RICE UNIVERSITY

# Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration

by

**R. B. Lehoucq**

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

**Doctor of Philosophy**

Approved, Thesis Committee:

Danny C. Sorensen, Chairman
Professor of Computational and Applied Mathematics

C. S. Burrus
Professor of Electrical and Computer Engineering

John E. Dennis
Noah Harding Professor of Computational and Applied Mathematics

William W. Symes
Professor of Computational and Applied Mathematics

Houston, Texas

May, 1995

# Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration

R. B. Lehoucq

## Abstract

The Arnoldi algorithm, or iteration, is a computationally attractive technique for computing a few eigenvalues and associated invariant subspace of large, often sparse, matrices. The method is a generalization of the Lanczos process and reduces to that when the underlying matrix is symmetric. This thesis presents an analysis of Sorensen's Implicitly Re-started Arnoldi iteration, (IRA-iteration), by exploiting its relationship with the QR algorithm. The goal of this thesis is to present numerical techniques that attempt to make the IRA-iteration as robust as the implicitly shifted QR algorithm. The benefit is that the Arnoldi iteration only requires the computation of matrix vector products $w = Av$ at each step. It does not rely on the dense matrix similarity transformations required by the EISPACK and LAPACK software packages.

Five topics form the contribution of this dissertation. The first topic analyzes re-starting the Arnoldi iteration in an implicit or explicit manner. The second topic is the numerical stability of an IRA-iteration. The forward instability of the QR algorithm and the various schemes used to re-order the Schur form of a matrix are fundamental to this analysis. A sensitivity analysis of the Hessenberg decomposition is presented. The practical issues associated with maintaining numerical orthogonality among the Arnoldi/Lanczos basis vectors is the third topic. The fourth topic is deflation techniques for an IRA-iteration. The deflation strategies introduced make it possible to compute multiple or clustered eigenvalues with a single vector re-start method. The block Arnoldi/Lanczos methods commonly used are not required. The final topic is the convergence typical of an IRA-iteration. Both formal theory and heuristics are provided for making choices that will lead to improved convergence of an IRA-iteration.

# Acknowledgments

Special thanks are due Andrew Machusko of the California University of Pennsylvania for all his help during my undergraduate years. At Virginia Polytechnic Institute and State University, Christopher Beattie introduced me to the fascinating area of numerical linear algebra. I thank him for this introduction and his interest and support over the last few years. Zhaojun Bai, Martin Berggren, Jim Demmel, Augustin Dubrulle, Karl Meerbergen, Cliff Nolan, Beresford Parlett, Axel Ruhe, Jennifer Scott, Nick Trefethen, Luis Vicente, David Watkins and the members of my thesis committee provided many helpful comments, references and thoughts along the way.

The two people who I am most indebted to are Danny Sorensen, my thesis advisor, and Richard Hanson. Richard encouraged me to return to school for which I am ever grateful. I also want to thank him for showing me the art of writing mathematical software. I want to give special thanks to Danny for setting high technical standards and then expecting them from me. Danny and Richard have set an excellent example for me through both their high scientific and personal integrity. This thesis is a direct consequence of their influence upon me.

Finally, I wish to thank my many friends, my brother and parents for all their support. My parents, through direct example, showed me the value of what hard work accomplishes.

Financial support for this work was provided in part by the National Science Foundation cooperative agreement CCR-9120008, by the Department of Energy contract DE-FG0f-91ER25103 and by ARPA contract number DAAL03-91-C-0047 administered by the U.S. Army Research Office.

# Contents

# Illustrations

# Tables

# Chapter 1

# Introduction

Many scientific and engineering problems lead to the matrix eigenvalue problem

$$Ax = \lambda Bx, \tag{1.0.1}$$

where $A$ and $B$ are real matrices of order $n$. The matrix $B$, when it arises, is usually symmetric positive semi-definite. In many situations $B = I$, the identity matrix, and this is the case assumed unless stated otherwise. For the remainder of the thesis, we suppose that $A$ is nonsymmetric and real with standard simplifications when the matrix is symmetric.

This thesis examines a promising variant of Arnoldi's method [3] for computing approximations to a few eigenpairs $(x, \lambda)$ of $A$. The Arnoldi method is an efficient procedure for approximating a subset of the eigensystem for a large, often sparse, matrix $A$. The method is a generalization of the Lanczos process [46] and reduces to that when $A$ is symmetric. The process, sequential in nature, produces an upper Hessenberg matrix $H_k$ of order $k$ at the $k$-th step. The eigenvalues of $H_k$ are used to approximate a few of the eigenvalues of $A$. Excellent approximations to some of the eigenvalues often appear for values of $k$ significantly smaller than the order of the matrix. The iteration only requires the computation of a matrix vector product $w = Av$ at each step. It does not rely on the dense matrix similarity transformations required by EISPACK [82] and LAPACK [1].

There are a number of numerical difficulties with Arnoldi/Lanczos methods. These include:

- Maintaining the orthogonality of the Arnoldi/Lanczos basis vectors.
- Reducing the storage requirements of the methods.
- The computation of multiple and clustered eigenvalues of $A$.
- Convergence to a selected group of eigenvalues of $A$.

- Handling spurious eigenvalues when orthogonality is not enforced.

Each of these issues is considered in detail during the course of the thesis. Over a decade of research has been devoted to understanding and overcoming the numerical difficulties of the Lanczos method. The works of Parlett [61], Cullum and Wiloughby [21] study in detail the many specifics of the Lanczos algorithm, while the paper by Grimes, Lewis and Simon [39] discusses the design and development of high quality software.

Development of the Arnoldi method lagged behind due to the inordinate computational and storage requirements associated with the original method when a large number of steps are required for convergence. The explicitly re-started Arnoldi iteration (ERA-iteration) was introduced by Saad [74] to overcome these difficulties, based on similar ideas developed for the Lanczos process by Paige [57], Cullum and Donath [20], and Golub and Underwood [37]. Karush [44] proposes what appears to be the first example of a re-started iteration.

A relatively recent variant was developed by Sorensen [83] as a more efficient and numerically stable way to implement restarting. This technique, the Implicitly Restarted Arnoldi iteration (IRA-iteration), may be viewed as a truncation of the standard implicitly shifted QR-iteration. This thesis presents an analysis of an IRA-iteration that exploits its relationship with the implicitly shifted QR algorithm. This viewpoint provides an alternate approach to study the Arnoldi/Lanczos iterations in which the power of the QR algorithm is utilized. The immediate impact is the improvement of the numerical accuracy and convergence properties of the ARPACK [49] software package.

The goal of this thesis is to present numerical techniques that are designed to make the IRA-iteration as robust as the implicitly shifted QR algorithm for dense problems. These schemes are analyzed with respect to numerical stability and computational results are presented.

## 1.1 Organization of the Thesis

The dissertation is organized as follows. Chapter 2 introduces Arnoldi's method as well as a few of the many associated fundamentals. The QR algorithm is the subject of Chapter 3. A connection between the Arnoldi method and the implicitly shifted QR-iteration is established that is exploited for the remainder of the thesis. The idea of re-starting an Arnoldi iteration is examined in Chapter 4. The IRA-iteration is introduced and a comparison between implicitly and explicitly re-starting an Arnoldi iteration is drawn. Chapter 5 examines the numerical stability of the IRA-iteration by considering the stability of a Hessenberg decomposition. Connections are made with the concept of the forward instability of the QR algorithm, re-orthogonalization methods and the various methods used to re-order the Schur form of a matrix. Deflation techniques for an IRA-iteration are treated in Chapter 6. A numerically stable scheme is introduced that implicitly deflates the converged approximations from the iteration. Two forms of implicit deflation are presented. Convergence of the iteration is improved and a reduction in computational effort is also achieved. The deflation strategies make it possible to compute multiple or clustered eigenvalues with a single vector restart method. A block method is not required. Maintaining orthogonality of the Arnoldi basis vectors is considered in Chapter 7. The convergence typical of an IRA-iteration is the subject of Chapter 8. Both formal theory and heuristics are provided for making choices that will lead to improved convergence of an IRA-iteration. Chapter 9 summarizes the dissertation and examines future work.

## 1.2 Notation and Fundamentals of Matrix Computations

We shall now establish the notation to be used during the course of this thesis. It is also necessary to review a number of details on the matrix factorizations and techniques that will be used.

We employ Householder notational conventions. Capital and lower case letters denote matrices and vectors, respectively, while lower case Greek letters denote scalars. The identity matrix in $\mathbf{R}^{n\times n}$ is denoted by $I_n$ and the subscript is dropped when the context is clear. The $j$-th canonical basis vector is denoted by $e_j$, the $j$-th column of the identity matrix. The transpose of a vector $x$ is denoted by $x^T$ and $x^H$ denotes the complex conjugate of $x^T$. The norms used are the Euclidean and Frobenius denoted by $\|\cdot\|$ and $\|\cdot\|_F$, respectively. The range of a matrix $A$ is denoted by $\mathcal{R}(A)$.

### 1.2.1 The Real Schur Form

Since we are especially concerned with algorithms that result in robust and efficient software, the following decomposition is a special case of the more general Schur decomposition. The special case allows us to compute strictly in real arithmetic. The proper resolution of complex conjugate pairs of eigenvalues comes from noting that if $A(x+iz) = (\nu + i\mu)(x+iz)$ where $x$ and $z$ are vectors in $\mathbf{R}^n$ with $\mu \neq 0$, then

$$A \begin{bmatrix} x & z \end{bmatrix} = \begin{bmatrix} x & z \end{bmatrix} \begin{bmatrix} \nu & \mu \\ -\mu & \nu \end{bmatrix} \equiv \begin{bmatrix} x & z \end{bmatrix} D_2. \tag{1.2.1}$$

The following decomposition proves central to the eigenvalue algorithms considered in this thesis.

**Theorem 1.1** (Real Schur Decomposition) If $A \in \mathbf{R}^{n \times n}$ then there exists an orthogonal $Q \in \mathbf{R}^{n \times n}$ such that

$$Q^T A Q = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{mm} \end{bmatrix} \equiv R, \tag{1.2.2}$$

where each $R_{ii}$ is a square block of order one or two. The blocks of order two contain the complex conjugate eigenvalues of $A$. The matrix $R$ is said to be in upper *quasi–triangular* matrix form.

**Proof** See [35, page 362]. □

Let $C$ be a quasi-diagonal orthonormal matrix with two by two blocks allowed only where $R$ has them. Then $(QC)^T AQC = C^T RC$ has diagonal blocks that are similar to those of $R$. Thus, apart from the eigenvalues of multiplicity larger than one, the decomposition is essentially unique given some ordering of the eigenvalues. Denote the leading principal matrix of $k$ blocks of $R$ by $R_k$ where no $R_{ii}$ is split. Let $Q_k \in \mathbf{R}^{n \times k}$ be the corresponding columns of $Q$. Then $AQ_k = Q_k R_k$ is a *partial* real Schur decomposition of $A$ of order $k$. The algorithms of this thesis attempt to compute a partial Schur decomposition for $A$ with a group of *wanted* eigenvalues located on the diagonal blocks of $R_k$. The $k \ll n$ eigenvalues of $A$ requiring approximation are typically contained within some convex set of interest in the complex plane. Examples include those nearest the origin, and of largest real part. An important exception might be the *dominant* eigenvalues of $A$, those largest in magnitude.

A *quasi-diagonal* form for $A$ exists if there is a nonsingular matrix $X \in \mathbf{R}^{n \times n}$ such that $AX = XD$ where $D$ is a block diagonal matrix with each block of order one or two. The blocks of order two contain the complex conjugate pair of eigenvalues as in equation (1.2.1) with $\mu$ positive. The columns of $X$ span the right eigenspace corresponding to diagonal values of $D$. For the blocks of order two on the diagonal of $D$ the corresponding complex eigenvector is stored in two consecutive columns of $X$, the first holding the real part, and the second the imaginary part. We also assume that the columns of $X$ are unit vectors. If we assume that $A$ is diagonalizable, the matrix $R$ may be further decomposed [1, 35, 86] as $RS = SD$ where $D = \mathrm{diag}(R_{11}, R_{22}, \ldots, R_{mm})$ and $S \in \mathbf{R}^{n\times n}$ is upper quasi-triangular and nonsingular. The matrix pair $(QS, D)$ represents a *quasi-diagonal* form for $A$.

### 1.2.2 Elementary Orthonormal Matrices

A real matrix $U \in \mathbf{R}^{n\times n}$ is *orthonormal* if $U^T U = I_n$. The matrix consisting of any of the columns of $U$ is called an *orthogonal* matrix. For example, define $U[e_1, \ldots, e_k] = U_k \in \mathbf{R}^{n\times k}$, and note that $U_k^T U_k = I_k$ but $U_k U_k^T \neq I_n$ unless $k = n$. Hence, $U_k$ is orthogonal for all values of $k$ but only orthonormal when $k = n$.

Givens rotations and Householder reflectors are two important classes of simple orthonormal matrices that will be used extensively in this thesis. We briefly introduce their fundamentals and refer the reader to the sources [47, 61, 101] for more comprehensive treatments including their numerically stable implementation.

A Householder reflector is a matrix of the form $W = I - \tau w w^T$ where $\tau = 2(w^T w)^{-1}$ if $w \neq 0$. Direct computation yields that $W$ is orthogonal and symmetric and hence $W^2 = I$. If we choose the vector $w = x \pm \|x\| e_1$ the Householder matrix $W$ is such that $Wx = \mp\|x\| e_1$ for $x \in \mathbf{R}^n$. Since $W$ is orthogonal and symmetric it follows that its first column (and row) contains $\pm x/\|x\|$. The geometrical interpretation of the transformation effected by $W$ is that it acts as a reflection in the subspace of dimension $n-1$ orthogonal to $w$.

A Givens rotation $G_{i,j} \in \mathbf{R}^{n\times n}$ acts as a rotation in the plane spanned by $e_i$ and $e_j$. The rotation differs from $I_n$ only in the $(i,i)$,$(j,j)$,$(i,j)$ and $(j,i)$ entries of $G_{i,j}$:

$$\begin{bmatrix} e_i^T G_{i,j} e_i & e_i^T G_{i,j} e_j \\ e_j^T G_{i,j} e_i & e_j^T G_{i,j} e_j \end{bmatrix} \equiv \begin{bmatrix} \sigma & \gamma \\ -\gamma & \sigma \end{bmatrix}.$$

An example that illustrates their use is to determine scalars $\gamma$ and $\sigma$ so that the first column of $G_{1,2}$ is equal to $\pm x/\|x\|$ for $x \in \mathbf{R}^2$. Equivalently, we solve $G_{1,2}^T \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \pm\|x\| e_1$ and a simple derivation shows that $\sigma = -\xi_1/\|x\|$ and $\gamma = \xi_2/\|x\|$ give the required result. Thus, the rotation acts as a matrix transformation that rotates $\mathbf{R}^2$ through a counterclockwise angle $\phi$ where $\tan\phi = -\xi_2/\xi_1$.

If $x \in \mathbf{R}^n$ then we may compute a sequence of Givens rotations so that

$$G_{1,2}^T G_{1,3}^T \cdots G_{1,n-1}^T x \quad = \quad \pm\|x\|e_1.$$

Note, that unlike the corresponding Householder reflector accomplishing the same task, the product $G_{1,n-1} \cdots G_{1,2}$ is not symmetric. However, since each Givens rotation is orthogonal, the product of them is also, which is the important property.

Returning to the previous example of constructing a Givens rotation so that $G_{1,2}^T x \ = \ \pm\|x\|e_1$ where $x \in \mathbf{R}^2$, allows us to determine a relationship with the Householder reflector accomplishing the same task. The relationship is

$$G_{1,2} = \begin{bmatrix} \sigma & \gamma \\ -\gamma & \sigma \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \sigma & \gamma \\ \gamma & -\sigma \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} W,$$

thus expressing a Givens rotation as a product of two reflectors.

### 1.2.3 The QR Factorization of Matrix

Given a matrix $B \in \mathbf{R}^{m \times n}$, it will prove useful to be able to factor $B$ into a product of an orthonormal and upper triangular matrix, respectively. Such a factorization allows an orthogonal represention of $B$'s column space.

> **Theorem 1.2** Suppose that $B \in \mathbf{R}^{m \times n}$ where $m$, the number of rows, is at least as large as $n$, the number of columns. If $l = \text{rank}(B)$ then there exist an unique orthogonal matrix $Q_1 \in \mathbf{R}^{m \times l}$ and an unique nonsingular upper triangular matrix $R_1 \in \mathbf{R}^{l \times l}$ with positive diagonal elements such that
>
> $$B = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 & R_{12} \\ 0 & 0 \end{bmatrix}, \tag{1.2.3}$$
>
> where $Q \in \mathbf{R}^{m \times m}$ is an orthonormal matrix.

**Proof** See Golub and Van Loan [35, pages 212, 214] for algorithmic derivations using either Givens rotations or Householder reflectors. □

It follows that $\mathcal{R}(B) = \mathcal{R}(Q_1)$ thus providing an orthogonal basis for the columns space of $B$. The unique factorization $B = Q_1 R_1$ results if $\text{rank}(B) = n$ and amounts to performing the Gram–Schmidt process to the columns of $B$. It is interesting to note that regardless of whether Givens rotations or Householder reflectors are used to compute the $QR$ factorization of $B$, both implementations result in the same $Q_1$ and $R_1$. However, the other orthogonal matrix, $Q_2$, and consequently $R_{12}$ are not uniquely defined. A word of caution: most algorithms computing the $QR$ factorization of a matrix are only unique up to a scaling of the columns of $Q_1$ and the corresponding rows of $R_1$ by a factor of $\pm 1$. The reason is that we may always compute a diagonal matrix $D \in \mathbf{R}^{n\times n}$ consisting of only $\pm 1$ and so that $B = QDD^{-1}R$ is another orthogonal factorization of $B$.

# Chapter 2

# The Arnoldi Method

Arnoldi's method [3] is an orthogonal projection method for approximating a subset of the eigensystem of a general square matrix. The method builds, step by step, an orthogonal basis for the *Krylov* subspace,

$$\mathcal{K}_m(A, v_1) \quad \equiv \quad \mathrm{Span}\{v_1, Av_1, \ldots, A^{m-1}v_1\},$$

for $A$ generated by the vector $v_1$. The original algorithm proposed was designed to compute the *Hessenberg decomposition*

$$U^T AU = H \qquad U^T U = I,$$

where $H$ is an upper Hessenberg matrix. As this chapter demonstrates, there is an intimate connection between Krylov subspaces and Hessenberg matrices.

The chapter is organized as follows. Some useful results concerning Hessenberg matrices are presented in § 2.1. The Arnoldi factorization is introduced in § 2.2. The Hessenberg decomposition of $A$ using other orthogonal reduction methods is reviewed in § 2.3. Truncated Arnoldi factorizations which lead to real partial Schur decompositions are treated in § 2.4. Determining how well an eigenvalue of the projected matrix $H_m$ approximates an eigenvalue of $A$ is considered in § 2.5. The convergence properties of Krylov subspaces are briefly reviewed in § 2.6.

## 2.1 Fundamentals of Hessenberg Matrices

Hessenberg matrices hold a fundamental role for the analysis presented in this thesis. This section reviews many of their most important properties.

We choose to label the $i$-th diagonal and sub-diagonal elements of $H \in \mathbf{R}^{n\times n}$, an Hessenberg matrix, as $\alpha_i$ and $\beta_{i+1}$, respectively:

$$\begin{bmatrix} \alpha_1 & \cdots & & \\ \beta_2 & \alpha_2 & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \beta_n & \alpha_n \end{bmatrix}.$$

A Hessenberg matrix is said to be *unreduced* if all of its sub-diagonal elements are nonzero. Both the left and right eigenvectors of unreduced upper Hessenberg matrices possess the following curious properties.

**Lemma 2.1** Suppose that $H \in \mathbf{R}^{n\times n}$ is an unreduced upper Hessenberg matrix. If $Hs = s\theta$ with $s \neq 0$ and $H^T u = u\theta$ with $u \neq 0$ then $e_n^T s \neq 0$ and $e_1^T u \neq 0$.

**Proof** The proof is by induction on the order of $H$. Suppose that $H_2$ is an unreduced matrix of order of order two. The last row results of the equation $H_2 s = s\theta$ is

$$e_2^T H_2 s = \beta_2\sigma_1 + \alpha_2\sigma_2 = \theta\sigma_2,$$

where $e_i^T s = \sigma_i$. If $\sigma_2 = 0$ then $\beta_2\sigma_1 = 0$. Since $H_2$ is unreduced, then $\sigma_1 = 0$ which is a contradiction since by definition eigenvectors are non zero.

Assume the lemma's truth for matrices of order $n-1$. Let $H_n \in \mathbf{R}^{n\times n}$ be an unreduced Hessenberg matrix and partition the equation $H_n s = s\theta$ as

$$\begin{bmatrix} H_{n-1} & h_n \\ \beta_n e_{n-1}^T & \alpha_n \end{bmatrix} \begin{bmatrix} s_{n-1} \\ \sigma_n \end{bmatrix} = \begin{bmatrix} s_{n-1} \\ \sigma_n \end{bmatrix} \theta,$$

where $H_{n-1} \in \mathbf{R}^{n-1\times n-1}$ and $s_{n-1} \in \mathbf{R}^{n-1}$. Note that $H_{n-1}$ is unreduced since $H_n$ is. Suppose $e_n^T s = \sigma_n = 0$ which implies that $\beta_n e_{n-1}^T s_{n-1} = 0$ and $H_{n-1}s_{n-1} = s_{n-1}\theta$. By the induction hypothesis $e_{n-1}^T s_{n-1} \neq 0$ and hence $\beta_n = 0$ which is a contradiction.

The proof for the result that $e_1^T u \neq 0$ where $H^T u = u\theta$ also follows from a similar proof by mathematical induction. □

Unreduced Hessenberg matrices have rank at least $n-1$ since the first $n-1$ columns are linearly independent. Thus the null space of $H - \mu I$ is of dimension one if $\mu$ is an eigenvalue of $H$ and zero otherwise. If the invariant subspace associated with an eigenvalue is of dimension greater than one, then the corresponding matrix is *derogatory* otherwise the matrix is *nonderogatory*. It follows then that a symmetric unreduced tridiagonal matrix cannot have a repeated eigenvalue since a repeated eigenvalue would imply that the eigenvectors of the symmetric matrix would not span $\mathbf{R}^n$. The previous discussion is summarized by the following result.

**Lemma 2.2** An unreduced Hessenberg matrix is nonderogatory. In particular, if $H$ is a symmetric matrix all its eigenvalues are distinct.

It follows that an unreduced nonsymmetric Hessenberg matrix is likely to have an ill conditioned basis of eigenvectors when it has nearly equal eigenvalues. When there is a repeated eigenvalue the lemma implies that $H \in \mathbf{R}^{n \times n}$ has less than $n$ linearly independent eigenvectors. If the eigenvectors of a matrix of order $n$ are not a basis for $\mathbf{R}^n$ then the matrix is called *defective*. Hence, if $H$ has a repeated eigenvalue it is a defective matrix.

Unreduced Hessenberg matrices reveal much information about the underlying eigen-system. Ericsson [29] and Parlett [59, 61] provide an abundance of results for Hessenberg matrices.

## 2.2 The Arnoldi Factorization

After $k$ steps, the Arnoldi method computes

$$AV_k = V_k H_k + f_k e_k^T, \tag{2.2.1}$$

where $V_k^T V_k = I_k$ and $H_k \in \mathbf{R}^{k \times k}$ is an upper Hessenberg matrix. The vector $f_k$ is the residual and is orthogonal to the columns of $V_k$, the *Arnoldi vectors*. The matrix $H_k = V_k^T A V_k$ is the orthogonal projection of $A$ onto the Range of $V_k$. Equation (2.2.1) defines a length $k$ Arnoldi factorization of $A$. If the residual $f_k$ is the zero vector then equation (2.2.1) is called a *truncated* Arnoldi factorization when $k < n$. Note that $f_n$ must vanish since $V_n^T f_n = 0$ and the columns of $V_n$ form an orthogonal basis for $\mathbf{R}^n$. In this case the Arnoldi method computes an Hessenberg decomposition.

