RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

Implicit Application of Polynomial Filters in a *k*-Step Arnoldi Method

D. C. Sorensen

RIACS Technical Report 90.43
October 1990

Implicit Application of Polynomial Filters in a *k*-Step Arnoldi Method

D. C. Sorensen

The Research Institute of Advanced Computer Science is operated by
Universities Space Research Association, The American City Building, Suite 212,
Columbia, MD 21044, (301)730-2656

Work reported herein was supported in part by Coopertive Agreement NCC2-387 between the
National Aeronautics and Space Administration (NASA) and
The Universities Space Research Association (USRA)
Work was performed at the Research Institute for Advanced Computer Science (RIACS),
NASA Ames Research Center, Moffett Field, California 94035-1000

Implicit Application of Polynomial Filters in a k-Step Arnoldi Method

D. C. Sorensen*
Department of Mathematical Sciences
Rice University
Houston, Texas 77251-1829

October 18, 1990

Abstract

The Arnoldi process is a well known technique for approximating a few eigenvalues and corresponding eigenvectors of a general square matrix. Numerical difficulties such as loss of orthogonality and assessment of the numerical quality of the approximations as well as a potential for unbounded growth in storage have limited the applicability of the method. These issues are addressed by fixing the number of steps in the Arnoldi process at a prescribed value k and then treating the residual vector as a function of the initial Arnoldi vector. This starting vector is then updated through an iterative scheme that is designed to force convergence of the residual to zero. The iterative scheme is shown to be a truncation of the standard implicitly shifted QR-iteration for dense problems and it avoids the need to explicitly restart the Arnoldi sequence. The main emphasis of this paper is on the derivation and analysis of this scheme. However, there are obvious ways to exploit parallelism through the matrix-vector operations that comprise the majority of the work in the algorithm. Preliminary computational results are given for a few problems on some parallel and vector computers.

AMS classification: Primary 65F15; Secondary 65G05

Key words and phrases: Arnoldi method, eigenvalues, polynomial filter, iterative refinement, parallel computing.

*This work was supported in part by by NSF cooperative agreement CCR-8809615 and also by RIACS under the NAS systems division of NASA and DARPA cooperative agreement MCC 2-387.

1 Introduction

Large scale eigenvalue problems arise in a variety of settings. Often these very large problems arise through the discretization of a linear differential operator in an attempt to approximate some of the spectral properties of the operator. However, there are a considerable number of sources other than PDE. Saad gives a number of examples in [28].

If one hopes to solve extremely large algebraic eigenvalue problems it is not possible to rely upon the proven methods for dense matrices such as the Q-R iteration due to the expense of storage requirements and arithmetic cost of an iteration. Fortunately, it is common to be interested only in a selected subset of the spectrum of a large matrix. In the symmetric setting one is typically interested in the extremes of the spectrum (i.e. a few of the largest or smallest eigenvalues). In the non-symmetric setting one is often concerned with determining eigenvalues with largest real part.

The Lanczos method [19] is a popular algorithm for solving large symmetric eigenvalue problems . The Arnoldi process [1] is a generalization of the Lanczos method which is appropriate for finding the eigenvalues of a large non-symmetric matrix. These methods only require one to compute action of the matrix on a vector through a matrix vector product. Often this may be accomplished without explicit storage of the matrix and this property along with a number of theoretical and computational features have contributed to the widespread appeal of these methods. However, both of these share some inherent numerical difficulties which have been the subject of considerable research over the last two decades [8, 16, 25, 27].

In this paper these methods will be discussed from a new perspective. The goal is to address the non-symmetric problem and thus the focus is on the Arnoldi algorithm. However, since the Arnoldi method reduces to the Lanczos method when the matrix is symmetric, everything that is developed here is applicable to the symmetric case as well with obvious savings in computational effort available through the exploitation of symmetry. Traditionally, the point of view has been to let the Arnoldi or the Lanczos sequence develop without bound while monitoring error estimates associated with the Ritz vectors to identify converged eigenvalues. However, if one explores the relation with the QR-iteration it is apparent that the Arnoldi (Lanczos) method is really a truncated reduction of the given matrix into upper Hessenberg (tridiagonal) form. The iterative phase of the QR-method does not have an analogy within the traditional treatment of these algorithms.

A variant of the Arnoldi method which includes such an iterative phase is developed here by analogy to the well-known implicitly shifted Q-R iteration [14, 33, 35] for dense matrices. Such an analogy may be developed if one treats the residual vector as a function of the initial Arnoldi (Lanczos) vector, and then attempts to iteratively improve this vector in a way to force the residual vector to zero. As shown here, this may be done by implicit application of a polynomial filter to the starting vector on each iteration. The implicit application of this polynomial filter is accomplished through a truncated version of the implicitly shifted Q-R iteration. Within this context, an updating scheme is developed which preserves an Arnoldi (Lanczos) factorization of predetermined size. The method generalizes explicit restart methods and it is possible to implement a mathematically equivalent im-

plicit method corresponding to all of the explicitly restarted methods that this author is aware of (See Section 5).

The idea of iteratively forcing the residual to zero is not new. Variants of this idea were introduced early by Karush in [18]. Cullum and her colleagues have investigated explicit restart methods for the symmetric case [5, 6, 8]. Most recently the idea has been explored by Saad in [28,29] by Chatelin and Ho in [2] and by Chronopoulos in [3] for the nonsymmetric case. All of these techniques use eigensystem information from the projected matrix to construct an updated starting vector for the Arnoldi (Lanczos) process, and then *restart* this process from scratch. Here, a computational framework is developed which updates the Arnoldi factorization instead of re-starting it. As just mentioned, this update procedure is completely analogous to the implicitly shifted QR-iteration. It is shown here that the update procedure will implicitly apply linear polynomial factors to the starting vector in a manner that will purge the starting vector of unwanted components. In this way invariant subspaces of predetermined dimension might be found.

This approach has several advantages over more traditional approaches. The number of eigenvalues that are sought is prespecified. This fixes the storage and computational requirements instead of allowing them to become arbitrarily large. It is expected that the number of eigenvalues that are sought will be modest, and in this situation, orthogonality of the Arnoldi (Lanczos) basis for the Krylov subspace can be maintained. Therefore, the questions of spurious eigenvalues and selective re-orthogonalization do not enter. Finally, the well understood deflation rules associated with the QR iteration may be carried over directly to the technique.

2 The Arnoldi Factorization

The Arnoldi factorization may be viewed as a truncated reduction of an $n \times n$ matrix A to upper Hessenberg form. After k steps of the factorization one has

$$AV = VH + re_k^T \tag{2.1}$$

where $V \in \mathbf{R}^{n\times k}$, $V^TV = I_k$; $H \in \mathbf{R}^{k\times k}$ is upper Hessenberg, $r \in \mathbf{R}^n$ with $0 = V^Tr$. An alternative way to write (2.1) is

$$AV = (V, v)\begin{pmatrix} H \\ \beta e_k^T \end{pmatrix} \quad \text{where } \beta = \|r\| \text{ and } v = \frac{1}{\beta}r\ . \tag{2.2}$$

From this representation, it is apparent that (2.2) is just a truncation of the complete reduction

$$A(V, \hat{V}) = (V, \hat{V})\begin{pmatrix} H & M \\ \beta e_1 e_k^T & \hat{H} \end{pmatrix} \tag{2.3}$$

where $(V, \hat{V})$ is an orthogonal $n \times n$ matrix and $\hat{H}$ is an upper Hessenberg matrix of order $n - k$. Equation (2.2) and hence (2.1) may be derived from (2.3) by equating the first k columns of both sides and setting $v = \hat{V}e_1$.

The factorization (2.1) may be advanced one step through the following

recursion formulas:

$$(2.3.1)\quad \beta = \|r\| \,;\quad v = \tfrac{1}{\beta} r \,;$$

$$(2.3.2)\quad V_+ = (V, v) \,;$$

$$(2.3.3)\quad w = Av \,;\quad \begin{pmatrix} h \\ \alpha \end{pmatrix} = {V_+}^T w \,;$$

$$(2.3.4)\quad H_+ = \begin{pmatrix} H & h \\ \beta e_k^T & \alpha \end{pmatrix} \,;$$

$$(2.3.5)\quad r_+ = w - V_+ \begin{pmatrix} h \\ \alpha \end{pmatrix} = (I - V_+ V_+^T) w \,.$$

From this development it is easily seen that

$$AV_+ = V_+ H_+ + r_+ e_{k+1}^T \,,\quad V_+^T V_+ = I_{k+1} \,, V_+^T r_+ = 0 \,.$$

In a certain sense, computation of the projection indicated at Step (2.3.5) has been the main source of research activity in this topic. The computational difficulty stems from the fact that $\|r\| = 0$ if and only if the columns of V span an invariant subspace of A. When V "nearly" spans such a subspace $\|r\|$ will be small. Typically, in this situation, a loss of significant digits will take place at Step (2.3.5) through numerical cancellation unless special care is taken. On the one hand, it is a delightful situation when $\|r\|$ becomes small because this indicates that the eigenvalues of H are accurate approximations to the eigenvalues of A. On the other hand, this "convergence" will indicate a probable loss of numerical orthogonality in V. The identification of this phenomenon and the first rigorous numerical treatment is due to Paige[22,23]. There have been several approaches to overcome this problem :

(1) *Complete Re-orthogonalization.*

This may be accomplished through maintaining V in product Householder form [15, 34]. It may also be accomplished through the Modified Gram-Schmidt processes with re-orthogonalization [9, 26]. More will be said of these alternatives later.

(2) *Selective Re-orthogonalization.*

This option has been proposed by Parlett and has been heavily researched by him and his students. Most notably, the thesis and subsequent papers and computer codes of Scott have developed this idea [24, 25, 31]. The general scheme is described in [25].

(3) *No Re-orthogonalization.*

This last option introduces the almost certain possibility of introducing spurious eigenvalues. Various techniques have been developed to detect the presence of spurious eigenvalues [7, 8]. However, they do not appear when even a modest level of linear independence has been imposed on the Arnoldi vectors.

Computational cost has been cited as the reason for not employing complete orthogonalization of the Arnoldi (or Lanczos) vectors. However, the cost will be reasonable if one is able to fix k at a modest size and then update the starting vector $v_1 = Ve_1$ while repeatedly doing k-Arnoldi steps. This approach has been explored to some extent in [2, 28]. In the symmetric case Cullum [6] relates a variant of this approach (which has been termed an s-Step method) to applying a fixed number of conjugate gradient steps to

a minimize (maximize) $\langle VV^T, A\rangle$ where $\langle B, A\rangle = trace(B^T A)$ is the Frobenius product functional with V restricted to the generalized block Krylov subspace. However, while this argument gives considerable credence to the restart procedure, it does not establish convergence.

Throughout the remainder of this paper, the k-step approach will be developed from a different point of view. An attempt will be made to iteratively update v_1 in order to force the residual vector $r(v_1)$ to zero. In order to make sense of this it will be necessary to understand when r is indeed a function of v_1 and also to determine its functional form and characterize the zeros of this function.

The classic simple result that explains when r is a function of v_1 is the Implicit Q-Theorem.

