 = \mathbf{y}^T\mathbf{R}$ and $1 = \mathbf{y}^T\mathbf{y}$ will give

$$\begin{aligned}\mathbf{Q}^T\mathbf{H}\mathbf{Q} &= (\mathbf{L}^T + \mathbf{g}\mathbf{y}^T)\mathbf{H}\mathbf{Q}\\ &= \mathbf{L}^T\mathbf{H}\mathbf{Q} + \mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{Q}\\ &= \mathbf{L}^T\mathbf{H}\mathbf{R} + \mathbf{L}^T\mathbf{H}\mathbf{y}\mathbf{e}_1^T + \theta\mathbf{g}\mathbf{e}_1^T.\end{aligned}$$

Since both $\mathbf{L}^T$ and $\mathbf{R}$ are upper triangular, it follows that $\mathbf{L}^T\mathbf{H}\mathbf{R}$ is upper Hessenberg with the first row and the first column each being zero due to $\mathbf{L}\mathbf{e}_1 = \mathbf{R}\mathbf{e}_1 = 0$. Therefore

$$\mathbf{Q}^T\mathbf{H}\mathbf{Q} = \begin{bmatrix} \theta & 0 \\ \mathbf{h} & \widehat{\mathbf{H}} \end{bmatrix}$$

where $\begin{bmatrix} 0 \\ \mathbf{h} \end{bmatrix} = \mathbf{L}^T\mathbf{H}\mathbf{y}$. Now, use the properties if the left-circular-shift operator $\mathbf{S}_L$ to see that

$$\mathbf{H}_+ = \mathbf{U}^T\mathbf{H}\mathbf{U} = \mathbf{S}_L^T\mathbf{Q}^T\mathbf{H}\mathbf{Q}\mathbf{S}_L = \begin{bmatrix} \widehat{\mathbf{H}} & \mathbf{h} \\ 0 & \theta \end{bmatrix},$$

with $\widehat{\mathbf{H}}$ upper Hessenberg. This concludes the proof. □

Observe that there is no requirement that $\mathbf{y}$ be an accurate left eigenvector for $\mathbf{H}$. However, it will be necessary for the residual $\mathbf{y}^T\mathbf{H}\mathbf{Q} = \theta\mathbf{e}_1^T$ to meet a componentwise accuracy condition that we shall discuss in § 7 . Moreover, there is no need for η to be small, but when it is not small, the implicit restart mechanism with exact shifts will suffice to purge θ. Finally, this procedure is valid in complex arithmetic with minor notational modifications.

6. Locking or Purging a Complex Conjugate Pair. When working with real nonsymmetric matrices, it is desirable to compute in real arithmetic and this requires the ability to work with complex conjugate pairs of eigenvalues as a unit. This theme is standard for the double implicit shift both in implicit QR and in implicit restarting [7].

Suppose $\mathbf{H}(\mathbf{x} + i\mathbf{y}) = (\mathbf{x} + i\mathbf{y})(\theta + i\mu)$ with $\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} = 1$ and $\|\mathbf{e}_k^T(\mathbf{x}, \mathbf{y})\| = \epsilon < \tau$. Then $(\mathbf{x} \pm i\mathbf{y})$ are eigenvectors corresponding to the complex conjugate pair of eigenvalues $\theta \pm i\mu$ and we may express the relationship in real arithmetic as

$$\mathbf{H}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y}) \begin{pmatrix} \theta & \mu \\ -\mu & \theta \end{pmatrix}.$$

As shown in [1, 2], it may not be possible to lock this pair using a set of orthogonal vectors. Here, we present a scheme that will lock the pair using orthogonal transformations if possible. If it is not possible to lock the pair, the scheme will attempt to lock one of the two while splitting the conjugate pair safely into two real eigenvalues. In some cases, no locking with an orthogonal matrix will be possible.

To develop this, let

$$(\mathbf{x},\mathbf{y}) = (\mathbf{q}_1,\mathbf{q}_2)\begin{pmatrix}\rho_1 & 0\\ 0 & \rho_2\end{pmatrix}\begin{pmatrix}\gamma & -\sigma\\ \sigma & \gamma\end{pmatrix}$$

be the singular value decomposition (SVD) of $(\mathbf{x},\mathbf{y})$ with $\rho_1 \geq \rho_2$ denoting the singular values of $(\mathbf{x},\mathbf{y})$. The columns of the orthogonal matrix $\begin{pmatrix}\gamma & \sigma\\ -\sigma & \gamma\end{pmatrix}$ are the right singular vectors and the columns of $(\mathbf{q}_1,\mathbf{q}_2)$ are the left singular vectors. Note

$$\|\mathbf{e}_k^T(\mathbf{q}_1\rho_1,\mathbf{q}_2\rho_2)\| = \|\mathbf{e}_k^T(\mathbf{x},\mathbf{y})\begin{pmatrix}\gamma & \sigma\\ -\sigma & \gamma\end{pmatrix}\| = \|\mathbf{e}_k^T(\mathbf{x},\mathbf{y})\| = \epsilon < \epsilon_D.$$

An important quantity for the analysis is $\rho \equiv \frac{\rho_2}{\rho_1}$, the reciprocal of $cond\{(\mathbf{x},\mathbf{y})\}$. It is easily checked that

$$\begin{pmatrix}\gamma & -\sigma\\ \sigma & \gamma\end{pmatrix}\begin{pmatrix}\theta & \mu\\ -\mu & \theta\end{pmatrix}\begin{pmatrix}\gamma & \sigma\\ -\sigma & \gamma\end{pmatrix} = \begin{pmatrix}\theta & \mu\\ -\mu & \theta\end{pmatrix}.$$