The following classical result explains that the Arnoldi factorization is completely specified by $v_1$.

**Theorem 2.1** (Implicit Q) Let two length $k$ Arnoldi factorizations be given by

$$\begin{aligned} AV_k &= V_k H_k + f_k e_k^T, \\ AU_k &= U_k G_k + r_k e_k^T, \end{aligned}$$

where $U_k$ and $V_k$ have orthonormal columns, and $G_k$ and $H_k$ are upper Hessenberg matrices with positive sub-diagonal elements. If the first column of $V_k$ and $U_k$ are equal then $G_k = H_k$, $U_k = V_k$, and $r_k = f_k$.

**Proof** See Golub and Van Loan [35, page 367]. □

The essential hypothesis is that $H_k$ is unreduced. We note that if $H_k$ has any negative sub-diagonal elements, a diagonal matrix $D_k$ consisting of $\pm 1$ is easily computed so that $D_k^{-1} H_k D_k$ has positive sub-diagonal elements. Equation (2.2.1) may then be updated to obtain another Arnoldi factorization

$$A V_k D_k \;=\; V_k D_k (D_k^{-1} H_k D_k) + \delta_k f_k e_k^T,$$

where $\delta_k = e_k^T D_k e_k$. The direction of $v_1$ is the important consideration.

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

**Algorithm 2.2**

function $[V_{k+p}, H_{k+p}, f_{k+p}] =$ `Arnoldi` $(A, V_k, H_k, f_k, k, p)$

Input: $AV_k - V_k H_k = f_k e_k^T$ with $V_k^T V_k = I_k$, $V_k^T f_k = 0$.

Output: $AV_{k+p} - V_{k+p} H_{k+p} = f_{k+p} e_{k+p}^T$ with $V_{k+p}^T V_{k+p} = I_{k+p}$, and $V_{k+p}^T f_{k+p} = 0$.

1. For $j = 1, 2 \ldots p$
   2. $\beta_{k+j} \leftarrow \|f_{k+j-1}\|$; if $\beta_{k+j} = 0$ then stop;
   3. $v_{k+j} \leftarrow f_{k+j-1} \beta_{k+j}^{-1}$; $V_{k+j} \leftarrow [V_{k+j-1}, v_{k+j}]$;
   4. $w \leftarrow A v_{k+j}$;
   5. $h_{k+j} \leftarrow V_{k+j-1}^T w$; $\alpha_{k+j} \leftarrow v_{k+j}^T w$ ;
   6. $$H_{k+j} \;\leftarrow\; \begin{bmatrix} H_{k+j-1} & h_{k+j} \\ \beta_{k+j} e_{k+j-1}^T & \alpha_{k+j} \end{bmatrix}$$
   7. $f_{k+j} \leftarrow w - V_{k+j-1} h_{k+j} - v_{k+j} \alpha_{k+j}$;

A few remarks are in order.

1. If $A$ is symmetric, then $H_k$ is a symmetric tridiagonal matrix so that $h_{k+j}^T = \beta_{k+j} e_{k+j-1}^T$ and hence a three term recurrence may be used to compute $f_{k+j}$.

2. If $k = 0$, then $V_1 = v_1$ represents the initial vector.

3. In order to ensure that the $k$-th residual is numerically orthogonal to the matrix of Arnoldi vectors $V_k$ in finite precision arithmetic, procedure `Arnoldi` requires some form of re-orthogonalization at Line 7. This is the subject of Chapter 7. Mathematically, the residual $f_{k+j}$ computed at Line 7 represents the projection of $w = Av_{k+j}$ onto the orthogonal compliment of $\mathcal{K}_{k+j}(A, v_1)$.

4. In exact arithmetic, the algorithm halts only if a residual vector vanishes, i.e. $f_j = 0$. The implications of a truncated Arnoldi factorization are discussed in § 2.4.

5. If $f_j = 0$ for $j < n$, then the factorization may be modified to extend the truncated factorization by using any unit vector orthogonal to the columns of $V_j$. The unit vector becomes the $j+1$-st Arnoldi vector and the $j$-th sub-diagonal element, $\beta_{j+1}$, is zero. Although $V_n^T A V_n = H_n$ is upper Hessenberg, it is not unreduced.

## 2.3 Orthogonal Reductions to Hessenberg Form

If Algorithm 2.2 is used to compute a length $n$ Arnoldi factorization, then the resulting factorization is also an Hessenberg decomposition of $A$. Theorem 2.1 indicates when the decomposition is unique. Other orthogonal methods for computing Hessenberg decompositions are based upon Givens rotations or Householder reflectors [35, 86].

The Householder reduction computes a sequence of Householder reflectors $W_j$ designed to introduce zeros in last $n-j-2$ elements of column $j$ of $W_{j-1}^T \cdots W_1^T A$. The product $U_n = W_1 \cdots W_{n-2}$ results in an orthogonal matrix so that $U_n^T A U_n$ is upper Hessenberg. The first column of $U_n$ is $e_1$ so that by Theorem 2.1, the Hessenberg decomposition computed by Algorithm 2.2 with $v_1 = e_1$ is equivalent to that computed by the Householder reduction. Given an arbitrary unit vector $v_1$, a Householder reduction to upper Hessenberg form is an orthogonal matrix away from being equivalent to an Arnoldi factorization as the following result shows.

**Lemma 2.3** Suppose an Hessenberg decomposition $AV_n = V_n H_n$ is computed by Algorithm 2.2 for $A \in \mathbf{R}^{n \times n}$. If $W_0$ is an orthogonal matrix such that $W_0 e_1 = V_n e_1$, then the orthogonal decomposition $(W_0^T A W_0) U_n = U_n G_n$ is such that $W_0 U_n = V_n$ and $G_n = H_n$.

**Proof**

Let $W_0$ be an orthogonal matrix so that $W_0 e_1 = V_n e_1$. Let $W_0^T A W_0 U_n = U_n G_n$ be a Hessenberg decomposition. Since $W_0 U_n e_1 = W_0 e_1 = V_n e_1$, Theorem 2.1 gives the necessary equalities. □

This simple observation allows us to establish a direct link between all orthogonal reductions, or factorizations, to upper Hessenberg form. As will be seen in Chapter 5, the various methods for computing the decomposition may produce drastically different results when computing in finite precision arithmetic.

## 2.4 Truncated Arnoldi Factorizations

The following section is concerned with finding conditions for the Arnoldi method terminating prematurely. This is a welcome event since if $AV_m = V_m H_m$ is a truncated Arnoldi factorization of length $m$, the eigenvalues of $H_m$ are a subset of those of $A$. Indeed, if $H_m Z_m = Z_m T_m$ is a real Schur decomposition, then $A(V_m Z_m) = (V_m Z_m) T_m$ is a partial one for $A$. A few results are needed before a theorem stating necessary and sufficient conditions for a truncated factorization is presented.

The first result needed is a slight modification of Theorem 7.4.3 proved in Golub and Van Loan [35]. It allows us to establish a connection between the *Krylov matrix*

$$K_m(A, v_1) \;=\; \begin{bmatrix} v_1 & Av_1 & \cdots & A^{m-1}v_1 \end{bmatrix}$$

and an Arnoldi factorization.

**Theorem 2.3** Suppose $Q \in \mathbf{R}^{n\times n}$ is orthogonal and $A \in \mathbf{R}^{n\times n}$ such that $AQ = QH$ is an upper Hessenberg decomposition. Partition $Q = [Q_m, \bar{Q}_{n-m}]$ where $Q_m \in \mathbf{R}^{n\times m}$ and set $H_m = Q_m^T A Q_m$.

Then $H_m$ is an unreduced Hessenberg matrix if and only if

$$Q_m^T K_m(A, v_1) \;\equiv\; R_m \in \mathbf{R}^{m\times m},$$

is nonsingular and upper triangular, for $m = 1, \ldots, n$.

**Proof** Let $AQ = QH$ be an Arnoldi factorization of length $n$. Partition $Q = [Q_m, \bar{Q}_{n-m}]$ where $Q_m \in \mathbf{R}^{n\times m}$ and set $H_m = Q_m^T A Q_m$. Note that $Q^T A^j v_1 = Q^T A Q \cdots Q^T A Q e_1 = H^j e_1$ for $j = 0, \ldots, n-1$, and then

$$\begin{aligned} Q^T K_n(A, v_1) &= \begin{bmatrix} Q^T v_1 & Q^T A v_1 & \ldots & Q^T A^{n-1} v_1 \end{bmatrix}, \\ &= \begin{bmatrix} e_1 & He_1 & \ldots & H^{n-1} e_1 \end{bmatrix}, \qquad (2.4.1) \\ &\equiv R, \end{aligned}$$

is an upper triangular of matrix order $n$. Thus $Q_m^T K_m(A, v_1) = R_m$ is the leading principal sub-matrix of $R$ of order $m$.

Suppose that $H_m$ is an unreduced Hessenberg matrix. The diagonal elements of $R_m$ are $e_i^T R_m e_i = \beta_1 \beta_2 \cdots \beta_i$ for $i = 1, \ldots, m$ with $\beta_1 \equiv 1$. The non-singularity of $R_m$ now follows.

For the converse, suppose that $R_m \in \mathbf{R}^{m \times m}$ is nonsingular and upper triangular. Since $H^j e_1 \in \mathrm{Span}\{e_1, \ldots, e_{j+1}\}$ is a linear combination of the first $j$ columns of $H$ it follows from equation (2.4.1) that $R_m e_{j+1} = H_m R_m e_j$ for $j = 1, \ldots, m-1$. Since $R_m$ is nonsingular and upper triangular, all its diagonal elements are nonzero. To show that $H_m$ is an unreduced upper Hessenberg matrix, consider $e_{j+1}^T H_m e_{j+1}$ for $j = 1, \ldots, m-1$. Since $R_m e_{j+1} = H_m R_m e_j$ it follows that

$$e_{j+1}^T H_m R_m e_j = \sum_{i=1}^{m} (e_{j+1}^T H_m e_i)(e_i^T R_m e_j) = (e_{j+1}^T H_m e_j)(e_j^T R_m e_j)$$

because $e_i^T R_m e_j = 0$ for $i > j$ and $e_{j+1}^T H_m e_i = 0$ for $i < j$. Thus $e_{j+1}^T H_m e_j = e_{j+1}^T R_m e_{j+1} / e_j^T R_m e_j \neq 0$ for $j = 1, \ldots, m-1$ since by assumption the diagonal elements of $R_m$ are nonzero. □

Theorem 2.3 implies that the residual $f_{m+1}$ vanishes at the first step $m$ such that the dimension of $\mathcal{K}_{m+1}(A, v_1)$ is equal to $m$ and hence is guaranteed to vanish for some $m \leq n$.

The monic polynomial $\psi(\lambda)$ of smallest degree such that $\psi(A)v_1 = 0$ is called the *minimal polynomial* of $A$ with respect to $v_1$. The degree of the minimal polynomial of $A$ with respect to $v_1$ is called the *grade* of $v_1$. Suppose that the grade of $v_1$ is $m$. Define $C_m = [e_2, \ldots, e_m, c_m] \in \mathbf{R}^{m \times m}$ where $c_m$ is the solution of the linear system $K_m(A, v_1)c_m = -A^m v_1$. We note that such a solution exist since $\psi_m(\lambda) = \lambda^m + \lambda^{m-1} e_m^T c_m + \cdots + e_1^T c_m$ is the minimal polynomial of $A$ with respect to $v_1$. It follows that

$$AK_m(A, v_1) = K_m(A, v_1) C_m. \tag{2.4.2}$$

The matrix $C_m$ is called a *companion matrix*. If we assume that the diagonal elements of $R_m$ are non-zero, Theorem 2.3 implies that $H_m$ is unreduced. From equation (2.4.2) the identity $AQ_m = Q_m(R_m C_m R_m^{-1})$ follows. By the Implicit Q theorem, $Q_m = V_m$ and $H_m = R_m C_m R_m^{-1}$ since the first columns of $Q_m$ and $V_m$ are equal. From equation (2.4.2) it follows that the characteristic polynomial for $C_m$ is equal to the minimal polynomial of $A$ with respect to $v_1$.

Ruhe [69] shows that $AK_j(A, v_1) = K_j(A, v_1)C_j$ where $C_j = [e_2, \ldots, e_j, c_j] \in \mathbf{R}^{j\times j}$ for $j \leq m$. The vector $c_j \in \mathbf{R}^j$ solves the least squares problem

$$\min_{c\in\mathbf{R}^j} \|A^j v_1 - K_j(A, v_1)c\| \quad = \quad \|A^j v_1 - K_j(A, v_1)c_j\|. \tag{2.4.3}$$

Denote the residual of the least squares problem by $r_j$ and note that $r_j = \psi_j(A)v_1$ where

$$\psi_j(\lambda) = \lambda^j - [1, \ldots, \lambda^{j-1}]^T c_j = \det(C_j - \lambda I_j).$$

It follows that

$$AK_j(A, v_1) \quad = \quad K_j(A, v_1)C_j + r_j e_j^T.$$

If $AV_j = V_j H_j + f_j e_j^T$ is an Arnoldi factorization of length $j$ with $V_j e_1 = v_1$ then Theorem 2.3 implies that $f_j e_j^T = r_j e_j^T$. Hence

$$f_j = (e_j^T R_j e_j)^{-1} r_j = (e_j^T R_j e_j)^{-1} \psi_j(A)v_1. \tag{2.4.4}$$

Saad [75] uses projection arguments to show that $\psi_j(\lambda)$ minimizes $\|\hat{\psi}_j(A)v_1\|$ over all monic polynomials $\hat{\psi}_j$ of degree $j$. This property is also a direct consequence of equation (2.4.3).

The following theorem summarizes the preceding discussion on the various relationships between an Arnoldi factorization and Krylov matrices. We remark that the previous discussion is in the spirit of that presented by Sorensen [83, pages 360–362].

**Theorem 2.4** Suppose the integer $m$ is the grade of the unit vector $v_1$ with respect to $A$. Let a sequence of Arnoldi factorizations be given by $AV_j = V_j H_j + f_j e_j^T$ for $j \leq m$ where $V_j e_1 = v_1$. If $K_j(A, v_1) = Q_j R_j$ where $R_j$ is upper triangular then $\psi_j(\lambda) = \det(C_j - \lambda I_j)$ solves

$$\min \|\hat{\psi}_j(A)v_1\|$$

over all monic polynomial of degree $j$. Moreover, $C_j = [e_2, \ldots, e_j, c_j] \in \mathbf{R}^{j\times j}$ is the companion matrix for $H_j$ where $AK_j(A, v_1) = K_j(A, v_1)C_j + r_j e_j^T$ with

$$\beta_2 \cdots \beta_j f_j = \psi_j(A)v_1 = r_j,$$

and if the sub-diagonal elements of $R_j$ are positive, then $H_j R_j = R_j C_j$, $V_j = Q_j$, and $e_j^T R_j e_j = \beta_2 \cdots \beta_j$.

We now state the main result of the section indicating when an exact truncated factorization occurs. This is desirable since the columns of $V_k$ form a basis for an invariant subspace and the eigenvalues of $H_k$ are a subset of those of $A$.

> **Theorem 2.5** Let equation (2.2.1) define a $k$-step Arnoldi factorization of $A$, with $H_k$ unreduced. Then $f_k = 0$ if and only if $v_1 = Q_k y$ where $AQ_k = Q_k R_k$ with $Q_k^T Q_k = I_k$, and $R_k$ an upper quasi-triangular matrix of order $k$.

**Proof** If $f_k = 0$ then $AV_k = V_k H_k$. Let $H_k Z_k = Z_k R_k$ be a real Schur decomposition where $Z_k^T Z_k = I_k$ and $R_k \in \mathbf{R}^{k\times k}$ is an upper quasi-triangular matrix. Then

$$v_1 = V_k e_1 = V_k Z_k Z_k^T e_1 \equiv Q_k y,$$

where $y = Z_k^T e_1$ and $V_k Z_k = Q_k$. Note that $AQ_k = Q_k R_k$.

Conversely, suppose that $AQ_k = Q_k R_k$ with $Q_k^T Q_k = I_k$ and $R_k$ is an upper quasi-triangular matrix of order $k$. Let $v_1 = Q_k y$ with $y \in \mathbf{R}^k$ arbitrary. Now, for any integer $m > 0$, $A^m Q_k = Q_k R_k^m$ and thus

$$A^m v_1 = A^m Q_k y = Q_k R_k^m y \in \mathcal{R}(Q_k).$$

Hence the dimension of the Krylov subspace $\mathcal{K}_m(A, v_1)$ is at most $k$. Since $H_k$ is unreduced, Theorem 2.3 implies that the dimension of $\mathcal{K}_{k+1}(A, v_1)$ is $k$ and hence $f_k = 0$. $\square$

The theorem's hypothesis indicates that the range of $Q_k$ represents an invariant subspace for $A$. The diagonal blocks of $R_k$ contain the eigenvalues of $A$. The complex conjugate pairs are in blocks of order two and the real eigenvalues are on the diagonal of $R_k$, respectively. The matrix equation $AQ_k = Q_k R_k$ is a partial real Schur decomposition of order $k$ for $A$. In particular, if the initial vector is a linear combination of $k$ linearly independent eigenvectors then the $k$-th residual vector vanishes. It is therefore desirable to to devise a method that forces the starting vector $v_1$ to be a linear combination of Schur vectors corresponding to wanted eigenvalues.

Theorem 2.5 gives conditions for the Arnoldi factorization to prematurely terminate. Computing in finite precision arithmetic blurs the exact conditions of the theorem. The Implicit Q theorem and the results of § 2.3 show that all orthogonal reductions to upper Hessenberg form are related. Thus the optimality property of Saad, and Ruhe's characterization of the Arnoldi factorization are fundamental results concerning the reduction of a matrix to upper Hessenberg form. Ruhe's analysis forms the basis of a perturbation theory for the Hessenberg reduction that is presented in Chapter 5. In particular, the theory developed determines the sensitivity, or degree of forward instability, of an Arnoldi or QR iteration upon the starting vector.

## 2.5 Stopping Criteria

This section considers the important question of determining when a length $k$ Arnoldi factorization has computed approximate eigenvalues. If the norm of $f_k$ is small, the $k$ eigenvalues of $H_k$ are approximations to $k$ eigenvalues of $A$. Numerical experience indicates that $\|f_k\|$ rarely becomes small let alone zero. Nevertheless, some of the eigenvalues of $H_k$ may be good estimates of the eigenvalues of $A$. Since the interest is in a small subset of the eigensystem of $A$, alternate criteria that allow termination for $k \ll n$ are needed. Let $H_k s = s\theta$ where $\|s\| = 1$. Define the vector $x_r = V_k s$ to be a *Ritz vector* and the scalar $\theta$ to be *Ritz value*. Then

$$\|AV_k s - V_k H_k s\| = \|Ax_r - x_r\theta\| = \|f_k\|\,|e_k^T s|, \tag{2.5.1}$$

indicates that if the last component of an eigenvector for $H_k$ is small the Ritz pair $(x_r, \theta)$ is an approximation to an eigenpair of $A$. We note that by Lemma 2.1, $|e_k^T s| > 0$ if $H_k$ is unreduced. This pair is exact for a nearby problem: it is easily shown that $(A + E)x_r = x_r\theta$ with $E = -(e_k^T s) f_k x_r^H$. The advantage of using the *Ritz estimate* $\|f_k\|\,|e_k^T s|$ is to avoid explicit formation of the direct residual $AV_k s - V_k s\theta$ when accessing the numerical accuracy of an approximate eigenpair. We remark that a small $\|E\|$ does not imply that the Ritz pair $(x_r, \theta)$ is an accurate approximation to an eigenpair $(x, \lambda)$ of $A$. The perturbation theory presented in § 5.2 of Chapter 5 considers these accuracy issues.

Recent work by Chatelin and Fraysée [18, 19] and Godet–Thobie [34] suggests that when $A$ is highly non-normal, the size of $e_k^T s$ is not an appropriate guide for detecting convergence. If the relative *departure from normality* defined by the Henrici number $\|AA^T - A^TA\|_F / \|A^2\|_F$, is large, the matrix $A$ is considered highly non-normal. Assuming that $A$ is diagonalizable, a large Henrici number implies that the basis of eigenvectors is ill-conditioned [18]. Bennani and Braconnier compare the use of the Ritz estimate and direct residual $\|Ax_r - x_r\theta\|$ in Arnoldi algorithms [12]. They suggest normalizing the Ritz estimate by the norm of $A$ resulting in a stopping criteria based on the *backward* error. The backward error is defined as the smallest, in norm, perturbation $\Delta A$ such that the Ritz pair is an eigenpair for $A + \Delta A$. Scott [80] presents a lucid account of the many issues involved in determining stopping criteria for the unsymmetric problem.

## 2.6 Convergence Properties of Krylov Spaces

In this section, we consider the rate at which the eigenvalues of $H_m$ emerge as approximations to those of $A$ as $m$ increases towards $n$. Since $H_m$ is the projection of $A$ with respect to the columns of $V_m$, Saad [74] proposes studying the convergence of the two residuals $(A-\theta I)x_r$ or $(V_m H_m V_m^T - \lambda I_n)x$, for some eigenpair $(x,\lambda)$ of $A$, to zero. Indeed, the former residual is that used in equation (2.5.1) of the previous section. Saad [78] uses the latter residual to obtain the inequality

$$\|(H_m - \lambda I_m)(V_m^T x)\| \;\; \leq \;\; \gamma_m \frac{\|(I - V_m V_m^T)x\|}{\|V_m^T x\|}, \tag{2.6.1}$$

where

$$\gamma_m \equiv \|V_m V_m^T A (I - V_m V_m^T)\| \leq \|A\|.$$

The quality of the approximation afforded by $V_m$ and $H_m$ is governed by the tangent of the angle between $\mathcal{K}_m(A, V_m e_1)$ and $x$, which is given by the ratio on the right hand side of equation (2.6.1). Thus, the size of the numerator $\|(I - V_m V_m^T)x\|$, the sine of the angle between $\mathcal{K}_m(A, V_m e_1)$ and $x$, is the quantity to estimate. The following theorem which we state without proof is due to Saad [75].

> **Theorem 2.6** Assume that $A$ is diagonalizable with eigenpairs $(x_j, \lambda_j)$ where each eigenvector is of unit length. If $V_m e_1 = x_1\zeta_1 + \cdots + x_n\zeta_n$ with $\zeta_1 \neq 0$, then there exist $m$ eigenvalues $\lambda_2, \ldots, \lambda_{m+1}$ of $A$ such that
>
> $$\|(I - V_m V_m^T)x_1\| \;\; \leq \;\; \sum_{j=2}^{n} \frac{|\zeta_j|}{|\zeta_1|}\varepsilon_1^m \tag{2.6.2}$$
>
> where
>
> $$\frac{1}{\varepsilon_1^m} \;\; \equiv \;\; \sum_{j=2}^{m+1} \prod_{l=2, l\neq j}^{m+1} \left| \frac{\lambda_l - \lambda_1}{\lambda_l - \lambda_j} \right|.$$

If $|\lambda_1 - \lambda_l| > |\lambda_j - \lambda_l|$ for $j,l = 2,\ldots,m+1$ then $\varepsilon_1^m < 1$. The geometrical interpretation is that $A$'s extremal eigenvalues that are well separated emerge as eigenvalues of $H_m$. This generalizes the well known convergence behavior of the Lanczos iteration [61, 73].