Theorem 2.4 *Suppose*

$$\begin{aligned} AV &= VH + re_k^T \\ AQ &= QG + fe_k^T \end{aligned}$$

where Q, V have orthonormal columns and G, H are both upper Hessenberg with positive subdiagonal elements.

If $Qe_1 = Ve_1$ and $Q^T f = V^T r = 0$, then $Q = V$, $G = H$, and $f = r$.

Proof: There is a straightforward inductive proof (or see [16,p367]). □

Of course the Krylov space

$$\mathcal{K}_k(A, v_1) = \text{Span } \{v_1, Av_1, A^2v_1, \ldots, A^{k-1}v_1\}$$

plays an important role along with the Krylov matrix

$$K = (v_1, Av_1, \ldots, A^{k-1}v_1) .$$

An alternate derivation of the Arnoldi process is to consider the companion (or Frobenius) matrix

$$F \equiv \begin{pmatrix} 0 & \gamma_0 \\ I & \hat{g} \end{pmatrix} = \begin{pmatrix} 0 & & & \gamma_0 \\ 1 & & & \gamma_1 \\ & 1 & & \\ & & \ddots & \vdots \\ & & 1 & \gamma_{k-1} \end{pmatrix}$$

and to observe that

$$AK - KF = \hat{r}e_k^T \tag{2.5}$$

where $\hat{r} = A^k v_1 - Kg$ with $g^T = (r_0, \hat{g}^T)$. Note that $\hat{r} = \hat{p}(A)v_1$ where $\hat{p}(\lambda) = \lambda^k + \sum_{j=0}^{k-1} \gamma_j \lambda^j$, and also that $\hat{p}(\lambda)$ is the characteristic polynomial of F. If g is chosen to solve $\min \|A^k v_1 - Kg\|_2$ then $\hat{r}$ is orthogonal to all vectors in $\mathcal{K}_k(A, v_1)$. Moreover $\hat{p}$ solves $\min_{p \in \mathcal{PM}_k} \{\|p(A)v_1\|\}$ where $\mathcal{PM}_k$ is the set of all monic polynomials of degree k.

To solve the minimization problem in (2.5), one would factor $K = QR$ where Q is orthogonal, R is upper triangular. Note that R is nonsingular if and only if K has linearly independent columns and that Q may be constructed so that $\rho_{jj} = e_j^T R e_j > 0$. One then solves

$$g = R^{-1}Q^T A^k v_1 .$$

This choice of g will minimize the residual and also will assure that $0 = Q^T \hat{r}$. Multiplying (2.5) on the right by R^{-1} gives

$$A(KR^{-1}) - (KR^{-1})RFR^{-1} = \hat{r}e_k^T R^{-1} ,$$

i.e.

$$AQ - QG = fe_k^T \tag{2.6}$$

where $Q = KR^{-1}$, $G = RFR^{-1}$ is upper Hessenberg with the same characteristic polynomial as F, and $f = \frac{1}{\rho_{kk}}\hat{r}$. It is easily verified that $v_1 = Qe_1 = Ve_1$, and $0 = Q^T f$. Thus, the Implicit Q-Theorem will imply that $Q = V$, $G = H$, and $f = r$. Putting $H = G$ yields

$$\beta_j \equiv e_{j+1}^T H e_j = e_{j+1}^T RFR^{-1} e_j = \frac{\rho_{j+1,j+1}}{\rho_{jj}} .$$

Moreover,

$$\frac{1}{\rho_{jj}}\|\hat{p}_j(A)v_1\| = \beta_j = \frac{\rho_{j+1,j+1}}{\rho_{jj}} .$$

gives

$$\rho_{j+1,j+1} = \|\hat{r}_j\| = \|\hat{p}_j(A)v_1\| .$$

This discussion establishes the following.

Theorem 2.7 *Let $AV_j = V_j H_j + r_j e_j^T$ be a sequence of successive Arnoldi steps $1 \leq j \leq k$ and suppose that* $\dim(\mathcal{K}_k(A, v_1)) = k$. *Then*

$$(1) \qquad r_j = \frac{1}{\|\hat{p}_{j-1}(A)v_1\|}\hat{p}_j(A)v_1 , \quad \beta_j = \frac{\|\hat{p}_j(A)v_1\|}{\|\hat{p}_{j-1}(A)v_1\|}$$

where $\hat{p}_j(\lambda)$ is the characteristic polynomial of H_j. Moreover,

$$(2) \qquad \hat{p}_j \text{ solves } \min_{p \in \mathcal{PM}_j}\{\|p(A)v_1\|\}$$

for $1 \leq j \leq k$.

The development leading to Theorem (2.7) follows and builds upon the development by Ruhe in [27]. The fact that $\|\hat{p}_k(A)v_1\|$ (the characteristic

polynomial of H_k acting on v_1) will minimize $\|p(A)v_1\|$ over all monic polynomials of degree k was proved by Saad in [29]. Theorem (2.7) points out that this minimum principle is is not useful in assessing the behavior of the residual obtained through the Arnoldi process.

Since $r_k = 0$ if and only if v_1 is in the null space of some k-degree polynomial in A, it is likely that $r_k = 0$ if and only if v_k is the sum of k-eigenvectors of A. We establish this result in the following.

Theorem 2.8 *Let $AV_k - V_kH_k = r_ke_k^T$ be a k-step Arnoldi factorization of A, with $r_j \neq 0$, $0 \leq j \leq k-1$. Then $r_k = 0$ and H_k is diagonalizable if and only if $v_1 = \sum_{i=1}^{k} x_i$, where $\{x_i\}$ is a set of k linearly independent eigenvectors for A.*

Proof: Suppose $v_1 = \sum_{j=1}^{k} x_j$, where $\{x_j\}$ are a set of linearly independent eigenvectors for A. Then

$$\mathcal{K}_{k+1}(A, v_1) \equiv \text{Span } \{v_1, Av_1, \ldots, A^k v_1\} \subset \text{Span } (\{x_j\})$$

and since $r_{k-1} \neq 0 \Rightarrow \hat{p}_{k-1}(A)v_1 \neq 0$

$$\|r_k\| = \frac{1}{\|\hat{p}_{k-1}(A)v_1\|}\|\hat{p}_k(A)v_1\| = 0$$

must hold because the $k+1$ vectors $\{v_1, Av_1, \ldots, A^k v_1\}$ lie in a k-dimensional subspace. Moreover, $r_j \neq 0$ for $0 \leq j \leq k-1$ implies $\dim \mathcal{K}_k(A, v_1) = k$ and since $\mathcal{K}_k(A, v_1) \subset \text{Span } \{x_j\}$ these two subspaces must be identical. Thus the columns of V_k form a basis for Span $\{x_j\}$ and hence $x_j = V_k y_j$ for $1 \leq j \leq k$. It follows easily now from the Arnoldi factorization that $\{y_j\}$ is a set of k linearly independent eigenvectors for H.

Suppose now that $r_k = 0$ and H_k is diagonalizable. Then

$$AV_k y_j = V H_k y_j = \lambda_j V_k y_j$$

for every eigenpair (y_j, λ_j) of H_k. Since H_k is diagonalizable, $\{y_j\}$ hence $\chi \equiv \{x_j : x_j = V_k y_j\}$ is a linearly independent set of eigenvectors for A. The set of vectors χ must be a basis for $\mathcal{K}_k(A, v_1)$ so $v_1 = \sum_{j=1}^{k} \theta_j x_j$. If any $\theta_j = 0$ for $1 \leq j \leq k$, then v_1 would be the sum of fewer than k eigenvectors so r_j would vanish for some $1 \leq j \leq k-1$ by the first part of the proof. □

The presence of non-trivial Jordan blocks in H can be dealt with by introducing generalized eigenvectors. There is an analogous statement and proof but the details are tedious.

Now that the nature of the residual has been exposed and now that a criterion for this residual to vanish has been set forth it is possible to devise algorithms to accomplish this goal. The point of view that shall be taken for the derivation of these algorithms has considerable analogy with the standard QR-iteration. In the next section this iteration is discussed in a framework that will aid in the derivation of the new algorithms.

3 Relation to the QR Algorithm

In order to motivate the point of view put forth in the remainder of this paper, it will be instructive to derive and analyze the QR iteration from a certain point of view.

To do this, suppose that there has been a complete reduction of A to

upper Hessenberg form. Thus

$$(3.1) \qquad AV - VH = 0\,, \quad V^TV = I_n\,, \quad H\text{-upper Hessenberg.}$$

The explicitly shifted QR algorithm consists of the following four steps. Let μ be the shift and let $(H - \mu I) = QR$ with Q orthogonal and R upper triangular. Then

$$(3.1.1) \quad (A-\mu I)V - V(H-\mu I) = 0$$
$$(3.1.2) \quad (A-\mu I)V - VQR = 0$$
$$(3.1.3) \quad (A-\mu I)(VQ) - (VQ)(RQ) = 0$$
$$(3.1.4) \quad A(VQ) - (VQ)(RQ+\mu I) = 0$$

After these four steps we have updated (3.1) to produce

$$(3.2) \qquad AV_+ - V_+H_+ = 0$$

where $V_+ = VQ$, and $H_+ = RQ + \mu I$ is upper Hessenberg. Note that from (3.1.2) and (3.2) it follows that

$$(A-\mu I)v_1 = v_1^+\rho_{11}$$

where $\rho_{11} = e_1^TRe_1$, $v_1^+ = V_+e_1$. Moreover, from (3.1.3)

$$(VQ)^{-1}(A-\mu I)^{-1} - (RQ)^{-1}(VQ)^{-1} = 0\,.$$

Hence

$$Q^TV^T(A-\mu I)^{-1} - Q^TR^{-1}V_+^T = 0\,,$$

i.e.

$$V^T(A-\mu I)^{-1} - R^{-1}V_+^T = 0$$

so that

$$(A - \mu I)^T v_n^+ = \rho_{nn} v_n$$

where $v_n = Ve_n$, $v_n^+ = V_+ e_n$. This proves the well known that the QR iteration is performing inverse iteration [33] with respect to A^T on the last column of V.

An implicitly shifted QR step starting with (3.1) consists of

$$A(VQ) - (VQ)(Q^T H Q) = 0 \tag{3.3}$$

where the orthogonal matrix Q is computed as a product of Givens transformations which are specified implicitly through the well known "bulge chase" sequence as described in [25,p159, 33] once the shift μ is specified. From the previous discussion, the application of p implicit shifts will result in the implicit application of a polynomial ψ of degree p to the vector v_1. Thus once the p shifts have been applied

$$AV_+ - V_+ H_+ = 0 \tag{3.4}$$

where $V_+ = VQ_1Q_2 \cdots Q_p$, $H_+ = Q_p^T \cdots Q_2^T Q_1^T H Q_1 Q_2 \cdots Q_p$ with $v_1^+ \equiv V_+ e_1$ satisfying

$$v_1^+ = \psi(A) v_1$$

where $\psi(\lambda) = \frac{1}{\tau} \prod_{j=1}^{p} (\lambda - \mu_j)$ with τ a normalizing factor to make $\|v_1^+\| = 1$ and $\{\mu_j\}$ the set of p implicit shifts.

