

IMPLICIT APPLICATION OF POLYNOMIAL FILTERS IN A K-STEP ARNOLDI METHOD*

D. C. SORENSSEN†

Dedicated to Gene Golub on the occasion of his 60th birthday

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.

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

AMS(MOS) subject classifications. 65F15, 65G05

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].

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 a few eigenvalues and corresponding eigenvectors of a large nonsymmetric 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, has 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 nonsymmetric 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

* Received by the editors October 29, 1990; accepted for publication (in revised form) July 30, 1991.

† Department of Mathematical Sciences, Rice University, Houston, Texas 77251-1829 (sorensen@rice.edu). This research was supported in part by National Science Foundation cooperative agreement CCR-8809615, Department of Energy contract DE-FG0F-91ER25103, and the Research Institute for Advanced Computer Science.

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 QR-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 such a way as to force the residual vector to zero.

In §2 we develop the Arnoldi factorization, expose the functional dependence of the residual on the starting vector, and give necessary and sufficient conditions for a starting vector to produce a zero residual. In §3 we show how to update the starting vector through implicit application of a polynomial filter to this vector on each iteration. The implicit application of this polynomial filter is accomplished through a truncated version of the implicitly shifted QR-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 as shown in §4, it is possible to implement a mathematically equivalent implicit method corresponding to all of the explicitly restarted methods that we are aware of. Convergence results for specific restart strategies are given in §5. Extension to the generalized problem is discussed in §6 and preliminary computational results are presented in §7.

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 that updates the Arnoldi factorization instead of restarting it.

This approach has several advantages over more traditional approaches. The number of eigenvalues that are sought is prespecified. This fixes the storage 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 reorthogonalization 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

$$(2.1) \quad AV = VH + re_k^T,$$

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

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

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

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

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 $v = \hat{V}e_1$.

Approximate eigenvalues and eigenvectors are readily available through this factorization. If $Hy = y\theta$ is an eigenpair for H , then the vector $x = Vy$ satisfies

$$\|Ax - x\theta\| = \|(AV - VH)y\| = |\beta e_k^T y|.$$

We call the vector x a Ritz vector and the approximate eigenvalue θ a Ritz value and note that the smaller $|\beta e_k^T y|$ is, the better these approximations are.

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

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

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

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

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

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

From this development it is easily seen that

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

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 in the symmetric case and the first rigorous numerical treatment is due to Paige [22], [23]. There have been several approaches to overcoming this problem in the symmetric case. They include: (1) complete reorthogonalization which may be accomplished through maintaining V in product Householder form [15], [34] or through the modified Gram–Schmidt processes with reorthogonalization [9], [26]; (2) Selective reorthogonalization, which 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]; (3) No reorthogonalization, which has been developed by Cullum and her colleagues. This last option introduces the almost certain possibility of introducing spurious eigenvalues. Various techniques have been developed to detect and deal with the presence of spurious eigenvalues [7], [8].

The appearance of spurious eigenvalues may be avoided through complete reorthogonalization of the Arnoldi (or Lanczos) vectors. Computational cost has been cited as the reason for not employing this option. 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 minimize (maximize) $\langle VV^T, A \rangle$ where $\langle B, A \rangle = \text{trace}(B^T A)$ 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, p. 367]). \square

Of course the Krylov space

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

plays an important role along with the Krylov matrix

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

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

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

and to observe that

$$(2.5) \quad AK - KF = \hat{r}e_k^T$$

where $\hat{r} = A^k v_1 - Kg$ with $g^T = (\gamma_o, \hat{g}^T)$. 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 and 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.,

$$(2.6) \quad AQ - QG = f e_k^T$$

where $Q = KR^{-1}$, $G = RFR^{-1}$ is upper Hessenberg with the same characteristic polynomial as F , and $f = (1/\rho_{kk})\hat{r}$. 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

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

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*

$$(2.7.1) \quad r_j = \frac{1}{\|\hat{p}_{j-1}(A)v_1\|} \hat{p}_j(A)v_1 , \quad \beta_j = \frac{\|\hat{p}_j(A)v_1\|}{\|\hat{p}_{j-1}(A)v_1\|}$$

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

$$(2.7.2) \quad \hat{p}_j \text{ solves } \min_{p \in \mathcal{PM}_j} \{\|p(A)v_1\|\}$$

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 the structure of β_j is a somewhat complicated function of A and v_1 . This theorem will provide the foundation for a convergence result to be presented later in §5.

The final result of this section will develop necessary and sufficient conditions for a particular starting vector to generate a k -dimensional invariant subspace.

THEOREM 2.8. *Let $AV_k - V_k H_k = r_k e_k^T$ be a k -step Arnoldi factorization of A , with H unreduced (i.e., $r_j \neq 0$, $1 \leq j \leq k-1$). Then $r_k = 0$ if and only if $v_1 = Xy$ where $AX = XJ$ with $\text{rank}(X) = k$ and J a Jordan matrix of order k (i.e., the direct sum of Jordan blocks).*

Proof. If $r_k = 0$, let $H\hat{X} = \hat{X}J$ be the Jordan canonical form of H and put $X = V_k \hat{X}$. Then $AX = XJ$, $\text{rank}(X) = k$ and

$$v_1 = V_k e_1 = V_k \hat{X} \hat{X}^{-1} e_1 = Xy, \quad \text{with } y = \hat{X}^{-1} e_1.$$

Suppose now that $AX = XJ$, $\text{rank}(X) = k$, and $v_1 = Xy$. Then $A^m X = XJ^m$ for any nonnegative integer m and it follows that

$$A^m v_1 = A^m Xy = XJ^m y \in \text{Range}(X)$$

for all m . Hence, $\dim \mathcal{K}_{k+1}(A, v_1) \leq \text{rank}(X) = k$. Now, H unreduced implies $\dim \mathcal{K}_j(A, v_1) = j$ for $1 \leq j \leq k-1$ and it follows from Theorem 2.7 that $r_k = 0$. \square

A similar result may be formulated in terms of Schur vectors instead of generalized eigenvectors. This result will be stated without its proof, which is very similar to the proof of the previous result.

THEOREM 2.9. *Let $AV_k - V_k H_k = r_k e_k^T$ be a k -step Arnoldi factorization of A , with H unreduced. Then $r_k = 0$ if and only if $v_1 = Qy$ where $AQ = QR$ with $Q^H Q = I_k$ and R upper triangular of order k .*

Theorem 2.8 provides the motivation for the algorithms we shall develop. It suggests that one might find an invariant subspace by iteratively replacing the starting vector with a linear combination of approximate eigenvectors corresponding to eigenvalues of interest. Such approximations are readily available through the Arnoldi factorization. This theorem also indicates that it will be impossible to force the residual to zero if v_1 has a component of a generator of a cyclic subspace of dimension greater than k . Theorem 2.9 indicates that our computations can be carried out within the framework of a truncated Schur decomposition and this leads to the development of an implicit restart method that is analogous to the implicitly shifted QR iteration.

3. Updating the Arnoldi factorization via QR-iterations. In this section a direct analogue of the implicitly shifted QR-iteration will be derived in the context of the k -step Arnoldi factorization. This will lead to an updating formula that may be used to implement iterative techniques designed to drive the residual r_k to zero by iteratively forcing v_1 into a subspace spanned by k Schur vectors of A .