The constant multiplying $\varepsilon_1^m$ consisting of the normalized sum of expansion coefficients on the righthand side of equation (2.6.2) reflects the possible ill-conditioning of the matrix of eigenvectors for $A$. This may be seen as follows. Suppose the left eigenvector [35, 101] corresponding to the right eigenvector $x_j$ is denoted by $y_j^H$, for $j = 1, \ldots, n$ where $\lambda_j$ is distinct from the other eigenvalues. Assume the left eigenvector is also of unit length. Using the Cauchy–Schwartz inequality, it follows that $|\zeta_i||y_i^H x_i| = |y_i^H v_1| \leq \|y_i\|\|v_1\| = 1$, giving $|\zeta_i| \leq \sec\varphi_i$ where $\varphi_i$ measures the angle between the corresponding left and right eigenvector. If the eigenvalue $\lambda_i$ is poorly conditioned, then $\sec\varphi_i$ is large and possibly so is the coefficient $|\zeta_i|$. If we assume that the eigenvalues of $A$ are distinct, then

$$\sum_{j=2}^{n} \frac{|\zeta_j|}{|\zeta_1|} \;\; \leq \;\; \sum_{j=2}^{n} \frac{|\sec\varphi_j|}{|\zeta_1|},$$

may be quite large. The conclusion is that a large factorization may need to be built for poorly conditioned eigenvalue problems in order for good estimates of $A$'s eigenvalues to emerge in $H_m$. In addition, if $A$ is defective, it may not possess a basis of eigenvectors. Numerically, problems are encountered when a basis for the desired invariant subspace is poorly conditioned. The recent thesis of Jia [43] extends Saad's results without the assumption that $A$ is diagonalizable.

Finally, we end with a theorem that combines $\gamma_m$ and $\beta_{m+1}$ to estimate how close $\mathcal{K}_m(A, V_m e_1)$ is to an invariant subspace of $A$. But first we provide a brief motivation for the theorem.

Suppose that $Z = \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}$ is an orthonormal matrix where the columns of $Z_1 \in \mathbf{R}^{n\times m}$ spans an invariant subspace for $A$. Partition $Z^T A Z = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ where $A_{ij} = Z_i^T A Z_j$ for $i, j = 1, 2$. Since $\mathcal{R}(Z_1)$ is invariant under $A$, there exist a matrix $G_1 \in \mathbf{R}^{m\times m}$ so that $AZ_1 = Z_1 G_1$. Thus, $A_{21} = Z_2^T A Z_1 = Z_2^T Z_1 G_1 = 0$ since $Z$ is orthonormal.

Stewart [85] considers the interesting question of how close $Z_1$ is to an invariant subspace of $A$ is if $\|A_{21}\|$ is small instead of zero. For example, if $\tilde{Z}$ and $\tilde{Z}^T A \tilde{Z}$ are partitioned conformably with $Z$ and $Z^T A Z$, respectively, can an orthonormal matrix

$Y$ deviating little from $I_n$ be found so that $Z = \tilde{Z}Y$ ? Stewart chooses

$$Y = \begin{bmatrix} I_m & -P^T \\ P & I_{n-m} \end{bmatrix} \begin{bmatrix} (I_m + P^T P)^{-1/2} & 0 \\ 0 & (I_{n-m} + PP^T)^{-1/2} \end{bmatrix},$$

where $P \in \mathbf{R}^{(n-m)\times m}$ and since both $I_m + P^T P$ and $I_{n-m} + PP^T$ are positive definite and symmetric matrices, the square roots are uniquely defined. The answer to whether the column space of $\tilde{Z}_1$ is an accurate approximation to that of $Z_1$ becomes that of analyzing the interaction among the matrices $P$ and $A_{ij}$ for $i, j = 1, 2$. The analysis presented by Stewart gives the following interesting interpretation with respect to an Arnoldi factorization.

**Theorem 2.7** Suppose that $AV_m = V_m H_m + f_m e_m^T$ is a length $m$ Arnoldi factorization that is extended to a Hessenberg decomposition of $A$:

$$A \begin{bmatrix} V_m & \bar{V}_{n-m} \end{bmatrix} = \begin{bmatrix} V_m & \bar{V}_{n-m} \end{bmatrix} \begin{bmatrix} H_m & M_m \\ \beta_{m+1} e_1 e_m^T & \bar{H}_{n-m} \end{bmatrix},$$

where $\beta_{m+1} = \|f_m\|$. Let

$$\delta_m = \operatorname{sep}(H_m, \bar{H}_{n-m}) \equiv \min_{X \neq 0} \frac{\|XH_m - \bar{H}_{n-m}X\|_F}{\|X\|_F}.$$

If $4\beta_{m+1}\gamma_m < \delta_m^2$, then there is a matrix $P$ that satisfies the bound

$$\|P\| \leq 2\frac{\beta_{m+1}}{\delta_m}$$

so that the columns of $Q_m = (V_m + \bar{V}_{n-m}P)(I + P^T P)^{-1/2}$ are an orthogonal basis for an invariant subspace of $A$.

**Proof** A simple derivation shows that $\gamma_m = \|V_m V_m^T A(I - V_m V_m^T)\| = \|M_m\|$. The conclusion now follows directly from Theorem 4.1 of Stewart [85]. $\Box$

The size of $\gamma_m$ measures the amount of coupling between the $\mathcal{R}(V_m)$ and $\mathcal{R}(\bar{V}_{n-m})$. The reciprocal of $\delta_m$ measures the sensitivity of the $\mathcal{R}(Q_m)$ as an invariant subspace. It may be shown that

$$\operatorname{sep}(H_m, \bar{H}_{n-m}) \leq \min_{k,l} |\lambda_k(H_m) - \lambda_l(\bar{H}_{n-m})|.$$

Moreover, Varah [94] shows that if the matrices involved are highly non-normal, the smallest difference between the spectrums of $H_m$ and $\bar{H}_{n-m}$ may be an over estimate of the actual separation.

Theorem 2.7 shows the dependence of $\beta_{m+1}$ upon $\gamma_m$ and $\delta_m$ in determining the quality of the $\mathcal{R}(V_m)$ as an eigenspace of $A$. Since $V_m^T Q_m = (I + P^T P)^{-1/2}$, Stewart [85] shows that the singular values of $P$ are the tangents of the canonical, or principal, angles [18, 35, 85] between the two spaces spanned by the columns of $V_m$ and $Q_m$, respectively.

Unfortunately, both Theorems 2.6 and 2.7 require information about $A$ that is not readily available. In addition, Theorem 2.7 requires that the sub-diagonal element $\beta_{m+1}$ of $H$ be small relative to $\delta_m$ and $\gamma_m$. The next two chapters give conditions under which we can expect this situation to occur.

# Chapter 3

# The QR Algorithm

The QR algorithm is a general purpose method for computing all the eigenvalues of dense matrices. The LR-iteration of Rutishauser [72], based on a triangular sequence of similarity transformation, preceded its discovery. The QR algorithm, developed independently by both Francis [32] and Kublanovskaya [45], instead uses a sequence of orthogonal similarity transformation The algorithm iteratively computes an approximation to the real Schur decomposition.

The chapter first examines the *explicitly shifted iteration* and some of its fundamental properties in § 3.1. The convergence of the iteration is considered in § 3.2. The well known duality of the QR iteration and inverse iteration is interpreted in terms of Krylov subspaces in § 3.3. The practical QR algorithm is the subject of § 3.4 which includes a discussion of the *implicitly* shifted version. There is wealth of excellent material on the QR algorithm. Thorough introductions are given by Golub and Van Loan [35], Parlett [61], Stewart [86] and Watkins [97, 98]. More advanced treatments include those by Parlett and Poole [65], Watkins and Elsner [100], and Wilkinson [101].

For the remainder of the chapter we assume that $A$ is factored into $AU = UH$ where $H$ is an unreduced upper Hessenberg matrix and $U^TU = I_n$. There is no loss of generality since if $H$ is reduced then for some $1 \le j < n$,

$$H = \begin{bmatrix} H_j & M_j \\ 0 & \bar{H}_{n-j} \end{bmatrix},$$

where $H_j$ is an unreduced Hessenberg matrix. The eigenvalues of $H$ are the eigenvalues of $H_j$ and $\bar{H}_{n-j}$ so that we may work with $H_j$ and then in turn $\bar{H}_{n-j}$. We remark that if Schur vectors or eigenvectors are desired for any of the eigenvalues of $\bar{H}_{n-j}$, the sub-matrix $M_j$ is required.

## 3.1 Explicitly Shifted QR Iteration

The explicitly shifted QR-iteration is defined by

**Algorithm 3.1**
Input: $H^{(1)} = H$ an unreduced upper Hessenberg matrix, and a sequence of shifts $\{\tau_j\}_{j=1}^{p}$.
Output: $H^{(p+1)}$ and $Z^{(p)} \leftarrow Q^{(1)} \cdots Q^{(p)}$.

1.1 For $j = 1, \ldots, p$

  2.1 Compute the $QR$ factorization :
  $Q^{(j)} R^{(j)} = H^{(j)} - \tau_j I$ ;

  2.2 $H^{(j+1)} \leftarrow R^{(j)} Q^{(j)} + \tau_j I$

One cycle of the iteration is said to be a QR step. Some of the most important properties in a QR step are summarized with the following lemma.

**Lemma 3.1** Let $H - \tau I = QR$ be a QR factorization where $H$ is an unreduced upper Hessenberg matrix and denote $e_i^T R e_i = \rho_i$. Then the following properties hold:

1. $Q$ is an upper Hessenberg matrix.
2. $\rho_i \neq 0$ for $i = 1, \ldots, n-1$
3. $\rho_n = 0$ if and only if $\tau$ is an eigenvalue of $H$.
4. $e_n^T (RQ + \tau I) = \tau e_n^T$ if and only if $\tau$ is an eigenvalue of $H$.

**Proof** A sequence of plane rotations $G_{i,i+1}$ are easily constructed so that

$$G_{n-1,n}^H \cdots G_{1,2}^H (H - \tau I)$$

is upper triangular [35, page 215]. Each $G_{i,i+1}$ is designed to annihilate the entry in the $(i+1, i)$ entry of $G_{i-1,i}^H \cdots G_{1,2}^H (H - \tau I))$. The product $G_{1,2} \cdots G_{n-1,n}$ is upper Hessenberg and $G_{n-1,n}^H \cdots G_{1,2}^H (H - \tau I)$ is upper triangular. Set $Q = G_{1,2} \cdots G_{n-1,n}$ and $R = Q^H (H - \tau I)$. Note that $Q$ is an upper Hessenberg matrix.

A simple derivation shows that $e_{i+1}^T H e_i = e_{i+1}^T Q e_i \rho_i$. Since $H$ is an unreduced upper Hessenberg matrix, $0 < |e_{i+1}^T H e_i| = |e_{i+1}^T Q e_i| |\rho_i| \leq |\rho_i|$ for $i = 1, \ldots, n-1$ establishing the second property.

The matrix $H - \tau I$ is singular if and only if $\tau$ is an eigenvalue of $H$. The third property follows immediately since $\det(H - \tau I) = \det(R) = \rho_1 \cdots \rho_n$ is zero if and only if $\rho_n$ is.

The third property gives $\rho_n = 0$ if $\tau$ is an eigenvalue of $H$. Since $e_n^T R = e_n^T \rho_n$ the final property holds. $\square$

The lemma allows us to conclude that all $H^{(j)}$ remain upper Hessenberg. The only sub-diagonal of $H^{(j)}$ that ever becomes zero is the last one and this is purely a function of the shift. If a shift is equal to an eigenvalue, then we no longer have an unreduced Hessenberg matrix and instead we only continue working with the leading sub-matrix of $H^{(j)}$ that remains unreduced. It should be emphasized that the conclusions of Lemma 3.1 hold in exact arithmetic. An elegant extension of Lemma 3.1 to the case where $p$ shifts are applied is proved by Miminis and Paige [53]. However, we show in Chapter 5 that computing in finite precision may have dramatic effects that degrade the expected performance of multiple shifts.

The following properties are consequences of the iteration. The first two are easily established using mathematical induction; see for example [86, 97]. The third is a standard result that does not depend upon the condition that $H$ is an Hessenberg matrix.

**Lemma 3.2** Let $Z^{(p)} = Q^{(1)} \cdots Q^{(p)}$. Then $HZ^{(p)} = Z^{(p)} H^{(p+1)}$.

**Proof** The result follows by a simple induction argument since it easily follows that $H^{(2)} = R^{(1)}Q^{(1)} + \tau_1 I = (Q^{(1)})^H H^{(1)} Q^{(1)} - \tau_1 I + \tau_1 I = (Q^{(1)})^H H Q^{(1)}$. $\square$

**Theorem 3.2** Let $Z^{(p)} = Q^{(1)} \cdots Q^{(p)}$ and $T^{(p)} = R^{(p)} \cdots R^{(1)}$. Then $Z^{(p)}T^{(p)} = \mathcal{P}(H)$ where $\mathcal{P}(\lambda) = (\lambda - \tau_1) \cdots (\lambda - \tau_p)$.

**Proof** For $p = 1$ the result is Line 2.1 of Algorithm 3.1. Suppose the result is true for $p - 1$. From Line 2.2 of Algorithm 3.1 and Lemma 3.2 we have

$$R^{(p)} = (H^{(p+1)} - \tau_p I)(Q^{(p)})^H = (Z^{(p)})^H (H - \tau_p I) Z^{(p)} (Q^{(p)})^H,$$

and note that $Z^{(p)}(Q^{(p)})^H = Z^{(p-1)}$. Thus

$$T^{(p)} = R^{(p)} T^{(p-1)} = (Z^{(p)})^H (H - \tau_p I) Z^{(p-1)} T^{(p-1)},$$

which results in $Z^{(p)}T^{(p)} = (H - \tau_p I) Z^{(p-1)} T^{(p-1)} = \mathcal{P}(H)$ by the induction hypothesis. $\square$

**Theorem 3.3** Suppose that $H \in \mathbf{R}^{n \times n}$ and let $\mathcal{P}(\lambda) = (\lambda - \tau_1) \cdots (\lambda - \tau_p)$ be a polynomial. If $Hs_i = s_i \lambda_i$ where $s_i \neq 0$ then

$$\mathcal{P}(H) s_i = s_i \mathcal{P}(\lambda_i). \tag{3.1.1}$$

Each $H^{(j)}$ computed by the iteration is orthogonally similar to the original $H$ according to Lemma 3.2. Theorem 3.2 tells us that the explicitly shifted QR-iteration computes the QR factorization of $\mathcal{P}(H)$. The proof is due to Stewart [86, page 353]. Since $T^{(p)}$ is an upper triangular matrix, the first $k$ columns of $Z^{(p)}$ are an orthogonal basis for the space spanned by the first $k$ columns of $\mathcal{P}(H)$.

Since any of the shifts might have a nonzero imaginary part, the matrix $Z^{(p)}$ is in general unitary. In practical computation, the $Z^{(p)}$ constructed is orthonormal as long as two of the shifts applied form a complex conjugate pair. The details of the application of a complex conjugate pair of shifts in real arithmetic are delayed until § 3.4. Unless otherwise stated, we assume that if a shift has a nonzero imaginary part then its complex conjugate pair is also applied.

If any shift $\tau_j$ is equal to an eigenvalue $\lambda_i$ of $H$, then Theorem 3.3 gives that $\mathcal{P}(\lambda_i) = 0$. Thus, the non-zero eigenvalues of $H$ used as shifts are zero eigenvalues of $\mathcal{P}(H)$. The previous three results will prove useful for the remainder of the thesis. For the present they allows us to establish the following theorem.

**Theorem 3.4** Suppose that $AU = UH$ is an upper Hessenberg decomposition of $A$ where $H$ has positive sub-diagonal elements. Suppose that Algorithm 3.1 is used with the $p$ shifts $\tau_1, \ldots, \tau_p$ on $H$ resulting in $H^{(p)}$. Let $Z^{(p)} = Q^{(1)} \cdots Q^{(p)}$ and $\mathcal{P}(\lambda) = (\lambda - \tau_1) \cdots (\lambda - \tau_p)$.

If $AV_k = V_k H_k + f_k e_k^T$ is an Arnoldi factorization with the first column of $V_k$ equal to $\varrho \mathcal{P}(A) U e_1$ where $\varrho^{-1} = \|\mathcal{P}(A) U e_1\|$, then $H_k$ is the same as the leading principal sub-matrix of order $k$ of $H^{(p)}$ and $V_k = UZ^{(p)}[e_1, \ldots, e_k]$ for $k = 1, \ldots, n$.

**Proof** Let $AU = UH$ be a upper Hessenberg decomposition of $A$ where $H$ has positive sub-diagonals elements. Using Lemma 3.2 it follows that

$$AUZ^{(p)} = UHZ^{(p)} = UZ^{(p)} H^{(p+1)}. \tag{3.1.2}$$

Partition $UZ^{(p)} = \begin{bmatrix} W_k & \bar{W}_{n-k} \end{bmatrix}$ and $H^{(p+1)} = \begin{bmatrix} H_k^{(p+1)} & M_k^{(p+1)} \\ \beta_k^{(p+1)} e_1 e_k^T & \bar{H}_{n-k}^{(p+1)} \end{bmatrix}$. Equate the first $k$ columns of equation (3.1.2) to obtain

$$AW_k = W_k H_k^{(p+1)} + \beta_k^{(p+1)} (\bar{W}_{n-k} e_1) e_k^T.$$

Theorem 3.2 gives $Z^{(p)} T^{(p)} = \mathcal{P}(H)$ where $T^{(p)} = R^{(p)} \cdots R^{(1)}$. But $\mathcal{P}(H) = \mathcal{P}(U^T AU) = U^T \mathcal{P}(A) U$ which implies that $UZ^{(p)} T^{(p)} = \mathcal{P}(A) U$. If $\hat{\rho}_1 = e_1^T T^{(p)} e_1$

then $UZ^{(p)}T^{(p)}e_1 = UZ^{(p)}e_1\hat{\rho}_1$ which gives $UZ^{(p)}e_1 = \hat{\rho}_1^{-1}\mathcal{P}(A)Ue_1$. Theorem 2.1 of Chapter 2 (Implicit Q) then gives that $H_k = H_k^{(p+1)}$, $V_k = W_k$, and $f_k = \beta_k^{(p+1)}(\bar{W}_{n-k}e_1)$ with $\varrho = \hat{\rho}_1^{-1}$. □

We remark that if in the theorem's hypothesis the $m$-th sub-diagonal of $H$ is zero, then the conclusion only holds for $k = 1, \ldots, m$. A fundamental identification between an Arnoldi factorization and an explicitly shifted QR iteration is established. The first $m$ columns of $Z^{(p)}$ are an orthogonal basis for the Krylov subspace $\mathcal{K}_m(A, \varrho\mathcal{P}(A)Ue_1)$. In words, every step of a QR-iteration defines a Krylov subspace and hence an Arnoldi factorization. The immediate benefit is to establish the convergence typical of an Arnoldi iteration.

## 3.2 Convergence of an Explicitly Shifted QR Iteration

The main result of this section gives conditions that determine the convergence of the explicitly shifted QR iteration on Hessenberg matrices. Parlett [60] presents the first set of comprehensive sufficient conditions for convergence of the QR-iteration on Hessenberg matrices while a portion of the paper by Parlett and Poole [65] considers a geometric convergence theory for Hessenberg matrices. A comprehensive geometric convergence theory for the shifted QR iteration is presented by Watkins and Elsner [100] within the more general framework of generic *GR algorithms*. A GR algorithm is an iterative procedure such as in Algorithm 3.1 where the QR factorization is replaced with any other decomposition of the form $GR = H - \tau I$ where $R$ is upper triangular and $G$ is a nonsingular matrix.

**Theorem 3.5** Let $H \in \mathbf{R}^{n\times n}$ be an unreduced upper Hessenberg matrix and $\Psi(\lambda)$ be a polynomial. Order the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of $H$ so that $|\Psi(\lambda_1)| \geq |\Psi(\lambda_2)| \geq \cdots \geq |\Psi(\lambda_n)|$. Let $HQ = QR$ a real Schur decomposition where the first $k$ columns of $Q$ span an eigenspace corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_k$. Suppose $k$ is a positive integer less than $n$ such that $\rho_k \equiv |\Psi(\lambda_{k+1})|/|\Psi(\lambda_k)| < 1$.

If a sequence of shifts $\{\tau_i\}_{i=1}^m$ has the properties that

$$\begin{aligned}
\mathcal{P}_m(\lambda_i) \equiv (\lambda_i - \tau_1)\cdots(\lambda_i - \tau_m) &\rightarrow \Psi(\lambda_i), \quad i = k+1, \ldots, n \\
\mathcal{P}_m(\lambda_i) &\neq 0, \quad i = 1, \ldots, k \\
\prod_{i=1}^{m} \tau_i &\in \mathbf{R},
\end{aligned}$$

as $m \to \infty$, then Algorithm 3.1 computes an upper Hessenberg matrix

$$H^{(m+1)} \equiv \begin{bmatrix} H_k^{(m+1)} & M_k^{(m+1)} \\ \beta_{k+1}^{(m+1)} e_1 e_k^T & \bar{H}_{n-k}^{(m+1)} \end{bmatrix},$$

and an orthogonal matrix $Z^{(m)}$ such that for every value of $\hat{\rho}_k$ satisfying $\rho_k < \hat{\rho}_k < 1$ there exist a constant $C$ such that

$$|\beta_{k+1}^{(m+1)}| \le C(\hat{\rho}_k)^m \quad \text{and} \quad \mathrm{dist}(Q_k, Z_k^{(m)}) \le C(\hat{\rho}_k)^m,$$

where $Z_k^{(m)} = Z^{(m)}[e_1, \ldots, e_k]$.

**Proof** See Theorems 5.4 and 6.2 of Watkins and Elsner [100]. □

The distance between the subspaces [18, 35] $\mathcal{R}(Q_k)$ and $\mathcal{R}(Z_k^{(p)})$ may be shown to be equal to $\sqrt{1 - \|Q_k^T Z_k^{(m)}\|^2}$. For increasing values of $m$, the approximating subspace $\mathcal{R}(Z_k^{(p)})$ aligns itself with $\mathcal{R}(Q_k)$. Thus the $\mathrm{dist}(Q_k, Z_k^{(m)}) \to 0$ and the eigenvalues of $H_k^{(m+1)}$ tend to $\lambda_1, \ldots, \lambda_k$. It follows from the theorem that for all values of $k$ such that $\rho_k < 1$, the $k$-th sub-diagonal element of $H^{(m+1)}$ tends to zero.

The hypothesis on the product of the shifts ensures that if one is applied with a nonzero imaginary part, then its complex conjugate is also a shift. The hypothesis on $\rho_k$ implies that complex conjugate pairs of eigenvalues are kept together; $\lambda_i = \bar{\lambda}_j$ only if $i, j \le k$.

The theorems proved by Watkins and Elsner in [100] identify the convergence of the QR algorithm with that of simultaneous iteration, or subspace iteration. The QR-iteration uses the starting subspace of $\mathrm{Span}\{e_1, e_2, \ldots, e_k\}$. This is easily seen by using Theorem 3.2 and equating the first $k$ columns of $Z^{(m)}T^{(m)} = \mathcal{P}(H)$. This forms the basis of a geometric convergence theory for the QR-iteration and other GR algorithms.

### 3.2.1 Implications for a Shifting Strategy

The following example is due to Watkins and Elsner [100, page 30] and illustrates the use of Theorem 3.5.