From this point of view, one may interpret the QR iteration as a process of rapidly determining an approximate root μ of the characteristic polynomial and then applying the linear factor $A - \mu I$ to v_1 to replace it with $v_1^+ \leftarrow \frac{1}{\tau}(A -$

$\mu I)v_1$ in order to purge the starting vector of components along eigenvectors associated with μ. As the iteration proceeds, subdiagonal elements of H must tend to zero according to Theorem (2.8).

In the next section, the mechanics of applying an implicit shift to an Arnoldi factorization will be developed. Eventually, the goal will be to choose shifts in a way that will damp or filter out the components of unwanted eigenvectors in the starting vector v_1. In the large scale setting, it is not viable to apply an implicit shift corresponding to each unwanted eigenvalue. Therefore, polynomial filtering techniques will have to be utilized.

4 Updating the Arnoldi Factorization

In this section a direct analogue of the QR iteration will be derived. This will lead to an updating formula that can be used to implement an iterative technique for forcing the residual r_k to zero by iteratively forcing v_1 into a subspace spanned by k eigenvectors.

Throughout this discussion, the integer k should be thought of as a fixed pre-specified integer of modest size. Let p be another positive integer, and consider the result of $k+p$ steps of the Arnoldi process applied to A which has resulted in the construction of an orthogonal matrix V_{k+p} such that

$$
\begin{aligned}
AV_{k+p} &= V_{k+p}H_{k+p} + r_{k+p}e_{k+p}^T \\
&= (V_{k+p}, v_{k+p+1}) \begin{pmatrix} H_{k+p} \\ \beta_{k+p}e_{k+p}^T \end{pmatrix} .
\end{aligned} \tag{4.1}
$$

An analogy of the explicitly shifted QR algorithm may be applied to this truncated factorization of A. It consists of the following four steps. Again,

let μ be the shift and let $(H - \mu I) = QR$ with Q orthogonal and R upper triangular. Then (putting $V = V_{k+p},\ H = H_{k+p}$)

$$(4.1.1)\quad (A - \mu I)V - V(H - \mu I) = r_{k+p}e_{k+p}^T$$

$$(4.1.2)\quad (A - \mu I)V - VQR = r_{k+p}e_{k+p}^T$$

$$(4.1.3)\quad (A - \mu I)(VQ) - (VQ)(RQ) = r_{k+p}e_{k+p}^T Q$$

$$(4.1.4)\quad A(VQ) - (VQ)(RQ + \mu I) = r_{k+p}e_{k+p}^T Q$$

Note that just as in (3.1.1) - (3.1.4), if one takes $V_+ = VQ$ and $H_+ = RQ + \mu I$, then H_+ is upper Hessenberg and

$$(A - \mu I)v_1 = v_1^+ \rho_{11}$$

where $\rho_{11} = e_1^T R e_1$, $v_1^+ = V_+ e_1$ just as before so long as $e_{k+p}^T Q e_1 = 0$. Since Q is an upper Hessenberg matrix of order $_{k+p}$. This idea may be extended for up to p shifts being applied successively. The development will continue using the implicit shift strategy.

The application of a QR iteration corresponding to an implicit shift μ produces an upper Hessenberg orthogonal $Q \in \mathbf{R}^{k+p}$ such that

$$AV_{k+p}Q = (V_{k+p}Q\ ,\ v_{k+p+1}) \begin{pmatrix} Q^T H_{k+p} Q \\ \beta_{k+p} e_{k+p}^T Q \end{pmatrix}.$$

An application of p implicit shifts therefore results in

$$(4.2)\qquad AV_{k+p}^+ = (V_{k+p}^+\ ,\ v_{k+p+1}) \begin{pmatrix} H_{k+p}^+ \\ \beta_{k+p} e_{k+p}^T \hat{Q} \end{pmatrix}$$

where $V_{k+p}^+ = V_{k+p}\hat{Q}$, $H_{k+p}^+ = \hat{Q}^T H_{k+p}\hat{Q}$, and $\hat{Q} = Q_1 Q_2 \cdots Q_p$, with Q_j the orthogonal matrix associated with the shift μ_j.

Now, partition

$$V_{k+p}^{+} = (V_k^{+}, \hat{V}_p)\,, \quad H_{k+p}^{+} = \begin{pmatrix} H_k^{+} & M \\ \hat{\beta}_k e_1 e_k^T & \hat{H}_p \end{pmatrix}\,, \tag{4.3}$$

and note

$$\beta_{k+p} e_{k+p}^T \hat{Q} = (\underbrace{0, 0, \ldots, \tilde{\beta}_{k+p}}_{k}, \underbrace{b^T}_{p})\,.$$

Substituting into (4.2) gives

$$A(V_k^{+}, \hat{V}_p) = (V_k^{+}, \hat{V}_p, v_{k+p+1}) \begin{bmatrix} H_k^{+} & M \\ \hat{\beta}_k e_1 e_k^T & \hat{H}_p \\ \tilde{\beta}_{k+p} e_k^T & b^T \end{bmatrix}\,. \tag{4.4}$$

Equating the first k columns on both sides of (4.4) gives

$$AV_k^{+} = V_k^{+} H_k^{+} + r_k^{+} e_k^T \tag{4.5}$$

so that

$$AV_k^{+} = (V_k^{+}, v_{k+1}^{+}) \begin{pmatrix} H_k^{+} \\ \beta_k^{+} e_k^T \end{pmatrix} \tag{4.6}$$

where $v_{k+1}^{+} = \frac{1}{\beta_k^{+}} r_k^{+}$, $r_k^{+} \equiv (\hat{V}_p e_1 \hat{\beta}_k + v_{k+p+1} \tilde{\beta}_{k+p})$ and $\beta_k^{+} = \|r_k^{+}\|$. Note that $(V_k^{+})^T \hat{V}_p e_1 = 0$ and $(V_k^{+})^T v_{k+p+1} = 0$ so $(V_k^{+})^T v_{k+1}^{+} = 0$. Thus (4.6) is a legitimate Arnoldi factorization of A. Using this as a starting point it is possible to use p additional steps of the Arnoldi recursions (2.3.1) - (2.3.5) to return to the original form (4.1). This requires only p evaluations of a matrix-vector products involving the matrix A and the p-new Arnoldi vectors. This is to be contrasted with the Tchebyshev-Arnoldi method of Saad [28] where the entire Arnoldi sequence is restarted. From the standpoint of numerical stability this updating scheme has several advantages:

(1) Orthogonality can be maintained since the value of k is modest.

(2) There is no question of spurious eigenvalues.

(3) There is a fixed storage requirement.

(4) The deflation techniques associated with the QR-iteration for dealing with numerically small subdiagonal elements of H_k may be taken advantage of directly.

For the sake of clarity, the Arnoldi iteration and the updating procedure will be defined:

Algorithm 4.7

function $[H, V, r] = Arnoldi\ (A, H, V, r, k, p)$

Input: $AV - VH = re_k^T$ *with* $V^TV = I_k$, $V^Tr = 0$.

Output: $AV - VH = re_k^T$ *with* $V^TV = I_{k+p}$, $V^Tr = 0$.

(1) For $j = 1, 2, \ldots, p$

(1) $\beta \leftarrow \|r\|$; *if* $\beta < tol$ *then stop;*

(2) $H \leftarrow \begin{pmatrix} H \\ \beta e_{k+j-1}^T \end{pmatrix}$; $v \leftarrow \frac{1}{\beta} r$; $V \leftarrow (V, v)$;

(3) $w \leftarrow Av$;

(4) $h \leftarrow V^Tw$; $H \leftarrow (H, h)$;

(5) $r \leftarrow w - Vh$;

(6) while $\|s\| > \epsilon\|r\|$;

(1) $s = V^Tr$;

(2) $r \leftarrow r - Vs$;

(3) $h \leftarrow h + s$;

Remark 1: Step (1.6) is Gram Schmidt with iterative refinement to assure orthogonality [9]. For details of implementation see Reichel and Gragg [26]. Computational experience with this device indicates that it is sufficient to do just one step of iterative refinement.

With the basic Arnoldi factorization defined, it is possible to describe the complete iteration:

Algorithm 4.8

function $[V, H, r] = Arnupd\ (A, k, p, tol)$.

(1) initialize $V(:,1) = v_1$; $H \leftarrow ({v_1}^T A v_1)$; $r \leftarrow Av_1 - v_1 H$;

(2) $[H, V, r] \leftarrow Arnoldi\ (A, H, V, r, 1, k)$

(3) For $m = 1, 2, \ldots$

(1) if $(\|r\| < tol)$ *then stop;*

(2) $[V, H, r] \leftarrow Arnoldi\ (A, H, V, r, p)$

(3) $u = Shifts\ (H, p)$

(4) $Q \leftarrow I_{k+p}$;

(5) for $j = 1, 2, \ldots, p$

(1) $H \leftarrow Q_j^T H Q_j$; *(Bulge-Chase corresponding to shift* $\mu_j = u(j)$*)*

(2) $Q \leftarrow QQ_j$:

(6) $v \leftarrow (VQ)e_{k+1}$; $V \leftarrow (VQ)\begin{pmatrix} I_k \\ 0 \end{pmatrix}$;

(7) $r \leftarrow (v\beta_k + r\sigma_k)$; *where* $\beta_k = e_{k+1}^T H e_k$, $\sigma_k = e_{k+p}^T Q e_k$

Remark 2: The Bulge Chase at step (3.4.1) is defined implicitly as usual so that $H - \mu_j I = Q_j R_j$; if the shifts are in complex conjugate pairs then the implicit double shift can be implemented to avoid complex arithmetic as usual.

Remark 3: During a Bulge Chase sweep at step (3.4.1), it may happen that a sub-diagonal element β_j becomes small. The deflation strategies associated with the QR algorithm are then employed. In this case,

$$H = \begin{pmatrix} H_j & M \\ \beta_j e_1 e_j{}^T & \hat{H}_j \end{pmatrix} \simeq \begin{pmatrix} H_j & M \\ 0 & \hat{H}_j \end{pmatrix} , \quad VQ = (V_j, \hat{V}_j) .$$

Thus, an invariant subspace of dimension j has been found. If $j \geq k$ then the iteration is halted. Otherwise H_j, V_j are retained and the iteration proceeds with V_j, H_j filling the role of V, H respectively.

As discussed at the beginning of this section, each application of an implicit shift μ_j will replace the starting vector v_1 with $(A - \mu_j I)v_1$. Thes after completion of each cycle of the loop at Step 2 in Algorithm (4.8):

$$Ve_1 = v_1 \leftarrow \psi(A)v_1 ;$$

where $\psi(\lambda) = \frac{1}{\tau}\prod_{j=1}^{p}(\lambda - \mu_j)$. Numerous choices are possible for the selection of these p shifts. Some possibilities will be discussed in Section 5. However, there is one immediate possibility to discuss and that is the case of choosing p "exact" shifts with respect to H. Thus the selection process might be

Algorithm 4.9

function [u] = Shifts (H, p)

(1) Compute $\lambda(H)$ *(by QR for example)*

(2) Select p unwanted *eigenvalues* $\{u(j) \leftarrow \mu_j : 1 \leq j \leq p\} \subset \lambda(H)$

Some obvious criterion for this selection might be

(i) Sort $\lambda(H)$ according to algebraically largest real part and discard the p smallest;

(ii) Sort $\lambda(H)$ according to largest modulus and discard the p eigenvalues of smallest modulus;

Selecting these exact shifts has interesting consequences in the iteration.