Since

$$\begin{aligned}\mathbf{H}(\mathbf{q}_1,\mathbf{q}_2) &= \mathbf{H}(\mathbf{x},\mathbf{y})\begin{pmatrix}\gamma & \sigma\\ -\sigma & \gamma\end{pmatrix}\begin{pmatrix}\rho_1^{-1} & 0\\ 0 & \rho_2^{-1}\end{pmatrix}\\ &= (\mathbf{x},\mathbf{y})\begin{pmatrix}\theta & \mu\\ -\mu & \theta\end{pmatrix}\begin{pmatrix}\gamma & \sigma\\ -\sigma & \gamma\end{pmatrix}\begin{pmatrix}\rho_1^{-1} & 0\\ 0 & \rho_2^{-1}\end{pmatrix}\\ &= (\mathbf{q}_1,\mathbf{q}_2)\begin{pmatrix}\rho_1 & 0\\ 0 & \rho_2\end{pmatrix}\begin{pmatrix}\gamma & -\sigma\\ \sigma & \gamma\end{pmatrix}\begin{pmatrix}\theta & \mu\\ -\mu & \theta\end{pmatrix}\begin{pmatrix}\gamma & \sigma\\ -\sigma & \gamma\end{pmatrix}\begin{pmatrix}\rho_1^{-1} & 0\\ 0 & \rho_2^{-1}\end{pmatrix},\end{aligned}$$

it follows that

$$\mathbf{H}(\mathbf{q}_1,\mathbf{q}_2) = (\mathbf{q}_1,\mathbf{q}_2)\begin{pmatrix}\theta & \frac{\mu}{\rho}\\ -\mu\rho & \theta\end{pmatrix} \tag{6.1}$$

with

$$\mathbf{e}_k^T(\mathbf{q}_1,\mathbf{q}_2) = (\mathbf{e}_k^T\mathbf{q}_1,\mathbf{e}_k^T\mathbf{q}_2) = (\frac{\epsilon_1}{\rho_1},\frac{\epsilon_2}{\rho_2}).$$

Here, $(\epsilon_1,\epsilon_2) \equiv \mathbf{e}_k^T(\mathbf{x}\gamma - \mathbf{y}\sigma, \mathbf{x}\sigma + \mathbf{y}\gamma)$, so $\epsilon_1^2 + \epsilon_2^2 = \epsilon^2$.

Now, $\frac{1}{\sqrt{2}} \leq \rho_1 \leq 1$ follows readily from the facts $\rho_1^2 + \rho_2^2 = trace\{(\mathbf{x},\mathbf{y})^T(\mathbf{x},\mathbf{y})\} = \mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} = 1$ and $\rho_1 \geq \rho_2$. Also,from (6.1), we have the relations

$$\mathbf{H}\mathbf{q}_1 = \mathbf{q}_1\theta - \mathbf{q}_2\mu\rho \quad , \quad \mathbf{H}\mathbf{q}_2 = \mathbf{q}_1\frac{\mu}{\rho} + \mathbf{q}_2\theta.$$

Using orthogonality of $\mathbf{q}_1$ and $\mathbf{q}_2$ gives

$$\mathbf{q}_2^T\mathbf{H}\mathbf{q}_1 = -\mu\rho \quad , \quad \mathbf{q}_1^T\mathbf{H}\mathbf{q}_2 = \frac{\mu}{\rho}$$

to see that $|\frac{\mu}{\rho}| \leq \|\mathbf{H}\|$, and hence that

$$|\mu| \leq \|\mathbf{H}\|\rho \;\text{ and }\; |\mu\rho| \leq \|\mathbf{H}\|\rho^2. \tag{6.2}$$

Locking a congugate pair $\theta \pm i\mu$: If $\theta + i\mu$ is "wanted" then we may wish to lock $\theta + i\mu$. This is accomplished with a block version of the single eigenvalue case but there may be some complications.

To begin the deflation, the short form of the SVD of $(\mathbf{x}, \mathbf{y})$ must be computed. A well known efficient way to do this is to compute the QR-factorization of $(\mathbf{x}, \mathbf{y})$ followed by a computation of the SVD of the 2×2 upper triangular R-matrix. In any case, Equation 6.1 will hold. Now, use *orthQ* twice to construct

$$\begin{array}{rrcl} i) & \mathbf{Q}_1 & = & \mathbf{Q}(\mathbf{q}_1) = \mathbf{R}_1 + \mathbf{q}_1\mathbf{e}_1^T, \\ ii) & \hat{\mathbf{q}}_2 & = & \mathbf{R}_1^T\mathbf{q}_2, \\ iii) & \mathbf{Q}_2 & = & \mathbf{Q}(\hat{\mathbf{q}}_2) = \mathbf{R}_2 + \hat{\mathbf{q}}_2\mathbf{e}_2^T + \mathbf{e}_1\mathbf{e}_1^T, \\ iv) & \mathbf{Q} & = & \mathbf{Q}_1\mathbf{Q}_2 = \mathbf{R}_1\mathbf{R}_2 + \mathbf{q}_1\mathbf{e}_1^T + \mathbf{q}_2\mathbf{e}_2^T. \end{array} \tag{6.3}$$

In this construction, $\mathbf{R}_2\mathbf{e}_j = 0$ for $j = 1, 2$ and it is easily seen that $\mathbf{q}_2 = \mathbf{Q}_1\mathbf{Q}_1^T\mathbf{q}_2 = \mathbf{R}_1\mathbf{R}_1^T\mathbf{q}_2 = \mathbf{R}_1\hat{\mathbf{q}}_2$.

This construction will provide a $\mathbf{Q}$ such that

$$\mathbf{Q} = (\mathbf{q}_1\mathbf{q}_2|\mathbf{W}) \quad \text{where} \quad (\mathbf{x}, \mathbf{y}) = (\mathbf{q}_1\mathbf{q}_2|\mathbf{W}) \begin{pmatrix} \mathbf{S} \\ 0 \end{pmatrix} \mathbf{U}^T$$

is a full SVD of $(\mathbf{x}, \mathbf{y})$ with

$$\mathbf{U}^T = \begin{pmatrix} \gamma & -\sigma \\ \sigma & \gamma \end{pmatrix} \quad \text{and} \quad \mathbf{S} = \begin{pmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{pmatrix}.$$