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

$$\begin{aligned} (3.1) \quad AV_{k+p} &= V_{k+p} H_{k+p} + r_{k+p} e_{k+p}^T \\ &= (V_{k+p}, v_{k+p+1}) \begin{pmatrix} H_{k+p} \\ \beta_{k+p} e_{k+p}^T \end{pmatrix}. \end{aligned}$$

An analogy of the explicitly shifted QR-algorithm may be applied to this truncated factorization of A . It consists of the following four steps. Let μ be a shift and let $(H - \mu I) = QR$ with Q orthogonal and R upper triangular. Then (putting $V = V_{k+p}$, $H = H_{k+p}$)

$$(3.1.1) \quad (A - \mu I)V - V(H - \mu I) = r_{k+p}e_{k+p}^T,$$

$$(3.1.2) \quad (A - \mu I)V - VQR = r_{k+p}e_{k+p}^T,$$

$$(3.1.3) \quad (A - \mu I)(VQ) - (VQ)(RQ) = r_{k+p}e_{k+p}^T Q,$$

$$(3.1.4) \quad A(VQ) - (VQ)(RQ + \mu I) = r_{k+p}e_{k+p}^T Q.$$

Let $V_+ = VQ$ and $H_+ = RQ + \mu I$. Then H_+ is upper Hessenberg and applying the matrices in (3.1.2) to the vector e_1 to expose the relationship of their first columns gives

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

where $\rho_{11} = e_1^T R e_1$, $v_1^+ = V_+ e_1$.

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

$$(3.2) \quad AV_{k+p}^+ = (V_{k+p}^+, v_{k+p+1}) \begin{pmatrix} H_{k+p}^+ \\ \beta_{k+p} e_{k+p}^T \hat{Q} \end{pmatrix}$$

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

$$(3.3) \quad V_{k+p}^+ = (V_k^+, \hat{V}_p), \quad H_{k+p}^+ = \begin{pmatrix} H_k^+ & M \\ \hat{\beta}_k e_1 e_k^T & \hat{H}_p \end{pmatrix},$$

and note

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

Substituting into (3.2) gives

$$(3.4) \quad A(V_k^+, \hat{V}_p) = (V_k^+, \hat{V}_p, v_{k+p+1}) \begin{bmatrix} H_k^+ & M \\ \hat{\beta}_k e_1 e_k^T & \hat{H}_p \\ \beta_{k+p} e_{k+p}^T & b^T \end{bmatrix}.$$

Equating the first k columns on both sides of (3.4) gives

$$(3.5) \quad AV_k^+ = V_k^+ H_k^+ + r_k^+ e_k^T$$

so that

$$(3.6) \quad AV_k^+ = (V_k^+, v_{k+1}^+) \begin{pmatrix} H_k^+ \\ \beta_k^+ e_k^T \end{pmatrix}$$

where $v_{k+1}^+ = (1/\beta_k^+)r_k^+$, $r_k^+ \equiv (\hat{V}_p e_1 \hat{\beta}_k + v_{k+p+1} \tilde{\beta}_{k+p})$, 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 (3.6) 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 (3.1). This requires only p evaluations of a matrix-vector product 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) Deflation techniques similar to those 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 3.7.

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

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

Output: $AV - VH = re_{k+p}^T$ with $V^T V = I_{k+p}$, $V^T r = 0$.

- (1) *For* $j = 1, 2, \dots, p$
 - (1) $\beta \leftarrow \|r\|$; *if* $\beta < \text{tol}$ *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) $h \leftarrow V^T w$; $H \leftarrow (H, h)$;
 - (5) $r \leftarrow w - Vh$;
 - (6) *while* $\|s\| > \epsilon\|r\|$;
 - (1) $s = V^T r$;
 - (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 3.8.

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

- (1) *initialize* $V(:, 1) = v_1$; $H \leftarrow (v_1^T Av_1)$; $r \leftarrow Av_1 - v_1 H$;
- (2) $[H, V, r] \leftarrow \text{Arnoldi}(A, H, V, r, 1, k)$
- (3) *For* $m = 1, 2, \dots$
 - (1) *if* $(\|r\| < \text{tol})$ *then stop*;
 - (2) $[V, H, r] \leftarrow \text{Arnoldi}(A, H, V, r, k, p)$;

- (3) $u = \text{Shifts}(H, p)$; (defined below)
- (4) $Q \leftarrow I_{k+p}$;
- (5) for $j = 1, 2, \dots, p$
 - (1) $H \leftarrow Q_j^T H Q_j$; (Bulge-Chase corresponding to shift $\mu_j = u(j)$)
 - (2) $Q \leftarrow Q Q_j$;
- (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(5(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.

Remark 3. During a Bulge-Chase sweep at step (3(5(1))), it may happen that a subdiagonal element β_j becomes small. The deflation strategies associated with the QR-algorithm are then employed. In this case, the matrix H is split, giving

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

Thus, an invariant subspace of dimension j has been found. If $j \geq k$ and all the shifts have been applied, then the iteration is halted. Otherwise H_j, V_j are retained and the iteration proceeds with \hat{V}_j, \hat{H}_j filling the role of V, H , respectively. However, H_j continues to participate in the shift selection strategy on subsequent iterations. That is, all of the eigenvalues of H are considered in the selection process. If some of the eigenvalues of H_j are selected as shifts then these are applied implicitly to H_j to split this matrix and the unwanted portion is discarded to form a submatrix of smaller size. If the matrix is nonsymmetric the factorization must be explicitly restarted at the $j + 1$ position with a vector that is orthogonal to the first j basis vectors. If the matrix A is symmetric then the corresponding columns of the (updated) matrix V_j are discarded and then \hat{V}_j and \hat{H}_j are moved (concatenated) to the left. The remaining shifts are applied implicitly to \hat{H}_j and then the Arnoldi factorization is completed to fill out the remainder of the $k + p$ columns of V . In this way the iteration is not terminated by deflation until the appropriate approximation to the wanted spectrum has appeared.

As discussed at the beginning of this section, each application of an implicit shift μ_j will replace the starting vector v_1 with $(A - \mu_j I)v_1$. Thus after completion of each cycle of the loop at step (3) in Algorithm 3.8:

$$V e_1 = v_1 \leftarrow \psi(A)v_1;$$

where $\psi(\lambda) = (1/\tau) \prod_{j=1}^p (\lambda - \mu_j)$ with τ a normalization factor. Numerous choices are possible for the selection of these p shifts. Some possibilities will be discussed in §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 as in Algorithm 3.9.

ALGORITHM 3.9.

function $[u] = \text{Shifts}(H, p)$

- (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 criteria for this selection might be

- (i) Sort $\lambda(H)$ according to the algebraically largest real part and select the p eigenvalues with the smallest real part as shifts.

- (ii) Sort $\lambda(H)$ according to the largest modulus and select the p eigenvalues with the smallest modulus as shifts.

Selecting these exact shifts has interesting consequences in the iteration.

LEMMA 3.10. *Let $\lambda(H) = \{\theta_1, \dots, \theta_k\} \cup \{\mu_1, \dots, \mu_p\}$ be a disjoint partition of the spectrum of H 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 $\beta_j \neq 0$, $1 \leq j \leq k-1$, then $\beta_k = 0$ and

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

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

$$v_1^+ = V Q e_1 = \sum x_j,$$

where each x_j is a Ritz vector corresponding to the Ritz value θ_j , i.e., $x_j = V y_j$, where $H y_j = y_j \theta_j$, $1 \leq j \leq k$.

Proof. After applying the p implicit shifts we have

$$H Q = Q H_+$$

with

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

Therefore $q_1 = \sum_{j=1}^k y_j \zeta_j$ where $H y_j = y_j \theta_j$, since $q_1 = \psi(H) e_1$ has annihilated any component of e_1 along an eigenvector of H associated with μ_j , $1 \leq j \leq p$. As a consequence of Theorem 2.8, $\beta_k = 0$ must hold. Moreover, $v_1^+ = V Q e_1 = V q_1 = \sum_{j=1}^k V y_j \zeta_j = \sum_{j=1}^k x_j \zeta_j$. \square

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 $v_1 \leftarrow \sum x_j \zeta_j$, 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 similar to the single vector s -step Lanczos process described by Cullum and Donath in [5] and expanded on in [6], [8]. The particular linear combination is apparently different. This variant has the advantage that a restart of the entire Lanczos sequence is not required. Approximate eigenvectors from a Krylov subspace of dimension $k + p$ are available at each iteration for a cost of p rather than $k + p$ matrix-vector products per iteration.