Lemma 4.10 *Let* $\lambda(H) = \{\theta_1, \ldots, \theta_k\} \cup \{\mu_1, \ldots, \mu_p\}$ *and let*

$$H_+ = Q^T H Q$$

where $Q = Q_1 Q_2 \cdots Q_p$ *with* Q_j *implicitly determined by the shift* μ_j. *If* $\beta_j \neq 0 \quad 1 \leq j \leq k-1$ *then* $\beta_k = 0$ *and*

$$H_+ = \begin{pmatrix} H_k^+ & M^+ \\ 0 & R_p \end{pmatrix}$$

where $\lambda(H_k^+) = \{\theta_1, \ldots, \theta_k\}$, $\lambda(R_p) = \{\mu_1, \mu_2, \ldots, \mu_p\}$. *Moreover,*

$$v_1^+ = VQe_1 = \sum y_j$$

where each y_j *is a "Ritz" vector corresponding to the Ritz value* θ_j *i.e.* $y_j = Vs_j$ *where* $Hs_j = s_j\theta_j \quad 1 \leq j \leq k$.

Proof: Note $HI = IH$ and after applying the p implicit shifts we have

$$HQ = QH_+$$

so that

$$q_1 \equiv Qe_1 = \psi(H)e_1 \,, \quad \psi(\lambda) = \frac{1}{\tau}\prod_{j=1}^{p}(\lambda - \mu_j) \,.$$

Therefore $q_1 = \sum_{j=1}^{k} s_j\zeta_j$ where $Hs_j = s_j\theta_j$ since $q_1 = \psi(H)e_1$ has annihilated any component of e_1 along an eigenvector of H associated with $\mu_j \,, \quad 1 \le j \le p$. As a consequence of Theorem (2.8), $\beta_k = 0$ must hold. Moreover, $v_1^+ = VQe_1 = Vq_1 = \sum_{j=1}^{k} Vs_j\zeta_j = \sum_{j=1}^{k} y_j\zeta_j$. □

This lemma provides a very nice interpretation of the iteration when exact shifts are chosen. Casting out the unwanted set of eigenvalues using exact shifts is mathematically equivalent to restarting the Arnoldi Factorization from the beginning after updating $v_1 \leftarrow \sum y_j\zeta_j$ a linear combination of Ritz vectors associated with the "wanted" eigenvalues. Thus the updated starting vector has been implicitly replaced by the sum of k approximate eigenvectors.

If A is symmetric and the p algebraically smallest eigenvalues of H are selected for deletion then this method is equivalent to the single vector s-step Lanczos process described by Cullum and Donath in [5] and expanded on in [6, 8]. This variant has the advantage that a restart of the entire Lanczos sequence is not required. Instead, it is updated in place and orthogonality is maintained.

5 Some Polynomial Filters

The previous discussion has indicated that it would be advantageous to construct polynomials $\psi(\lambda)$ of degree p which filter out certain portions of the spectrum of A. Several researchers have considered such schemes [5,8,28]. Related ideas appear throughout the literature of iterative methods for linear systems [17,21,30].

A particularly appealing polynomial filter may be constructed using Tchebychev polynomials. In this case, one constructs an ellipse containing the unwanted eigenvalues of H then at step (2.2) of Algorithm 4.9 the shifts μ_j are taken to be the zeroes of the Tchebyshev polynomial of degree p associated with this ellipse (i.e. the polynomial of degree p which gives the best approximation to 0 in the max norm). Construction of such an ellipse and the associated polynomials is discussed by Saad [29] and is based on Manteuffel's scheme[20]. Variants of this are presented and discussed by Chatelin and Ho in [2].

An alternative is to use exact shifts as described earlier in Section 4. When $A \in \mathbf{R}^{n \times n}$ one should take these exact shifts in complex conjugate pairs in order to avoid complex arithmetic by using the implicit double shift technique. This use of exact shifts is quite effective in the symmetric case and may be analyzed in that setting. That analysis will be done in the next section.

One may observe that both filters will have the feature of weighting extreme eigenvalues most heavily. An alternative is to construct polynomial approximations to step functions which take the value zero in unwanted

regions and one in wanted regions of the complex plane. One also might construct polynomials which produce an updated v_1^+ which is a weighted linear combination of approximate eigenvectors corresponding to the wanted eigenvalues.

In order to construct these sorts of filters it is advantageous to be able to apply the filter polynomial which is specified by its coefficients when expanded in the basis of polynomials constructed through the Arnoldi (Lanczos) process. To make this more precise, suppose ψ is any polynomial of degree less than or equal to p. Then expand ψ in the form

$$\psi(\lambda) = \sum_{j=1}^{p+1} \eta_j p_{j-1}(\lambda)$$

where $\{p_j\}$ are the Arnoldi (Lanczos) polynomials. Observe that

$$\psi(A)v_1 = Vy$$

where $y^T = (\eta_1, \eta_2, ..., \eta_{p+1}, 0, 0, ..., 0)$ since

$$Vy = \sum_{j=1}^{p+1} v_j \eta_j = \sum_{j=1}^{p+1} \eta_j p_{j-1}(A)v_1, \quad v_j = p_{j-1}(A)v_1.$$

Unfortunately, the technique developed in Section 4 for the implicit application of $\psi(A)$ to v_1 is not directly applicable because the roots of ψ are unknown. However, there is an analogous way to apply this polynomial implicitly. Assume that $\|y\| = 1$ and construct a vector w_o such that

$$\left(I - 2w_o {w_o}^T\right) e_1 = y. \tag{5.1}$$

Replace H by

$$\hat{H} = \left(I - 2w_o {w_o}^T\right) H \left(I - 2w_o {w_o}^T\right). \tag{5.2}$$

Now, apply the Householder reduction of $\hat{H}$ to upper Hessenberg form so that

$$\hat{H} \leftarrow Q^T H Q$$

where

$$(5.3) \qquad Q = \left(I - 2w_o {w_o}^T\right)\left(I - 2w_1 {w_1}^T\right) \ldots \left(I - 2w_{k+p-2} {w_{k+p-2}}^T\right)$$

with each $\left(I - 2w_j {w_j}^T\right)$ being a Householder transformation constructed to introduce zeros below the $(j+1) - st$ element of the $j - th$ column. Now, consider the application of Q to the Arnoldi Factorization:

$$AVQ - VQ(Q^T H Q) = r {e_{k+p}}^T Q$$

In order to fit within the updating framework developed in Section 4, the condition

$${e_{k+p}}^T Q e_j = 0, 1 \leq j < k.$$

must hold. This is established by the following

Lemma 5.2 *The matrix Q displayed in (5.3) satisfies ${e_{k+p}}^T Q e_j = 0$, $1 \leq j < k$.*

Proof: Let $Q_j = I - 2w_j {w_j}^T$ for $0 \leq j \leq k+p-2$, and let $H^{(j+1)} = {Q_j}^T H^{(j)} Q_j$ with $H^{(o)} = H$. From (5.1) it follows that $w_o = \theta(y - e_1)$, with $\frac{1}{\theta} = \|y - e_1\|$. Thus, ${e_i}^T Q_o = {e_i}^T$ for $i > p+1$. Since

$$Q_o H^{(1)} = H Q_o$$

and since H is upper Hessenberg, it follows that

$${e_i}^T H^{(1)} = {e_i}^T H Q_o = {e_i}^T H$$

for $i > p+2$. From this one may conclude that ${e_i}^T w_1 = 0$ for $i > p+2$ and thus ${e_i}^T Q_1 = {e_i}^T$ for $i > p+2$. Now, suppose that ${e_i}^T Q_j = {e_i}^T$ and that ${e_i}^T H^{(j)} = {e_i}^T H$for $i > p+j+1$. Since $Q_j H^{(j+1)} = H^{(j)} Q_j$ it follows that

$$ {e_i}^T H^{(j+1)} = {e_i}^T H^{(j)} Q_j = {e_i}^T H $$

for $i > p+j+2$, and again, one may conclude that ${e_i}^T w_{j+1} = 0$ so that ${e_i}^T Q_{j+1} = {e_i}^T$ for $i > p+j+2$. This inductive argument continues to hold until $j = k-1$. Hence,

$$ {e_{k+p}}^T Q = {e_{k+p}}^T Q_{k-1} Q_k \cdots Q_{k+p-2} $$

Now, observe that $Q_i e_j = e_j$ for $k-1 \leq i \leq k+p-2$ and for $1 \leq j < k$ to establish the result. □

This observation allows the application of a polynomial filter when the polynomial can be expanded in the Arnoldi basis. It provides an opportunity to implement at some interesting options. The idea is to construct a weighted linear combination of approximate eigenvectors corresponding to the wanted eigenvalues of the matrix A. One may do this by taking a linear combination of the eigenvectors of the leading $p \times p$ principal submatrix of H corresponding to the k eigenvalues of this matrix that are the best approximations to the wanted spectrum. It has been assumed that $p \geq k$. Let these vectors be $\hat{y}_j$ and form

$$ \hat{y} = \sum_{j=1}^{k} \hat{y}_j \gamma_j $$

and put $y^T = \frac{1}{\|\hat{y}\|}(\hat{y}^T, 0)$. Note that

$$ Vy = \frac{1}{\|\hat{y}\|} \sum_{j=1}^{k} V_p \hat{y}_j \gamma_j $$

and the vectors $V_p\hat{y}_j$ are the approximate eigenvectors constructed by the Arnoldi process from the Krylov subspace of dimension p. In [28] Saad discuses some heuristics for choosing the weights γ_j. One possibility is to alternate the application of this polynomial with the application of a Tchebychev polynomial or with the polynomial constructed with exact shifts. Let us denote that polynomial by $\hat{\psi}$ then it would be natural to take

$$\gamma_j = 1/\hat{\psi}(\theta_j)$$

where the θ_j are the approximate eigenvalues in the wanted spectrum. The rational for such a choice is that after these two successive steps

$$\begin{aligned} {v_1}^{++} &= \psi(A)\hat{\psi}(A)v_1 \\ &= \sum_{j=1}^{k} \psi(A)\hat{\psi}(A)V_p\hat{y}_j \\ &= \sum_{j=1}^{k} \psi(\lambda_j)\hat{\psi}(\theta_j)q_j + \psi(A)\hat{\psi}(A)z \\ &= \sum_{j=1}^{k} q_j + \psi(A)\hat{\psi}(A)z \end{aligned}$$

Where $\{q_j\}$ is the set of normalized eigenvectors corresponding to the eigenvalues of A that are approximated by θ_j and the vector z is orthogonal to $\{q_j\}$. Since z will nearly belong to the subspace spanned by the eigenvectors corresponding to eigenvalues belonging to a region in the complex plane where the polynomial $\psi(\lambda)\hat{\psi}(\lambda)$ is small, this choice will have the desired effect. Namely, the starting vector is forced to be the sum of k eigenvectors of A.