Note that $\mathbf{e}_k^T\mathbf{Q} = (\frac{\epsilon_1}{\rho_1}, \frac{\epsilon_2}{\rho_2}, \tau\mathbf{e}_k^T)$. Since $0 < \rho_2 < \rho_1$, the conjugate pair may be safely locked if $|\mathbf{e}_k^T\mathbf{q}_2| \le \epsilon_D$. Now,

$$\mathbf{Q}^T\mathbf{H}\mathbf{Q} = \begin{pmatrix} \mathbf{D} & \widehat{\mathbf{H}}_1 \\ 0 & \widehat{\mathbf{H}}_2 \end{pmatrix} \quad \text{with} \quad \mathbf{D} = \begin{pmatrix} \theta & \frac{\mu}{\rho} \\ -\mu\rho & \theta \end{pmatrix},$$

and

$$\mathbf{A}(\mathbf{V}_1, \widehat{\mathbf{V}}) = (\mathbf{V}_1, \widehat{\mathbf{V}}) \begin{pmatrix} \mathbf{D} & \widehat{\mathbf{H}}_1 \\ 0 & \widehat{\mathbf{H}}_2 \end{pmatrix} + \mathbf{f}(\frac{\epsilon_1}{\rho_1}, \frac{\epsilon_2}{\rho_2}, \tau\mathbf{e}_{k-2}^T) \quad \text{with} \quad \mathbf{V}_1 \equiv \mathbf{V}(\mathbf{q}_1, \mathbf{q}_2).$$

To complete the locking process, construct $\mathbf{U}_2$ such that $\mathbf{e}_{k-1}^T\mathbf{U}_2 = \mathbf{e}_{k-1}^T$, and $\mathbf{H}_2 = \mathbf{U}_2^T\widehat{\mathbf{H}}_2\mathbf{U}_2$ is upper Hessenberg and put

$$\mathbf{H}_1 = \widehat{\mathbf{H}}_1\mathbf{U}_2$$
$$\mathbf{V}_2 = \widehat{\mathbf{V}}\mathbf{U}_2.$$

Then

$$\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1\mathbf{D} + \mathbf{f}(\frac{\epsilon_1}{\rho_1}, \frac{\epsilon_2}{\rho_2}),$$

where $\mathbf{V}_1^T\mathbf{f} = 0$, and

$$\mathbf{A}\mathbf{V}_2 = (\mathbf{V}_1\mathbf{V}_2) \begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{pmatrix} + \mathbf{f}\tau e_{k-2}^T.$$

Again, this means that implicit restarting takes place as if

$$\mathbf{A}\mathbf{V}_2 = \mathbf{V}_2\mathbf{H}_2 + \mathbf{f}\eta\mathbf{e}_{k-2}^T$$

with all subsequent orthogonal transformations and column deletions applied to $\mathbf{H}_1$ never disturbing the relation

$$\mathbf{A}\mathbf{V}_1 = \mathbf{V}_1\mathbf{D} + \mathbf{f}(\frac{\epsilon_1}{\rho_1}, \frac{\epsilon_2}{\rho_2}).$$

In subsequent Arnoldi steps, $\mathbf{V}_1$ must continue to participate in the orthogonalization to prevent the introduction of spurious Ritz values.

In some cases it may be appropriate to lock just one eigenvalue, and to split the conjugate pair into two real eigenvalues. A conservative approach shall be adopted here to only allow perturbations relative to ϵ_M. Suppose $\rho < \sqrt{\epsilon_M}$, then $|\mu\rho| < ||\mathbf{H}||\epsilon_M$ and

$$\mathbf{A}\mathbf{v}_1 = \mathbf{v}_1\theta + \mathbf{g} \ \text{ with } \ \mathbf{v}_1^T\mathbf{g} = 0$$

where $\mathbf{g} = -\mathbf{v}_2\mu\rho + \mathbf{f}\frac{\epsilon_1}{\rho_1}$ with $\mathbf{v}_1 = \mathbf{V}\mathbf{q}_1$ and $\mathbf{v}_2 = \mathbf{V}\mathbf{q}_2$, We then construct an orthogonal matrix $\mathbf{Q}(\mathbf{q}_1)$ such that $\mathbf{Q} = \mathbf{R} + \mathbf{q}_1\mathbf{e}_1^T$. Note

$$\mathbf{e}_k^T\mathbf{R} = \tau e_{k-1}^T \ \text{ with } \ \tau^2 = 1 - \epsilon^2$$

where $\epsilon^2 = \frac{\epsilon_1}{\rho_1}^2$. Now proceed as in the locking step described above for a single eigenvalue.

Purging a congugate pair $\theta \pm i\mu$:

If the conjugate pair is "unwanted" but converged then it may be necessary to purge the pair $\theta \pm i\mu$ directly, since the implicit restart strategy may fail to do this [2].

Suppose $(\mathbf{x}^T + i\mathbf{y}^T)\mathbf{H} = (\theta + i\mu)(\mathbf{x}^T + i\mathbf{y}^T)$. Then $\mathbf{Y}^T\mathbf{H} = \mathbf{D}\mathbf{Y}^T$, with $\mathbf{Y} = (\mathbf{x}, \mathbf{y})$ and $\mathbf{D} = \begin{pmatrix} \theta & -\mu \\ \mu & \theta \end{pmatrix}$.

Purging is much easier. Simply compute a QR-factorization $(\mathbf{x}, \mathbf{y}) = \widehat{\mathbf{Q}}\mathbf{R}$ with $(\mathbf{q}_1, \mathbf{q}_2) = \widehat{\mathbf{Q}}$. Now compute $\mathbf{U} = \mathbf{Q}\mathbf{S}_L^2$ with $\mathbf{Q}$ constructed as in Equation 6.3 above.

With essentially the same process as the single vector case, we have

$$\mathbf{U}^T\mathbf{H}\mathbf{U} = \begin{pmatrix} \widehat{\mathbf{H}} & \mathbf{H}_1 \\ 0 & \widehat{\mathbf{D}} \end{pmatrix} \quad \text{where } \ \widehat{\mathbf{D}} = \mathbf{R}^{-1}\mathbf{D}\mathbf{R},$$

and

$$\mathbf{A}(\widehat{\mathbf{V}}, \mathbf{V}_k) = (\widehat{\mathbf{V}}, \mathbf{V}_k)\begin{pmatrix} \widehat{\mathbf{H}} & \mathbf{H}_1 \\ 0 & \widehat{\mathbf{D}} \end{pmatrix} + f(\tau\mathbf{e}_{k-2}^T, \epsilon_1, \epsilon_2)$$

Now, simply delete the last two columns on both sides to get

$$\mathbf{A}\widehat{\mathbf{V}} = \widehat{\mathbf{V}}\widehat{\mathbf{H}} + \mathbf{f}\eta\mathbf{e}_{k-2}^T,$$

and then construct $\mathbf{U}_2$ such that $\mathbf{e}_{k-2}^T\mathbf{U}_2 = \mathbf{e}_{k-2}^T$ and $\mathbf{H} \leftarrow \mathbf{U}_2^T\widehat{\mathbf{H}}\mathbf{U}_2$ is upper Hessenberg. Finally, replace $\mathbf{V} \leftarrow \widehat{\mathbf{V}}\mathbf{U}_2$.