4. Some polynomial filters. The previous discussion has indicated that it would be advantageous to construct polynomials $\psi(\lambda)$ of degree p that 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].

We have just described the use of exact shifts to construct such filters. Another particularly appealing polynomial filter may be constructed using Tchebyshev polynomials. In this case, one constructs an ellipse containing the unwanted eigenvalues of H , then at step (3(3)) of Algorithm 3.8 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 that 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]. Since each of these schemes specifies the filter polynomials by their roots, they may all be implemented within the framework of the algorithms developed in the previous section. At some point in the iteration one might consider fixing the roots used as shifts and continuing with a stationary iteration. The convergence of this strategy is analyzed in the following section.

One may observe that these filters each have the feature of weighting the eigenvalues of the wanted spectrum quite unevenly. For example, in the symmetric case where the wanted spectrum consists of the k largest eigenvalues, the eigenvalues closest to the right end of the spectrum are weighted 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 that 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 $y^T = (\eta_1, \eta_2, \dots, \eta_{p+1}, 0, 0, \dots, 0)$ since

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

The technique developed in §3 for the implicit application of $\psi(A)$ to v_1 is not directly applicable because the roots of ψ are unknown. One could perhaps compute these roots and then apply the scheme of §3. However, there is an alternate way to implicitly apply this polynomial directly from its expansion in the Arnoldi basis. Assume that $\|y\| = 1$ and construct a vector w_o such that

$$(4.1) \quad (I - 2w_o w_o^T) e_1 = y.$$

Replace H by

$$(4.2) \quad \hat{H} = (I - 2w_o w_o^T) H (I - 2w_o w_o^T).$$

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

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

where

$$(4.3) \quad Q = (I - 2w_0w_0^T)(I - 2w_1w_1^T) \cdots (I - 2w_{k+p-2}w_{k+p-2}^T),$$

with each $(I - 2w_jw_j^T)$ 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:

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

In order to fit within the updating framework developed in §3, the condition

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

must hold. This is established by the following lemma.

LEMMA 4.4. *The matrix Q displayed in (4.3) satisfies $e_{k+p}^T Q e_j = 0$, $1 \leq j < k$.*

Proof. Let $Q_j = I - 2w_jw_j^T$ for $0 \leq j \leq k+p-2$, and let $H^{(j+1)} = Q_j^T H^{(j)} Q_j$ with $H^{(0)} = H$. From (4.1) it follows that $w_0 = \theta(y - e_1)$, with $\frac{1}{\theta} = \|y - e_1\|$. Thus, $e_i^T Q_0 = e_i^T$ for $i > p+1$. Since

$$Q_0 H^{(1)} = H Q_0$$

and since H is upper Hessenberg, it follows that

$$e_i^T H^{(1)} = e_i^T H Q_0 = e_i^T H$$

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. \square

This observation allows the application of any polynomial filter of degree p when the polynomial is expanded in the Arnoldi basis.

Moreover, it provides the means for implicit application of all of the suggested restart methods known to this author which are not specified by roots of polynomials.

5. Some convergence results. In this section, we analyze the algorithm just developed with respect to two strategies for constructing the filter polynomials. The first analysis applies to general (nonsymmetric) matrices but the filter polynomials are stationary. The second analyzes the “exact shift” polynomials but applies only to symmetric matrices. In this sense, each of the two is incomplete. However, they both give insight to the nature of the convergence of this method.

Let us begin with an analysis of the stationary iteration. To this end we define

$$\lambda(A) = \lambda_W(A) \cup \lambda_U(A)$$

where $\lambda_W(A) = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$ and $\lambda_U(A) = \{\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n\}$. By a stationary iteration, we mean that the set of shifts $\{\mu_1, \mu_2, \dots, \mu_p\}$ used to construct the filter polynomial remains fixed. This means that the filter polynomials are all multiples of a fixed (i.e., stationary) polynomial $\psi(\lambda) = \prod_{i=1}^k (\lambda - \mu_i)$.

We define

$$v_1^{(0)} = v \quad \text{and} \quad v_1^{(j)} = \psi(A)v_1^{(j-1)} / \|\psi(A)v_1^{(j-1)}\|,$$

where v is an arbitrary starting vector of length one. We define $\pi_j = \beta_1^{(j)}\beta_2^{(j)}\dots\beta_k^{(j)}$ where the $\beta_i^{(j)}$ are the subdiagonal elements of H in the Arnoldi factorization resulting from the starting vector $v_1^{(j)}$.

THEOREM 5.1. *Assume that the fixed polynomial ψ satisfies*

$$|\psi(\lambda_1)| \geq |\psi(\lambda_2)| \geq \dots \geq |\psi(\lambda_k)| > |\psi(\lambda_{k+1})| \geq \dots \geq |\psi(\lambda_n)|,$$

with

$$\gamma = |\psi(\lambda_{k+1})|/|\psi(\lambda_k)| < 1,$$

and assume that the starting vector v_1 is not a member of the invariant subspace corresponding to $\{\lambda_{k+1}, \dots, \lambda_n\}$. Then the sequence $\{\pi_j\}$ converges to zero. Moreover, there is a fixed constant K and a positive integer J such that

$$0 \leq \pi_j \leq \gamma^j K$$

for all $j > J$.

Proof. Let

$$(5.2) \quad A(Q_1, Q_2) = (Q_1, Q_2) \begin{pmatrix} R_1 & M \\ 0 & R_2 \end{pmatrix}$$

be a Schur decomposition of A with $\lambda(R_1) = \lambda_W(A)$. The hypothesis on ψ assures $\lambda_W(A) \cap \lambda_U(A)$ is empty and thus there is a unique $k \times (n - k)$ matrix solution Γ to the Sylvester equation

$$\Gamma R_2 - R_1 \Gamma = M.$$

Since

$$\begin{pmatrix} I & \Gamma \\ 0 & I \end{pmatrix} \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix} = \begin{pmatrix} R_1 & M \\ 0 & R_2 \end{pmatrix} \begin{pmatrix} I & \Gamma \\ 0 & I \end{pmatrix},$$

it follows that

$$\phi(A)(Q_1, Q_1\Gamma + Q_2) = (Q_1, Q_1\Gamma + Q_2) \begin{pmatrix} \phi(R_1) & 0 \\ 0 & \phi(R_2) \end{pmatrix}$$

for any polynomial ϕ . Let $\hat{Q}_2 = Q_1\Gamma + Q_2$ and let $v_1 = Q_1y_1 + \hat{Q}_2y_2$. Then,

$$\psi^j(A)v = Q_1\psi^j(R_1)y_1 + \hat{Q}_2\psi^j(R_2)y_2 \equiv Q_1y_1^{(j)} + \hat{Q}_2y_2^{(j)}.$$

Recall that $v_1^{(j)} = \psi^j(A)v_1/\|\psi^j(A)v_1\|$ and that from Theorem 2.7, one obtains $\pi_j = \min_{p \in \mathcal{PM}_k} \{\|p(A)v_1^{(j)}\|\}$. Thus, putting \hat{p} to be the characteristic polynomial of R_1 gives

$$\begin{aligned}\pi_j \|\psi^j(A)v_1\| &\leq \|\hat{p}(A)\psi^j(A)v_1\| \leq \|\hat{p}(A)Q_1y_1^{(j)} + \hat{p}(A)\hat{Q}_2y_2^{(j)}\| \\ &= \|Q_1\hat{p}(R_1)y_1^{(j)} + \hat{p}(A)\hat{Q}_2y_2^{(j)}\| = \|\hat{p}(A)\hat{Q}_2y_2^{(j)}\|.\end{aligned}$$