6 The Symmetric Case

The symmetric case is important and therefore, it is worthwhile to analyze the k-step method in this setting. Throughout this section it will be assumed that

$$A = A^T .$$

The Arnoldi Factorization then reduces to its predecessor the Lanczos factorization

$$AV - VT = re_k^T$$

where T is a symmetric tridiagonal matrix. In order to analyze the iteration induced by Algorithm (4.8) when exact shifts are taken some notation and a preliminary lemma must be established.

Lemma 6.1 *Let*

$$M = \begin{pmatrix} T & \beta e_k e_1^T \\ \beta e_1 e_k^T & \hat{T} \end{pmatrix}$$

be a symmetric tridiagonal matrix. Then the roots of the equation

$$\beta^2 e_k^T (T - \lambda I)^{-1} e_k = \frac{1}{e_1^T (\hat{T} - \lambda I)^{-1} e_1}.$$

are eigenvalues of M.

Proof: Define $M_\lambda = M - \lambda I$, $T_\lambda = T - \lambda I$ and $\hat{T}_\lambda = \hat{T} - \lambda I$. Then for any $\lambda \notin \lambda(T) \cup \lambda(\hat{T})$

$$M_\lambda = \begin{pmatrix} I_k & 0 \\ \beta_k e_1 e_k^T T_\lambda^{-1} & I_p \end{pmatrix} \begin{pmatrix} T_\lambda & \beta_k e_k e_1^T \\ 0 & \hat{T}_\lambda - \beta^2 e_k^T T_\lambda^{-1} e_k e_1 e_1^T \end{pmatrix}.$$

Thus

$$\begin{aligned} detM_\lambda &= detT_\lambda det(\hat{T}_\lambda - \beta^2 e_k^T T_\lambda^{-1} e_k e_1 e_1^T) \\ &= detT_\lambda det\hat{T}_\lambda det(I - \beta^2 e_k^T T_\lambda^{-1} e_k e_1 e_1^T \hat{T}^{-1}) \\ &= detT_\lambda det\hat{T}_\lambda [1 - \beta^2 (e_k^T T_\lambda^{-1} e_k)(e_1^T \hat{T}^{-1} e_1)]. \end{aligned}$$

Since $\lambda(T) \cup \lambda(\hat{T})$ is a discrete set, the expression developed for $detM_\lambda$ is seen to be valid in general by a continuity argument. □

With this lemma established it will be possible to analyze the iterations using polynomial filters constructed from exact shifts. The selection rule to be analyzed is to retain the k extreme eigenvalues of T_{k+p} (i.e. an equal number of the smallest and largest eigenvalues of T_{k+p}). Let m denote the iteration number. Then $v_1^{(m)}$ is the starting vector, and

$$AV_{k+p}^{(m)} - V_{k+p}^{(m)} T_{k+p}^{(m)} = r_{k+p}^{(m)} e_{k+p}^T .$$

Let

$$T_{k+p}^{(m)} = \begin{pmatrix} T_k^{(m)} & \beta_k^{(m)} e_k e_1^T \\ \beta_k^{(m)} e_1 e_k^T & \hat{T}^{(m)} \end{pmatrix}$$

have eigenvalues

$$\begin{aligned} \{\theta_{1,m+1} < \theta_{2,m+1} < \cdots < \theta_{\ell,m+1} < \mu_{1,m+1} < \cdots \\ < \mu_{p,m+1} < \theta_{\ell+1,m+1} < \cdots < \theta_{k,m+1}\} \end{aligned}$$

and let $T_k^{(m)}$ have eigenvalues

$$\{\theta_{1m} < \theta_{2m} < \cdots < \theta_{\ell m} < \theta_{\ell+1 m} < \cdots < \theta_{km}\}$$

and let $e_1^T(\hat{T}^{(m)} - \lambda I)^{-1}e_1$ have zeroes $\{\hat{\mu}_{1m} < \cdots < \hat{\mu}_{p-1,m}\}$. Then

$$Q^{(m)T}\, T_{k+p}^{(m)}\, Q^{(m)} = \begin{pmatrix} T_k^{(m+1)} & 0 \\ 0 & \Theta_p^{(m+1)} \end{pmatrix}$$

where $Q^{(m)} = Q_{1m}Q_{2m}\cdots Q_{pm}$ are the orthogonal matrices constructed to apply the implicit shifts $\mu_{1m}, \ldots, \mu_{pm}$ at step 2.4 of Algorithm 4.8. And step (2.5) gives

$$V_k^{(m+1)} = [V_{k+p}^{(m)}\, Q^{(m)}] \begin{bmatrix} I_k \\ 0 \end{bmatrix} .$$

From Lemma (6.1),

$$\{\theta_{j,m+1}\} \cup \{\mu_{j,m+1}\}$$

are the $k+p$ roots of the equation

$$(\beta_k^{(m)})^2 e_k^T (T_k^{(m)} - \lambda I)^{-1} e_k = \frac{1}{e_1^T(\hat{T}^{(m)} - \lambda I)^{-1}e_1} . \tag{6.2}$$

Lemma 6.3 *Each* $\{\theta_{jm}\}$, $m = 1, 2, \ldots$ *is a decreasing sequence for* $1 \le j \le \ell$, *and an increasing sequence for* $\ell + 1 \le j \le k$. *Moreover,* $\theta_{\ell m} < \mu_{1(m+1)}$ *and* $\mu_{p,(m+1)} < \theta_{\ell+1,m}$ *for all* m *sufficiently large.*

Proof: It will first be shown that $\theta_{\ell m} < \hat{\mu}_{1m}$ for all m sufficiently large. To see this, suppose that

$$\hat{\mu}_{1m} < \theta_{1m} .$$

Define

$$\phi(\lambda) = \frac{1}{e_1^T(\hat{T}^{(m)} - \lambda I)^{-1}e_1}.$$

Note that the zeroes $\{\hat{\mu}_{1m} < \cdots < \hat{\mu}_{p-1,m}\}$ of the function $e_1^T(\hat{T}^{(m)} - \lambda I)^{-1}e_1$ are poles of ϕ. One can check that $(\beta_k^{(m)})^2 e_k^T (T_k^{(m)} - \lambda I)^{-1} e_k$ is an increasing positive continuous function on the interval $(-\infty, \theta_{1m})$ and that

ϕ is unbounded above as $\lambda \to -\infty$ and that $\phi(\hat{\mu}_{1m} - \tau) \to -\infty$ while $\phi(\hat{\mu}_{1m} + \tau) \to +\infty$ as $\tau \to 0$. From these facts, it follows that the root $\theta_{1,m+1} < \hat{\mu}_{1m}$ and also $\theta_{2,m+1} \leq \theta_{1m}$ due to Lemma (6.1) and the rational structure of equation (6.2). Therefore,

$$\theta_{1,m+1} < \hat{\mu}_{1m} < \theta_{2,m+1} \leq \theta_{1m}$$

The interlace theorem given by Golub and VanLoan in [16,p479] assures the existence of an eigenvalue λ_{jm} of A between $\theta_{1,m+1}$ and $\theta_{2,m+1}$ and hence λ_{jm} must also lie between $\theta_{1,m+1}$ and $\theta_{1,m}$. However, this situation can occur at most n times since each occurrence isolates a distinct eigenvalue of A. The same argument may be applied in succession for θ_{jm} $1 \leq j \leq \ell$ to see that $\hat{\mu}_{1m} < \theta_{jm}$ at most n-times. A similar argument will show $\theta_{jm} < \hat{\mu}_{pm}$ at most n times. For m sufficiently large this will imply that

$$\theta_{1m} < \cdots < \theta_{\ell m} < \hat{\mu}_{1m} < \cdots < \hat{\mu}_{p-1,m} < \theta_{\ell+1,m} < \cdots < \theta_{km}$$

It follows readily from Lemma (6.1) and the rational structure of equation (6.2) that there is exactly one zero of the equation in each of the intervals

$$(-\infty, \theta_{1m}], (\theta_{1m}, \theta_{2m}], \cdots (\theta_{(l-1)m}, \theta_{lm}], [\theta_{\ell+1,m}, \theta_{(l+2)m}), \cdots, [\theta_{(k-1)m}, \theta_{km}), [\theta_{km}, \infty).$$

Moreover, since $(\beta_k^{(m)})^2 e_k^T (T_k^{(m)} - \lambda I)^{-1} e_k$ is a strictly increasing continuous function on the interval $(\theta_{lm}, \theta_{l+1,m})$ which tends to $-\infty$ at the left endpoint and $+\infty$ at the right endpoint of the interval and since $\phi(\lambda)$ alternates sign on crossing each pole $\hat{\mu}_{jm}$, it follows that there are p zeros of equation (6.2) in the interval $(\theta_{lm}, \theta_{l+1,m})$ and hence

$$\{\mu_{j(m+1)}\} \subset (\theta_{lm}, \theta_{l+1,m}).$$

persists for all m sufficiently large. □

The following theorem results directly from Lemma 1.2.

Lemma 6.4 *$\{\theta_{jm}\}$ is decreasing and $\lim_{m\to\infty}\theta_{pm} = \theta_j \quad 1 \le j \le \ell$ while $\{\theta_{jm}\}$ is increasing and $\lim_{m\to\infty}\theta_{jm} = \theta_j \quad \ell+1 \le j \le k$.*

Proof: The proof of Lemma (6.2) implies that $\theta_{j,m+1} \le \theta_{jm}$. for $j \le l$ while $\theta_{j,m+1} \ge \theta_{jm}$ for $j > l$ for all m sufficiently large and $\lambda_1 \le \theta_{jm} \le \lambda_n$ for all j and all m. Since bounded monotone sequences converge the result is proved. □

The convergence of the sequences $\{\theta_{jm}\}$ has been established but it is still not clear that the limit points will be eigenvalues of the matrix A. This shall be established by showing that the sequence $\{\beta_k^{(m)}\}$ is not bounded away from zero.