Observe that there is no requirement that $\mathbf{V}\mathbf{Y}$ be a "good" set of left eigenvectors for $\mathbf{A}$ or even that it is an accurate eigenvector matrix. All we require is that the norm

$$||\mathbf{Y}^T\mathbf{H} - \mathbf{D}\mathbf{Y}^T|| \quad \text{is} \quad \text{small}$$

7. Error Analysis. This section will give a brief error analysis indicating the important features of the locking and purging schemes. We shall first analyze the numerical stability of the transformations for locking and purging. Then we give an analysis of the effect of locking on the accuracy of the computed eigenvalues of the original problem.

Stability of $\mathbf{Q}^T\mathbf{H}\mathbf{Q}$: Since the orthogonal transformations developed in Section 4 are clearly stable (i.e. componentwise relatively accurate representation of the transformation one would obtain in exact arithmetic), there is no question that the similarity transformation $\mathbf{Q}^T\mathbf{H}\mathbf{Q}$ numerically preserves the eigenvalues of $\mathbf{H}$. However, there is a serious question about how well these transformations perform numerically in preserving tridiagonal or Hessenberg form during locking and/or purging. For simplicity, we shall restrict our discussion of this question to real θ for both the symmetric and nonsymmetric case. There is no essential change to extend this to a complex θ in complex arithmetic.

It is necessary to show that when $\mathbf{H}$ is symmetric and $\mathbf{H}\mathbf{y} = \mathbf{y}\theta$, then $\mathbf{H}_+ \equiv \mathbf{Q}^T\mathbf{H}\mathbf{Q}$ is symmetric and numerically tridiagonal, and when $\mathbf{H}$ is Hessenberg and $\mathbf{y}^T\mathbf{H} = \theta\mathbf{y}^T$, then $\mathbf{H}_+ \equiv \mathbf{U}^T\mathbf{H}\mathbf{U}$ is numerically upper Hessenberg. (i.e. that the entries below the subdiagonal are all tiny relative to $||\mathbf{H}||$).

It will suffice to discuss the case of purging in the nonsymmetric upper Hessenberg case. The proof that $\mathbf{H}$ is returned to tridiagonal form in the symmetric case will follow from symmetry if we show it is numerically upper Hessenberg. In both cases a left eigenvector $\mathbf{y}$ with norm 1, determines the transformation $\mathbf{Q}$ developed in Section 4. In the symmetric case, the proof given in Lemma 5.1 that $\mathbf{Q}^H\mathbf{H}\mathbf{Q}$ is tridiagonal relied upon the term $\mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{R}$ vanishing in the expression

$$\mathbf{Q}^T\mathbf{H}\mathbf{Q} = \mathbf{L}^T\mathbf{H}\mathbf{R} + \mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{R} + \theta\mathbf{e}_1\mathbf{e}_1^T.$$

Also, in the nonsymmetric purging case, in the proof that Hessenberg form is preserved in Lemma 5.2 relied upon $\mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{Q} = \theta\mathbf{g}\mathbf{e}_1^T$ in the expression

$$\mathbf{Q}^T\mathbf{H}\mathbf{Q} = \mathbf{L}^T\mathbf{H}\mathbf{Q} + \mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{Q}$$

However, on closer examination, we see that

$$\mathbf{e}_1^T\mathbf{Q} = \mathbf{e}_1^T\mathbf{L} + (\mathbf{e}_1^T\mathbf{y})\mathbf{g}^T = \eta_1\mathbf{g}^T,$$

where η_1 is the first component of $\mathbf{y}$. Therefore,

$$||\mathbf{g}|| = \frac{1}{|\eta_1|}, \tag{7.1}$$

so there may be numerical difficulty when the first component of $\mathbf{y}$ is small. To be specific, $\mathbf{y}^T\mathbf{H} = \theta\mathbf{y}^T$ and thus $\mathbf{y}^T\mathbf{H}\mathbf{R} = 0$ in exact arithmetic. However, in finite precision, the computed $fl(\mathbf{y}^T\mathbf{H}) = \theta\mathbf{y}^T + \mathbf{e}^T$. The error $\mathbf{e}$ will be on the order of ϵ_M relative to $||\mathbf{H}||$, but

$$||\mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{R}|| = ||\mathbf{g}|| \cdot ||\mathbf{e}^T\mathbf{R}|| = \frac{1}{|\eta_1|}||\mathbf{e}^T\mathbf{R}||,$$

so this term may be quite large. It may be as large as order $\mathcal{O}(1)$ if $\eta_1 = \mathcal{O}(\epsilon_M)$. This is of serious concern and has been observed in practice. Therefore, we give an analysis and offer a remedy to this dilemma.

The remedy is to introduce a step-by-step acceptable rescaling of the vector $\mathbf{y}$ to simultaneously force the conditions

$$\mathbf{Q}^T\mathbf{y} = \mathbf{e}_1, \quad \text{and} \quad \mathbf{y}^T\mathbf{H}\mathbf{Q} = \theta\mathbf{e}_1^T$$

to hold with sufficient accuracy in finite precision. To accomplish this, we shall devise a scheme to achieve

$$\mathbf{y}^T\mathbf{H}\mathbf{q}_j = 0, \quad \text{for} \;\; j > 1$$

numerically and then prove that this scheme is sufficient to establish $\mathbf{q}_i^T\mathbf{H}\mathbf{q}_j = 0$ numerically, relative to $||\mathbf{H}||$ for $j < i-1$.