Dividing both sides by $|\psi^j(\lambda_k)|$ gives

$$\pi_j (\|\psi^j(A)v_1\|/|\psi^j(\lambda_k)|) \leq \|\hat{p}(A)\hat{Q}_2y_2^{(j)}\|/|\psi^j(\lambda_k)| \leq \left\| \hat{p}(A)\hat{Q}_2 \left[\frac{1}{\psi(\lambda_k)}\psi(R_2) \right]^j y_2 \right\|.$$

Note the spectral radius of $(1/\psi(\lambda_k))\psi(R_2)$ is less than the number γ according to the hypothesis. Using the fact that there is a consistent matrix norm ν such that $\nu([(1/\psi(\lambda_k))\psi(R_2)]^j) \leq \gamma^j$, together with the equivalence of matrix norms, implies the existence of a positive constant K_o such that

$$\left\| \hat{p}(A)\hat{Q}_2 \left[\frac{1}{\psi(\lambda_k)}\psi(R_2) \right]^j y_2 \right\| \leq K_o \gamma^j \|y_2\|.$$

Thus

$$\pi_j \leq \frac{K_o \gamma^j \|y_2\|}{(\|\psi^j(A)v_1\|/|\psi^j(\lambda_k)|)}.$$

Moreover, by hypothesis v_1 is not in the invariant subspace corresponding to $\{\lambda_{k+1}, \dots, \lambda_n\}$, and it follows that $y_1 \neq 0$. Since the spectral radius of $\psi(R_2)/\psi(\lambda_k)$ is less than one and since every eigenvalue (i.e., every diagonal element) of the triangular matrix $\psi(R_1)/\psi(\lambda_k)$ is greater than or equal to one, there is a J such that $j > J$ implies

$$\begin{aligned}\|\psi^j(A)v_1\|/|\psi^j(\lambda_k)| &= \|Q_1\psi^j(R_1)y_1 + \hat{Q}_2\psi^j(R_2)y_2\|/|\psi^j(\lambda_k)| \\ &\geq \|Q_1\psi^j(R_1)y_1\|/|\psi^j(\lambda_k)| - \|\hat{Q}_2\psi^j(R_2)y_2\|/|\psi^j(\lambda_k)| \\ &\geq |\eta| - \frac{1}{2}|\eta|,\end{aligned}$$

where η is the last nonzero component of y_1 . Thus, the result is established with $K = 2K_o(\|y_2\|/|\eta|)$. \square

This result applies directly and generally to cases where there is a priori information about the location of the spectrum and where a single polynomial might be constructed (e.g., Tchebyshev polynomials). The analysis does not apply to the adaptive algorithms. However, it does give an indication of how these might behave near the final stages of the iteration where the filter polynomials tend to become stationary.

Next, we analyze the “exact shift” filter polynomials in the symmetric case. This analysis makes heavy use of the interlace property of eigenvalues of symmetric matrices modified by low-rank changes. The analysis relies on two technical lemmas, which we must establish first.

LEMMA 5.3. *Let*

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

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

$$(5.4) \quad \beta^2 e_k^T (T - \lambda I)^{-1} e_k = \alpha - \lambda$$

are eigenvalues of M .

See [25] for a proof.

If $T = Y\Theta Y^T$ is the eigendecomposition of T then (5.4) becomes

$$(5.5) \quad \beta^2 \sum_{i=1}^k \frac{\eta_i^2}{(\theta_i - \lambda)} = \alpha - \lambda$$

where $(\eta_1, \eta_2, \dots, \eta_k) = e_k^T Y$ and $\Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_k)$. It is easily verified that this equation has exactly one root in each of the intervals

$$(-\infty, \theta_1), (\theta_1, \theta_2), \dots, (\theta_{k-1}, \theta_k), (\theta_k, \infty)$$

and that the k largest roots $\hat{\theta}_i$ of this equation satisfy

$$\theta_i < \hat{\theta}_i \quad \text{for } i = 1, \dots, k.$$

Also, we note for the sequel that if the subdiagonals of T are all nonzero, then none of the η_j are zero and the θ_j are distinct.

The next lemma shows that when the starting vector v_1 nears a subspace of dimension less than k , then deflation must occur.

LEMMA 5.6. *Suppose $AV = VH + re_k^T$ is an Arnoldi factorization of A and let β_j be the j th subdiagonal element of H . If $v_1 = q\gamma + w\sigma$ with $\gamma^2 + \sigma^2 = 1$, $\|q\| = \|w\| = 1$, $q^T w = 0$, and $q = \sum_{j=1}^i q_j \gamma_j$, $Aq_j = q_j \lambda_j$ (where $\{\lambda_j\}$ are an arbitrary set of i eigenvalues of A), then*

$$\prod_{j=1}^i \beta_j \leq \sigma \left\| \prod_{j=1}^i (A - \lambda_j I) \right\|.$$

Proof. From Theorem 2.7 and the fact that $\prod_{j=1}^i (A - \lambda_j I)q = 0$, we have

$$\begin{aligned} \prod_{j=1}^i \beta_j &= \min_{p \in \mathcal{PM}_i} \|p(A)v_1\| \\ &\leq \left\| \prod_{j=1}^i (A - \lambda_j I)(q\gamma + w\sigma) \right\| \\ &= \left\| \prod_{j=1}^i (A - \lambda_j I)w\sigma \right\| \\ &\leq \sigma \left\| \prod_{j=1}^i (A - \lambda_j I) \right\|. \end{aligned} \quad \square$$

With these technical lemmas established it will be possible to analyze the iterations using polynomial filters constructed from exact shifts in the symmetric case.

We assume throughout the remainder of this section that the matrix A is symmetric and hence that $H = T$ is tridiagonal. The selection rule to be analyzed is to retain the k 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

$$\mu_{1,m+1} < \cdots < \mu_{p,m+1} < \theta_{1,m+1} < \cdots < \theta_{k,m+1}$$

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

$$\theta_{1m} < \theta_{2m} < \cdots < \theta_{km}.$$

Then, the exact implicit shift strategy provides

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

where $Q^{(m)} = Q_{1m} Q_{2m} \cdots Q_{pm}$ are the orthogonal matrices constructed to apply the implicit shifts $\mu_{1m} \cdots \mu_{pm}$ at step (3(5)) of Algorithm 3.8. Step (3(6)) gives

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

LEMMA 5.7. *Each $\{\theta_{j,m} : m = 1, 2, \dots\}$ is an increasing convergent sequence for each $j = 1, 2, \dots, k$.*

Proof. Since $T_{k+p}^{(m)}$ is obtained through a succession of p borderings of $T_k^{(m)}$, it follows from p successive applications of Lemma 5.3 that

$$\theta_{j,m} < \theta_{j,m+1} \quad \text{for } j = 1, \dots, k.$$

Since each $\theta_{j,m}$ is a Raleigh quotient with respect to A it follows that $\lambda_1 \leq \theta_{j,m} \leq \lambda_n$ for all j, m . Since bounded increasing sequences are convergent the lemma is established. \square

We now establish that the limits of the convergent sequences $\theta_{j,m}$ are eigenvalues of the matrix A .