Theorem 6.5 *The limit points $\{\theta_j\}$ of the sequences $\{\theta_{jm}\}$ are eigenvalues of the matrix A.*

Proof: The sequence of vectors $\{v_1{}^{(m)}\}$ lie on the unit ball in $\mathbf{R}^n$ and hence have a convergent subsequence $\{v_1{}^{(m_i)}\}$ Let $\hat{v_1}$ be the limit of this subsequence. It is sufficient to show that the corresponding subsequence $\{\beta_k^{(m_i)}\}$ converges to zero. Let ϵ be a specified acceptance tolerance in the sense that the iteration is halted, or deflation occurs when a subdiagonal of H falls below ϵ. Then, without loss of generality, it may be assumed that $\beta_j{}^{(m)} > \epsilon$ for all $j \ne k$ (otherwise a deflation would have occurred at indices $j < k$ or the iteration would halt if $\beta_j{}^{(m)} < \epsilon$ for $j > k$). Suppose that $\beta_k^{(m_i)} > \epsilon$ for all i. It follows from the implicit-Q Theorem that the sequence of

matrices $\{T_k^{(m_i)}\}$, $\{\hat{T}^{(m_i)}\}$ and the sequence of vectors $\{r_k^{(m_i)}\}$ must converge to limits T_k, $\hat{T}$ and r_k respectively. Moreover, $\beta_k^{(m_i)} \rightarrow \beta_k = \|r_k\|$. Now, the subsequences $\{\theta_{j(m_i+1)}\}$ each must converge to a root of the equation

$$\beta_k^2 e_k^T (T_k - \lambda I)^{-1} e_k = \frac{1}{e_1^T (\hat{T} - \lambda I)^{-1} e_1}.$$

But, at the same time $\theta_{j(m_i+1)} \rightarrow \theta_j$ which leads to a contradiction. The contradiction arises since $\{\theta_j\} = \lambda(T_k)$ and hence no θ_j can be a root of the equation above. This is assured since $\beta_k \neq 0$ and the last component of each eigenvector of T_k as well as the first component of each eigenvector of $\hat{T}$ must be nonzero due to the nonzero off diagonal elements of T_k, $\hat{T}$. It follows that $\beta_k^{(m_i)} \rightarrow 0$.

The assumption that $\beta_j{}^{(m)} > \epsilon$ for $j < k$ and the implicit-Q Theorem imply that $\beta_k^{(m_i)} \rightarrow \beta_k = ||r_k||$ and since this nonnegative subsequence is not bounded away from zero it follows that $||r_k^{(m_i)}|| \rightarrow ||r_k|| = \beta_k = 0$. From this one may conclude that $\{\theta_j\}$ must be eigenvalues of A. □

These results which are based upon compactness arguments are not very satisfying. They do not reveal much about the behavior of of the iteration. A better understanding of the nature of the convergence is found in the following results. The following discussion essentially shows that the expansion coefficients $\gamma_j^{(m)} = q_j^T v_1^{(m)}$ must converge to zero for $\lambda_j \in (\theta_l, \theta_{l+1})$.

Define: $\Psi_m(\lambda) = \prod_{i=1}^m \psi_i(A)$. Then $v_1^m = \frac{\Psi_m(A)v_1}{\|\Psi_m(A)v_1\|}$.

Lemma 6.6 *Assume* $\ell \geq 2$, $\ell+1 \leq k-1$. *Then (i)* $|\Psi_m(\lambda_n)| \geq \left(\frac{\lambda_n - \lambda_{n-1}}{\lambda_{n-1} - \theta_\ell} + 1\right)^{pm} |\Psi_m(\lambda_{n-1})|$

(ii) $|\Psi_m(\lambda_1)| \geq \left(\frac{\lambda_2 - \lambda_1}{\theta_{\ell+1} - \lambda_2} + 1\right)^{pm} |\Psi_m(\lambda_2)|$

.

Proof: Assume m is sufficiently large that $\{\mu_{jm}\} \subset (\theta_\ell, \theta_{\ell+1})$ holds. Consider the normalized polynomial

$$\hat{\Psi}_m(\lambda) = \frac{\Psi_m(\lambda)}{\Psi_m(\lambda_{n-1})} = \prod_{\ell=1}^{m} \left(\prod_{j=1}^{p} \left(\frac{\lambda - \mu_{j\ell}}{\lambda_{n-1} - \mu_{j\ell}} \right) \right)$$

Note, that $\theta_l < \lambda_{n-1}$ will imply that

$$\frac{\lambda - \mu_{j\ell}}{\lambda_{n-1} - \mu_{j\ell}} = \frac{\lambda - \lambda_{n-1}}{\lambda_{n-1} - \mu_{j\ell}} + 1 \geq \frac{\lambda - \lambda_{n-1}}{\lambda_{n-1} - \theta_\ell} + 1$$

for $\lambda > \lambda_{n-1}$ and (i) follows. A similar argument using $\frac{\Psi_m(\lambda)}{\Psi_m(\lambda_2)}$ will establish (ii). □

Let us now define $\gamma_j^{(m)} = q_j^T v_1^{(m)}$ where $\{q_j\}$ are the orthonormal eigenvectors for A. Then the following lemma may be established.

Lemma 6.7 *Suppose that neither of $v_1^T q_1$ or $v_1^T q_n$ are equal to zero. Then $\gamma_j^{(m)} \to 0$ for every j such that $\lambda_{n-1} \geq \lambda_j > \theta_{\ell+1}$ and for every j such that $\lambda_2 \leq \lambda_j < \theta_\ell$.*

Proof: Observe that

$$\begin{aligned} \gamma_j^{(m)} &= \frac{q_j^T \Psi_m(A) v_1^{(1)}}{\|\Psi_m(A) v_1^{(1)}\|} \\ &= \frac{\gamma_j^{(1)} \Psi_m(\lambda_j)}{\left(\sum \Psi_m^2(\lambda_i) \gamma_i^{(1)2}\right)^{\frac{1}{2}}}. \end{aligned}$$

Hence

$$(\gamma_j^{(m)})^2 = \frac{(\gamma_j^{(1)})^2 \left[\Psi_m(\lambda_j)/\Psi_m(\lambda_n)\right]^2}{\gamma_n^{(1)2} + \sum_{i=1}^{n-1} \left(\frac{\Psi_m(\lambda_i)}{\Psi_m(\lambda_n)}\right)^2 \gamma_i^{(1)2}}$$

$$
\begin{aligned}
&\leq \left(\frac{\gamma_j^{(1)}}{\gamma_n^{(1)}}\right)^2 \left[\frac{\Psi_m(\lambda_j)}{\Psi_m(\lambda_n)}\right]^2 \\
&\leq \left(\frac{\gamma_j^{(1)}}{\gamma_n^{(1)}}\right)^2 \left[\frac{\Psi_m(\lambda_j)}{\Psi_m(\lambda_{n-1})}\right]^2 \left(\frac{\lambda_n - \lambda_{n-1}}{\lambda_{n-1} - \theta_\ell} + 1\right)^{-2pm} \\
&\to 0
\end{aligned}
$$

for all j such that $\left|\frac{\Psi_m(\lambda_j)}{\Psi_m(\lambda_{n-1})}\right|$ is bounded. Now, $\Psi_m(\lambda)$ is monic of degree mp and has all roots contained in the interval $(\theta_\ell, \theta_{\ell+1})$. Therefore,

$$(|\Psi_m(\lambda)| \ / \ |\Psi_m(\lambda_{n-1})|) < 1$$

for all $\lambda \in (\theta_{\ell+1}, \lambda_{n-1}]$,and

$$(|\Psi_m(\lambda)| \ / \ |\Psi_m(\lambda_2)|) < 1$$

for all $\lambda \in [\theta_2, \theta_\ell)$ and the result follows. □

We are now able to prove the main result.

Theorem 6.8 *Suppose that neither of $v_1^T q_1$ or $v_1^T q_n$ are equal to zero. Then $\theta_1 = \lambda_1$ and $\theta_k = \lambda_n$.*

Proof: Since the sequence $\{v_1^{(m)}\}$ is bounded it must have a limit point $\hat{v}_1$. From our previous result, we know that

$$q_j^T \hat{v}_1 = 0, \ \lambda_j \in (\theta_{\ell+1}, \lambda_{n-1}] \text{ and } \lambda_j \in [\lambda_2, \theta_\ell) \ .$$

Moreover, $p_k^{(m)}(\lambda) \to p(\lambda) = \prod_{j=1}^k (\lambda - \theta_j)$ and each $p_k^{(m)}$ satisfies

$$v_1^{(m)^T} p_k^{(m)^2}(A) v_1^{(m)} \leq v_1^{(m)^T} \hat{p}^2(A) v_1^{(m)}$$

for all monic polynomials $\hat{p}$ of degree k. Since,

$$v_1^{(m)^T} p_k^{(m)^2}(A) v_1^{(m)} \rightarrow v_1^{(m)^T} p^2(A) v_1^{(m)}$$

it follows that

$$v_1^T p^2(A) \hat{v}_1 \leq \hat{v}_1^T \hat{p}^2(A) \hat{v}_1$$

for all monic polynomials $\hat{p}$ of degree k. But

$$\hat{v}_1^T p^2(A) \hat{v}_1 = \sum_{\lambda_j \in (\theta_\ell, \theta_{\ell+1})} p^2(\lambda_j)\gamma_j^2 + \gamma_1^2 p^2(\lambda_1) + \gamma_n^2 p^2(\lambda_n)$$

where $\gamma_j = q_j^T v_1$, $\sum \gamma_j^2 = 1$. This leads to a contradiction, since it is possible to construct another monic polynomial $\hat{p}$ such that

$$\begin{aligned} \hat{v}_1^T \hat{p}(A)^2 \hat{v}_1 &= \sum_{\lambda_j \not\in (\theta_\ell, \theta_{\ell+1})} \hat{p}(\lambda_j)^2 \gamma_j^2 + \gamma_1^2 \hat{p}(\lambda_1)^2 + \gamma_n^2 \hat{p}^2(\lambda_n) \\ &< \hat{v}_1^T p(A)^2 v_1^T . \end{aligned}$$

as the following Lemma shows. □

Lemma 6.9 *Suppose k is even, $k \geq 4$, with ℓ s.t. $p(\lambda) < 0$ on $(\theta_\ell, \theta_{\ell+1})$. Then there is a quadratic polynomial ϕ such that*

$$\hat{p}(\lambda) = p(\lambda) - \phi(\lambda)$$

is monic and satisfies both

$$0 < \hat{p}(\lambda_1) < p(\lambda_1) , \quad 0 < \hat{p}(\lambda_n) < p(\lambda_n)$$

and

$$0 > \hat{p}(\lambda) > p(\lambda) , \quad \lambda \in (\theta_\ell, \theta_{\ell+1}) .$$

Proof: Take $\phi(\lambda) = \epsilon(\lambda - \theta_\ell)(\lambda - \theta_{\ell+1})$, $\epsilon > 0$. Consider the polynomial $\hat{p}$. Note that $\hat{p}$ is monic since $\deg \phi < \deg p$. Moreover, for ϵ sufficiently small

$$0 < \hat{p}(\lambda) = p(\lambda) - \phi(\lambda) < p(\lambda)$$

since $\phi(\lambda) = p(\lambda) - \hat{p}(\lambda)$ has at most two roots $\theta_\ell, \theta_{\ell+1}$, it cannot change sign on $(\theta_\ell, \theta_{\ell+1})$.