We begin by examining the inner product $\mathbf{y}^T\mathbf{H}\mathbf{q}_j$. We define $\hat{\mathbf{q}}_j^T \equiv (\gamma_j\mathbf{y}_{j-1}^T, \rho_j)$, and note that the zero/non-zero structure of $\mathbf{H}$ and $\mathbf{q}_j$ gives

$$\mathbf{y}^T\mathbf{H}\mathbf{q}_j = \mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j + \rho_j\beta_j\eta_{j+1},$$

where $\mathbf{H}_j = \mathbf{H}(1:j, 1:j)$, $\beta_j = \mathbf{H}(j, j+1)$. Through the remainder of this discussion, we shall treat the quantity $\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j$ as a computed term and assume that $fl(\mathbf{y}^T\mathbf{H}\mathbf{q}_j)$ is the floating point result of computing $\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j$ first and then adding $\rho_j\beta_j\eta_{j+1}$ to it. In the analysis, we shall not attempt to represent the round-off error associated with each floating point operation. It is the accuracy of the addition of these two terms that determines the success (or failure) of the computation.

If the computed quantity $fl(\mathbf{y}^T\mathbf{H}\mathbf{q}_j)$ is not zero then we may adjust it to become zero by scaling the vector $\mathbf{y}_j$ by a number ϕ and the component η_{j+1} by a number ψ. Thus, prior to the computation of $\mathbf{q}_{j+1}$, we have $\mathbf{y}_j \leftarrow \mathbf{y}_j\phi$ and $\eta_{j+1} \leftarrow \eta_{j+1}\psi$. Certainly, $||\mathbf{y}||$ should not be altered with this scaling. The following system of equations will determine ϕ and ψ according to these conditions:

$$\begin{aligned} (\tau_j\phi)^2 + (\eta_{j+1}\psi)^2 &= \tau_{j+1}^2 \\ \mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j\phi + \rho_j\beta_j\eta_{j+1}\psi &= 0. \end{aligned}$$

Let $\sigma = \rho_j\beta_j\eta_{j+1}/(\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j)$. A little algebraic manipulation gives the following alternative expressions for ϕ and ψ. Either use

$$\psi = \pm\frac{\tau_{j+1}}{\sqrt{(\tau_j\sigma)^2 + \eta_{j+1}^2}}, \quad \phi = -\sigma\psi,$$

or

$$\phi = \pm\frac{\tau_{j+1}}{\sqrt{(\tau_j)^2 + (\eta_{j+1}/\sigma)^2}}, \quad \psi = -\phi/\sigma.$$

Observe that none of the previously computed $\mathbf{q}_i$ $2 \le i < j$ will be changed due to this procedure. After step j, the vector $\mathbf{y}_j$ is simply rescaled in subsequent steps, and the the formulas defining $\mathbf{q}_i$, $2 \le i \le j$ are invariant with respect to scaling of $\mathbf{y}_j$. Also, note that one of the two formulas given above for computing ϕ and ψ will be well defined even when one of the quantities $\rho_j\beta_j\eta_{j+1} = 0$ or $\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j = 0$.

It remains to show that $fl(\mathbf{q}_{j+1}^T\mathbf{H}\mathbf{q}_i)$ is small. We shall demonstrate that

$$|fl(\mathbf{q}_{j+1}^T\mathbf{H}\mathbf{q}_i)| \le 2 \cdot ||\mathbf{H}\mathbf{q}_i||\epsilon_M \;\; \text{for} \;\; 1 < i < j.$$

To establish this, we first show that the effect of re-scaling is to provide a new vector $\mathbf{y}$ such that

$$|fl(\mathbf{y}^T\mathbf{H}\mathbf{q}_j)| \le 2 \cdot \|\mathbf{H}\mathbf{q}_j\|\tau_{j+1}\epsilon_M$$

To see this, first note that prior to the scaling,

$$|\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j| \le \|\mathbf{H}_j\hat{\mathbf{q}}_j\|\|\mathbf{y}_j\| \le \|\mathbf{H}\mathbf{q}_j\|\tau_j,$$

and that

$$|\rho_j\beta_j\eta_{j+1}| \le \|\mathbf{H}\mathbf{q}_j\||\eta_{j+1}|.$$

Thus

$$\begin{aligned}|\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j\phi| &\le \|\mathbf{H}\mathbf{q}_j\|\tau_j|\phi|\\ &= \|\mathbf{H}\mathbf{q}_j\|\frac{\tau_j\tau_{j+1}}{\sqrt{(\tau_j)^2+(\eta_{j+1}/\sigma)^2}}\\ &\le \|\mathbf{H}\mathbf{q}_j\|\tau_{j+1},\end{aligned}$$

and

$$\begin{aligned}|\rho_j\beta_j\eta_{j+1}\psi| &\le \|\mathbf{H}\mathbf{q}_j\||\eta_{j+1}||\psi|\\ &= \|\mathbf{H}\mathbf{q}_j\|\frac{|\eta_{j+1}|\tau_{j+1}}{\sqrt{(\tau_j\sigma)^2+\eta_{j+1}^2}}\\ &\le \|\mathbf{H}\mathbf{q}_j\|\tau_{j+1}.\end{aligned}$$

Now, using standard results on floating point addition, we see that

$$\begin{aligned}|fl(\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j\phi+\rho_j\beta_j\eta_{j+1}\psi)| &\le 2\cdot max(|\mathbf{y}_j^T\mathbf{H}_j\hat{\mathbf{q}}_j\phi|, |\rho_j\beta_j\eta_{j+1}\psi|)\epsilon_M\\ &\le 2\cdot\|\mathbf{H}\mathbf{q}_j\|\tau_{j+1}\epsilon_M.\end{aligned}$$

Thus, if each previous $\mathbf{q}_i$ is constructed as above for $i = 2, 3, \ldots, j$, we have that the vector $\mathbf{y}$ in place at the j-th step will satisfy

$$|fl(\mathbf{y}^T\mathbf{H}\mathbf{q}_i)| \le 2 \cdot \|\mathbf{H}\mathbf{q}_i\|\tau_{i+1}\epsilon_M \tag{7.2}$$

for $1 \le i \le j$. Now, for $i < j$,

$$|fl(\mathbf{q}_{j+1}^T\mathbf{H}\mathbf{q}_i)| = |\gamma_{j+1}\mathbf{y}_i^T\mathbf{H}_i\hat{\mathbf{q}}_i| = |\mathbf{y}_i^T\mathbf{H}_i\hat{\mathbf{q}}_i\left(\frac{-\eta_{j+1}}{\tau_j\tau_{j+1}}\right)|. \tag{7.3}$$