Suppose that $\{\lambda_1, \ldots, \lambda_k\} \cup \{\lambda_{k+1}, \ldots, \lambda_n\}$ is a disjoint partition of the spectrum an unreduced upper Hessenberg matrix $H \in \mathbf{R}^{n \times n}$. We also assume that the complex conjugate pairs of eigenvalues are kept together; $\lambda_i = \bar{\lambda}_j$ implies that $i, j \le k$ or $i, j > k$. Define the polynomials

$$\Psi(\lambda) = (\lambda - \lambda_{k+1}) \cdots (\lambda - \lambda_n) \quad \text{and} \quad \mathcal{P}_m(\lambda) = (\lambda - \tau_1) \cdots (\lambda - \tau_m).$$

The shifts $\tau_i$ are chosen so that $\Psi(\lambda_i) - \mathcal{P}_m(\lambda_i) \to 0$ as $m \to \infty$, for $i = k+1, \ldots, n$ but $\mathcal{P}_m(\lambda_j) \neq 0$ for $j = 1, \ldots, k$ and that there exist a positive integer $m_0$ such that for all integers $m > m_0$,

$$\min_{j=1,\ldots,k} |\mathcal{P}_m(\lambda_j)| \;>\; \max_{j=k+1,\ldots,n} |\mathcal{P}_m(\lambda_j)|. \tag{3.2.1}$$

It is also assumed that if any of the shifts has a nonzero imaginary part, its complex conjugate is also a shift. If $\rho_k \equiv |\Psi(\lambda_{k+1})|/|\Psi(\lambda_k)| < 1$, then Theorem 3.5 gives that Algorithm 3.1 computes a sequence of Hessenberg matrices $H^{(m)}$ and orthogonal matrices $Z^{(m)}$ such that

$$\beta_{k+1}^{(m+1)} \to 0 \quad \text{and} \quad \operatorname{dist}(Q_k, Z_k^{(m)}) \to 0$$

where $Z_k^{(m)} = Z^{(m)}[e_1, \ldots, e_k]$. It follows that $HZ_k^{(m)} = Z_k^{(m)} H_k^{(m)}$ is converging to the partial real Schur decomposition of interest.

The search is for the best approximating polynomial $\mathcal{P}_m(\lambda)$ or equivalently, a proper set of shifts. If, for example, $\tau_i = \lambda_{k+i}$ for $i = 1, \ldots, n-k$ then $\mathcal{P}_{n-k}(\lambda) = \Psi(\lambda)$ and $\rho_k = 0$. Thus, after application of the $n-k$ shifts, the leading principal submatrix of order $k$ of the upper Hessenberg matrix $H^{(n-k+1)}$ computed by Algorithm 3.1 contains the eigenvalues $\lambda_1, \ldots, \lambda_k$.

### 3.2.2 Implications for an Arnoldi Factorization

As mentioned in § 1.2.1 of Chapter 1, computing a partial real Schur decomposition corresponding to a small subset of the eigenvalues of $A$ is the major goal of this thesis. Since the size of $A$ is often so large as to prevent using the QR algorithm, let us consider the possibility of computing the just the leading portion of the iteration. Let $AU = UH$ be a Hessenberg decomposition. Using the notation of Theorem 3.5, we may write a length $k$ Arnoldi factorization as

$$AUZ_k^{(m)} \;=\; UZ_k^{(m)} H_k^{(m+l)} + \beta_{k+1}^{(m+l)} (Z^{(m)} e_{k+1}) e_k^T. \tag{3.2.2}$$

Suppose that $AQ_k = Q_k R_k$ is a partial real Schur decomposition of order $k$. Expand

$$UZ_k^{(m)} e_1 \;=\; Q_k x_k + r, \tag{3.2.3}$$

where $Q_k^T r = 0$. Note that $r = (I - Q_k Q_k^T) UZ_k^{(m)} e_1$: The norm of $r$ measures the sine of the angle between the $\mathcal{R}(Q_k)$ and the first column of $UZ_k^{(m)}$. If $H_k^{(m+l)}$ is unreduced, then Theorem 2.5 shows that $r$ approaches zero if and only if $\beta_{k+1}^{(m+l)}$ does. Theorem 3.5 gives the convergence rate of $\beta_{k+1}^{(m+l)}$ to zero given a shifting strategy. However, the shifting strategy has the effect of replacing the starting vector—which re-starts the factorization of equation (3.2.2). The IRA-iteration, introduced in the next chapter, is motivated by precisely this idea.

## 3.3 Duality of the QR iteration and Krylov Spaces

The following theorem establishes a fundamental relationship between the QR algorithm and inverse iteration.

> **Theorem 3.6** Suppose that $H-\tau I \in \mathbf{R}^{n\times n}$ is a nonsingular Hessenberg matrix. If $H-\tau I = QR$ where $Q \in \mathbf{R}^{n\times n}$ is orthogonal and $R \in \mathbf{R}^{n\times n}$ is upper triangular, then
>
> $$(H-\tau I)^{-T} = QL, \tag{3.3.1}$$
>
> where $L = R^{-T}$.

**Proof** The result follows easily by first inverting the equation $H-\tau I = QR$ and then taking the transpose of both sides. □

The proof of the theorem shows that the hypothesis that $H$ is an upper Hessenberg matrix may be removed. The only crucial hypothesis is that of nonsingularity.

Post-multiplying both sides of equation (3.3.1) with $e_n$ results in

$$(H-\tau I)^{-T} e_n \rho_n = Q e_n$$

where $\rho_n = e_n^T R e_n$. Apparently, one step of the QR-iteration amounts to inverse iteration with $(H-\tau I)^{-T}$ on the vector $e_n$. The implication is that while the QR-iteration builds an orthogonal factorization for the Krylov subspace $\mathcal{K}_n(A-\tau I, UQe_1)$ it is also building one for $\mathcal{K}_n((A-\tau I)^{-T}, UQe_n)$, where $AU = UH$ is an Hessenberg decomposition. We call the latter space the *dual* Krylov subspace of $\mathcal{K}_n(A-\tau I, UQe_1)$. This duality and the convergence theory developed for Krylov subspaces in § 2.6 of Chapter 2, helps to explain why the Hessenberg decomposition helps to sort the spectral information of $A$. Indeed, practical shifting strategies for the QR algorithm use information in $\mathcal{K}_j((A-\tau I)^{-T}, UQe_n)$ for $j = 1, 2$.

## 3.4 The Practical QR algorithm

This section briefly reviews some of the practical issues affecting the convergence of Algorithm 3.1. The issues considered include:

- Deflation.
- Selection of shifts.
- The *implicitly shifted* QR iteration.
- Computing eigenvectors and reordering the Schur decomposition.

Our discussion is patterned after those in Demmel [24], Golub and Van Loan [35], and Stewart [86].

For the remainder of the section we continue to assume that $AU = UH$ is an Hessenberg decomposition of $A$ and that $H$ is an unreduced upper Hessenberg matrix.

### 3.4.1 Deflation

Suppose that after $m$ steps of Algorithm 3.1 we have

$$H^{(m+1)} = \begin{bmatrix} H_{11}^{(m+1)} & H_{12}^{(m+1)} \\ \epsilon e_1 e_j^T & H_{22}^{(m+1)} \end{bmatrix}$$

where $H_{11}^{(m+1)} \in \mathbf{R}^{j\times j}$ for $1 \le j < n$. If $\epsilon$ is suitably small we may set it to zero—this is called *deflation*. This is justified since

$$AUZ^{(m)} = UZ^{(m)} \begin{bmatrix} H_{11}^{(m+1)} & H_{12}^{(m+1)} \\ 0 & H_{22}^{(m+1)} \end{bmatrix} + \epsilon(UZ^{(m)}e_1)e_j^T,$$

and setting $E = -\epsilon(UZ^{(m)}e_1)(UZ^{(m)}e_j)^T$ it follows that

$$(A+E)UZ^{(m)} = UZ^{(m)} \begin{bmatrix} H_{11}^{(m+1)} & H_{12}^{(m+1)} \\ 0 & H_{22}^{(m+1)} \end{bmatrix}.$$

Since $\|E\| = \epsilon$, deflating the sub-diagonal element is equivalent to computing the eigenvalues of a matrix near $A$. After deflation, two unreduced Hessenberg matrices remain. Since computing the eigenvalues of a matrix determines the roots of its characteristic polynomial, deflation is equivalent to factoring the characteristic polynomial for a nearby matrix.

A criterion used by both EISPACK [82] and LAPACK [1] is to check if $|\beta_j^{(m+1)}| \leq \eta(|\alpha_{j-1}^{(m+1)}| + |\alpha_j^{(m+1)}|)$ for $2 \leq j \leq n$ where $\eta$ is the machine precision. Since $\alpha_j^{(m+1)} \leq \|A\|$ the criterion deflates sub-diagonals that are small relative to the matrix. Every sub-diagonal element of $H^{(m+1)}$ element that satisfies the above inequality is set to zero. If $\beta_n^{(m+1)}$ is negligible, then $\alpha_n^{(m+1)}$ is an approximate eigenvalue and we continue the QR-iteration on the leading sub-matrix of $H^{(m+1)}$ of order $n-1$. Francis [33] also explains how deflation may be performed if the product of two consecutive sub-diagonal elements is small.

### 3.4.2 Shift selection

Although Theorem 3.5 indicates the convergence expected of Algorithm 3.1 given a set of shifts, the important question of selecting one has gone unanswered. From Lemma 3.1 we expect the last sub-diagonal element to become small after a QR step with a shift close to an eigenvalue of $H$. According to the results of § 3.3, the lower right hand corner of $H^{(m)}$ contains some important spectral information. Consider the residual of using $(e_n, \tau)$ as an approximation to an eigenpair of $(H^{(m)})^T$:

$$\|(H^{(m)} - \tau I)^T e_n\| = \|(H^{(m)})^T e_n - \tau e_n\| = \|(\alpha_n - \tau)e_n + \beta_n e_{n-1}\| \geq |\beta_n|.$$

The Rayleigh quotient $\tau \equiv e_n^T H^{(m)} e_n$ results in the minimum residual. Since the eigenvalues of $(H^{(m)})^T$ are the same as those of $H^{(m)}$, the previous discussion suggests that Algorithm 3.1 use the sequence of Rayleigh quotients $e_n^T H^{(m)} e_n$ as shifts. Assuming the hypothesis of Theorem 3.5 on $\rho_{n-1}$ are met, then $\beta_n^{(m)}$ tends toward zero. In fact, a straight forward calculation shows that before the last plane rotation necessary for the QR factorization of $H^{(m)} - \tau I$ we have

$$G_{n-1,n-2}^T \cdots G_{1,2}^T (H^{(m)} - \tau I) \;=\; \begin{bmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & \hat{\alpha}_{n-1}^{(m)} & \gamma_n \\ 0 & 0 & 0 & \beta_n^{(m)} & 0 \end{bmatrix},$$

where the $G_{i-1,i}$ are plane rotations. After completing the QR step it follows that

$$|\beta_n^{(m+1)}| \equiv \frac{(\beta_n^{(m)})^2 |\gamma_n|}{(\hat{\alpha}_{n-1}^{(m)})^2 + (\beta_n^{(m)})^2} \leq \frac{|\gamma_n|}{(\hat{\alpha}_{n-1}^{(m)})^2} (\beta_n^{(m)})^2.$$

Once $|\hat{\beta}_n^{(m)}| < 1$ then $\beta_n^{(m+1)}$ goes to zero at a quadratic rate. In particular if $H^{(m)}$ is symmetric, then $\gamma_n = \beta_n^{(m)}$ and the rate improves to a cubic one.

A shifting strategy due to Wilkinson [101, 35] has generally been adopted for the practical implementations of the QR algorithm [1, 82]. Suppose that $\nu_1$ and $\nu_2$ are the eigenvalues of the two by two block in the South–East corner of $H^{(m)}$. If $\nu_1$ and $\nu_2$ are both real numbers, then Wilkinson's shift is defined to be the value of $\nu_i$ closet to $\alpha_n^{(m)}$. Otherwise, the eigenvalues of this two by two block form a complex conjugate pair. The next section considers an efficient manner in which a complex conjugate pair of shifts are applied.

### 3.4.3 The Implicitly Shifted QR iteration

Theorem 2.1 (Implicit Q) of Chapter 2 gives conditions under which the Hessenberg decomposition of $(H - \nu_1 I)(H - \nu_2 I)$ is unique. If $H$ is an unreduced Hessenberg matrix, then the decomposition $(H - \nu_1 I)(H - \nu_2 I)$ is specified by the first column of $U$. Francis [33] also observed that

$$(H - \nu_1 I)(H - \nu_2 I)e_1 \quad = \quad \eta_1 e_1 + \eta_2 e_2 + \eta_3 e_3. \tag{3.4.1}$$

From Theorem 3.2 we have

$$Z^{(2)}\hat{R}^{(2)} = Q^{(1)}Q^{(2)}R^{(2)}R^{(1)} = (H - \nu_1 I)(H - \nu_2 I) = H^2 - 2\mathrm{Real}(\nu_1)H + |\nu_1|^2 I.$$

This implies that $\eta_1$,$\eta_2$, and $\eta_3$ are real numbers when $\nu_1$ is the complex conjugate of $\nu_2$. Thus, in theory, two consecutive QR steps with a complex conjugate pair of shifts may be applied in real arithmetic by computing the similarity transformation $H^{(2)} = (Z^{(2)})^T H Z^{(2)}$. But there is a more efficient manner in which to apply a complex conjugate pair of shifts. By the Implicit Q Theorem Francis concluded that only the values of $\eta_{1,2,3}$ are needed since only they are used when computing the first column of a QR factorization of $(H - \nu_1 I)(H - \nu_2 I)$.

Suppose that $W_0$ is a Householder reflector that transforms the vector defined by the right hand side of equation 3.4.1 into $\|\eta_1 e_1 + \eta_2 e_2 + \eta_3 e_3\| e_1$. Computing $W_0^T H W_0$ has the unfortunate side affect of destroying the Hessenberg structure in the leading principal sub-matrix of order four. The *implicitly shifted* QR iteration is defined by computing a Householder matrix $W_i$ so that $(W_0 \cdots W_i)^T H W_0 \cdots W_i$ for $i = 0, \ldots, n-1$ is an upper Hessenberg through its first $i$ columns. It may be easily shown [35] that $W_i e_1 = e_1$ for $1 \le i \le n-1$ and so $W_0 \cdots W_{n-1} e_1 = W_0 e_1 =$

$\|\eta_1 e_1 + \eta_2 e_2 + \eta_3 e_3\| e_1$. Hence, as long as $H$ is an unreduced upper Hessenberg matrix, the explicit and implicit QR-iterations are the same.

Finally, the QR algorithms of EISPACK [82] and LAPACK [1] also implicitly apply Wilkinson's shift. Implicitly applying a shift avoids subtracting the shift from the diagonal elements of $H$, possibly preventing loss of information due to cancelation. An example of this is presented in § 5.3 of Chapter 5. We also remark that a number of shifts may be implicitly applied. This is the basis for the multi-shift QR-iteration [8] of Bai and Demmel. Both Dubrulle [27] and Watkins [96] discuss the multi-shift algorithm and present explanations of why the algorithm performs poorly when the number of shifts applied is large.

### 3.4.4 Computing Eigenvectors and Reordering the Schur Decomposition

Suppose that $A \in \mathbf{R}^{n\times n}$ is reduced to upper quasi-triangular form by the QR algorithm:

$$Q^T A Q = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \equiv R, \tag{3.4.2}$$

where $Q$ is the orthogonal matrix computed by the algorithm. Equation (3.4.2) is a Schur form for $A$ of order $n$ where the sub-matrices $R_{11}$ and $R_{22}$ are of order $k$ and $n-k$, respectively. Assume that the spectrums of $R_{11}$ and $R_{22}$ are distinct. In practice, the order in which the computed eigenvalues of $A$ appear on the diagonal of $R$ is determined by Theorem 3.5.

If all the eigenvectors of $A$ are required an upper quasi-triangular matrix $S$ may be computed so that $RS = SD$ where $D$ is the quasi-diagonal portion of $R$. It follows that $AQS = QSD$. Further details are considered by Demmel [24] and Golub and Van Loan [35].

In many situations, only a small number, say $k$, eigenvectors are requested. If the corresponding eigenvalues are found in $R_{11}$, then the first $k$ columns of $Q$ are an orthogonal basis subspace corresponding to the eigenvalues of $R_{11}$. The eigenvectors are easily determined by computing those of $R_{11}$. Suppose that $R_{11}S_1 = S_1 D_1$ is a quasi-diagonal form; then $AQS_1 = QS_1D_1$.

If eigenvalues of interest are located in $R_{22}$ and a basis for the associated eigenspace is wanted then we must either increase the number of columns of $Q$ used or somehow place them at the top of $R$. Algorithms for re-ordering a Schur form accomplish this task by using orthogonal matrices to move the wanted eigenvalues to the top of $R$.

The recent work of Bai and Demmel [9] attempts to correct the occasional numerical problems encountered by Stewart's algorithm [87] EXCHNG. Their work was motivated by that of Ruhe [68] and that of Dongarra, Hammarling, and Wilkinson [25]. Both algorithms swap consecutive $1 \times 1$ and $2 \times 2$ blocks of a quasi-triangular matrix to attain the desired ordering.

Suppose that the matrix $R$ of equation (3.4.2) is of order two. EXCHNG constructs a plane rotation that zeros the second component of the eigenvector corresponding to the eigenvalue $\lambda_2 = R_{22}$. A similarity transformation is performed on $R$ with the plane rotation and the diagonal blocks are interchanged. We refer to a strategy that constructs an orthogonal matrix and performs a similarity transformation to interchange the eigenvalues as a *direct* swapping algorithm.

Consider the following alternate *iterative* swapping algorithm: Perform a similarity transformation on $R$ with an arbitrary orthogonal matrix followed by one step of the QR-iteration with shift equal to $\lambda_2$. The arbitrary orthogonal similarity transformation introduces a non-zero off-diagonal element in the $(2,1)$ entry so that the transformed $R$ is an unreduced upper Hessenberg matrix with the diagonal blocks coupled. Lemma 3.1 implies that the $(2,1)$ entry is zeroed since an eigenvalue is used as a shift and hence $\lambda_1$ and $\lambda_2$ are switched.

If the order of $R_{22}$ is equal to two, EXCHNG uses the iterative swapping strategy using a standard double shift to re-order the diagonal blocks. The direct swapping algorithm, instead, computes an appropriate orthogonal matrix by computing the QR factorization of a basis of two vectors that span the desired invariant subspace. The reader is referred to [9, 25] for further details.

An example and explanation for the occasional failure of Stewart's algorithm is considered in § 5.4 of Chapter 5.

# Chapter 4

# Re-starting an Arnoldi Iteration

The previous two chapters considered in detail two fundamental algorithms for computing approximations to the eigenvalues of $A$. The Arnoldi/Lanczos algorithms are appropriate when the matrix $A$ is so large that storage and computational requirements prohibit completing anything but a length $k \ll n$ factorization with Algorithm 2.2. If only a small subset of the eigenvalues are desired, the length $k$ Arnoldi factorization may suffice. The analysis of Chapter 2 indicates dictates that a strategy for finding $k$ eigenvalues in a length $k$ factorization is to find an appropriate starting vector that forces $f_k$ to vanish. However, working in finite precision arithmetic generally removes the possibility of the computed residual ever vanishing exactly—even if a length $n$ factorization is built.

The QR algorithm, on the other hand, computes an approximation to a real Schur decomposition of $A$. All the eigenvalues of $A$ are approximated and the eigenvectors are readily available. Theorem 3.4 of Chapter 3 shows the relationship between the Arnoldi/Lanczos and QR algorithms. In exact arithmetic, when using the same starting vector, both algorithms generate the same orthogonal and upper Hessenberg matrices. Forcing the residual to zero for the Arnoldi/Lanczos algorithms has the effect of deflating a sub-diagonal element during the QR algorithm.

The idea of re-starting the Arnoldi iteration is motivated by Theorems 2.1 and 2.5. Our goal will be to construct a starting vector that is a member of the invariant subspace of interest. Theorem 2.5 then gives that the residual vector associated with the truncated Arnoldi factorization vanishes. This chapter considers two re-starting variants. The first variant, introduced by Saad [74], *explicitly* re-starts the Arnoldi factorization and is the subject of § 4.1. The second approach is to *implicitly* re-start the factorization. This IRA-iteration, introduced by Sorensen [83], is the subject of § 4.2. A numerical example is presented in § 4.3 that serves to illustrate how both variants perform in practice. The important subject of polynomial iterations or acceleration methods is examined in § 4.4. This includes the polynomial iterations of

Saad and a careful look at the IRA-iteration. Finally, § 4.4.3 examines an interesting explicitly re-started approach recently introduced by Scott [80].

## 4.1 Explicitly Re-starting the Arnoldi Iteration

Suppose that $k \ll n$ eigenvalues of $A$ require approximation. As explained in § 1.2.1 of Chapter 1, the $k$ eigenvalues of $A$ of interest are called the wanted ones. The ERA-iteration starts by building an Arnoldi factorization of length $k+p$ for some positive integer $p$. An improved starting vector is then obtained by using a specific linear combination of the columns of $V_{k+p}$. The linear combination is determined by the spectral information of $H_{k+p}$. The ERA-iteration is defined by repeating the above process. Algorithm 4.1 outlines the procedure followed by some comments.

**Algorithm 4.1** (Explicitly Re-started Arnoldi Iteration)
Input: An unit vector $v_1^{(1)}$.

1.1 For $j = 1, 2, \ldots$ until convergence

  2.1 Build an Arnoldi factorization of length $k+p$ given a starting vector $v_1^{(j)}$:
  $AV_{k+p}^{(j)} = V_{k+p}^{(j)} H_{k+p}^{(j)} + f_{k+p}^{(j)} e_{k+p}^T$ ;

  2.2 Compute the decomposition :
  $H_{k+p}^{(j)} S_{k+p}^{(j)} = S_{k+p}^{(j)} D_{k+p}^{(j)}$
  where $(S_{k+p}^{(j)}, D_{k+p}^{(j)})$ is a quasi-diagonal form for $H_{k+p}^{(j)}$ ordered so that the wanted eigenvalues are in leading portion of $D_{k+p}^{(j)}$ ;

  2.3 If $k$ wanted eigenvalues $\{\theta_i^{(j)}\}_{i=1}^k$ of $H_{k+p}^{(j)}$ have converged then exit the current loop ;

  2.4 Compute the unit vector : $v_1^{(j+1)} \leftarrow V_{k+p}^{(j)} s^{(j)}$
  where $s^{(j)} \leftarrow \gamma_1^{(j)} S_{k+p}^{(j)} e_1 + \cdots + \gamma_k^{(j)} S_{k+p}^{(j)} e_k$ and $\|s^{(j)}\| = 1$.

1.2 End For

1.3 If desired, compute the Ritz vectors $\hat{x}_i^{(j)} = V_{k+p}^{(j)} s_i^{(j)}$
where $H_{k+p}^{(j)} s_i^{(j)} = s_i^{(j)} \theta_i^{(j)}$.

We briefly address the issues of determining convergence, the choice of $p$, and how the coefficients $\gamma_i^{(j)}$ for $i = 1, \ldots, k$ are selected.

An eigenvalue of $H_{k+p}^{(j)}$ (or equivalently, a Ritz value of $A$) is converged when it satisfies the stopping criterion of § 2.5 of Chapter 2. A practical implementation of Algorithm 4.1 would include the deflation of converged Ritz values during the course of the iteration. Chapter 6 discusses deflation rules in detail.

The choice of $p$ is usually a tradeoff between the length of a factorization that may be tolerated and the rate of convergence. From the results on the convergence of Krylov spaces in § 2.6, the accuracy of the Ritz values typically increases as $p$ does. However, for increasing $p$, the number of Arnoldi vectors stored as well as the size of the Hessenberg matrix increases. For most problems, the size of $p$ is determined experimentally.

The selection of the expansion coefficients is the most unsettling decision that needs to be made. Saad first suggested [74] choosing the coefficients so that the slowest converging Ritz vectors are favored the most. For example, let $\hat{\gamma}_i^{(j)}$ be the Ritz estimate for the $i$-th wanted Ritz value during the $j$-th iteration of the loop. The $\gamma_i^{(j)}$ are the properly normalized $\hat{\gamma}_i^{(j)}$ that result in the unit vector $s^{(j)}$. The use of *polynomial filters* that better employ the spectral information of $H_{k+p}^{(j)}$ to determine an improved starting vector is addressed in § 4.4. Since the new starting vector computed at line 2.4 is a linear combination of the columns of $V_{k+p}^{(j)}$, there is a unique vector $c_{k+p} \in \mathbf{R}^{k+p}$ such that the relation $K_{k+p}(A, v_1^{(j)})c_{k+p} = v_1^{(j+1)}$ holds. In other words, the new starting vector is determined by applying a polynomial of at most degree $k+p-1$ in $A$ to the current starting one.