LEMMA 5.8. *Let $T_k^{(m)} = Y^{(m)} \Theta^{(m)} Y^{(m)T}$ where $(\eta_1^{(m)}, \eta_2^{(m)}, \dots, \eta_k^{(m)}) = e_k^T Y^{(m)}$. Assume θ_j are distinct, where $\theta_{j,m} \rightarrow \theta_j$. Then*

$$\beta_k^{(m)} \eta_j^{(m)} \rightarrow 0 \quad \text{as } m \rightarrow \infty, \quad \text{for } j = 1, \dots, k$$

and as a consequence

$$\|AV^{(m)} y_j^{(m)} - y_j^{(m)} \theta_{j,m}\| = |\beta_k^{(m)} \eta_j^{(m)}| \rightarrow 0,$$

where $y_j^{(m)} = Y^{(m)}e_j$ for $j = 1, \dots, k$.

Proof. Consider the leading $(k+1) \times (k+1)$ submatrix of $T_{k+p}^{(m)}$

$$M^{(m)} = \begin{pmatrix} T_k^{(m)} & \beta_k^{(m)} e_k \\ \beta_k^{(m)} e_k^T & \alpha^{(m)} \end{pmatrix}.$$

From Lemma 5.3 it follows that the k largest eigenvalues $\hat{\theta}_{j,m}$ of $M^{(m)}$ satisfy

$$\theta_{j,m} < \hat{\theta}_{j,m} < \theta_{j,m+1}.$$

Moreover, a straightforward algebraic manipulation of (5.5) gives

$$(\beta \eta_j)^2 = (\theta_j - \lambda) \left[\frac{(\alpha - \lambda) - \beta^2 \sum_{i=j+1}^k \frac{\eta_i^2}{(\theta_i - \lambda)}}{1 + \sum_{i=1}^{j-1} \frac{\eta_i^2 (\theta_j - \lambda)}{\eta_i^2 (\theta_i - \lambda)}} \right]$$

for any root λ . Substituting the appropriate quantities indexed by m from the matrix $M^{(m)}$ and putting $\lambda = \hat{\theta}_{j,m}$ gives

$$(\beta_k^{(m)} \eta_j^{(m)})^2 < |(\theta_{j,m} - \hat{\theta}_{j,m})| \left| (\alpha^{(m)} - \hat{\theta}_{j,m}) - \beta_k^{(m)^2} \sum_{i=j+1}^k \frac{\eta_i^{(m)^2}}{(\theta_{i,m} - \hat{\theta}_{j,m})} \right|.$$

The assumption that the limits θ_j are distinct implies that the quantities

$$\left| (\alpha^{(m)} - \hat{\theta}_{j,m}) - \beta_k^{(m)^2} \sum_{i=j+1}^k \frac{\eta_i^{(m)^2}}{(\theta_{i,m} - \hat{\theta}_{j,m})} \right|$$

have finite limits for each j . Hence, for m sufficiently large there is a positive constant K such that

$$(\beta_k^{(m)} \eta_j^{(m)})^2 < K |\theta_{j,m} - \hat{\theta}_{j,m}| < K |\theta_{j,m} - \theta_{j,m+1}| \rightarrow 0$$

as $m \rightarrow \infty$. \square

The final task is to show that not only will the limits θ_j be eigenvalues of A , but they will be the k eigenvalues of interest. To do this, we shall show that if deflation does not occur, then the coefficients $\gamma_j^{(m)} = q_j^T v_1^{(m)}$ must converge to zero for $j = 1, \dots, n-k$ where q_j is the eigenvector of A corresponding to the eigenvalue λ_j and $v_1^{(m)}$ is the Arnoldi starting vector for the m th iteration.

Define $\psi_\ell(\lambda) = \prod_{i=1}^p (\lambda - \mu_{i,\ell})$ and $\Psi_m(\lambda) = \prod_{i=1}^m \psi_i(A)$. Then note that $v_1^{(m)} = (\Psi_m(A)v_1 / \|\Psi_m(A)v_1\|)$.

THEOREM 5.9. *Suppose that the initial starting vector v_1 satisfies $q_j^T v_1 = \gamma_j \neq 0$ for $j = n-k+1, \dots, n$, where q_j is the eigenvector of A corresponding to the eigenvalue λ_j with the eigenvalues of A listed in increasing order. Let $\beta_i^{(m)}$ be the i th subdiagonal element of $T_k^{(m)}$ and assume that $\beta_i^{(m)} > \epsilon > 0$ for all i, m . Then the sequences*

$$\theta_{j,m} \rightarrow \theta_j = \lambda_{n-k+j} \quad \text{as } m \rightarrow \infty.$$

Proof. The assumption $\beta_i^{(m)} > \epsilon > 0$ assures that separation of the $\theta_{j,m}$ is uniform over all m so that the limits θ_j are distinct. This implies that each θ_j is an eigenvalue

of A . Moreover, the assumption implies a uniform lower bound on the quantities $|\eta_j^{(m)}|$ and thus $\beta_k^{(m)} \rightarrow 0$. (All due to remarks following Lemma 5.3.)

Let $\phi_m(\lambda) = \prod_{i=1}^k (\lambda - \theta_{j,m})$, and let $\phi(\lambda) = \prod_{i=1}^k (\lambda - \theta_j)$ be the limit polynomial of the ϕ_m . Then

$$\|\phi_m(A)v_1^{(m)}\| = \prod_{j=1}^k \beta_j^{(m)} \rightarrow 0$$

and thus

$$\phi_m(\lambda_j)\gamma_j^{(m)} = q_j^T \phi_m(A)v_1^{(m)} \rightarrow 0.$$

Hence, either

$$\phi(\lambda_j) = 0 \quad \text{or} \quad \gamma_j^{(m)} \rightarrow 0$$

for $j = 1, \dots, n$. This means that $n - k$ of the expansion coefficients $\gamma_j^{(m)}$ tend to 0 as $m \rightarrow \infty$. Moreover, Lemma 5.7 implies that the k expansion coefficients corresponding to the eigenvalues θ_j must all be bounded away from zero due to the assumption $\beta_j^{(m)} > \epsilon > 0$ for all j, m .

Now, suppose that $\lambda_{j_k} = \theta_k < \lambda_n$. Then the expansion coefficient

$$\gamma_{j_k}^{(m)} = q_{j_k}^T v_1^{(m)} = q_{j_k}^T \frac{\Psi_m(A)v_1}{\|\Psi_m(A)v_1\|} = \frac{\gamma_{j_k} \Psi_m(\theta_k)}{\sqrt{\sum_{i=1}^n \gamma_i^2 \Psi_m^2(\lambda_i)}}.$$

Hence

$$(\gamma_{j_k}^{(m)})^2 = \frac{(\gamma_{j_k} \Psi_m(\theta_k) / \Psi_m(\lambda_n))^2}{\gamma_n^2 + \sum_{i=1}^{n-1} \gamma_i^2 \Psi_m^2(\lambda_i) / \Psi_m^2(\lambda_n)} \leq \left(\frac{\gamma_{j_k} \Psi_m(\theta_k)}{\gamma_n \Psi_m(\lambda_n)} \right)^2$$

where the γ_i are the expansion coefficients of $v_1^{(o)}$. Now, the roots $\mu_{i,m}$ of the filter polynomials all satisfy $\lambda_1 \leq \mu_{i,m} < \theta_{k,m} \leq \theta_k < \lambda_n$ so that

$$0 \leq \frac{\Psi_m(\theta_k)}{\Psi_m(\lambda_n)} = \prod_{\ell=1}^m \left(\prod_{i=1}^p \left(\frac{\theta_k - \mu_{i\ell}}{\lambda_n - \mu_{i\ell}} \right) \right) \leq \left(\frac{\theta_k - \lambda_1}{\lambda_n - \lambda_1} \right)^{mp} \rightarrow 0,$$

since $(\theta_k - \lambda_1 / \lambda_n - \lambda_1) < 1$. This is a contradiction. We conclude that $\theta_k = \lambda_n$.