Thus, for $i < j$,

$$\begin{aligned}|fl(\mathbf{q}_{j+1}^T\mathbf{H}\mathbf{q}_i)| &\le 2\cdot(\|\mathbf{H}\mathbf{q}_i\|\tau_{i+1}\epsilon_M)\left(\frac{|\eta_{j+1}|}{\tau_j\tau_{j+1}}\right)\\ &= 2\cdot\|\mathbf{H}\mathbf{q}_i\|\left(\frac{\tau_{i+1}}{\tau_j}\right)\left(\frac{|\eta_{j+1}|}{\tau_{j+1}}\right)\epsilon_M\\ &\le 2\cdot\|\mathbf{H}\mathbf{q}_i\|\epsilon_M,\end{aligned}$$

since $0 \le \frac{\tau_{i+1}}{\tau_j} \le 1$, and $0 \le \frac{|\eta_{j+1}|}{\tau_{j+1}} \le 1$.

Since a right eigenvector is also a left eigenvector in the symmetric case, this argument suffices for that case as well.

Analysis of IRA with Locking: We analyze the case of a single vector. The generalization to more than one vector is straightforward. After locking, we have

$$\mathbf{A}(\mathbf{v}_1, \mathbf{V}_2) = (\mathbf{v}_1, \mathbf{V}_2) \begin{pmatrix} \theta_1 & \mathbf{h}^T \\ 0 & \mathbf{H}_2 \end{pmatrix} + (\mathbf{f}\epsilon | \mathbf{f}\tau \mathbf{e}_{m-1}^T),$$

with $\mathbf{v}_1^T \mathbf{f} = 0$ and $\mathbf{V}_2^T \mathbf{f} = 0$.

An implicit restart step will take the form

$$\mathbf{A}\mathbf{V}_2\mathbf{Q} = (\mathbf{v}_1, \mathbf{V}_2 Q) \begin{pmatrix} \mathbf{h}^T\mathbf{Q} \\ \mathbf{Q}^T\mathbf{H}_2\mathbf{Q} \end{pmatrix} + \mathbf{f}\tau\mathbf{e}_m^T\mathbf{Q},$$

followed by truncation to a k-step factorization

$$\mathbf{A}\widehat{\mathbf{V}}_2 = (\mathbf{v}_1, \widehat{\mathbf{V}_2}) \begin{pmatrix} \hat{\mathbf{h}} \\ \widehat{\mathbf{H}} \end{pmatrix} + \hat{\mathbf{f}}\mathbf{e}_k^T.$$

This is then built out to an m-step factorization using a standard Arnoldi process with all basis vectors, including $\mathbf{v}_1$, participating. Repeated implicit restarting will usually yield convergence of an eigenvalue in $\mathbf{H}_2$ and then we will have

$$\mathbf{A}(\mathbf{v}_1, \tilde{\mathbf{V}}_2) = (\mathbf{v}_1, \tilde{\mathbf{V}}_2) \begin{pmatrix} \theta_1 & \tilde{\mathbf{h}}^T \\ 0 & \tilde{\mathbf{H}}_2 \end{pmatrix} + (\mathbf{f}\epsilon \mid \tilde{\mathbf{f}}\mathbf{e}_{m-1}^T),$$

Now, suppose there is a converged Ritz value $\theta_2 \in \sigma(\tilde{\mathbf{H}}_2)$. Then we compute an approximate eigenvector $\mathbf{x}_2 = \tilde{\mathbf{V}}_2\tilde{\mathbf{y}}$ where $\tilde{\mathbf{y}} = (\eta, \mathbf{y}^T)$ with

$$\begin{pmatrix} \theta_1 & \tilde{\mathbf{h}}^T \\ 0 & \tilde{\mathbf{H}}_2 \end{pmatrix} \begin{pmatrix} \eta \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \eta \\ \mathbf{y} \end{pmatrix} \theta_2.$$

Assuming $||\tilde{\mathbf{y}}|| = 1$, we have

$$\mathbf{A}\mathbf{x}_2 - \mathbf{x}_2\theta_2 = \mathbf{f}\epsilon\eta + \tilde{\mathbf{f}}\mathbf{e}_{m-1}^T\mathbf{y}$$

so that

$$||\mathbf{A}\mathbf{x}_2 - \mathbf{x}_2\theta_2|| \le 2\epsilon_D$$

if $||\tilde{\mathbf{f}}|||\mathbf{e}_{m-1}^T\mathbf{y}| < \epsilon_D$.

Since the orthogonality condition $\mathbf{v}_1^T\tilde{\mathbf{V}}_2 = 0$ has been enforced, it is easily seen that the deflated Arnoldi factorization is exact for a slightly perturbed problem. This is demonstrated by the observation that

$$(\mathbf{A} - \mathbf{f}\epsilon\mathbf{v}_1^T)(\mathbf{v}_1, \tilde{\mathbf{V}}_2) = (\mathbf{v}_1, \tilde{\mathbf{V}}_2) \begin{pmatrix} \theta_1 & \tilde{\mathbf{h}}^T \\ 0 & \tilde{\mathbf{H}}_2 \end{pmatrix} + \tilde{\mathbf{f}}\mathbf{e}_m^T.$$

In the symmetric case, we may either account for $\tilde{\mathbf{h}}$ or discard it. $\tilde{\mathbf{H}}_2 = \tilde{\mathbf{V}}_2^T\mathbf{A}\tilde{\mathbf{V}}_2$ is tridiagonal in any case, and a similar argument accounting for symmetry will give

$$(\mathbf{A} - \mathbf{v}_1(\tilde{\mathbf{V}}_2\tilde{\mathbf{h}})^T - (\tilde{\mathbf{V}}_2\tilde{\mathbf{h}})\mathbf{v}_1^T)\tilde{\mathbf{V}}_2 = \tilde{\mathbf{V}}_2\tilde{\mathbf{H}}_2 + \tilde{\mathbf{f}}\mathbf{e}_{m-1}^T.$$

Thus, if $\tilde{\mathbf{h}}$ is discarded we will have computed eigenvalues of a perturbed problem and the computed eigenvectors will be orthogonal to full working precision ϵ_M. On the other hand, if $\tilde{\mathbf{h}}$ is kept, then eigenvectors will have to be re-orthogonalized at the end of the computation.

```
3.6314e-01       0            0            0            0
4.2585e+00   1.3738e-01       0            0            0
1.3738e-01   3.4903e+00   4.2605e-11       0            0
    0        4.2605e-11   1.7126e+00   2.7572e+00       0
    0            0        2.7572e+00   6.4175e+00   3.1738e-01
    0            0            0        3.1738e-01   3.9050e+00
    0            0            0            0        3.3433e+00
```

FIG. 8.1. *A submatrix of the tridiagonal matrix* **H**.