Finally, Saad [78, page 234] suggests using a *deflated* algorithm when computing $k > 1$ Ritz values. The idea is to compute one Ritz value and approximate Schur vector at a time. The process uses an ERA-iteration to compute an approximate Ritz pair—taking care that the Arnoldi vectors are orthogonalized against the approximate Schur vectors. During each cycle of the iteration, the approximate Schur vectors are kept in the leading portion of $V_{k+p}^{(j)}$, and the corresponding part of $H_{k+p}^{(j)}$ is upper quasi-triangular. As each Ritz value converges, the corresponding Ritz vector is orthogonalized against the approximate Schur basis to obtain another approximate Schur vector. This orthogonalization procedure is further discussed in § 6.5 of chapter 5 within the context of deflation. When the converged portion of the Arnoldi factorization of the $j$-th cycle of the iteration contains a basis for an approximate invariant subspace of dimension $k$, the deflated algorithm is halted. This procedure, analogous to the one used by Scott [80], is considered in more detail in § 4.4.3.

## 4.2 The Implicitly Restarted Arnoldi Iteration

The IRA-iteration is motivated by re-starting the factorization in an *implicit* manner as suggested in § 3.2.2 of Chapter 3. The scheme is called implicit because the updating of the starting vector is accomplished with an implicitly shifted QR mechanism on $H_k$. This will allows us to update the starting vector by working with orthonormal matrices that live in $\mathbf{R}^{k\times k}$ rather than in $\mathbf{R}^{n\times n}$.

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

As an example, suppose that $p = 1$ and $k$ represents the dimension of the desired invariant subspace. Let $\mu$ be a real shift and let $H_{k+1} - \mu I = QR$ with $Q$ orthogonal and $R$ upper triangular matrices, respectively. From equation (2.2.1) of Chapter 2,

$$
\begin{aligned}
(A-\mu I)V_{k+1} - V_{k+1}(H_{k+1}-\mu I) &= f_{k+1}e_{k+1}^T, \\
(A-\mu I)V_{k+1} - V_{k+1}QR &= f_{k+1}e_{k+1}^T, \\
(A-\mu I)(V_{k+1}Q) - (V_{k+1}Q)(RQ) &= f_{k+1}e_{k+1}^T Q, \\
A(V_{k+1}Q) - (V_{k+1}Q)(RQ+\mu I) &= f_{k+1}e_{k+1}^T Q. \qquad (4.2.1)
\end{aligned}
$$

The matrices are updated via $V_{k+1}^+ \leftarrow V_{k+1}Q$ and $H_{k+1}^+ \leftarrow RQ + \mu I$ and the latter matrix remains upper Hessenberg since $R$ is upper triangular and $Q$ is upper Hessenberg. However, equation (4.2.1) is not quite a legitimate Arnoldi factorization. Equation (4.2.1) fails to be an Arnoldi factorization since the matrix $f_{k+1}e_{k+1}^T Q$ has a non-zero $k$-th column. Partitioning the matrices in the updated equation results in

$$
\begin{aligned}
A\begin{bmatrix} V_k^+ & v_{k+1}^+ \end{bmatrix} &= \begin{bmatrix} V_k^+ & v_{k+1}^+ \end{bmatrix}\begin{bmatrix} H_k^+ & h_{k+1}^+ \\ \beta_{k+1}^+ e_k^T & \alpha_{k+1} \end{bmatrix} \\
&+ f_{k+1}\begin{bmatrix} \sigma_{k+1}e_k^T & \gamma_{k+1} \end{bmatrix}, \qquad (4.2.2)
\end{aligned}
$$

where $\sigma_{k+1} = e_{k+1}^T Q e_k$ and $\gamma_{k+1} = e_{k+1}^T Q e_{k+1}$. Equating the first $k$ columns of equation (4.2.2) gives

$$
AV_k^+ = V_k^+ H_k^+ + (\beta_{k+1}^+ v_{k+1}^+ + \sigma_{k+1} f_{k+1})e_k^T. \qquad (4.2.3)
$$

Performing the update $f_{k-1}^+ \leftarrow \beta_k^+ v_k^+ + \sigma_k f_k$ and noting that $(V_{k-1}^+)^T f_{k-1}^+ = 0$ it follows that equation (4.2.3) is a length $k$ Arnoldi factorization.

The following elementary but technical result shows that the previous idea may be extended for up to $1 < p < k$ shifts and a new length $k$ Arnoldi factorization remains. A similar result was proved by Paige, Parlett and Van der Vorst in Lemma 1 of [58] for the Lanczos factorization.

**Lemma 4.1** Let $AV_{k+p} = V_{k+p}H_{k+p} + f_{k+p}e_{k+p}^T$ be a length $k+p$ Arnoldi factorization where $H_{k+p}$ is an unreduced upper Hessenberg matrix. If

$$\psi_p(\lambda) \;=\; \prod_{i=1}^{p}(\lambda - \mu_i),$$

then

$$\begin{aligned} \psi_p(A)V_{k+p} &= V_{k+p}\psi_p(H_{k+p}) \\ &+ \sum_{j=1}^{p}\psi_{j+1}^p(A)fe_m^T\psi_{p-j}(H_{k+p}), \end{aligned} \tag{4.2.4}$$

where $\psi_j(\lambda) = \prod_{i=1}^{j}(\lambda - \mu_i)$ and $\psi_j^p(\lambda) = \prod_{i=j}^{p}(\lambda - \mu_i)$.

Moreover,

$$\psi_p(A)V_k \;=\; V_{k+p}\psi_p(H_{k+p})\begin{bmatrix} e_1 & e_2 & \cdots & e_k \end{bmatrix}. \tag{4.2.5}$$

**Proof** The proof is by mathematical induction. Define $m \equiv k+p$ and the subscripts are suppressed on $V_{k+p}$ and $H_{k+p}$ for the proof. Since $\psi_1(A)V = V\psi_1(H) + fe_m^T$ where $\psi_1(\lambda) = \lambda - \mu_1$, the base case for $p = 1$ is established. Assume the lemma's truth for polynomials $\psi_j(\lambda)$ of degree $j \le p$. Let $\psi_{p+1}(\lambda) = (\lambda - \mu_{p+1})\psi_p(\lambda)$. Using the induction hypothesis, it follows that

$$\begin{aligned} \psi_{p+1}(A)V &= (A - \mu_{p+1}I)\psi_p(A)V \\ &= (A - \mu_{p+1}I)\left\{V\psi_p(H) + \sum_{j=1}^{p}\psi_{j+1}^p(A)fe_m^T\psi_{p-j}(H)\right\} \\ &= V(H - \mu_{p+1}I)\psi_p(H) + fe_m^T\psi_p(H) \\ &+ (A - \mu_{p+1}I)\sum_{j=1}^{p}\psi_{j+1}^p(A)fe_m^T\psi_{p-j}(H) \\ &= V\psi_{p+1}(H) + \sum_{j=1}^{p+1}\psi_{j+1}^{p+1}(A)fe_m^T\psi_{p+1-j}(H), \end{aligned}$$

which the desired result.

Since $H$ is unreduced it follows that $e_i^T \psi_{p-1}(H) e_j = 0$ for $i+p-1 > j$. Moreover, the last matrix on the right-hand side of equation (4.2.4) is zero through its first $k$ columns, equation (4.2.5) is established. $\square$

Denote the QR factorization of $\psi_p(H_{k+p}) = Z^{(p)} T^{(p)}$. Since $H_{k+p}$ is an unreduced upper Hessenberg matrix, $e_i^T \psi_p(H_{k+p}) e_j = 0$ for $i + p > j$ and hence $Z^{(p)}$ shares this same property. Partitioning

$$\psi(H_{k+p}) \;=\; \begin{bmatrix} Z_k^{(p)} & \bar{Z}_p^{(p)} \end{bmatrix} \begin{bmatrix} T_k^{(p)} & M_r \\ 0 & \bar{T}_p^{(p)} \end{bmatrix}$$

allows us to rewrite equation (4.2.5) as

$$\psi_p(A) V_k \;=\; V_{k+p} Z_k^{(p)} T_k^{(p)}. \tag{4.2.6}$$

In words, an IRA-iteration is equivalent to performing simultaneous iteration on the matrix $V_k$ while working only with matrices of order $k+p$ ! The column space of $V_{k+p} Z_k^{(p)}$ is an orthogonal basis for $\psi_p(A) V_k$. This is analogous to the well known connection between subspace and QR-iterations. Post-multiplication of an Arnoldi factorization of length $k+p$ with $Z^{(p)}$ results in

$$A V_{k+p} Z^{(p)} \;=\; V_{k+p} Z^{(p)} (Z^{(p)})^T H_{k+p} Z^{(p)} + f_{k+p} e_{k+p}^T Z^{(p)}.$$

Equating the first $k$ columns results in

$$A V_k^+ \;=\; V_k^+ H_k^+ + f_k^+ e_k^T.$$

In direct analogy with the single shift case, the updated residual is

$$f_k^+ \;=\; \beta_k^+ V_{k+p} Z^{(p)} e_{k+1} + \sigma_k^{(p)} f_{k+p},$$

where $\beta_k^+ = (Z^{(p)} e_{k+p})^T H_{k+p} Z^{(p)}$ and $\sigma_k^{(p)} = e_{k+p}^T Z^{(p)} e_k$. The norm of $f_k^+$ is easily seen to be $\sqrt{(\beta_k^+)^2 + (\sigma_k^{(p)})^2 \|f_{k+p}\|^2}$.

Application of the shifts is performed implicitly as in the QR algorithm. If the shifts are in complex conjugate pairs, the implicit double shift is used to avoid complex arithmetic. The following procedure outlines the scheme.

**Algorithm 4.2** (Implicitly Re-started Arnoldi Iteration)
Input: A length $k$ Arnoldi factorization $A V_k^{(1)} = V_k^{(1)} H_k^{(1)} + f_k^{(1)} e_k^T$.

1.1 For $j = 1, 2, \ldots$ until convergence

2.1 Extend the length $k$ Arnoldi factorization by $p$ steps :
$AV_{k+p}^{(j)} = V_{k+p}^{(j)} H_{k+p}^{(j)} + f_{k+p}^{(j)} e_{k+p}^T$ ;

2.2 If $k$ wanted eigenvalues $\{\theta_i^{(j)}\}_{i=1}^k$ of $H_{k+p}^{(j)}$ have converged exit the current loop ;

2.3 Apply $p$ implicitly QR steps with shifts $\mu_1^{(j)}, \ldots, \mu_p^{(j)}$ to $H_{k+p}^{(j)}$ to obtain $H_{k+p}^{(j)} Z^{(p)} = Z^{(p)} H_{k+p}^{(j+1)}$ ;

2.4 Update the length $k+p$ Arnoldi factorization of Line 2.1 :
$AV_{k+p}^{(j)} Z^{(p)} = V_{k+p}^{(j)} Z^{(p)} H_{k+p}^{(j+1)} + f_{k+p}^{(j)} e_{k+p}^T Z^{(p)}$ ;

2.5 Obtain a length $k$ Arnoldi factorization by retaining only the first $k$ columns of the factorization in Line 2.4 :
$AV_k^{(j+1)} = V_k^{(j+1)} H_k^{(j+1)} + f_k^{(j+1)} e_k^T$

1.2 End For

1.3 If desired, compute the Ritz vectors $\hat{x}_i^{(j)} = V_{k+p}^{(j)} s_i^{(j)}$
where $H_{k+p}^{(j)} s_i^{(j)} = s_i^{(j)} \theta_i^{(j)}$.

One cycle of the iteration is illustrated in Figures 4.1— 4.3. Theorem 3.4 implies that after each cycle of the $j$ loop,

$$\begin{aligned}
v_1^{(j+1)} &= V_k^{(j+1)} e_1, \\
&= V_{k+p}^{(j)} Z^{(p)} e_1, \\
&= \psi_p^{(j)}(A) V_{k+p}^{(j)} e_1, \quad \text{(Theorem 3.4)} \\
&= \psi_p^{(j)}(A) v_1^{(j)}, \\
&= \psi_p^{(j)}(A) \cdots \psi_p^{(1)}(A) v_1^{(1)}, \\
&\equiv \mathcal{P}_{jp}(A) v_1^{(1)}, \qquad (4.2.7)
\end{aligned}$$

where $\psi_p^{(j)}(\lambda) = \tau^{(j)}(\lambda - \mu_1^{(j)}) \cdots (\lambda - \mu_p^{(j)})$ with $\tau^{(j)}$ a normalization factor. The results of Theorem 3.5 determine the rate of convergence of the IRA-iteration given a set of shifts. Recall the example at the end of § 3.2 concerning a specific choices for $\Psi$ and $\mathcal{P}_m$ used by Theorem 3.5. The example implies that Algorithm 4.2 drives $f_k^{(j)}$ to zero if the discrete min-max problem (3.2.1) of Chapter 3 is solved. If the sequence of shifts $\{\mu_1^{(i)}, \ldots, \mu_p^{(i)}\}_{i=1}^j$ defining the polynomial $\mathcal{P}_{jp}(\lambda) = \psi_p^{(j)}(\lambda) \cdots \psi_p^{(1)}(\lambda)$ is a good approximation to $\Psi(\lambda) = (\lambda - \lambda_{k+1}) \cdots (\lambda - \lambda_n)$, then the IRA-iteration converges to the desired invariant subspace. Theorem 3.5 implies that the magnitude of the ratio of $\Psi(\lambda_k)$ to $\Psi(\lambda_{k+1})$ gives the convergence rate.

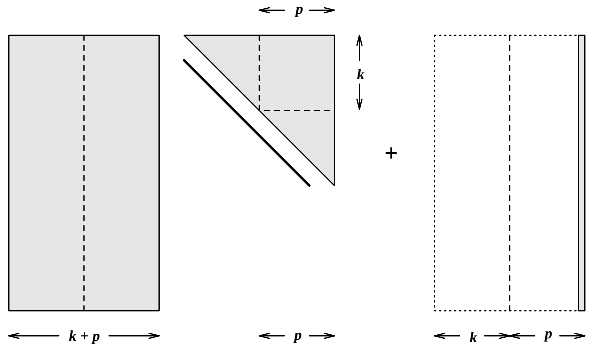


**Figure 4.1** The set of rectangles represents the matrix equation $V_{k+p}^{(j)} H_{k+p}^{(j)} + f_{k+p}^{(j)} e_{k+p}^T$ of an Arnoldi factorization. The unshaded region on the right is a zero matrix of $k + p - 1$ columns.

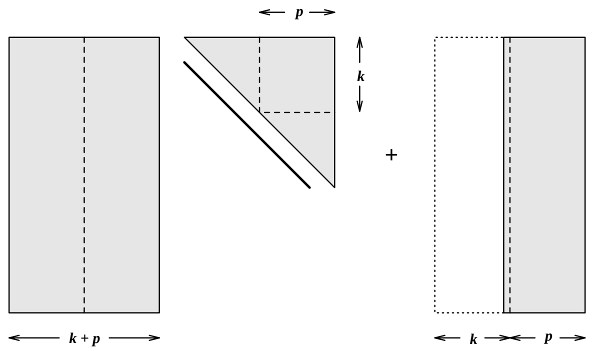


**Figure 4.2** After performing $p$ implicitly shifted QR steps on $H_{k+p}^{(j)}$, the middle set of pictures illustrates $V_{k+p}^{(j)} Z^{(p)} (Z^{(p)})^T H_{k+p}^{(j)} Z^{(p)} + f_{k+p}^{(j)} e_{k+p}^T Z^{(p)}$. The last $p + 1$ columns of $f_{k+p} e_{k+p}^T Z^{(p)}$ are non-zero because of the QR-iteration.

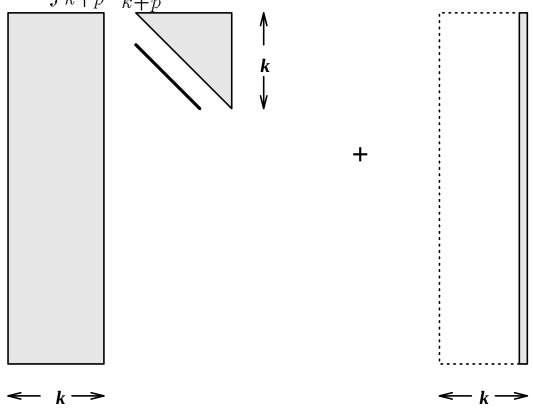


**Figure 4.3** After discarding the last $p$ columns, the final set represents $V_k^{(j+1)} H_k^{(j+1)} + f_k^{(j+1)} e_k^T$ of a length $k$ Arnoldi factorization.

Numerous choices are possible for the selection of the $p$ shifts. One immediate choice is to use the $p$ unwanted eigenvalues of $H_{k+p}^{(j)}$. This *exact* shifting scheme and others are discussed in § 4.4 on polynomial iterations. Exact shifts are equivalent to Rayleigh quotients: If $H_{k+p}^{(j)}s = s\theta$ then the identity $(V_{k+p}^{(j)}s)^T A V_{k+p}^{(j)} s = \theta$ follows from Arnoldi factorization of length $k+p$. Unlike the QR-iteration, the IRA-iteration or partial QR-iteration, does not have access to the spectral information necessary for the rapid convergence of the practical QR algorithm.

As for the ERA-iteration, the number of shifts to apply at each cycle of the above iteration is problem dependent. The only formal requirement is that $1 \le p \le n-k$. However, computational experience indicates that $p \ge k$ is preferable. Chapter 7 discusses the many tradeoffs when trying to select the size of $p$ relative to $k$.

The following result and Lemma 4.1 were communicated* by C. A. Beattie†.

> **Theorem 4.3** Assume the same hypothesis of Lemma 4.1. Suppose that $\{\mu_i\}_{i=1}^m$ is a set of shifts such that if $\mu_i$ has a nonzero imaginary part, then $\bar{\mu}_j$ is also a shift for some $i \ne j$. If
>
> $$\psi_p(\lambda) \;=\; \prod_{i=1}^{p}(\lambda - \mu_i),$$
>
> and $\psi_p(A)$ is non-singular, then
>
> $$\mathcal{K}_k(A, v_1^+)^{\perp} \;=\; \psi_p(A^T)^{-1}\mathcal{K}_k(A, v_1)^{\perp}, \tag{4.2.8}$$
>
> where $v_1^+ = V_{k+p}Z^{(p)}e_1$ and $\psi_p(H_{k+p}) = Z^{(p)}T^{(p)}$ is a QR factorization.

**Proof** Suppose that $w \in \psi_p(A^T)^{-1}\mathcal{K}_k(A, v_1)^{\perp}$. Then $w = \psi_p(A)^{-T}y$ from some vector $y \in \mathcal{K}_k(A, v_1)^{\perp}$. If $z \in \psi_p(A)\mathcal{K}_k(A, v_1)$ then $z = \psi_p(A)x$ for some vector $x \in \mathcal{K}_k(A, v_1)$. Thus,

$$w^T z = y^T \psi_p(A)^{-1}\psi_p(A)y = x^T y = 0,$$

and hence $w \in \{\psi_p(A)\mathcal{K}_k(A, v_1)\}^{\perp}$ establishing

$$\psi_p(A^T)^{-1}\mathcal{K}_k(A, v_1)^{\perp} \;\subset\; \{\psi_p(A)\mathcal{K}_k(A, v_1)\}^{\perp}. \tag{4.2.9}$$

---

*Workshop on Krylov Subspace Methods and Applications, Raleigh, NC, March 17–18, 1995

†Department of Mathematics, Virginia Polytechnic Institute and State University

Since $\psi_p(A)$ is nonsingular by hypothesis, it follows that $\psi_p(A^T)^{-1}$ exists and hence

$$\dim\{\psi_p(A^T)^{-1}\mathcal{K}_k(A,v_1)^\perp\} \;=\; \dim\{\mathcal{K}_k(A,v_1)^\perp\},$$

and

$$\dim\{\psi_p(A)\mathcal{K}_k(A,v_1)\} \;=\; \dim\{\mathcal{K}_k(A,v_1)\}.$$

Along with equation (4.2.9), the previous relations imply that $\{\psi_p(A)\mathcal{K}_k(A,v_1)\}^\perp = \psi_p(A^T)^{-1}\mathcal{K}_k(A,v_1)^\perp$.

By the second conclusion of Lemma 4.1, equation (4.2.5),

$$\begin{aligned}
\psi_p(A)\mathcal{K}_k(A,v_1) &= \mathcal{R}\{\psi_p(A)V_k\} \\
&= \mathcal{R}\{V_{k+p}\psi_p(H)\begin{bmatrix} e_1 & e_2 & \cdots & e_k \end{bmatrix}\} \\
&\equiv \mathcal{R}\{V_{k+p}Z^{(p)}T^{(p)}\begin{bmatrix} e_1 & e_2 & \cdots & e_k \end{bmatrix}\},
\end{aligned}$$

where a QR factorization of $\psi_p(H_{k+p})$ is $Z^{(p)}T^{(p)}$. The theorem is proved since $\psi_p(A)\mathcal{K}_k(A,v_1) = \mathcal{K}_k(A,v_1^+)$ where $v_1^+ = V_{k+p}Z^{(p)}e_1$. □

Suppose that $A$ is non-singular and that the grade of $v_1$ is $n$; in other words, the dimension of $\mathcal{K}_n(A,v_1)$ is $n$. If $AV_n = V_nH_n$ is an Hessenberg decomposition with $V_ne_1 = v_1$ then $\mathcal{K}_{n-k}(A^{-T},V_ne_n) = \mathcal{K}_k(A,v_1)^\perp$. The theorem shows that analogous to the duality of the QR-iteration discussed in § 3.3, during each cycle of an IRA-iteration, another IRA-iteration takes place on the Krylov subspace dual to $\mathcal{K}_k(A,v_1)$.

## 4.3 Explicit and Implicit Re-starting

This section presents a striking example that compares the ERA- and IRA-iterations. Let $A \in \mathbf{R}^{10\times 10}$ be zero everywhere except for diagonal elements

$$\alpha_{11} = 1, \alpha_{22} = 1, \alpha_{33} = 0, \alpha_{44} = 0, \alpha_{ii} = (i-1)\cdot 10^{-1}, \text{ for } i = 1,\ldots,9,$$

and ones on the sub-diagonal. Suppose that the vector $e_1$ is used to start both Algorithms 4.1 and 4.2 with $k=2$ and $p=2$ and the interest is to compute the two eigenvalues equal to one. Using an exact shift strategy, Algorithm 4.2 computes the approximate partial real Schur decomposition $AQ_2 = Q_2R_2$ where

$$R_2 \;\approx\; \begin{bmatrix} .94919 & .95789 \\ -2.6952\cdot 10^{-3} & 1.0508 \end{bmatrix},$$

with eigenvalues equal to $1 \pm i1.129168612228906 \cdot 10^{-8}$. The number of iterations needed was four and a total of ten matrix vector products were computed.

But Algorithm 4.1 stagnates. In fact, the same information was computed during every cycle of the iteration. For $j \geq 1$,

$$H_4^{(j)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The MATLAB function `EIG` computes the two eigenvectors

$$\begin{aligned} s_1^T &= \begin{bmatrix} 0 & .57735 & .57735 & .57735 \end{bmatrix}, \\ s_2^T &= \begin{bmatrix} 1.7 \cdot 10^{-18} & -.57735 & -.57735 & -.57735 \end{bmatrix}, \end{aligned}$$

corresponding to the two eigenvalues equal to one. If the expansion coefficients are chosen equal to the corresponding normalized Ritz estimates, the vector $s^{(j)} = e_1$ is computed during every cycle of the ERA-iteration.

The major drawback of using a linear combination of the eigenvectors of $H_{k+p}^{(j)}$ is that they may form a poor choice for the starting vector. If $H_{k+p}^{(j)}$ is defective, then there might not be enough eigenvectors corresponding to the wanted eigenvalues. As the previous example demonstrated, computing in finite precision arithmetic blurs this sharp characterization. A pair of approximate eigenvectors is produced that are aligned to working precision. On the other hand, using an expansion in terms of the Schur vectors of $H_{k+p}^{(j)}$ is a better behaved numerical process. As explained in § 4.4.2 the IRA-iteration implicitly uses a Schur basis of $H_{k+p}^{(j)}$. Golub and Wilkinson [38] examine the many practical difficulties involved when computing invariant subspaces. As the above example shows, computing in floating point arithmetic generally removes the possibility of ever detecting a defective matrix.