A similar argument may be carried out for each j in turn for the cases $\theta_j < \lambda_{n-k+j}$ and this concludes the proof. \square

6. 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

$$Ax = \lambda Mx.$$

In many cases the matrix M is symmetric and positive definite, and this condition shall be assumed in this section. The basic iterative method will carry over to this setting with very little modification. In this setting we maintain and update a factorization of the form

$$(6.1) \quad AV - MVH = re_k^T$$

where

$$V^T M V = I \quad \text{and} \quad V^T r = 0.$$

It is easily seen that one may apply the algorithm for the standard case to the matrix $M^{-1}A$ in place of A . Of course this would be implemented through factorization of M at the outset and solution of the appropriate linear system instead of applying M^{-1} .

There are two key consequences of maintaining the form (6.1):

1. $Q^T V^T M V Q = I$ is preserved so the implicit QR-shift strategy may be applied.
2. If $A = A^T$ is symmetric, then

$$H = V^T A V$$

follows from $V^T M V = I$, $V^T r = 0$ 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.

7. 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 §6. The other implements the algorithm for the standard nonsymmetric 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 3.7. Step (3) requires a user-supplied matrix-vector product. Steps (4) and (5) are implemented through calls to the level-two BLAS [11], [12] routine DGEMV. One step of iterative refinement is carried out at step (6) of Algorithm 3.7 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 $\|V^T V - I\|$ 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-two BLAS has provided the means to achieving 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

TABLE 7.1
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

the filter polynomials in order to reduce the required number of user-supplied matrix-vector products. Parallelism may be invoked through the level-two BLAS and also through the user-supplied matrix-vector product.

In all of the results reported below, exact shifts were used as described in §3. 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 five-point discretization of the convection-diffusion operator on the unit square Ω . The partial differential equations (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 that 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 its 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 Tables 7.1 and 7.2, 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.

In Table 7.2 below, the problems of orders 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.

The second set of results will briefly describe two problems that arise in the context of solving PDEs. The first of these involves a discretization of a membrane

TABLE 7.2
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

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 that 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 bilinear 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 compute $A^{-1}M(\rho)v$ for all ρ . The eigenvalue iteration also benefited from the reuse of the converged starting vector from the previous problem but this did not appear to be of great consequence in this case. Table 7.3 gives results for the same subproblem on a variety of machines.

TABLE 7.3
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 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 three to four iterations, with three 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 Edwards and Tuckerman that 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

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

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 pseudospectral 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.

This behavior of the algorithm on these two problems seems to be typical of 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 Figs. 7.1 and 7.2. The second example is the banded Toeplitz matrix used for test purposes by Grcar [17]. This matrix is nonnormal and has a nontrivial pseudospectrum, as discussed in [21]. (The ϵ pseudospectrum of a matrix A is $\{\lambda \in \mathbb{C} : \|(\lambda I - A)^{-1}\| \geq \epsilon^{-1}\}$.) The matrix is a five-diagonal matrix with the value -1 on the first subdiagonal and the value 1 on the main diagonal and the next three superdiagonals. The results for this problem are displayed in Figs. 7.3 and 7.4.

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 .