8. Computational Results and Conclusions.

In this section we give examples to illustrate the error analysis of the previous section. First we give an example to illustrate the the numerical stability of the transformations for locking and purging. This will show the need for the rescaling technique. Then we illustrate the effectiveness of the locking and purging in the full IRAM iteration on some examples with multiple eigenvalues.

Numerical performance of the locking and purging transformations: The first example will illustrate deflation on a symmetric tridiagonal matrix that leads to an extreme case of small leading components in $\mathbf{y}$. The matrix $\mathbf{H}$ is a 50×50 symmetric tridiagonal matrix resulting from 50 steps of Lanczos on an order 100 matrix $\mathbf{A}$ that was derived from the central difference discretization of the 2-dimensional Laplacian on the unit square $[0,1] \times [0,1]$ with zero Dirichlet boundary conditions. The starting vector $\mathbf{v}_1$ was set to the vector of all ones and then normalized to have unit length.

In this test, the six smallest eigenvalues of $\mathbf{H}$ were locked. We show the details of locking the second smallest eigenvalue after the first locking has already been done. As we shall see, this presents a very severe test problem. The matrix $\mathbf{H}$ has a small element on the subdiagonal (and superdiagonal) after the leading block of order 15. This is illustrated in Figure 8.1.

After the first deflation, the leading 15 components of the second eigenvector to be locked are quite small (all are nearly ϵ_M in magnitude). This is illustrated in Figure 8.2.

In Table 8.3 we show a graph of the absolute values of the weights γ_j, the components of $\mathbf{Q}^T\mathbf{H}\mathbf{y}$ and their products. We see that even in this extremely severe case, the products are at the level of ϵ_M. This indicates that the size of $\mathbf{g}\mathbf{y}^T\mathbf{H}\mathbf{Q}$ is at the level of roundoff $\epsilon_M||\mathbf{H}||$ as predicted by the analysis of the previous section.

In Table 8 we show the locked values that appear on the diagonal of the matrix $\tilde{\mathbf{H}}$ after locking the lowest six eigenvalues. These appear in the first column. The second and third columns show the lowest six eigenvalues of $\mathbf{H}_1 = \mathbf{H}(1:15, 1:15)$ and of $\mathbf{H}_2 = \mathbf{H}(16:50, 16:50)$ respectively. We see that two of the locked values came from the leading block and the remaining ones came from the trailing block. Also, note that both instances of the multiple eigenvalue $7.7129e-01$ were locked successfully.

The norm of the residual after deflation was

$$||\mathbf{H}\mathbf{Q} - \mathbf{Q}\tilde{\mathbf{H}}|| = 1.7e - 14,$$

where $\mathbf{Q}$ is the product of the six orthogonal matrices used to deflate the six lowest eigenvalues of $\mathbf{H}$ and $\tilde{\mathbf{H}} = \mathbf{Q}^T\mathbf{H}\mathbf{Q}$. The leading 6×6 block of $\tilde{\mathbf{H}}$ is diagonal and the remainder tridiagonal after these locking steps.

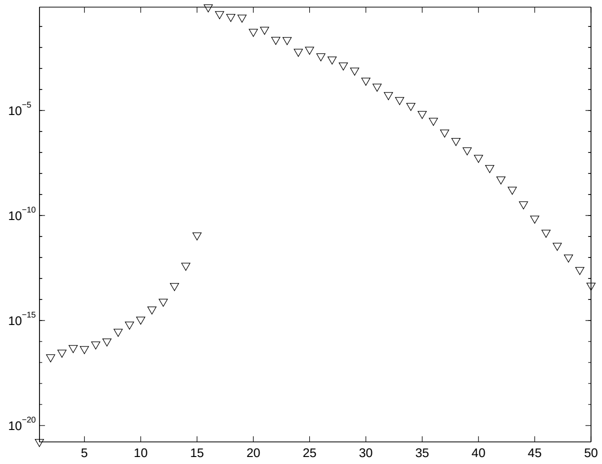

FIG. 8.2. Log of the components of $|\mathbf{y}|$

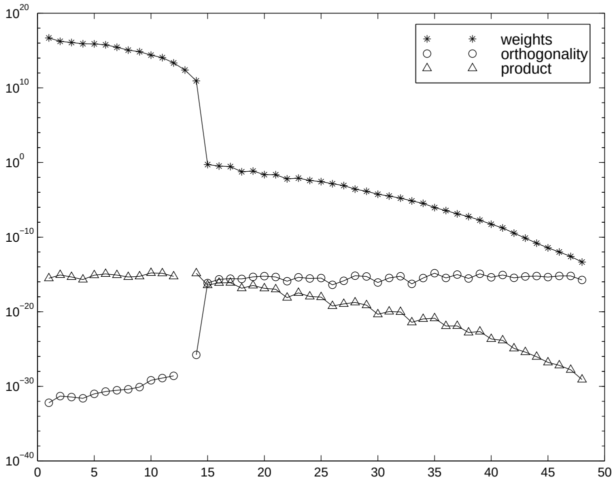

FIG. 8.3. Weights γ_j and orthogonality $\mathbf{y}^T\mathbf{H}\mathbf{q}_j$

Locked Values	Values from $\mathbf{H}_1$	Values from $\mathbf{H}_2$
1.6203e-01	1.6203e-01	3.9851e-01
3.9851e-01	7.7129e-01	6.3499e-01
6.3499e-01	1.3806e+00	7.7129e-01
7.7129e-01	1.7964e+00	1.0078e+00
7.7129e-01	2.4056e+00	1.2502e+00
1.0078e+00	2.9118e+00	1.4867e+00