Among the several advantages an implicit updating scheme possess over an explicit one are:

- Only $p$ matrix vector products are required during each iteration instead of $k + p$.
- Maintaining a prescribed level of orthogonality for only $p$ additional Arnoldi vectors during each iteration instead of $k + p$.

- Re-starting with a linear combination of Schur vectors instead of eigenvectors.
- Ability to avoid explicit application of $\psi(A)$.
- The incorporation of the well understood numerical and theoretical behavior of the practical QR algorithm.

The last point was first mentioned by Sorensen [83]: This thesis makes a detailed study of the relationship with the QR algorithm. In particular, application of a shift may result in one of the sub-diagonal elements of $H_{k+p}^{(j)}$ becoming small. The impact of the deflation strategies associated with the QR-iteration upon the IRA-iteration is the subject of chapter 6. The convergence of the iteration to selected portions of the spectrum of $A$ may then be answered by appealing to the theory developed in Chapter 3.

## 4.4 Polynomial Iterations

As explained in the § 4.2, each iteration of Algorithms 4.1 and 4.2 implicitly replaces the starting vector of an Arnoldi factorization with $\psi(A)v_1$ where subscripts are dropped for ease of notation. If $A$ is diagonalizable where $z_j$ for $j = 1, \ldots, n$ are the eigenvectors, then it follows that $v_1 = z_1\zeta_1 + \cdots + z_n\zeta_n$ and then

$$\psi(A)v_1 = z_1\psi(\lambda_1)\zeta_1 + \cdots + z_n\psi(\lambda_n)\zeta_n. \tag{4.4.1}$$

Assuming that the eigenpairs $(z_i, \lambda_i)$ are ordered so that the $k$ wanted ones are at the beginning of the expansion, a polynomial of degree $p$ is sought so that the

$$\max_{i=k+1,\ldots,n} |\psi(\lambda_i)| < \min_{i=1,\ldots,k} |\psi(\lambda_i)|. \tag{4.4.2}$$

A good polynomial $\psi(\lambda)$ acts as a *filter*. Components in the direction of unwanted eigenvectors are damped or equivalently, components in the direction of wanted eigenvectors are amplified. We remark that according to Theorem 3.5 the convergence of the QR-iteration is also dependent upon the same discrete min-max polynomial approximation problem.

It should be emphasized that even if a good approximate solution is computed for the discrete min–max problem defined by equation (4.4.2), the unwanted products $\psi(\lambda_i)\zeta_i$ may not be small. This can only happen if the unwanted coefficients $\zeta_i$ are large. As we demonstrated in § 2.6 of Chapter 2, if the corresponding eigenvalue

$\lambda_i$ is poorly conditioned, then $\zeta_i$ may be large. The conclusion is that unwanted components in the direction of an eigenvector corresponding to a poorly conditioned eigenvalue may not be expected to become negligible. In addition, if $A$ is defective, it may not possess enough eigenvectors corresponding to the wanted eigenvalues $\lambda_1, \ldots, \lambda_k$. Numerically, problems are encountered when a basis for the desired invariant subspace is poorly conditioned.

### 4.4.1 The Polynomial Iterations of Saad

The acceleration techniques and hybrid methods presented by Saad in Chapter seven of [78] are motivated by attempting compute a reasonable solution of the min–max problem defined by equation (4.4.2). Saad suggests a two stage process for calculating approximations to wanted eigenvectors.

First, an Arnoldi factorization of length $k+p$ is built. The spectrum of the upper Hessenberg matrix of order $k+p$ is used to determine a polynomial $p_m(\lambda)$ of degree $m$. Examples include using the Chebyshev [76], based on Manteuffel [50] scheme, and least squares [77] polynomials introduced by Saad. Second, the polynomial $p_m(A)$ of degree $m$ is applied to a linear combination of the wanted eigenvectors of the upper Hessenberg matrix of order $k+p$. The resulting vector is said to be *filtered*. A Ritz vector is then determined using the the filtered one. Within the context of Algorithm 4.1, the filtered starting vector is just another choice for $s^{(j)}$ in line 2.4. The above process is repeated until $k$ wanted Ritz values converge. As mentioned at the end of § 4.1, the above iterated process may be used within a deflated ERA algorithm.

### 4.4.2 Implicit Polynomial Iterations

The IRA-iteration implicitly applies a polynomial iteration to a linear combination of Schur vectors spanning a wanted eigenspace of $H_{k+p}$. This section presents several results that serve to motivate the the exact shifting strategy introduced in § 4.2. The first theorem presented is a generalization of Lemma 3.10 proved by Sorensen [83]. The major difference is that there is no assumption on the existence of a basis of eigenvectors for the desired invariant subspace. Only a Schur basis is used.

**Theorem 4.4** Suppose $H \in \mathbf{R}^{m \times m}$ is an unreduced upper Hessenberg matrix corresponding to a length $m$ Arnoldi factorization $AV = VH + fe_m^T$

and that the eigenvalues of $H$ are in the disjoint partition

$$\{\theta_1, \ldots, \theta_k\} \cup \{\theta_{k+1}, \ldots, \theta_m\}.$$

Assume that the complex conjugate pairs of eigenvalues are kept together; $\theta_i = \bar{\theta}_j$ implies that $i, j \leq k$ or $i, j > k$.

If $m - k$ QR steps are performed with the shifts $\theta_{k+1}, \ldots, \theta_m$ producing an orthogonal matrix $Q \in \mathbf{R}^{m \times m}$ then

$$Q^T H Q = \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \tag{4.4.3}$$

where the eigenvalues of $H_{22}$ are $\theta_{k+1}, \ldots, \theta_m$.

Moreover, the updated starting vector produced by Algorithm 4.2, given an exact shifting strategy, is

$$V Q e_1 \in \mathcal{R}(V Q_1 Z_1), \tag{4.4.4}$$

where $Q_1 = Q[e_1, \ldots, e_k]$ and $H_{11} Z_1 = Z_1 T_1$ is a partial real Schur decomposition and

$$A(VQ_1) = (VQ_1) H_{11} + (e_m^T Q_1 e_k) f e_k^T, \tag{4.4.5}$$

is the updated Arnoldi factorization of length $k$.

**Proof** The matrix equation (4.4.3) is a direct result of Theorem 3.5.

Partition $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ where $HQ_1 = Q_1 H_{11}$. Let $H_{11} Z_1 = Z_1 T_1$ be a real Schur decomposition and it follows that

$$VQe_1 = VQ_1 e_1 = VQ_1 Z_1 Z_1^T e_1 \equiv VQ_1 Z_1 y,$$

where $y = Z_1^T e_1$. Partition the updated length $m$ Arnoldi factorization of the hypothesis as

$$AV \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} = V \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix} + f e_m^T \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}. \tag{4.4.6}$$

Equating the first $k$ columns of equation (4.4.6) results in

$$A(VQ_1) = (VQ_1) H_{11} + (e_m^T Q_1 e_k) f e_k^T,$$

since, by construction, $\left[ \begin{array}{cc} Q_1 & Q_2 \end{array} \right]$, is zero below that $(m-k)$-th sub-diagonal. $\square$

Using the exact shifting strategy during the IRA-iteration, replaces the starting vector with a linear combination of the wanted approximate Schur vectors. The ERA-iteration also has the same goal, but the IRA-iteration performs this replacement implicitly in a stable fashion using a Schur basis of $H$. In addition, the IRA-iteration avoids the need to re-start the next factorization from scratch. Note that as $m \to n$, Theorem 3.5 implies that the exact shifting strategy places improving approximations of the wanted eigenvalues in $H_{11}$ in a stable manner.

The restriction that keeps the complex conjugate pairs of eigenvalues together is only needed so that the iteration may be done in real arithmetic. The hypothesis of Theorem 4.4 concerning the disjoint partition of the eigenvalues of $H$ may be removed. A result by Miminis and Paige [53, pages 391–395], briefly mentioned in § 3.1, makes this hypothesis superfluous. They prove that if $m-k$ QR steps are performed then the matrix equation (4.4.3) results if and only if the $m-k$ shifts are eigenvalues of $H$, regardless of their multiplicity.

Algorithm 4.2 with the exact shift strategy, builds an orthogonal basis for a number of Krylov subspaces simultaneously. The following is a slight generalization of Theorem 3 proved by Morgan [54].

**Theorem 4.5** Assume the same hypothesis and notation as Theorem 4.4 with the additional hypothesis that $f \neq 0$. Suppose that $m-k$ QR steps are performed with the shifts $\theta_{k+1}, \ldots, \theta_m$. Let $M$ be a positive integer less than or equal to $n-m$ and greater than $k$. If

$$AV^+ \;=\; V^+H^+ + f^+e_M^T,$$

is the length $M$ Arnoldi factorization that results from extending the compressed factorization of equation (4.4.5) and $H^+$ is unreduced then

$$\mathcal{R}(V^+) \;=\; \mathrm{Span}\{VQ_1, Az_j, \ldots, A^{M-k}z_j\} \qquad (4.4.7)$$

holds for each Ritz vector $z_j = Vs_j$ such that $Hs_j = s_j\theta_j$ for $j = 1, \ldots, k$. In particular, if the eigenvectors $s_1, \ldots, s_k$ of $H$ are linearly independent, then

$$\mathcal{R}(V^+) \;=\; \mathrm{Span}\{z_1, \ldots, z_k, Az_i, \ldots, A^{M-k}z_i\}. \qquad (4.4.8)$$

**Proof** Partition the eigenvalues of $H$ as in the hypothesis of Theorem 4.4. Let $(s_j, \theta_j)$ be an eigenpair for $H$ where $\|s_j\| = 1$ and set $z_j = Vs_j$. Define $v_{m+1} \equiv f/\|f\|$ and $V^+ e_j \equiv v_j^+$ for $j = 1, \ldots, M$. Note, that by Theorem 2.4 of Chapter 2 it follows that $v_{k+1}^+ = \psi_k(A)v_1^+$ for some polynomial $\psi_k(\lambda)$ of degree $k$.

Note that by equation (4.4.5) of Theorem 4.4, we have $v_{k+1}^+ = v_{m+1}$. It also follows that $A^i v_{k+1}^+ = A^i \psi_k(A) v_1^+ \in \mathcal{K}_{k+i+1}(A, v_1^+)$ for $i = 1, \ldots, M-k-1$ which implies that

$$\text{(4.4.9)} \qquad \mathrm{Span}\{VQ_1, v_{k+1}^+, \ldots, A^{M-k-1}v_{k+1}^+\} \;\subset\; \mathcal{R}\{V^+\}.$$

We now show that these two sets share the same dimension. Suppose that $VQ_1y_1 + K_{M-k}(A, v_{k+1}^+)y_2 = 0$ for some $y \equiv [y_1^T, y_2^T]^T \in \mathbf{R}^M$. Thus, there exists a polynomial $\psi(\lambda)$ of degree less than $M$ so that $\psi(A)v_1^+ = 0$. However, since $H^+$ is unreduced the grade of $v_1^+$ is at least $M$ and hence $y \equiv 0$ which implies that the two sets in equation (4.4.9) are equal.

Using mathematical induction we show that

$$A^i z_j \;\in\; \mathrm{Span}\{z_j, v_{k+1}^+, \ldots, A^{i-1}v_{k+1}^+\},$$

for $i = 1, \ldots, M-k$. From the length $m$ Arnoldi factorization, it follows that

$$Az_j = z_j\theta_j + f(e_m^T s_j) = z_j\theta_j + v_{m+1}(e_m^T s_j)\|f\| \in \mathrm{Span}\{z_j, v_{k+1}^+\},$$

establishing the base case. Suppose that the result is true for positive integers $i-1$. The inductive hypothesis implies that

$$\begin{aligned} A^i z_j &\in AA^{i-1}z_j \\ &\in A\mathrm{Span}\{z_j, v_{k+1}^+, \ldots, A^{i-2}v_{k+1}^+\} \\ &\in \mathrm{Span}\{z_j, v_{k+1}^+, \ldots, A^{i-1}v_{k+1}^+\}, \end{aligned}$$

and the desired result follows. Now, since $z_j \in \mathcal{R}\{VQ_1\}$ and $v_{k+1}^+ = \psi_k(A)v_1^+$ it follows from the established equality of the two sets in equation (4.4.9) that

$$\text{(4.4.10)} \qquad \mathrm{Span}\{VQ_1, Az_j, \ldots, A^{M-k}z_j\} \;\subset\; \mathcal{R}\{V^+\}.$$

Using a similar argument as the one that followed equation (4.4.9), the two sets in equation (4.4.10) are equal. The first conclusion of the theorem in equation (4.4.7) is proved and the second one in equation (4.4.8) easily follows when the eigenvectors of $H$ are linearly independent. $\square$

The Krylov subspace of length $k+p$ generated during cycle of Algorithm 4.2 using exact shifts contains all the Krylov subspaces of dimension $p+1$ generated from a wanted Ritz vector:

$$\mathcal{K}_{p+1}(A, z_i^{(j)}) \quad \subset \quad \mathcal{K}_{k+p}(A, V_{k+p}^{(j+1)} e_1) \equiv \mathcal{R}(V_{k+p}^{(j+1)}),$$

corresponding to the $i$ wanted Ritz values $\theta_1^{(j)}, \ldots, \theta_k^{(j)}$. Morgan infers that the method builds an orthogonal basis for a Krylov subspace without favoring any particular Ritz vector.

The next result shows that the polynomial implicitly applied by an IRA-iteration using exact shifts is of minimal degree when we wish to re-start an Arnoldi factorization with a vector that is a linear combination wanted spectral information of $H_{k+p}^{(j)}$.

**Theorem 4.6** Assume the same hypothesis of Theorem 4.4 with the addition that the eigenvalues of $H$ are distinct. Let

$$\psi(\lambda) \quad = \quad \prod_{j=k+1}^{m} (\lambda - \theta_j)$$

and denote the Ritz vectors by $z_j = V s_j$ where $H s_j = s_j \theta_j$. If $\hat{v}_1 \in \text{Span}(z_1, \ldots, z_k)$ then for some polynomial $\phi(\lambda)$ of degree not exceeding $m-1$

$$\hat{v}_1 \quad = \quad \phi(A) v_1,$$

where $\phi(\lambda) = \psi(\lambda)\chi(\lambda)$ for some polynomial $\chi(\lambda)$ of degree at most $k-1$.

**Proof** Let $z_j \in \mathcal{K}_m(A, v_1)$. Then, for every $j$, there is polynomial $p_j(\lambda)$ of degree not exceeding $m-1$ such that $p_j(A)v_1$. Thus $\hat{v}_1 = \phi(A)v_1$ where the degree of $\phi(\lambda)$ does not exceed $m-1$. Using Lemma 4.1 it follows that $\hat{v}_1 = \phi(A)v_1 = \phi(A)Ve_1 = V\phi(H)e_1$. Expand $e_1 = s_1\xi_1 + \cdots + s_m\xi_m$ and hence $\phi(H)e_1 = s_1\phi(\theta_1)\xi_1 + \cdots + s_m\phi(\theta_m)\xi_m$. Since $\hat{v}_1 \in \text{Span}(z_1, \ldots, z_k)$ it follows that $\phi(\theta_j)\xi_j = 0$ for $j = k+1, \ldots, n$. Denote the left eigenvectors of $H$ by $u_j$ indexed so that $u_j^H H = u_j^H \theta_j$. Since the eigenvalues of $H$ are distinct, the biorthogonality of the left and right eigenvectors of $H$ gives that $u_j^H e_1 = u_j^H s_j \xi_j$ and $u_j^H s_j \neq 0$ for $j = 1, \ldots, m$. Lemma 2.1 of Chapter 2 implies that $u_j^H e_1 \neq 0$ and hence $\xi_j \neq 0$ and so $\phi(\theta_j) = 0$ for $j = 1, \ldots, m$. Thus $\psi(\lambda)$ must be a divisor of $\phi(\lambda)$ and the theorem is proved. $\square$

Theorem 4.4 implies that an IRA-iteration using the exact shift strategy builds a new Arnoldi factorization using only the wanted spectral information from a previous Arnoldi factorization. Theorem 4.6 states that any other re-started scheme that uses spectral information from an Arnoldi factorization introduces unwanted components if the degree of the polynomial is greater than $m-k$. From equation (4.4.5), an IRA-iteration with the exact shift strategy uses a linear combination of the first $m-k$ columns of $V$.

Further research on alternate shifting strategies is needed. In particular, the implicit application of the Chebyshev and Least squares filtering techniques of Saad [76, 77] should be investigated. Calvetti, Reichel, and Sorensen [17] have examined the use of Leja points during an implicitly re-started Lanczos iteration.

### 4.4.3 Explicitly Re-starting with Schur Vectors

Scott [80] presents an interesting version of a deflated ERA-iteration discussed at the end of § 4.1. Suppose the first $l-1$ columns of an Arnoldi factorization are approximate Schur vectors that satisfy the convergence criterion. At every cycle of the iteration, an Arnoldi factorization of length $k+p-l+1$ is built where the $l-1$ approximate Schur vectors occupy the leading portion of the factorization. For example, consider the $j$-th cycle of the iteration where $V_{k+p}^{(j)} = \left[ \begin{array}{cc} V_{l-1} & \bar{V}_{k+p-l+1}^{(j)} \end{array} \right]$ and $H_{k+p}^{(j)} = \left[ \begin{array}{cc} T_{l-1} & M_{l-1} \\ 0 & \bar{H}_{k+p-l+1}^{(j)} \end{array} \right]$ are generated. The leading portion of both $V_{k+p}^{(j)}$ and $H_{k+p}^{(j)}$ define an approximate partial real Schur decomposition of $A$. Let $\bar{H}_{k+p-l+1}^{(j)} \bar{Z}_{k+p-l+1}^{(j)} = \bar{Z}_{k+p-l+1}^{(j)} \bar{T}_{k+p-l+1}^{(j)}$ be a real Schur decomposition ordered so that the wanted eigenvalues are in leading portion of $\bar{T}_{k+p-l+1}^{(j)}$. When the first column of $\bar{V}_{k+p-l+1}^{(j)} \bar{Z}_{k+p-l+1}^{(j)}$ satisfies the convergence criterion, it is accepted as the $l$-th approximate Schur vector.

Scott's version differs from Algorithm 4.7 given below at Line 3.2. The real Schur decomposition computed by Scott is ordered with the eigenvalues in descending order of magnitude along the diagonal blocks of $T_{k+p-l+1}^{(j)}$. Scott also provides what appears to be robust implementions of almost all the major explicitly re-started Arnoldi variants; Block, Chebyshev acceleration, and pre-conditioned Arnoldi. The resulting software is available as the code `EB12` in the Harwell Subroutine Library [2]. The following procedure summarizes Scott's re-started single vector approach.

**Algorithm 4.7**
Input: An unit vector $v_1^{(1)}$.

1.1 For $l = 1, \ldots, k$

2.1 For $j = 1, 2, \ldots$ until convergence

3.1 Build an Arnoldi factorization of length $k+p-l+1$ given a starting vector $v_l^{(j)}$ in the $l$-th column of $V_{k+p}^{(j)}$ :
$AV_{k+p}^{(j)} = V_{k+p}^{(j)} H_{k+p}^{(j)} + f_{k+p}^{(j)} e_{k+p}^T$ ;

3.2 Compute the real Schur decomposition :
$\bar{H}_{k+p-l+1}^{(j)} \bar{Z}_{k+p-l+1}^{(j)} = \bar{Z}_{k+p-l+1}^{(j)} \bar{T}_{k+p-l+1}^{(j)}$
ordered so that the wanted eigenvalues are in leading portion of $\bar{T}_{k+p-l+1}^{(j)}$ ;

3.3 Set

$$\begin{aligned} Z_{k+p}^{(j)} &\leftarrow \begin{bmatrix} I_{l-1} & 0 \\ 0 & \bar{Z}_{k+p-l+1}^{(j)} \end{bmatrix}, \\ T_{k+p}^{(j)} &\leftarrow (Z_{k+p}^{(j)})^T H_{k+p}^{(j)} Z_{k+p}^{(j)}, \end{aligned}$$

3.4 Update the length $k+p$ Arnoldi factorization of Line 3.1 :
$AV_{k+p}^{(j)} Z_{k+p}^{(j)} = V_{k+p}^{(j)} Z_{k+p}^{(j)} T_{k+p}^{(j+1)} + f_{k+p}^{(j)} e_{k+p}^T Z_{k+p}^{(j)}$ ;

3.5 Obtain a length $l$ Arnoldi factorization by retaining only the first $l$ columns of the factorization in Line 3.4 :
$AV_l^{(j+1)} = V_l^{(j+1)} H_l^{(j+1)} + f_l^{(j+1)} e_l^T$ ;

3.6 If the $l$-th column of $V_l^{(j+1)}$ converges as a Ritz vector, increase $l$ by one and go to Line 2.3.

2.2 End For

2.3 If $l = k$ then stop.

1.2 End For

During each cycle of the iteration in Algorithm 4.7, the Arnoldi factorization is re-started with $\bar{V}_{k+p-l+1}^{(j)} \bar{Z}_{k+p-l+1} e_1$ while maintaining orthogonality against the approximate Schur vectors already computed. Equating the last $k+p-l+1$ columns of the length $k+p$ Arnoldi factorization of Line 3.1 results in

$$A\bar{V}_{k+p-l+1}^{(j)} = V_{l-1} M_{l-1} + \bar{V}_{k+p-l+1}^{(j)} \bar{H}_{k+p-l+1}^{(j)} + f_{k+p}^{(j)} e_{k+p-l+1}^T. \tag{4.4.11}$$

Using the orthogonality of the columns of $V_{k+p}^{(j)}$ gives that $M_{l-1} = V_{l-1}^T A \bar{V}_{k+p-l+1}^{(j)}$ and hence

$$(I - V_{l-1} V_{l-1}^T) A\bar{V}_{k+p-l+1}^{(j)} = \bar{V}_{k+p-l+1}^{(j)} \bar{H}_{k+p-l+1}^{(j)} + f_{k+p}^{(j)} e_{k+p-l+1}^T. \tag{4.4.12}$$

This prompts Saad [78, page 182] to make the observation that the Hessenberg matrix $\bar{H}^{(j)}_{k+p-l+1}$ of the deflated Arnoldi factorization of equation (4.4.11) appears at the front of the Arnoldi factorization applied to $(I - V_{l-1}V_{l-1}^T)A$. Thus,

$$\bar{V}^{(j)}_{k+p-l+1}\bar{Z}_{k+p-l+1}e_1 = K_{k+p-l+1}((I - V_{l-1}V_{l-1}^T)A, v_l^{(j)})c_{k+p-l+1},$$

where $v_l^{(j)} = \bar{V}^{(j)}_{k+p-l+1}e_1 = V^{(j)}_{k+p}e_l$, a polynomial of degree at most $k+p-l$ in $(I - V_{l-1}V_{l-1}^T)A$ is applied to the starting vector $V^{(j)}_{k+p}e_l$.

In theory, there is no difference between explicitly and implicitly re-starting an Arnoldi iteration. However, the numerical behavior of mathematically equivalent schemes may quite different. An example of this was given in § 4.3 comparing the ERA- and IRA-iterations. A more comprehensive numerical study comparing Algorithm 4.7 and Algorithm 4.2 is planned [48]. Another alternative is the work of Baglama, Calvetti and Reichel [4]. They discuss a deflated implicitly re-started Lanczos iteration using Leja shifts.

# Chapter 5

# Numerical Stability of an IRA-iteration

This chapter examines the particulars of computing an IRA-iteration in finite precision arithmetic. The underlying theme of this thesis is that QR- and IRA-iterations are one and the same. The chapter discusses the numerical stability of an IRA-iteration by appealing to that of the QR-iteration.