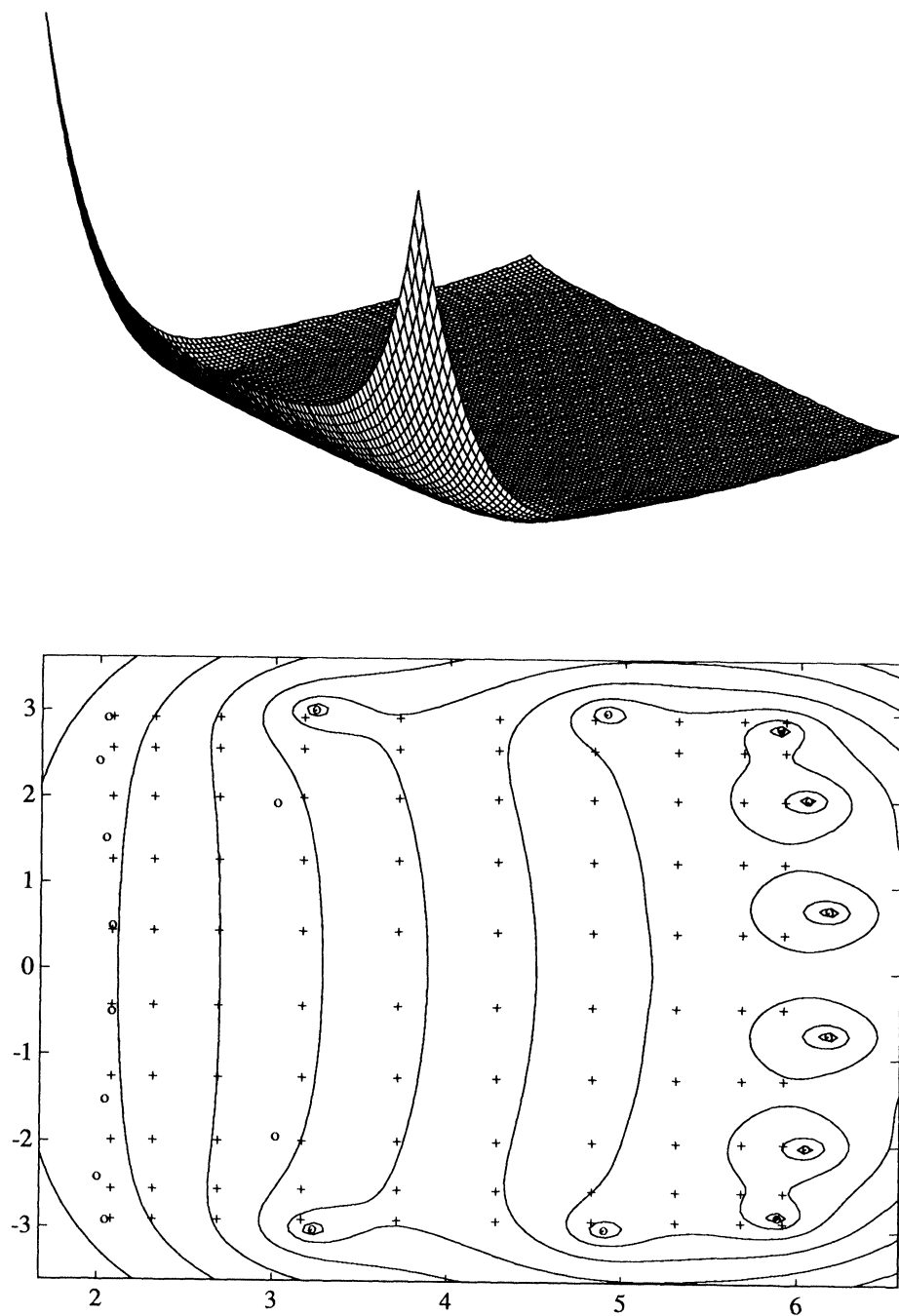


FIG. 7.1. *Convection diffusion: Iteration 1.*

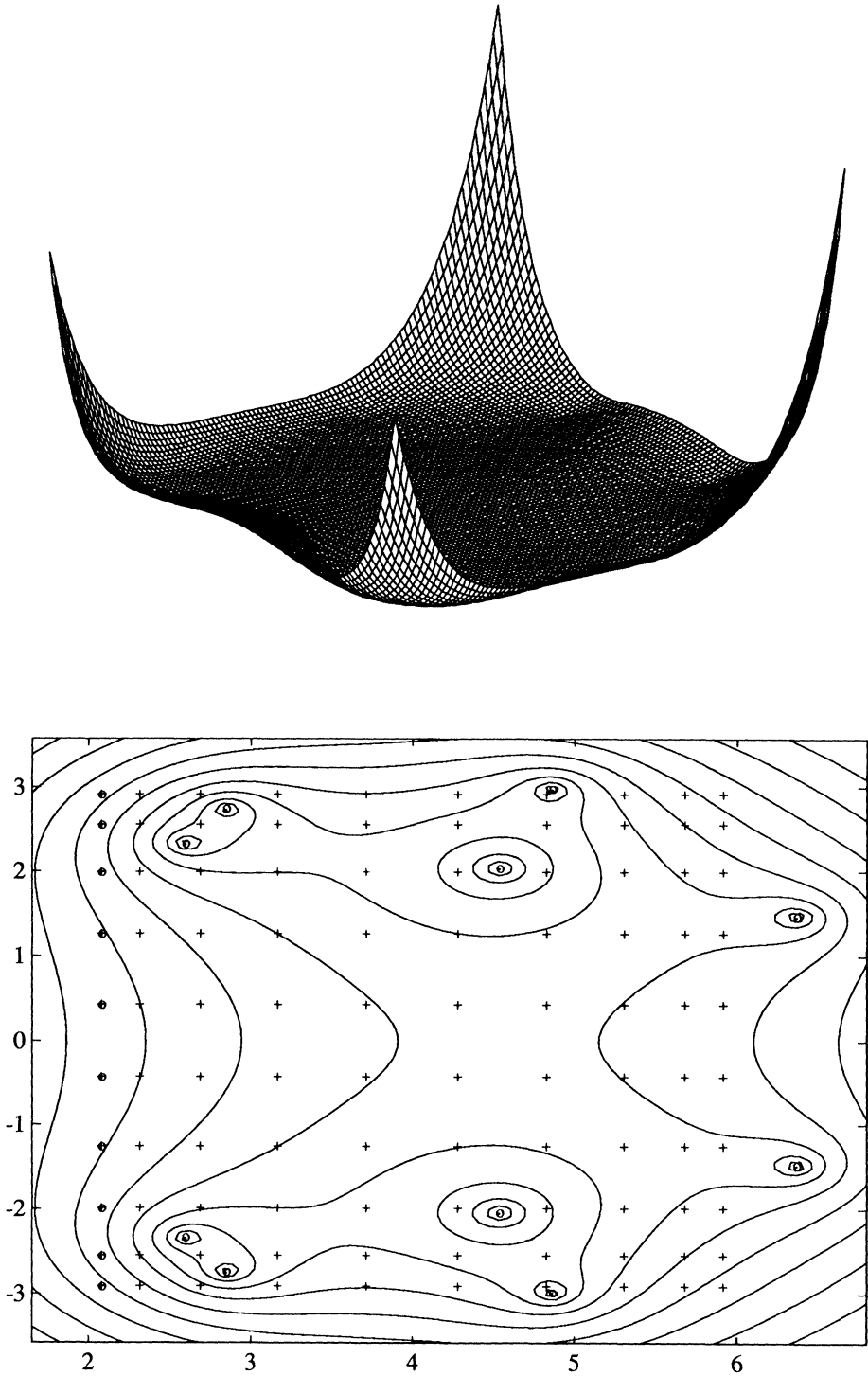


FIG. 7.2. Convection diffusion: At convergence.