TABLE 8.1
Locked Eigenvalues of $\mathbf{H}$

IRAM-iteration on $\mathbf{L}_{4096}$				
	$\rho = 5,$	$k = 8$	$p = 12$	
	M-V	M-V	Ritz Vals	Ritz Vals
ϵ_D	Products	Found	Locked	Purged
10^{-3}	661	599	13	3
10^{-5}	888	756	13	3
10^{-7}	1084	1036	12	2
10^{-9}	1487	1404	11	1
$\|\mathbf{L}_{4096}\mathbf{V}_8 - \mathbf{V}_8\mathbf{R}_8\| \approx \epsilon_D$				

TABLE 8.2
Convergence history for Convection Diffusion

Locking and purging in IRAM: The deflation strategy adopted here is considerably more conservative than the one proposed in [2]. However, the performance is comparable at the same level of accuracy. We demonstrate here that very low tolerances can be specified without missing multiple or clustered eigenvalues. We also give an indication of the computational savings resulting from this ability.

Our deflation scheme is to do the following:

1. Lock a single Ritz value each time one converges until k values have been locked.
2. Continue to iterate and lock each newly converged Ritz value that is a "better" value than the existing ones. Follow each locking operation with a purge operation to delete the least wanted but locked Ritz value.
3. Continue Step 2 until the next Ritz value to converge is not a "better" value. Replace the $k+1$-st basis vector with a randomly generated one and orthogonalize this against the previous ones and then build a new Arnoldi factorization. Repeat Step 2.
4. When Step 3 has been executed two consecutive times with no replacement of existing locked Ritz values the iteration is halted.

We shall give results of this scheme on several eigenvalue problems arising from a discrete form of 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 Dirichlet boundary conditions. We use a standard five-point scheme with centered finite differences to obtain a matrix $\mathbf{L}_{n^2}$ of order n^2 where $h = 1/(n+1)$ is the cell size. The eigenvalues of $\mathbf{L}_{n^2}$ are

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

for $1 \le i,j \le n$ where $\gamma = \rho h/2$. Of course, when $\rho = 0$ this is a discrete form of the Laplacian operator.

The results reported in Table 8.2 are for $\mathbf{L}_{4096}$ with $\rho = 5$ with $k = 8$ and $p = 12$ in the IRAM iteration. This required storage for 21 Arnoldi vectors (including the residual vector). We made runs with $\epsilon_D = 10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}$. All computations were performed on a Sun SparcStation 20 Model 61 with 64 megabytes of RAM using

MATLAB Version 5.1.0.421 [1]. Machine precision in this system is $\epsilon_M \approx 10^{-16}$. In all cases, the two single eigenvalues and the three pairs of eigenvalues with multiplicity two were computed. These are the smallest 8 eigenvalues of $\mathbf{L}_{4096}$. We made runs with various other values of ρ including $\rho = 0$ with comparable performance. When $\rho = 0$ the problem is symmetric and we used a symmetric version of the code for locking. To purge, we just needed to delete unwanted but converged values. In all cases, $||\mathbf{V}_8^T\mathbf{V}_8 - \mathbf{I}_8|| \approx \epsilon_M$ with $||\mathbf{L}_{4096}\mathbf{V}_8 - \mathbf{V}_8\mathbf{R}_8|| \approx \epsilon_D$ as desired. The column labeled "M-V Products" indicates the total number of matrix-vector products required for the computation. The column labeled " M-V Found" indicates the number of matrix-vector products required to lock all of the wanted eigenvalues. The remaining matrix-vector products were required to verify that these were all of the wanted values. This verification consisted of finding that subsequent convergence produced "worse" Ritz values than the ones which were already locked. The columns labeled "Ritz Vals Locked" and "Ritz Vals Purged" show the total number of locking and purging operations required. It is interesting to note that fewer locking and purging operations were required at the stricter tolerances. This is due to the self locking tendency of IRAM as convergence take place. The multiple instances were already present in the projected matrix by the time the initial locking took place. Hence, there were fewer instances of purging converged but unwanted Ritz values.

We also ran the case $n = 625$, $\rho = 25$, $k = 6$ and found all of the smallest eigenvalues with 372 matrix-vector products and completed the verification with a total of 480 matrix-vector products, This is comparable to the performance for the same problem reported in [2]. This new technique does not seem to improve upon the number of matrix-vector products required to lock the wanted vectors at comparable accuracy levels over the methods in [2]. However, it is possible to specify far more relaxed tolerances with the methods presented here.

The new deflation schemes developed here are more efficient than those of [2] but that is only of mild interest in this context. Usually, $k+p$ is quite small with respect to n and computations on the projected matrix $\mathbf{H}$ are inconsequential when compared to operations involving $\mathbf{A}$ or $\mathbf{V}$. However, the methods presented are very tidy. The purging operation using left eigenvectors seems quite a bit more attractive than the one presented in [2]. Surely, the most important feature of these new methods is the way in which the deflation error is structured. It is this that allows us to achieve the goal of deflating at relaxed tolerances.

Acknowledgement. The author would like to thank Prof. Trond Steihaug, and Dr. Chao Yang for reading the manuscript and making several useful suggestions.

REFERENCES

[1] R. B. Lehoucq. *Analysis and Implementation of an Implicitly Restarted Iteration.* PhD thesis, Rice University, Houston, Texas, May 1995. Also available as Technical Report TR95-13, Dept. of Computational and Applied Mathematics.

[2] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Analysis and Applications*, 17(4):789–821, October 1996.

[3] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM Publications, Phildelphia, PA, 1998.

[4] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Englewood Cliffs, N.J., 1980.

[1] Matlab is a registered trademark of the MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760, USA, tel. 508-647-7000, fax 508-647-7001, `info@mathworks.com`, `http://www.mathworks.com`.

[5] B. N. Parlett and J. Le. Forward instability of tridiagonal QR. *SIAM Journal on Matrix Analysis and Applications*, 14(1):279–316, 1993.

[6] B. N. Parlett and D. Scott. The Lanczos algorithm with selective orthogonalization. *Mathematics of Computation*, 33:217–238, 1979.

[7] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Analysis and Applications*, 13(1):357–385, January 1992.