The concepts of the *backward* and *forward* stability of the QR algorithm are explained in § 5.1. The relevant perturbation theory associated with matrix eigenvalue problem is the subject of § 5.2. The forward instability of the QR algorithm is taken up in § 5.3. A connection is made with the algorithms used to re-order the Schur form of a matrix in § 5.4. The final section of the chapter presents a sensitivity analysis of orthogonal reductions of a matrix to upper Hessenberg form.

## 5.1 Backward and Forward Stability of the QR Algorithm

Robust implementations of a practical QR algorithm, such as those found in the software packages EISPACK [82] and LAPACK [1], compute a real Schur form for a matrix $A \in \mathbf{R}^{n\times n}$ such that

$$(A+E)Q_b \;=\; Q_b\hat{R}, \tag{5.1.1}$$

where $Q_b^T Q_b = I$ is exactly orthogonal and $\|E\| \approx \epsilon_M \|A\|$. The machine precision is denoted by $\epsilon_M$. The upper quasi-triangular matrix $\hat{R}$ is that computed by a robust implementation of the QR algorithm. The computed orthogonal matrix $\hat{Q}$ satisfies $\|\hat{Q}^T\hat{Q} - I\| \approx \epsilon_M$. In other words, the real Schur form of a matrix near $A$ is computed. This is what makes the QR algorithm backward stable.

Suppose that the same algorithm is computed in exact arithmetic. Let $AQ = QR$ denote this ideal computation. Assume that the ordering of the eigenvalues on the diagonal of $\hat{R}$ and $R$ is the same. We emphasize that it does not follow that

$$\|R - \hat{R}\| \;\approx\; \epsilon_M \|A\|.$$

Indeed, the diagonal elements of $R$ and $\hat{R}$ may have few if any digits of agreement. If, on the other hand, the ratio of the above norm difference and the norm of $A$ is on the order of machine precision, then the QR algorithm is forward stable.

In particular, consider one step of the shifted QR-iteration. Suppose $H$ is an unreduced upper Hessenberg matrix. As discussed above, the computed output results in

$$(H+E)\hat{Q} \;\approx\; \hat{Q}\hat{H}^{+},$$

where $\|\hat{Q}^T\hat{Q} - I\| \approx \epsilon_M$ and $\|E\| \approx \epsilon_M\|H\|$. Let $HQ = QH^{+}$ be the exact QR-step computed in exact arithmetic. Is it reasonable to expect that

$$\|H^{+} - \hat{H}^{+}\| \;\leq\; \epsilon_M\|H^{+}\| \;?$$

As we shall see, the shifted QR algorithm may be very sensitive to shift. Equivalently, orthogonal reductions to upper Hessenberg form may be very sensitive to tiny perturbations in the starting vector.

## 5.2 Perturbation Theory

This section briefly addresses the question that a perturbation theory answers: How does an eigenvalue and eigenvector change subject to changes in the matrix ? An understanding of these issues is important since it helps us determine the accuracy of the eigenvalue approximations computed.

The analysis of § 2.5 of Chapter 2 shows that when the product of the last component of a normalized eigenvector for $H_k$ and the norm of $\|f_k\|$ is suitably small, the IRA-iteration has computed an approximate eigenpair. If $H_k s = s\theta$ then $(A+E)x_r = x_r\theta$ with $E = -(e_k^T s)f_k x_r^H$. It follows that $\|E\| = |e_k^T s|\beta_{m+1}$, the size of the backward error, bounds the distance to the nearest matrix that has the Ritz pair $(x_r, \theta)$ as an eigenpair. The following theorem indicates what accuracy might be expected to an eigenvalue of $A$.

**Theorem 5.1** Suppose that $\lambda$ is an eigenvalue of $A$ nearest the eigenvalue $\theta$ of $A+E$. Denote the left and right eigenvectors for $\lambda$ by $y$ and $x$, respectively, each of unit length. Then

$$|\lambda - \theta| \;\leq\; \frac{\|E\|}{|y^H x|} + O(\|E\|^2)$$

**Proof** See page 68 of Wilkinson [101]. $\square$

The secant of the angle between $x$ and $y$, the reciprocal of $|y^H x|$, determines the conditioning of $\lambda$. If the left and right eigenvectors are nearly orthogonal, then even if $\|E\| \approx \epsilon_M \|A\|$, where $\epsilon_M$ is machine precision, $\theta$ may contain few digits, if any, of accuracy. Note that if $A$ is symmetric, then $x = y$ and $\theta$ is an excellent approximation to $\lambda$.

The question of how close the Ritz vector $x_r$ is to $x$ is complicated by the fact that an eigenvector is not an unique quantity. Any scaling of an eigenvector by a complex number of unit modulus remains one.

> **Theorem 5.2** Suppose that $AQ = Q \begin{bmatrix} \lambda & r_{12}^T \\ 0 & R_{22} \end{bmatrix}$ is a Schur form for $A$ and let $\lambda$ be the eigenvalue of $A$ nearest the eigenvalue $\theta$ of $A + E$. If $\varphi$ measures the positive angle between $x$ and $x_r$ then
>
> $$\varphi \leq \frac{2\|E\|}{\mathrm{sep}(\lambda, R_{22})} + O(\|E\|_F^2).$$

**Proof** See Lemma 7.8 of Demmel [23]. $\square$

Varah [94] shows that

$$\begin{aligned} \mathrm{sep}(\lambda, R_{22}) &\leq \min_{\lambda_i \neq \lambda} |\lambda - \lambda_i|, \\ \mathrm{sep}(\lambda, R_{22}) &\leq \|r_{12}\| \frac{|y^H x|}{\sqrt{1 - |y^H x|^2}}, \end{aligned} \tag{5.2.1}$$

where the latter bound is only defined for nonzero $r_{12}$. Thus, the conditioning of the eigenvector problem depends upon both the distance to the other eigenvalues of $A$ and the sensitivity of $\lambda$. Varah also notes that both upper bounds may be significant over estimates. Note that when $A$ is symmetric, $r_{12} = 0$ and it may be shown that the first bound is an equality. The conclusion we must draw is that the computation of the eigenvalues for a nonsymmetric matrix is potentially an ill conditioned process.

Multiple and clusters of eigenvalues cause further complications and the answer is to study the conditioning of invariant subspaces. In fact, if the angle between the left and right eigenvector approaches ninety degrees, then equation (5.2.1) implies that $\lambda$ is not a distinct eigenvalue of $A$. The same result is essentially proved by Wilkinson [102]. He shows that if $\lambda$ is a distinct eigenvalue of $A$, then a perturbation matrix $F$ exists so that $\lambda$ is a repeated eigenvalue of $A + F$ and $\|F\| \leq |y^H x|/(\sqrt{1 - |y^H x|^2})$. If $|y^H x|$ is equal to zero then $\lambda$ is a repeated eigenvalue of $A$.

Saad [78] presents an excellent comprehensive introduction to perturbation theory within the context of large scale eigenvalue problems. The works of Chatelin [18], Stewart and Sun [90] are sources for more general study with many citations to the literature. In particular, the work of Bai, Demmel, and McKinney [7] examines the construction of the LAPACK software used to estimate the various condition numbers.

Finally, the possible ill conditioning of the nonsymmetric eigenvalue problem leads Toh and Trefethen to suggest that the Arnoldi iteration be used to estimate the *pseudospectra* of a matrix [93]. The eigenvalues of $A+E$ where $\|E\| \leq \epsilon$ are members of $A$'s pseudospectra.

## 5.3 Forward Instability of the QR Algorithm

This section investigates how the theory of Chapter 2 behaves when computing in floating point arithmetic. By understanding what causes the forward instability of the QR algorithm, we may possibly prevent its deleterious effects. These include:

- introducing perturbations that lead to unnecessary loss of accuracy in the computed spectral information.
- Increasing the number of iterations required for convergence.

Since the last two chapters demonstrate that the IRA-iteration is equivalent to the QR-iteration, we are directly led to an understanding of the effect applying shifts during a cycle of Algorithm 4.2.

Parlett and Le [63] carefully examine the forward instability of the QR algorithm on symmetric tridiagonal matrices. However, we shall see that their results appear to carry over directly to the QR algorithm on upper Hessenberg matrices. The analysis and numerical experiments suggest a sensitivity analysis for the orthogonal reduction of a matrix to upper Hessenberg form.

Suppose, for the moment, that $H \in \mathbf{R}^{n\times n}$ is an unreduced symmetric tridiagonal matrix and set $H^{(1)} = RQ + \tau I$ where $QR = H - \tau I$ is a QR factorization. Denote by $H_k$ the leading principal matrix of order $k$ of $H \equiv H_n$. The main result proved by Parlett and Le is a necessary and sufficient condition for the onset of forward instability. The instability occurs if and only if the shift $\tau$ is close to an eigenvalue of $H_k$ with a small last component of the corresponding normalized eigenvector. Parlett and Le present numerous examples illustrating the forward instability. Before continuing, we present three examples for the nonsymmetric problem that serve to motivate these ideas.

Consider the matrix

$$H = \begin{bmatrix} 3 & 1 & 1 & -1 \\ 1 & 3 & -1 & 1 \\ 0 & 10^{-12} & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}. \tag{5.3.1}$$

Table 5.1 displays spectral information of $H$; the notation $\omega_{i,n}$ stands for the last component of the $i$-th normalized eigenvector of $H$.

Suppose that two separate explicit QR steps are performed on $H$ with shifts 3 and 4. Computing in MATLAB, Version 4.2a, on a SUN SPARC station IPX results in

$$\hat{H}(3) \approx \begin{bmatrix} 3 & -1 & -1.4 & -1.1 \cdot 10^{-16} \\ -1 & 3 & -1.4 & -7.1 \cdot 10^{-13} \\ 0 & 1 & 1 & -10^{-12} \\ 0 & 0 & 0 & 3 \end{bmatrix}. \tag{5.3.2}$$

and

$$\hat{H}(4) \approx \begin{bmatrix} 2 & -1.4 & 1.4 & -3.2 \cdot 10^{-4} \\ 7.1 \cdot 10^{-13} & 2 & 1 & -7.1 \cdot 10^{-13} \\ 0 & 1 & 2 & 6.7 \cdot 10^{-4} \\ 0 & 0 & 6.7 \cdot 10^{-4} & 3.9 \end{bmatrix}, \tag{5.3.3}$$

where $\hat{H}(\tau) \equiv R(\tau)Q(\tau) + \tau I$. The floating point arithmetic is IEEE standard double precision with machine precision of $\epsilon_M \equiv 2^{-52} \approx 2.2204 \cdot 10^{-16}$. The results of equation (5.3.3) are in stark contrast to Lemma 3.1 of Chapter 3 where as those of equation 5.3.2 conform. The last property of Lemma 3.1 implies that for shifts that

| $i$ | Eigenvalue | Condition number | $\omega_{i,4}$ |
|---|---|---|---|
| 1 | 4 | 1.2 | $O(10^{-13})$ |
| 2 | 1.999999999999 | 1.2 | $O(10^{-13})$ |
| 3 | 1.000000000001 | 1.5 | $O(10^{-1})$ |
| 4 | 3 | 2.8 | $O(10^{-1})$ |

**Table 5.1** Eigenvalues and some sensitivity measures for $H$.

are nearly eigenvalues of $H$, the last row of $e_4^T(R(\tau)Q(\tau)+\tau I) \approx \lambda e_4^T$, where $\lambda$ is an eigenvalue of $H$. We note that the eigenvalues of both matrices are still equal to those of Table 5.1.

Let $(s_i^{(k)}, \lambda_i^{(k)})$ be an eigenpair for $H_k$, the leading principal sub-matrix of order $j$ of $H$, and let $\omega_{i,k} \equiv e_k^T s_i^{(k)}$ be the last component of the corresponding eigenvector. Assume that $s_i^{(k)}$ is a unit vector for $k = 1, \ldots, n$. Parlett and Le's analysis formally extended for an unreduced Hessenberg matrix states that there are entries of $\hat{H}(\tau) = R(\tau)Q(\tau) + \tau I$ whose derivatives are $O(1/\omega_{i,k})$ with respect to changes in $\tau$ when $\tau$ is nearly equal to $\lambda_i^{(k)}$. This analysis is corroborated when $\tau = 4$ since it is an eigenvalue of $H_2$ and $\omega_{i,k} \approx 10^{-13}$. The last sub-diagonal of $\hat{H}$ should be on the order of machine precision; however $|\hat{\beta}_4| \approx 10^{13}\epsilon_M$. Parlett and Le also observe that a small $\omega_{i,k}$ is an indicator that the first $k$ columns of $H - \lambda_i^{(k)} I$ are almost linearly dependent. Since $H - \lambda_i^{(k)} I = QR$, it follows that the condition number of $H_j - \lambda_i^{(k)} I$ is that of $R_j$ where $R_j$ is the leading principal sub-matrix of order $j$ of $R$. The condition numbers $\kappa(H_j - \tau I_j) = \|H_j - \tau I_j\| \|(H_j - \tau I_j)^{-1}\|$ are displayed in Table 5.2. We believe this geometric interpretation predicting forward instability should immediately come to mind when considering the size of $\omega_{i,k}$.

It is instructive to consider performing a QR step on $H$ with an implicitly shifted variant of the QR algorithm. Let $\check{H}(\tau)$ be the computed result of performing the QR step implicitly with shift $\tau$:

$$(5.3.4) \qquad \check{H}(4) \approx \begin{bmatrix} 2 & -1.4 & 1.4 & 0 \\ 7.1 \cdot 10^{-13} & 2 & 1 & -7.1 \cdot 10^{-13} \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}.$$

Performing the step implicitly prevents the forward instability in this example.

| j | $\kappa(H_j - 4I_j)$ | $\kappa(H_j - 3I_j)$ |
|---|---|---|
| 1 | 1 | 1 |
| 2 | $O(10^{12})$ | $O(1)$ |
| 3 | $O(10^{12})$ | $O(1)$ |
| 4 | $O(10^{16})$ | $+\infty$ |

**Table 5.2** Condition numbers for the shifted matrices.

| $i$ | Eigenvalue | Condition number | $\omega_{i,4}$ |
|---|---|---|---|
| 1 | 2.999999999999999 | $O(10^2)$ | $O(10^{-1})$ |
| 2 | .9999990001706701 | $O(10^6)$ | $O(10^{-7})$ |
| 3 | 1.00000099982933 | $O(10^6)$ | $O(10^{-7})$ |
| 4 | 3 | $O(10^2)$ | $O(10^{-1})$ |

**Table 5.3** Eigenvalues and and some sensitivity measures for $G$.

| j | $\kappa(G_j - I_j)$ |
|---|---|
| 1 | 1 |
| 2 | $+\infty$ |
| 3 | $O(10^{12})$ |
| 4 | $O(10^{12})$ |

**Table 5.4** Condition numbers for the shifted matrices.

Our second example shows that both explicit and implicit implementations are both sensitive to the shift used: Let

$$G = \begin{bmatrix} 2 & 1 & -1 & 1 \\ 1 & 2 & 1 & -1 \\ 0 & 10^{-12} & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \tag{5.3.5}$$

with spectral information given by Table 5.3.

If Wilkinson's shift is used, then we obtain

$$\hat{G}(1) \approx \begin{bmatrix} 3 & 0 & 10^{-16} & 0 \\ 7.1 \cdot 10^{-13} & 2 & .58 & .82 \\ 0 & 1.7 & .67 & -.47 \\ 0 & 0 & .94 & 2.3 \end{bmatrix}, \tag{5.3.6}$$

and

$$\check{G}(1) \approx \begin{bmatrix} 3 & 0 & 0 & 0 \\ 7.1 \cdot 10^{-13} & 2 & .58 & .82 \\ 0 & 1.7 & .67 & -.47 \\ 0 & 0 & .94 & 2.3 \end{bmatrix}. \tag{5.3.7}$$

Although Wilkinson's shift shares seven digits of accuracy with two of the eigenvalues of $G$, the last sub-diagonal elements of both $\hat{G}$ and $\check{G}$ are order unity. This is predicted by Parlett and Le's analysis since $10^{-7} \cdot \omega_{2,4}^{-1} = 1 \approx .94$. The condition numbers of the eigenvalues measure the possible loss of accuracy subject to changes in the matrix elements. Since the orthogonal matrices effecting the explicit and implicit QR steps are only numerically so, perturbations are introduced. Sorting the computed eigenvalues of $\hat{G}$ and $\check{G}$ into ascending order gives for $i = 1, 2$

$$\frac{\|\hat{G} - \check{G}\|}{10^6} = O(10^{-10}) \approx |\lambda_i(\hat{G}) - \lambda_i(\check{G})| = O(10^{-11}),$$

where $\|\hat{G} - \check{G}\| \approx 10^{-4}$. In words, the accuracy of the computed eigenvalues is essentially the ratio of the norm difference of the two matrices produced by the implicit and explicit QR-iterations and the condition number of the eigenvalue. Table 5.4 gives an alternative measure for the amount of forward instability that the QR algorithm may undergo.

The third and final example shows that small sub-diagonal entries are not needed for the QR algorithm to undergo forward instability. Let

$$F = \begin{bmatrix} 200 & 100 & 0 & 1 \\ 100 & 200 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \tag{5.3.8}$$

with spectral information given by Table 5.5. We also add that the matrix of eigenvectors for $F$ has condition number $O(1)$. Computing an implicit QR step with shift

| $i$ | Eigenvalue | Condition number | $\omega_{i,4}$ |
|---|---|---|---|
| 1 | 300.0000056304403 | $O(1)$ | $O(10^{-6})$ |
| 2 | 99.9994793289591 | $O(1)$ | $O(10^{-5})$ |
| 3 | 3.001734106345232 | $O(1)$ | $O(10^{-1})$ |
| 4 | .9983123303188788 | $O(1)$ | $O(10^{-1})$ |

**Table 5.5** Eigenvalues and and some sensitivity measures for $F$.

100 leads to

$$(5.3.9) \qquad \check{F}(100) \approx \begin{bmatrix} 300 & 0 & .7071 & 0 \\ .7071 & 2 & 1 & -.7071 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & .7071 & 100 \end{bmatrix}.$$

Although the relative error of the shift 100 with respect to the nearest eigenvalue of $F$ is $O(10^{-7})$, the last sub-diagonal element of $\check{F}(100)$ is $O(10^{-1})$. Since the eigenvalue (and eigenvector) problem for $F$ are extremely well conditioned, shifting with the numerically exact shift $\hat{\lambda}_2 = 99.9994793289591$ given in Table 5.5 should result in an $O(\epsilon_M)$ term in the last sub-diagonal entry of $\check{F}(\hat{\lambda}_2)$. Instead,

$$(5.3.10) \qquad \check{F}(\hat{\lambda}_2) \approx \begin{bmatrix} 300 & 3.8 \cdot 10^{-9} & .7071 & .0051 \\ .7071 & 2.0001 & 1.0051 & -.7071 \\ 0 & 1 & 2 & -.7071 \\ 0 & 0 & 6.6 \cdot 10^{-10} & \check{\zeta} \end{bmatrix},$$

is computed where $\check{\zeta} = .9999994793290061$. Note that the relative error in $\check{\zeta}$ to $\hat{\lambda}_2$ is $O(10^{-14})$ but that an order $O(10^{-10})$ element emerges in the last sub-diagonal entry. Once again, the sensitivity is measured by the reciprocal of $\omega_{2,4}$ since $\epsilon_M \omega_{2,4}^{-1} = O(10^{-10})$.

In a study examining the deterioration of forward stability during an implicit QR step, Watkins [99] investigates the transmission of the shift through the matrix. Watkins' analysis also shows that small sub-diagonal elements are not reliable indicators for predicting the loss of forward stability. This is substantiated by the previous examples. It is also shown that even when the QR step does undergo forward instability, the shift still manages to get propagated through the entire matrix. The only manner in which a shift can fail to be transmitted is when it is small and the entries in the leading portion of the matrix are large. Stewart observed this phenomenon for the QR algorithm on symmetric tridiagonal matrices [84].

### 5.3.1 Premature Deflation

Parlett and Le showed that if forward instability occurs during an implicit QR step, it is preceded by *premature deflation*. Before defining premature deflation, we review some necessary details concerning an implicit QR step. An implicit QR step with a real shift is calculated by forming $(U_1 \cdots U_{n-1})^T H U_1 \cdots U_{n-1}$ where each $U_i$ is an orthogonal matrix. The orthogonal matrices most commonly used are plane, or Givens', rotations. The first rotation is constructed so that $U_1^T(H - \tau I)e_1 = e_1\sqrt{(\alpha_1 - \tau)^2 + \beta_2^2}$. The similarity transformation $U_1^T H U_1$ introduces a nonzero entry, or bulge, in the $(3,1)$ entry. The remaining plane rotations chase the bulge successively down the sub-diagonal.

Suppose that the following $3 \times 3$ sub-matrix of $(U_1 \cdots U_i)^T H U_1 \cdots U_i$ arises:

| | column i | column i+1 | column i+2 |
|---|---|---|---|
| row i | $\hat{\alpha}_i$ | x | x |
| row i+1 | $\epsilon_1$ | $\hat{\tau}$ | x |
| row i+2 | $\hat{\beta}$ | $\epsilon_2$ | $\alpha_{i+2}$. |

If both $\epsilon_1$ and $\epsilon_2$ are small and $\hat{\tau}$ is nearly equal to the shift $\tau$ used, then premature deflation has occurred. Watkins' shows that the entries marked by an "x" and $\hat{\beta}$ are not relevant to the analysis. As an example, the sequence of intermediate matrices computed during the QR step with Wilkinson's shift that results in $\check{G}(1)$ undergoes premature deflation. Starting with the first Givens' rotation designed to annihilate the $(2,1)$ entry, the sequence is

$$U_1^T G U_1 = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 1.4 & -1.4 \\ 7.1 \cdot 10^{-13} & 7.1 \cdot 10^{-13} & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix},$$

$$(U_1 U_2)^T G U_1 U_2 = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 7.1 \cdot 10^{-13} & 2 & -7.1 \cdot 10^{-13} & 1 \\ 0 & -1.4 & 1 & 1.4 \\ 0 & 1 & 0 & 2 \end{bmatrix},$$

and finally $(U_1 U_2 U_3)^T G U_1 U_2 U_3 = \check{G}(1)$. Notice that for $U_1^T G U_1$ the $(2,1)$ entry is zeroed out, the $(3,2)$ entry is small and the shift emerges in the $(2,2)$ position. This is premature deflation. Parlett and Le's analysis shows that premature deflation is necessary for the implicitly shifted QR algorithm on symmetric tridiagonal matrices to undergo forward instability. Watkins demonstrates that along with premature deflation, certain sub-diagonal entries must undergo a significant reduction in size after the QR step. This is evident in the above example since $e_2^T G e_1 / e_2^T \check{G}(1) e_1 = O(10^{13})$. It is shown that the only way that a sub-diagonal element becomes tiny is through a cancelation error.

## 5.4 Re-ordering the Real Schur Form of a Matrix

Suppose that the upper Hessenberg matrix $H \equiv H_{k+p}^{(j)}$ computed during a cycle of an IRA-iteration is reduced to upper quasi-triangular form by the QR algorithm:

$$
\begin{aligned}
Q^T H Q &= R, \\
&\equiv \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},
\end{aligned} \tag{5.4.1}
$$

where $Q$ is the orthogonal matrix computed by the algorithm. Equation (5.4.1) is a real Schur form for $H$ of order $k+p$ where the sub-matrices $R_{11}$ and $R_{22}$ are of order $k$ and $p$, respectively. Assume that the spectrums of $R_{11}$ and $R_{22}$ are distinct. In practice, the order in which the computed eigenvalues of $H$ appear on the diagonal of $R$ depends upon the shifts applied. Two algorithms for re-ordering the real Schur form of a matrix, an iterative and direct variant, were presented in § 3.4.4 of Chapter 3.