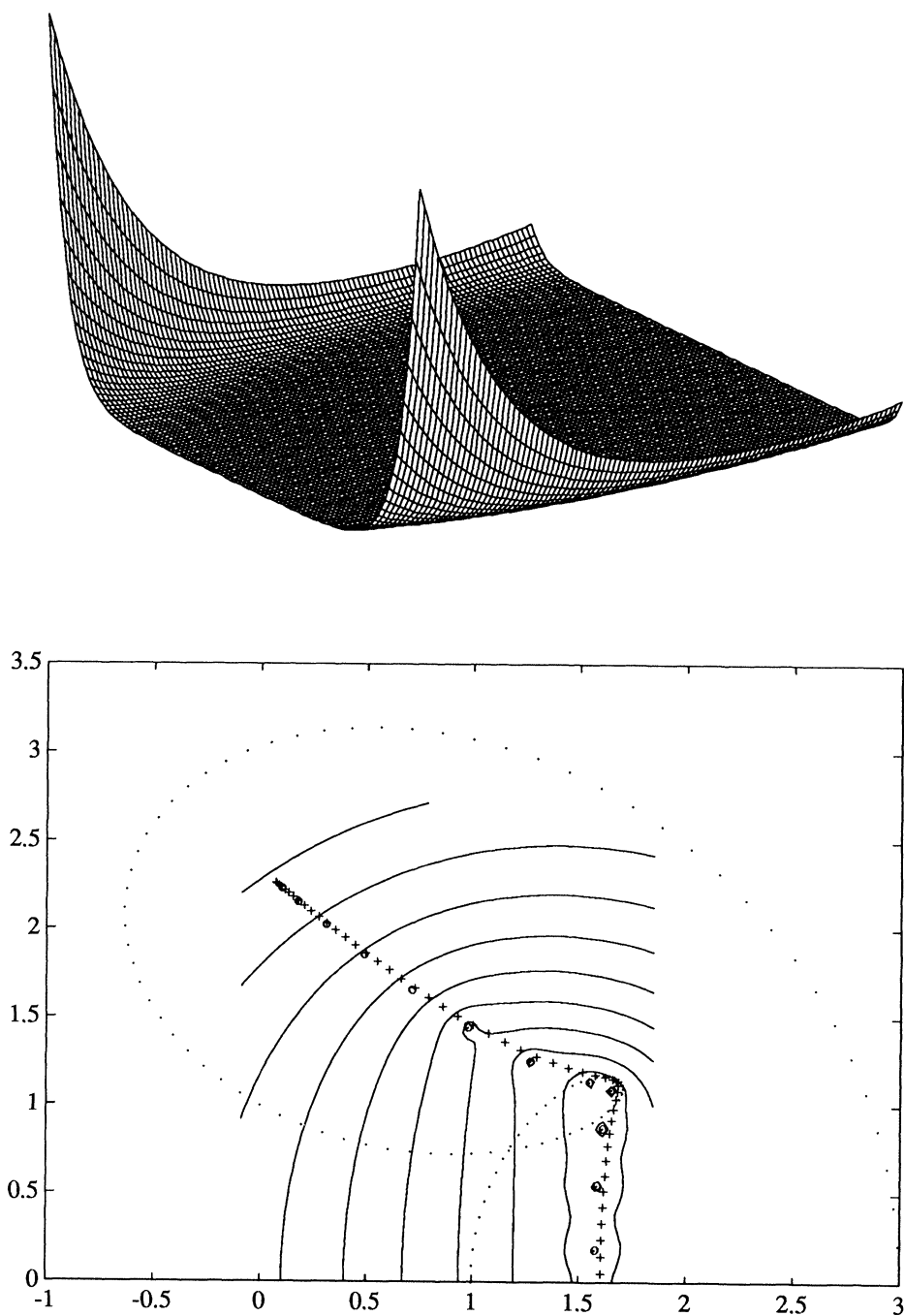


FIG. 7.3. *Grcar matrix: Iteration 1.*

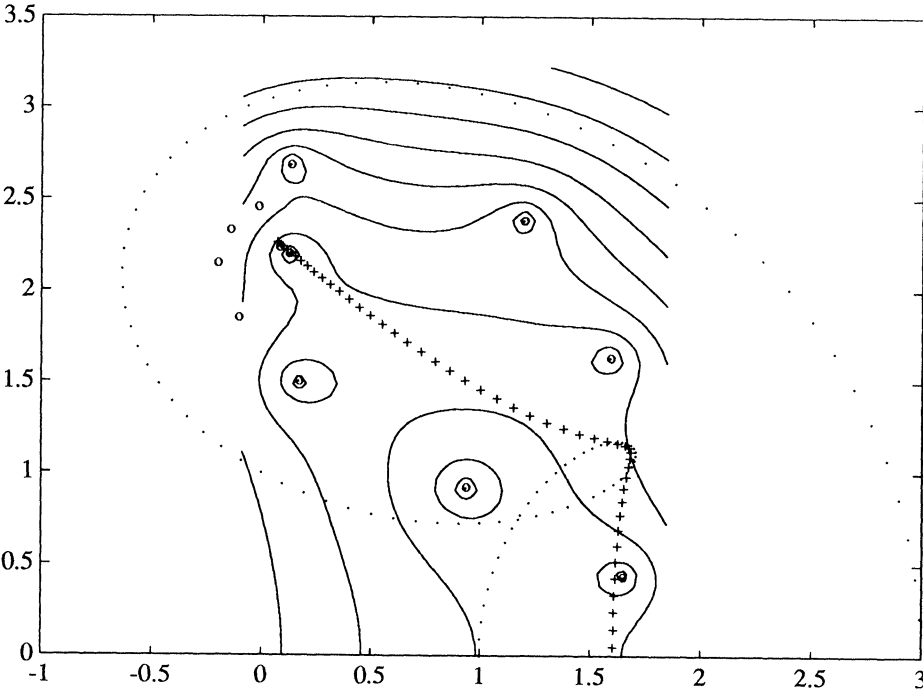
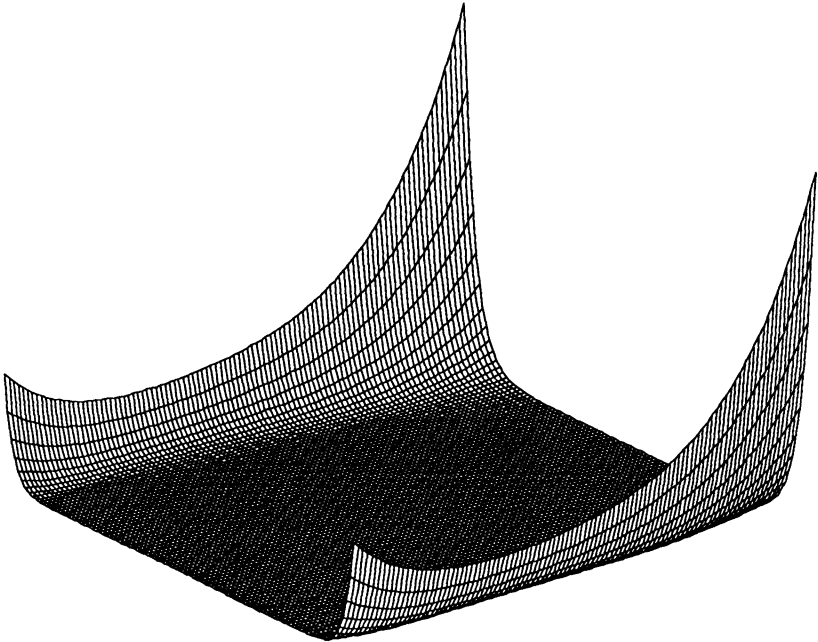


FIG. 7.4. *Grear matrix: At convergence.*

In Figs. 7.1 and 7.2 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 completely understood at the moment.

In Figs. 7.3 and 7.4 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 (i.e., the ten eigenvalues of smallest real part). However, as demonstrated in the figure, these are pseudoeigenvalues. 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 that research is needed in constructing filter polynomials that 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 (i.e., 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 under way and will be the topic of subsequent papers.

Acknowledgments. 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. The deflation strategy described in Remark 3 of §3 arose from a discussion with Dr. Vu. I would also like to thank Dr. S. Parter, Dr. L. Reichel and Dr. P. Vu 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 (1951), pp. 17–29.
- [2] F. CHATELIN AND D. HO, *Arnoldi–Tchebychev procedure for large scale nonsymmetric matrices*, Math. Model. Numer. Anal., 24 (1990), pp. 53–65.

- [3] A. T. CHRONOPOULOS, *s-Step Orthomin and GMRES implemented on parallel computers*, Report TR 90-15, Department of Computer Science, University of Minnesota, Minneapolis, MN, 1989.
- [4] S. COX, *The symmetry, regularity, and computation of a membrane interface*, Department of Mathematics and Science Report, 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, New York, 1974, pp. 505–509.
- [6] J. CULLUM, *The simultaneous computation of a few of the algebraically largest and smallest eigenvalues of a large, symmetric, sparse matrix*, BIT, 18 (1978), pp. 265–275.
- [7] J. CULLUM AND R. A. WILLOUGHBY, *Computing eigenvalues of very large symmetric matrices—an implementation of a Lanczos algorithm with no reorthogonalization*, J. Comput. Phys., 434 (1981), pp. 329–358.
- [8] ———, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, Vol. I. Theory, Birkhäuser, Boston, 1985.
- [9] J. DANIEL, W. B. GRAGG, L. KAUFMAN, AND G. W. STEWART, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.
- [10] J. W. DEMMEL, J. J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, AND D. SORESENSEN, *A prospectus for the development of a linear algebra library for high performance computers*, Report ANL-MCS-TM-97, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 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. Software 14 (1988), pp. 1–17.
- [12] ———, *Algorithm 656. An extended set of fortran basic linear algebra subprograms: Model implementation and test programs*, ACM Trans. Math. Software, 14 (1988), pp. 18–32.
- [13] W. S. EDWARDS, S. R. BEANE, AND S. VARMA, *Onset of wavy vortices in the finite-length Couette–Taylor problem*, Phys. Fluids, submitted.
- [14] J. G. F. FRANCIS, *The QR transformation: A unitary analogue to the LR transformation, Parts I and II*, Comput. J., 4 (1961), pp. 265–272; 332–345.
- [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 University, Stanford, CA, 1972.
- [16] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.
- [17] J. F. GRCAR, *Operator coefficient methods for linear equations*, Sandia National Laboratory Report SAND89-8691, Livermore, CA, 1989.
- [18] W. KARUSH, *An iterative method for finding characteristic vectors of a symmetric matrix*, Pacific J. Math., 1 (1951), pp. 233–248.
- [19] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.
- [20] T. A. MANTEUFFEL, *Adaptive procedure for estimating parameters for the nonsymmetric Tchebychev iteration*, Numer. Math., 31 (1978), pp. 183–208.
- [21] N. NACHTIGAL, L. REICHEL, AND L. N. TREFETHEN, *A hybrid GMRES algorithm for nonsymmetric matrix iterations*, Numerical Analysis Report 90-7, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1990.
- [22] C. C. PAIGE, *The computation of eigenvalues and eigenvectors of very large sparse matrices*, Ph.D. thesis, University of London, London, UK, 1971.
- [23] ———, *Computational variants of the Lanczos method for the eigenproblem*, J. Inst. Math. Appl., 10 (1972), pp. 373–381.
- [24] B. N. PARLETT AND D. S. SCOTT, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 311–328.
- [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 Trans. Math. Software, to appear.
- [27] A. RUHE, *Rational Krylov sequence methods for eigenvalue computation*, Linear Algebra Appl., 58 (1984), pp. 391–405.
- [28] Y. SAAD, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.
- [29] ———, *Projection methods for solving large sparse eigenvalue problems*, in Matrix Pencil Proceedings, B. Kagstrom and A. Ruhe, eds., Springer-Verlag, Berlin, 1982, pp. 121–144.

- [30] Y. SAAD AND M. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [31] D. SCOTT, *Analysis of the symmetric Lanczos algorithm*, Ph.D. thesis, Department of Mathematics, University of California, Berkeley, CA, 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, 6, Second 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. Sci. Statist. Comput., 9 (1988), pp. 152–163.
- [35] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.