Now, it is clear for $\epsilon > 0$ sufficiently small

$$0 < \hat{p}(\lambda_1) = p(\lambda_1) - \epsilon(\lambda_1 - \theta_\ell)(\lambda_1 - \theta_{\ell+1}) < p(\lambda_1)$$

and

$$0 > \hat{p}(\lambda_n) = p(\lambda_n) - \epsilon(\lambda_n - \theta_\ell)(\lambda_n - \theta_{\ell+1}) > p(\lambda_n)$$

since $\phi < 0$ for $\lambda \in (\theta_\ell, \theta_{\ell+1})$. □

These results indicate that the sort of convergence that takes place will typically be slow linear convergence of $\|r_k^{(m)}\|$ to zero with a rate governed by the ratio $\psi^{(m)}(\theta_\ell)/\psi^{(m)}(\lambda_j)$, where λ_j is the eigenvalue of A in $(\theta_\ell, \theta_{\ell+1})$. This may be seen through an analysis similar to the proof of Lemma(6.7). While this iteration does seem to perform reasonably well in practice if one monitors the Ritz estimates and halts when these are small for a significant percentage of the k eigenvalues actually sought. However, other iterations such as the one developed in Section 5, might do a better job of evenly distributing the components of the vector v_1 along the k wanted eigenvectors.

7 The Generalized Eigenvalue Problem

In this section the generalized eigenvalue problem will briefly be discussed. The generalized problem is to find (x, λ) such that

$$Ax = \lambda Mx \ .$$

Most often, the matrix M arises from applying a Galerkin principle to a linear operator $\mathcal{L}$ which leads to

$$A = \langle \mathcal{L}\phi_i, \phi_j \rangle \ , \quad M = \langle \phi_i, \phi_j \rangle$$

where $\langle \ , \rangle$ is an inner product on a linear space $\mathcal{H}$ and and $\{\phi_j\} \in W^n$ is a basis for a finite element subspace $W^n \subset \mathcal{H}$. The matrix M is thus symmetric and positive definite. This condition shall be assumed in this section. The basic iterative method will carry over to this setting with very little modification. It will be necessary to set aside storage for two basis matrices V and W and to maintain and update a factorization of the form

$$AV - WH = re_k^T \tag{7.1}$$

where

$$W = MV \ , \quad V^T W = I \ , \ \text{and} \ V^T r = 0 \ .$$

Again a simple recursion is available to advance the factorization (7.1) one step. Just note that

$$A(V, v) - (W, w) \begin{pmatrix} H & h \\ \beta e_k^T & \alpha \end{pmatrix} = r_+ e_{k+1}^T \tag{7.2}$$

leads to

$$AV - WH - \beta w e_k^T = 0$$

so

$$(r - \beta w) e_k^T = 0 \, .$$

But $w = Mv$, so by solving

$$M\hat{v} = r \quad \text{and putting} \quad \beta = (\hat{v}^T r)^{\frac{1}{2}} \tag{7.3}$$

one may scale by β to get

$$v \leftarrow \frac{1}{\beta}\hat{v} \, , \quad w \leftarrow \frac{1}{\beta} r \, .$$

Since $V^T r = 0$, and $v^T w = 1$, it follows that

$$V_+^T W_+ = I_{k+1} \, ,$$

where $V_+ = (V, v)$ and $W_+ = (W, w)$. Then one has

$$\begin{pmatrix} h \\ \alpha \end{pmatrix} = \begin{pmatrix} V^T \\ v^T \end{pmatrix} Av$$

follows from equating the $(k+1)$-st column of (7.2) and the updated residual

$$r_+ = Av - (V, v) \begin{pmatrix} h \\ \alpha \end{pmatrix} \, .$$

Again this step may be accomplished through two matrix vector products followed by two more to accomplish one step of iterative refinement to ensure the biorthogonality of V, W.

There are two key consequences of this arrangement:

1. $Q^T V^T W Q = I$ is preserved so the implicit Q-R shift strategy may be applied.

2. If $A = A^T$ is symmetric, then

$$H = V^T A V$$

follows from $V^T W = I$, $V^T r = 0$ so that $H = H^T$ will be symmetric and tridiagonal when A is symmetric.

With these observations, it is straightforward to adapt the algorithms previously discussed to solve the generalized eigenproblem. Some limited computational experience with this approach is the subject of the following section.

8 Computational Results and Conclusions

Computational results for this technique are quite promising but are certainly preliminary. There is a Fortran implementation of the algorithms developed here. Two versions of the code have been produced. One of these implements the strategy for the generalized symmetric eigenvalue problem as described in Section 7. The other implements the algorithm for the standard non-symmetric eigenproblem. In addition to exhibiting behavior on some test problems, two experiences with applications will be discussed. Finally, some very interesting illustrations of the shapes of the filter polynomials that are constructed through exact shifts shall be reported.

There are some important details of the Fortran implementation of Algorithm (4.7). Step 3 requires a user supplied matrix vector product. Steps

4 and 5 are implemented through calls to the level 2 BLAS [11,12] routine DGEMV. One step of iterative refinement is carried out at Step 6 of Algorithm (4.7) rather than iterating until the test $\|s\| \leq \epsilon\|r\|$ is passed. Steps (6.1) and (6.2) were also implemented through calls to DGEMV. In all of the computations observed there was never a loss of orthogonality in the columns of V. In all cases $\|V^TV - I\|$ was on the order of unit roundoff error. Eigenvalue calculations used a slight modification of EISPACK [32] subroutines TQL in the symmetric case and HQR in the nonsymmetric case. These may be replaced by the corresponding block routines from LAPACK [10] to enhance performance in the future.

Expressing the algorithm in terms of the level 2 BLAS has provided the means to achieve high performance portable Fortran code. The code has been run on SUN SPARC, CONVEX C1, Stardent Titan, CRAY 2, and CRAY YMP computers. The cost of operations were clearly dominated by the user supplied matrix vector products (and system solves in the generalized problem). The time spent in the user supplied portion was orders of magnitude over the time spent in the other parts of the eigenvalue calculations. This performance characteristic is a direct consequence of the performance of DGEMV on the architectures of the machines listed above. The crucial point for improving the algorithm is to better understand the construction of the filter polynomials in order to reduce the required number of user supplied matrix vector products . Parallelism may be invoked through the level 2 BLAS and also through the user supplied matrix vector product.

In all of the results reported below, exact shifts were used as described in Algorithm (4.10). The iteration was halted when $\|(e_k^T y_j) r_k\| < 10^{-7}, 1 \leq$

$j \leq k-3$ where y_j is the $j-th$ Ritz vector corresponding to Ritz values approximating the wanted spectrum. This ad hoc stopping rule allowed the iteration to halt quite early in cases where it was difficult to make a clean separation between the wanted and unwanted spectrum. This ad hoc criterion will have to be replaced with a more rigorous one in the future.

In the first set of test problems the matrix A arises from a standard 5-point discretization of the convection-diffusion operator on the unit square Ω. The PDE is

$$-\Delta u + \rho u_x = \lambda u, \quad \text{in } \Omega, \quad u|_{\partial\Omega} = 0$$

When $\rho = 0$ the matrix A is the discrete Laplacian and for $\rho > 0$ A has distinct complex eigenvalues which appear in a rectangular grid in the complex plane when the cell size $h = 1/(n+1)$ is large enough with respect to the parameter ρ. However, the boundary conditions of the continuous problem do not admit eigenfunctions corresponding to complex eigenvalues, so the eigenvalues of the matrix A become real when the mesh size becomes small enough. The order of the discrete operator A is $N = n^2$ and since it's eigenvalues are distinct, it is diagonalizable. These problems allowed testing of the algorithm for accuracy and performance in some interesting but well understood cases. In both of the tables below, the values $k = 10$ and $p = 10$ were used. The two columns on the right of the tables give the norm of the residual vector r and the norm of the true residual $\|Ax - x\lambda\|$ for the sixth eigenvalue. Typically, the eigenvalues of smaller index had residuals that were smaller than this one. For the symmetric problems the residual estimates were uniformly small for the eight smallest eigenvalues.

Table 8.1

Discrete Laplacian

Dimension	Niters	$\|r\|$	$\|Ax - x\lambda\|$
100	12	1.4-06	3D-15
256	23	3.4-06	5D-15
400	29	6.5-06	5D-15
625	25	7.1-06	3D-14
900	29	6.2-06	2D-14
1600	43	2.9-06	6D-14
2500	50	1.1-05	9D-13
3600	63	9.9-06	4D-11
4900	92	8.9-06	1D-11
8100	237	1.1-05	1D-11
10000	165	1.1-05	8D-12

In Table 8.2 below, the problems of order 256 and 400 did not satisfy the convergence test before the maximum number of iterations allowed had been reached. In all cases the ten eigenvalues of smallest real part were sought. In both of the problems just mentioned, five or more eigenvalues had been determined to high accuracy. In all cases the iterations could have halted much earlier if a better stopping criterion were devised.

Table 8.2

Convection Diffusion

Dimension	Niters	$\|r\|$	$\|Ax - x\lambda\|$
100	61	5.3-06	1D-12
256	100	.23	1D-5
400	100	5.2-03	2D-10
625	77	2.3-06	8D-12
900	153	8.9-06	2D-14
1600	103	7.4-06	6D-14

The second set of results will briefly describe two problems that arise in the context of solving partial differential equations. The first of these involves a discretization of a membrane problem in which the membrane is composed of two materials. On an open bounded connected set $\Omega \subset \mathbf{R}^2$ we consider

$$-\Delta u = \lambda \rho u, \quad \text{in } \Omega, \quad u|_{\partial\Omega} = 0$$

where the density ρ is of the form

$$\rho = \alpha\chi_S + \beta(1 - \chi_S)$$

where χ_S is the characteristic function of a subset $S \subset \Omega$ with area γ. The problem is to determine the density function ρ which minimizes the lowest eigenvalue $\lambda_1(\rho)$ of this PDE. Here α and β are the known (constant) densities of two given materials in respective volume fractions $\gamma/|\Omega|$ and $1-\gamma/|\Omega|$ and the set S is occupied by the material with density α. Cox [4] has formulated an algorithm to solve this minimization problem. The algorithm generates a

sequence of symmetric generalized eigenvalue problems

$$Av = \lambda M(\rho)v$$

which arise through a bi-linear finite element discretization of the PDE. The density function ρ is modified at each iteration with the set S determined through level sets of the corresponding eigenfunction. The matrix A is positive definite and independent of the density function ρ so the problem was cast in the form

$$M(\rho)v = \frac{1}{\lambda}Av.$$

Since only matrix vector products are required of M the dependence on ρ presented no additional computational burden. The matrix A was factored once and this factorization was subsequently used repeatedly to solve (7.3) (with A playing the role of M in that equation). The eigenvalue iteration also benefited from the re-use of the converged starting vector from the previous problem but this did not appear to be of great consequence in this case. The following table gives results for the same sub-problem on a variety of machines.

Table 8.3

Membrane Problem on Various Machines

	Sun	Convex	Titan	Y-MP
Time (secs)	240	81	40.9	5.4
matrix vector	40	40	40	40
$\|V^TW - I\|$	10^{-14}	10^{-14}	10^{-14}	10^{-11}

The overall performance was excellent on this problem. Grid sizes of of 64 by 64, 100 by 100, and 200 by 200 were used. Both minimization of $\lambda_1(\rho)$ and $\lambda_2(\rho)$ were done. The number of matrix vector products was typically around 32-40 regardless of the dimension of the matrix. That is, with $k = 8$ and $p = 8$ the eigenvalue solver required 3 to 4 iterations with 3 being the usual number. The Ritz estimates for $\|Ax - M(\rho)x\lambda\|$ were on the order of $10D - 14$ for the lowest six eigenvalues.

The second application leads to a nonsymmetric eigenvalue problem. The PDE arises in a study of bifurcations in a Couette-Taylor wavy vortex instability calculation. This work described in [13] is based upon a method of W.S. Edwards and L.S Tuckerman which is designed to study these bifurcations from Taylor vortices to wavy vortices. The discrete problem is obtained by first linearizing the Navier-Stokes equations about a (numerically) known steady state solution U corresponding to Taylor vortices. The perturbation u corresponding to wavy vortices is found by solving the linearized Navier-Stokes problem

$$\frac{\partial u}{\partial t} = -(U \cdot \nabla)u - (u \cdot \nabla)U - \nabla p + \nu \nabla^2 u$$

with

$$\nabla \cdot u = 0 \text{and} \quad u|_{\partial\Omega} = 0$$

where Ω is the annular region between two concentric rotating cylinders. This PDE is discretized to then yield a nonsymmetric eigenvalue problem

$$A(\nu)v = \lambda v$$

Since a pseudo-spectral method is used, the discrete matrix is dense rather

than sparse. However, matrix vector products can still be performed rapidly using Fourier transforms. The discrete problem involved a matrix of order 2380 . The eigenvalue code with $k = 16$ and $p = 40$ required 60 iterations to produce eight eigenvalues and corresponding eigenvectors with largest real part. This entailed about 2400 matrix vector products. The accuracy of these were confirmed to be at least five significant digits. Edwards in a private communication remarked that in his opinion "the high accuracy could not have been achieved by other methods I might have tried."

This behavior of the algorithm on these two problems seems to be typical on more difficult problems. The number of matrix vector products tends to be near n for difficult nonsymmetric problems. Symmetric generalized eigenvalue problems from finite element analysis of structures or membranes seem to be solved very rapidly if posed in terms of finding the largest eigenvalues.

To close this section, the interesting behavior of filtering polynomials associated with the choice of exact shifts will be presented. Two problems will be discussed. The first example arises from the convection diffusion above with $\rho = 40$. The grid size was 1/30 leading to a nonsymmetric matrix of order 900 . The results for this problem are displayed in Figures 8.1 and 8.2. The second example is the banded Toeplitz matrix used for test purposes by Grcar [17]. This matrix is non-normal and has a nontrivial pseudo-spectrum as discussed in [21]. (The ϵ pseudo-spectrum of a matrix A is $\{\lambda \in \mathbf{C} : \|(\lambda I - A)^{-1}\| \geq \epsilon^{-1}\}$). The matrix is a 5-diagonal matrix with the value -1 on the first sub-diagonal and the value 1 on the main diagonal and the next three super diagonals. The results for this problem are displayed in Figures 8.3 and 8.4.

The graphs shown below depict the filter polynomial $\psi(\lambda)$ for values of λ over a region containing the eigenvalues of A. The surface plot is of $|\psi|$ and the contour plots are of $log(|\psi|)$ The + symbols show the location of the true eigenvalues of A The o symbols mark the location of the eigenvalues of H that are "wanted". These will eventually converge to eigenvalues of A. The $*$ symbols show the roots of the polynomial ψ.

Figure 8.1

Convection Diffusion: iteration 1

Figure 8.2

Convection Diffusion: at convergence

In Figures 8.1 and 8.2 the values k = 10, p = 10 were used. One may observe convergence by looking at the 10 leftmost o symbols enclosing the + symbols. The interesting features of these filter polynomials is that they are remarkably well behaved in terms of being very flat in the region that is to be damped and very steep outside that region. The reason for this desirable behavior is not understood at the moment.

Figure 8.3 Grcar matrix: iteration 1

Figure 8.3

Grcar matrix : iteration 1

Figure 8.4

Grcar matrix : at convergence

In Figures 8.3 and 8.4 the corresponding behavior of the filter polynomials is shown. In these figures only the upper half-plane is shown. The dotted line shows the boundary of the practical spectrum [21] for this matrix. It is interesting to note how the contours of the filter polynomial obtained through the exact shifts mimic the shape of this boundary. The algorithm claimed convergence of the leftmost eigenvalues (ie. the ten eigenvalues of smallest real part). However, as demonstrated in the figure, these are pseudo-eigenvalues. Interestingly enough, HQR from Eispack will give the same behavior if applied to the transpose of the Grcar matrix. HQR will give the correct eigenvalues when applied to the Grcar matrix directly and it was used to calculate the values of the "true" spectrum shown above.

In conclusion, it seems that this is quite a promising approach. A direct relationship to the implicitly shifted QR iteration has been established and several problems inherent to the traditional Arnoldi method have been addressed through this new approach. The most important of these are the fixed storage, maintenance of orthogonality, and avoidance of spurious eigenvalues. The computational results are clearly preliminary. The limited experience indicates research is needed in constructing filter polynomials which have better properties with respect to the wanted part of the spectrum. Moreover, a better understanding of the Ritz convergence estimates in the nonsymmetric case would be helpful. These estimates have been very important in terminating the iteration early (ie. before the residual is very small) in the symmetric (generalized) eigenproblem. A criterion for choosing the values of k and p is also required. At present, ad hoc choices are made and there is little understanding of the relation of these two parameters to each other

and to the given problem. They have been chosen through experimentation for these results.

Future research on this topic might include a blocked variant to better deal with multiple eigenvalues. Investigations of the use of a preconditioner would also be interesting. Finally, extensions of this idea to other settings such as the solution of linear systems would seem to be a promising area of research as well. These investigations are underway and will be the topic of subsequent papers.

Acknowledgements

I would like to acknowledge Dr. Phuong Vu and Cray Research for providing access to CRAY-2 and CRAY Y-MP computers and for help in performing a number of numerical experiments with the computer codes. I would also like to thank Dr. Lothar Reichel for reading the manuscript and making some useful comments and corrections.

References

1. W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9, 17-29,(1951) .

2. F. Chatelin and D. Ho, Arnoldi-Tchebychev procedure for large scale nonsymmetric matrices, Math. Modeling and Num. Analysis, 24,53-65,(1990).

3. A.T. Chronopoulos, s-Step Orthomin and GMRES implemented on parallel computers, Dept. Computer Science Report TR 90-15,University of Minnesota, Minneapolis, Minn.,(1989).

4. S. Cox, The symmetry, regularity, and computation of a membrane interface, Dept. Math. Sci. Rept., Rice University, Houston, TX, (1990).

5. 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, IEEE Press, New York, 505-509, (1974).

6. J. Cullum, The simultaneous computation of a few of the algebraically largest and smallest eigenvalues of a large, symmetric, sparse matrix, BIT 18, 265-275, (1978).

7. J. Cullum and R.A. Wiloughby, Computing eigenvalues of very large symmetric matrices - an implementation of a Lanczos algorithm with no reorthogonalization, J. Comput. Phys. 434, 329-358, (1981).

8. J. Cullum and R.A. Wiloughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol. I Theory, Birkhauser Boston, Inc.,(1985).

9. J. Daniel, W.B. Gragg, L. Kaufman, G.W. Stewart, Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, Math. Comp. 30, 772-795,(1976).

10. J.W. Demmel, J.J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, and D. Sorensen, A prospectus for the development of a linear algebra library for high performance computers, Mathematics and Computer Science Division Rept. ANL-MCS-TM-97, Argonne National Laboratory, (1987).

11. J.J. Dongarra, J. Du Croz, S. Hammarling, and R.J. Hanson, An extended set of fortran basic linear algebra subprograms, ACM Trans. Math. Soft. 14, 1-17, (1988).

12. J.J. Dongarra, J. Du Croz, S. Hammarling, and R.J. Hanson, Algorithm 656 An extended set of fortran basic linear algebra subprograms: Model implementation and test programs, ACM Trans. Math. Soft. 14, 18-32, (1988).

13. W.S. Edwards, S.R. Beane, S. Varma, Onset of Wavy Vortices in the Finite-Length Couette-Taylor Problem, submitted to Physics of Fluids,(1990)

14. J.G.F. Francis, The QR transformation: A unitary analogue to the LR transformation, Parts I and II, Comp. J. 4, 265-272, 332-345, (1961).

15. G.H. Golub, R. Underwood, and J.H. Wilkinson, The Lanczos algorithm for the symmetric $Ax = \lambda Bx$ problem, Report STAN-CS-72-270, Department of Computer Science, Stanford U. Stanford, California ,(1972).

16. G.H. Golub and C.F. Van Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, Maryland (1983).

17. J.F. Grcar, Operator coefficient methods for linear equations, Sandia National Lab. Rept. SAND89-8691, Livermore, California,(1989).

18. W. Karush, An iterative method for finding characteristic vectors of a symmetric matrix, Pacific J. Math. 1, 233-248, (1951).

19. C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Stand. , 45, 255-282, (1950).

20. T.A. Manteuffel, Adaptive procedure for estimating parameters for the nonsymmetric Tchebychev iteration, Numer. Math. 31, 183-208,(1978).

21. N. Nachtigal, L. Reichel, L.N. Trefethen, A hybrid GMRES algorithm for nonsymmetric matrix iterations, Numerical Analysis Rept. 90-7, Dept. Mathematics, MIT, Cambridge, Mass., (1990).

22. C.C. Paige, The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices, Ph.D. thesis, Univ. of London,(1971).

23. C.C. Paige, Computational variants of the Lanczos method for the eigenproblem, J. Inst. Math. Appl. 10, 373-381, (1972).

24. B.N. Parlett and D. S. Scott, The Lanczos algorithm with selective orthogonalization, Math. Comp. 33, 311-328, (1979).

25. B.N. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, NJ. (1980).

26. L. Reichel and W.B. Gragg, Fortran subroutines for updating the QR Decomposition of a matrix, ACM-TOMS, (to appear).

27. A. Ruhe, Rational Krylov sequence methods for eigenvalue computation, Linear Algebra Apps. , 58, 391-405, (1984).

28. Y. Saad, Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems, Math. Comp., 42, 567-588, (1984).

29. Y. Saad, Projection methods for solving large sparse eigenvalue problems, in Matrix Pencil Proceedings (B. Kagstrom , and A. Ruhe, eds),Springer-Verlag, Berlin, 121-144 (1982).

30. Y. Saad and M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Scientific and Stat. Comp. 7, 856-869, (1986).

31. D. Scott, Analysis of the symmetric Lanczos algorithm, Ph.D. dissertation, Dept. of Mathematics, University of California, Berkeley, (1978).

32. B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe, V.C. Klema, and C.B. Moler, Matrix Eigensystem Routines - EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, 2nd edition, Springer-Verlag, Berlin, 1976.

33. G.W. Stewart, Introduction to Matrix Computations, Academic Press, New York, 1973.

34. H.F. Walker, Implementation of the GMRES method using Householder transformations, SIAM J. Scientific and Stat. Comp. 9, 152-163,(1988).

35. J.H. Wilkinson, The Algebraic Eigenvalue Problem, Claredon Press, Oxford, England (1965).

Implicit Application of Polynomial Filters in a *k*-Step Arnoldi Method

D. C. Sorensen

RIACS Technical Report 90.43
October 1990