The iterative swapping algorithm is equivalent to the implicit re-starting technique used by the IRA-iteration since both depend upon an implicitly shifted QR step applied to an unreduced upper Hessenberg matrix to interchange $R_{11}$ and $R_{22}$. The direct swapping algorithm is equivalent to a deflation technique, *locking*, presented in Chapter 6. An orthogonal matrix is constructed from a basis for the invariant subspace corresponding to $R_{22}$. When this is applied as a similarity transformation the diagonal blocks of $R$ are swapped. In exact arithmetic, both swapping variants result in a matrix that is upper quasi-triangular with the blocks interchanged.

The following example demonstrates that the two variants may produce drastically different output matrices when computed in floating point arithmetic. We compute under the same conditions as in the last section. Let

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

An eigenvector corresponding to $\lambda_2 = 1$ is $\begin{bmatrix} -1 \\ 10\epsilon_M \end{bmatrix}$. Denote by $Z$ the plane rotation that transforms this eigenvector to a multiple of the first column of the identity matrix in $\mathbf{R}^{2\times 2}$. Let

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

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

$$
\begin{aligned}
Z^TTZ &= \begin{bmatrix} 1 & -1 \\ 0 & 1+10\epsilon_M \end{bmatrix}, \\
\hat{T} &= \begin{bmatrix} 1.400000000000003 & -7.999999999999996 \cdot 10^{-1} \\ 2.000000000000002 \cdot 10^{-1} & 6.000000000000001 \cdot 10^{-1} \end{bmatrix}.
\end{aligned}
$$

The computed eigenvalues of $\hat{T}$ are

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

which both lost eight digits of accuracy. If another QR-step is performed on the matrix $\hat{T}$ with the same shift, $\begin{bmatrix} 1.000000000000003 & 1.000000000000001 \\ \approx 1.09 \cdot 10^{-15} & 1 \end{bmatrix}$ is computed. Note that the off-diagonal element is slightly larger than machine precision so that a standard QR algorithm does not set it to zero. But even if the off-diagonal element is set to zero, the iterative swapping algorithm fails to interchange the eigenvalues. Continuing to apply QR-steps with the shift equal to $\lambda_1$ does not result in a properly interchanged matrix.

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

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

Thus the small relative errors on the order of machine precision that occur when computing $U^TTU$ produce a nearby matrix in which both the eigenvalues differ by eight digits of accuracy. Performing a shifted QR step with $\lambda_1$ incurs forward instability since the last components of the eigenvectors for $U^TTU$ are on the order of $\sqrt{\epsilon_M}$. This is the necessary and sufficient condition of Parlett and Le [63]. Another QR step with the same shift on $\hat{T}$ almost zeros out the sub-diagonal element since the last components of the eigenvectors for $\hat{T}$ are order $10^{-1}$ and the shift is almost the average of the eigenvalues of $\hat{T}$ and quite close to both. We emphasize that the loss of accuracy of the computed eigenvalues is one of the deleterious effects of forward instability.

Bai and Demmel [9] present an example which compares their direct swapping approach with Stewart's algorithm EXCHNG. The matrix considered is

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

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

Bai and Demmel presents a rigorous analysis of their direct swapping algorithm. Although backward stability is not guaranteed, it appears that only when both $T_{11}$ and $T_{22}$ are both of order two and have almost indistinguishable eigenvalues [15] is stability lost. In this case, the interchange is not performed. Bojanczyk and Van Dooren [15] present an alternate swapping algorithm that appears to be backward stable.

## 5.5 Implications for an IRA-Iteration

A robust implementation of an IRA-iteration relies upon the proper transmission of shifts during the implicit application shift application. The discussion that followed Algorithm 4.2 used the convergence theory for the QR-iteration developed in § 3.2 to conclude that all the sub-diagonal elements of $H_k^{(j+1)}$, not including those corresponding to complex conjugate pairs, go to zero if the polynomial min-max problem (3.2.1) of Chapter 3 is approximately solved. In particular, if an exact shift strategy is used for Algorithm 4.2 in Chapter 4, Theorem 4.4 implies that the sub-diagonal entry $\beta_k^{(j+1)}$ is zeroed out during the $j$-th iteration. However, as the examples in § 5.3– 5.4 demonstrate, $\beta_k^{(j+1)}$ may not even be small, let alone negligible.

The theory reviewed and developed in the first three chapters of this thesis present an analysis of what occurs in exact arithmetic. Computing in finite precision arithmetic, however, complicates the situation. The phenomenon of the forward instability of the QR algorithm examined in the last two sections could have a possibly detrimental effect upon the accuracy in the computed eigenvalues. Since the IRA-iteration is a truncation of the implicitly shifted QR algorithm, it also is susceptible to loss of accuracy through forward instability. This indicates that it may be impossible to filter out unwanted Ritz values with the implicit re-starting technique in practical computations. This is the motivation for developing the deflation techniques of Chapter 7. In particular, using a converged Ritz value as a shift may incur forward instability. Since the norm of $f_k^{(j+1)}$ is the sub-diagonal entry $\beta_{k+1}^{(j+1)}$, forward instability may prevent the residual vectors of the successive Arnoldi factorizations from ever approaching zero.

For example, consider the following thought experiment. Suppose that the exact shift strategy is used for Algorithm 4.2 and $p > 1$ shifts are to be applied. According to Theorem 4.4, the computed $k$-th sub-diagonal entry $\hat{\beta}_{k+1}$ should be zero. Computing in floating point arithmetic, though, gives that all we may expect is that the computed $k$-th sub-diagonal entry $\hat{\beta}_{k+1}$ be on the order of $\epsilon_M$ relative to the norm of the matrix. However, the forward instability of the QR algorithm may prevent the computed $k$-th sub-diagonal entry $\hat{\beta}_{k+1}$ from becoming small. Application of the first shift possibly introduces perturbations so that the remaining shifts are no longer eigenvalues of the updated matrix. Thus, further QR steps may not lead to a negligible $\hat{\beta}_{k+1}$ after $p$ implicit shifts. The examples of the previous section illustrate this behavior. The possible ill conditioning of the nonsymmetric eigenvalue problem also exacerbates the situation since inaccurate eigenvalues may result from the computed errors in the matrix elements due to forward instability. An obvious, but expensive solution, is to recompute the eigenvalues of the deflated matrix after every implicit shift application.

## 5.6 The Sensitivity of the Hessenberg Decomposition

Theorem 2.5 of Chapter 2 determines conditions for a length $k$ truncated Arnoldi factorization. The following geometric result indicates the dependence of the residual vector upon the starting one used during the Hessenberg decomposition. Simply stated, if the starting vector for an Arnoldi factorization, or any other orthogonal reduction to Hessenberg form, is nearly in an invariant subspace for $A$ of dimension $m$, the residual vector associated with the length $m$ Arnoldi factorization may not be small as exact arithmetic leads us to expect.

**Theorem 5.3** Let $A \in \mathbf{R}^{n\times n}$. Suppose that $AQ_m = Q_m T_m$ is a real partial Schur factorization of order $m$, and that $AV_m = V_m H_m$ is a length $m$ Arnoldi factorization where $H_m$ is unreduced, and that $v_1 = V_m e_1 = Q_m y$, and let $K_m(A, v_1)c_m = A^m v_1$. If $\tau\hat{v}_1 = v_1 + w$ is an unit vector with $Q_m^T w = 0$ such that $A\hat{V}_j = \hat{V}_j \hat{H}_j + \hat{f}_j e_j^T$ is the corresponding Arnoldi factorization with $\hat{V}_j e_1 = \tau\hat{v}_1$, and

$$\epsilon \;=\; \max\{\frac{\|K_m(A,w)\|}{\|K_m(A,v_1)\|}, \frac{\|A^m w\|}{\|A^m v_1\|}\} \;<\; 1,$$

then

$$\hat{\rho}_m \hat{\beta}_{m+1} \;\leq\; \{1 + 2\kappa_2(K_m(A, v_1))\}\|A^m v_1\|\epsilon + O(\epsilon^2), \tag{5.6.1}$$

where $\hat{\rho}_m = \hat{\beta}_2 \cdots \hat{\beta}_m$.

**Proof** Suppose that $AV_j = V_j H_j + f_j e_j^T$ is an Arnoldi factorization with $v_1 = V_m e_1 = Q_m y$ where $AQ_m = Q_m T_m$ is a real partial Schur factorization of order $m$. Let $\tau\hat{v}_1 = v_1 + w$ be an unit vector such that $Q_m^T w = 0$, and $A\hat{V}_j = \hat{V}_j \hat{H}_j + \hat{f}_j e_j^T$ is the corresponding Arnoldi factorization with $\hat{V}_j e_1 = \tau\hat{v}_1$.

Using Ruhe's characterization of the Hessenberg decomposition in equation (2.4.3) of Chapter 2 it follows that

$$\begin{aligned}
\|A^j v_1 - K_j(A, v_1)c_j\| &= \min_{c\in\mathbf{R}^j} \|A^j v_1 - K_j(A, v_1)c\|, \\
&\equiv \|r_j\|, \\
\|A^j \tau\hat{v}_1 - K_j(A, \tau\hat{v}_1)\hat{c}_j\| &= \min_{c\in\mathbf{R}^j} \|A^j \tau\hat{v}_1 - K_j(A, \tau\hat{v}_1)c\|, \\
&\equiv \|\hat{r}_j\|.
\end{aligned}$$

But,

$$\|\hat{r}_j\| \;=\; \|A^j(v_1 + w) - \{K_j(A, v_1) + K_j(A, w)\}\hat{c}_j\|. \tag{5.6.2}$$

Standard results [35, page 228] on the sensitivity of the least squares problem give

$$\|r_j - \hat{r}_j\| \;\leq\; \|A^j v_1\|\{1 + 2\kappa_2(K_j(A, v_1))\}\epsilon + O(\epsilon^2).$$

In particular, when $j = m$ it follows that $r_m = 0$. Theorem 2.3 implies that the QR factorizations of $K_m(A, v_1)$ and $K_m(A, \tau\hat{v}_1)$ are $V_m R_m$ and $\hat{V}_m \hat{R}_m$, respectively, where both $R_m$ and $\hat{R}_m$ are nonsingular upper triangular matrices of order $m$.

Equation (2.4.4) gives $\|\hat{r}_m\| = \hat{\rho}_m \|\hat{f}_m\| = \hat{\rho}_m \hat{\beta}_{m+1}$, where $\hat{\rho}_m = e_m^T \hat{R}_m e_m$. The proof of Theorem 2.3 computes the equality $e_i^T \hat{R}_m e_i = \hat{\beta}_2 \cdots \hat{\beta}_i$ for $i = 2, \ldots, m$. $\square$

The sensitivity of the product of the sub-diagonal elements of the perturbed Arnoldi factorization depends linearly on $\kappa_2(K_m(A, v_1))$. Since $\kappa_2(K_{j-1}(A, v_1)) \leq \kappa_2(K_j(A, v_1))$, the theorem argues against building large factorizations. Also note that $\|A^m v_1\| = \|A^m Q_m y\| = \|T_m^m y\|$.

Suppose the solution to the perturbed least squares problem is $\hat{c}_j \equiv c_j + \delta c_j$. When $j = m$, equation (5.6.2) of the proof leads to

$$\hat{\beta}_2 \cdots \hat{\beta}_{m+1} = \|\hat{r}_m\| = \|A^m w - K_m(A, v_1)\delta c_m + K_m(A, w)c_m\|,$$

where second-order terms are ignored. It is this combination of vectors that is responsible for the possible amplification of the perturbation.

There is an interesting connection between Arnoldi factorizations and moment matrices that gives a lower bound on the product $\rho_m = \beta_2 \cdots \beta_m$. Nachtigal [55, page 36] discusses a similar connection between moment matrices and the nonsymmetric Lanczos process. Since $K_m$ is of full column rank, $K_m^T K_m$ is a positive definite symmetric matrix. By Theorem 2.3 of Chapter 2,

$$I_m = V_m^T V_m = R_m^T K_m^T K_m R_m,$$

where $K_m \equiv K_m(A, v_1)$ results in

$$R_m^{-T} R_m^{-1} \quad = \quad K_m^T K_m \equiv M_m.$$

Defining $L_m \equiv R_m^{-T}$, the Cholesky factorization $M_m = L_m L_m^T$ is determined by the inverse of the Fourier coefficient matrix $R_m$. Since the $i$-th sub-diagonal element of $H_m$ is $\beta_i$ for $i = 2, \ldots, m$ define $\beta_1 \equiv 1 (= \|v_1\|)$. Thus, the reciprocal of the product $\beta_1 \cdots \beta_i$ is the $i$-th pivot used during the Cholesky factorization of the moment matrix $M_m$. A standard result [35, page 145] on the numerical stability of the Cholesky factorization implies that

$$e_i^T R_m^{-1} e_j \leq \sqrt{e_j^T M_m e_j} = \sqrt{v_1^T (A^{(j-1)})^T A^{(j-1)} v_1},$$

and hence, $(v_1^T (A^{(j-1)})^T A^{(j-1)} v_1)^{-1/2} \leq \beta_2 \cdots \beta_m$. Note that $e_{m+1}^T R_{m+1} e_{m+1} = 0$ since $A^m v_1$ is a linear combination of the columns of $K_m(A, v_1)$. Hence the Cholesky factorization of $K_{m+1}$ is not defined since the diagonal element $e_{m+1}^T L_{m+1} e_{m+1}$ does not exist.

Let $\hat{K}_{m+1}^T \hat{K}_{m+1} = \hat{R}_{m+1}^{-T} \hat{R}_{m+1}^{-1}$ be the Cholesky factorization of the perturbed Krylov matrix $K_{m+1}(A, \tau\hat{v}_1)$ using the notation of Theorem 5.6. Since $e_i^T \hat{R}_{m+1}^{-1} e_i$ is just the reciprocal of $e_i^T \hat{R}_{m+1} e_i$ for $i = 1, \ldots, m+1$, the implication is that if $(\tau\hat{v}_1)^T A^{2m} \tau\hat{v}_1$ is not large then $\hat{\beta}_2 \cdots \hat{\beta}_{m+1}$ is not small. Since $\tau\hat{v}_1 = v_1 + w$, it follows that $(\tau\hat{v}_1)^T A^{2m} \tau\hat{v}_1$ will be not be large when the $e_{m+1}^T \hat{R}_{m+1}^{-1} e_{m+1}$ is not small—precisely the situation that indicates that forward instability occurred during the orthogonal reduction of $A$ to upper Hessenberg form.

Finally, we remark that the sensitivity of a Hessenberg decomposition via orthogonal matrices can help explain the perplexing numerical behavior of the Arnoldi iteration for computing eigenvalues. Suppose that $AV_m = V_m H_m + f_m e_m^T$ is an Arnoldi factorization of length $m$. It is often observed that although $k$ Ritz estimates of the factorization may be suitably small, the residual vector $f_m$ may not be—even for values of $m$ slightly larger than or equal to $k$. Since a step of a shifted QR-iteration is equivalent to replacing the starting vector, the potential forward instability of the QR algorithm examined in this chapter may also be explained by Theorem 5.3. Extreme sensitivity of some of the matrix elements to the shift during a QR step is equivalent to a starting vector having a small perturbation in an unwanted direction.

# Chapter 6

# Deflation Techniques within an IRA-iteration

The connection between the IRA and QR-iterations motivates us to take advantage of the well understood deflation rules of the QR algorithm and adapt them to the former iteration. These deflation techniques are extremely important with respect to convergence and numerical properties. Deflation rules have contributed greatly to the emergence of the practical QR algorithm as the method of choice for computing the eigen-system of dense matrices. This chapter introduces deflation schemes that may be used within an IRA-iteration. The iteration is designed to compute a selected subset of the spectrum of $A$ such as the $k$ eigenvalues of largest real part. We refer to this selected subset as *wanted* and the remainder of the spectrum as *unwanted*. As the iteration progresses, some of the Ritz value approximations to eigenvalues of $A$ may converge long before the entire set of wanted eigenvalues have. These converged Ritz values may be part of the wanted or the unwanted portion of the spectrum. In either case it is desirable to *deflate* the converged Ritz values and corresponding Ritz vectors from the unconverged portion of the factorization. If the converged Ritz value is wanted then it is necessary to keep it in the subsequent Arnoldi factorizations. This is called *locking*. If the converged Ritz value is unwanted then it must also be removed from the current and subsequent Arnoldi factorizations. This is called *purging*. These notions will be made precise during the course of the chapter. For the moment we note that the advantages of a numerically stable deflation strategy include:

- Reduction of the *working* size of the desired invariant subspace.
- The ability to determine clusters of nearby eigenvalues without need for a block Arnoldi/Lanczos method [39, 79, 80].
- Preventing the effects of the forward instability of the Arnoldi/Lanczos algorithm discussed in Chapter 5.

Deflating within the IRA-iteration is examined in § 6.1. The deflation scheme for converged Ritz values is presented in § 6.2. The practical issues associated with our deflation scheme are examined in § 6.3. These include block generalizations of the ideas examined in § 6.2 for dealing with clusters of Ritz values, avoiding the use of complex arithmetic when a complex conjugate pair of Ritz values converges and an error analysis. A brief survey of other deflation strategies is given in § 6.5. An interesting connection with the various algorithms used to re-order a Schur form of matrix is presented in § 5.4. Numerical results are presented in § 6.6.

## 6.1 Deflation within an IRA-iteration

As the iteration progresses the Ritz estimates (2.5.1) decrease at different rates. When a Ritz estimate is small enough, the corresponding Ritz value is said to have converged. The converged Ritz value may be wanted or unwanted. In either case, a mechanism to deflate the converged Ritz value from the current factorization is desired. Depending on whether the converged Ritz value is wanted or not, it is useful to define two types of deflation. Before we do this, it will prove helpful to illustrate how deflation is achieved. Suppose that after $m$ steps of the Arnoldi algorithm we have

$$A \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} H_1 & M \\ \epsilon e_1 e_j^T & H_2 \end{bmatrix} + f e_m^T, \tag{6.1.1}$$

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

### 6.1.1 Locking

If deflation has taken place and all of the deflated Ritz values are wanted, they are considered *locked*. This means that subsequent implicit restarting is done on the basis $V_2$. The sub-matrices effected during implicit restarting are $M$, $H_2$ and $V_2$. However, during the phase of the iteration that extends the Arnoldi factorization from $k$ to $k+p$ steps, all of the columns of $\begin{bmatrix} V_1 & V_2 \end{bmatrix}$ participate—just as if no deflation had occurred. This assures that all of the new Arnoldi basis vectors are orthogonalized against converged Ritz vectors and prevents the introduction of spurious eigenvalues into the subsequent iteration. Moreover, this provides a means to safely compute multiple eigenvalues when they are present. A block method is not required if deflation and locking are used. The concept of locking was introduced by Jennings and Stewart [92] as a deflation technique for simultaneous iteration.

### 6.1.2 Purging

If deflation has occurred but some of the deflated Ritz values are unwanted, a further mechanism, purging, must be introduced to remove the unwanted Ritz values and corresponding vectors from the factorization. In exact arithmetic this would not be necessary because the implicit shift technique would accomplish the removal of the unwanted Ritz pair from the leading portion of the iteration. However, computing with finite precision arithmetic may make it impossible to accomplish the removal because of the forward instability [63, 99] of the QR algorithm discussed in Chapter 5. The basic idea of purging is perhaps best explained with the case of a single deflated Ritz value.

Let $j = 1$ in (6.1.1) and equate the first columns of both sides to obtain

$$Av_1 = v_1\alpha_1 + \epsilon V_2 e_1, \tag{6.1.2}$$

where $v_1 = V_1 e_1$ and $H_1 = \alpha_1$. Equation (6.1.2) is an Arnoldi factorization of length one. The Ritz value $\alpha_1$ has Ritz estimate $|\epsilon|$.

Equating the last $m-1$ columns of (6.1.1) results in

$$AV_2 = V_1 M + V_2 H_2 + f e_{m-1}^T, \tag{6.1.3}$$

Suppose that $\alpha_1$ represents an unwanted Ritz value. If $A$ were symmetric then $M = \epsilon e_1^T$ and equation (6.1.3) becomes

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

where $E = -\epsilon v_1 (V_2 e_1)^T - \epsilon (V_2 e_1) v_1^T$. A simple derivation shows that $\|E\| = \epsilon$ and hence equation (6.1.3) defines a length $m-1$ Arnoldi factorization for a nearby problem. The unwanted Ritz pair $(v_1, \alpha_1)$ may be *purged* from the factorization simply by taking $V = V_2$ and $H = H_2$ and setting $M = 0$ in (6.1.3). If $A$ is not symmetric, the $1 \times (m-1)$ matrix $M$ couples $v_1$ to the rest of the basis vectors $V_2$. This vector may be decoupled using the standard Sylvester equation approach [9, 35]. Purging then takes place as in the symmetric case. However, the new set of basis vectors must be re-orthogonalized in order to return to an Arnoldi factorization. This procedure is developed in § 6.2 and § 6.3 including the case of purging several vectors.

### 6.1.3 Complications

An immediate question is: Do any sub-diagonal elements in the Hessenberg matrix of the factorization (6.1.1) become negligible as an IRA-iteration progresses ? Since a cycle of the Arnoldi iteration involves performing a sequence of QR steps, the question is answered by considering the behavior of the QR-iteration upon upper Hessenberg matrices. In exact arithmetic under the assumption that the Hessenberg matrix is unreduced, only the last sub-diagonal element may become zero when shifting. But the other sub-diagonal elements may become arbitrarily small. In addition, as discussed in Chapter 5, the forward instability of an IRA-iteration possibly renders the sub-diagonal entries of $H$ meaningless.

## 6.2 Deflating Converged Ritz Values

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

> **Lemma 6.1** Let $Hs = s\theta$ where $H \in \mathbf{R}^{k\times k}$ is an unreduced upper Hessenberg matrix and $\theta \in \mathbf{R}$ with $\|s\| = 1$ . Let $W$ be a Householder matrix such that $Ws = e_1\zeta$ where $\zeta = \pm 1$. Then
>
> (6.2.1) $$e_k^T W \quad = \quad e_k^T + w^T,$$
>
> where $\|w\| \le \sqrt{2}|e_k^T s|$ and
>
> (6.2.2) $$W^T H W e_1 \quad = \quad e_1\theta.$$

**Proof** The required Householder matrix has the form $W = I - \gamma(s-\zeta e_1)(s-\zeta e_1)^T$, where $\gamma = (1+|e_1^T s|)^{-1}$. A direct computation reveals that

(6.2.3) $$e_k^T W \quad = \quad e_k^T + w^T,$$

where $w^T = \gamma e_k^T s(\zeta e_1^T - s^T)$. Estimating

$$\|w\| = \frac{|e_k^T s|}{1+|e_1^T s|}\|s-\zeta e_1\| = \frac{|e_k^T s|}{1+|e_1^T s|}\sqrt{2(1+|e_1^T s|)} \le \sqrt{2}|e_k^T s|,$$

establishes the bound on $\|w\|$. The final assertion (6.2.2) follows from

$$W^T H W e_1 = \zeta^{-1} W^T H s = \zeta^{-1}\theta W^T s = \zeta^{-1}\theta W s = \theta e_1.$$

□

The hypothesis that $H$ is unreduced assures that $|e_k^T s| \neq 0$ by Lemma 2.1. Lemma 6.1 indicates that the last row and column of $W$ differ from the last row and column of $I_k$ by terms of order $|e_k^T s|$. The Ritz estimate (2.5.1) will indicate when the corresponding Ritz value $\theta$ may be deflated.

Rewriting (2.2.1) as

$$AVW \quad = \quad VWW^T HW + fe_k^T W,$$

and using both (6.2.1) and (6.2.2) and partitioning we obtain

$$AVW \quad = \quad VW \begin{bmatrix} \theta & \bar{h}^T \\ 0 & \bar{H} \end{bmatrix} + fe_k^T + fw^T. \tag{6.2.4}$$

Equation (6.2.4) is not an Arnoldi factorization. The matrix $\hat{H}$ of order $k-1$ needs to be returned to upper Hessenberg form. Care must be taken not to disturb the matrix $fe_k^T$ and the first column of $W^T HW$. To start the process we compute a Householder matrix $W_1$ such that

$$W_1^T \bar{H} W_1 \quad = \quad \begin{bmatrix} \bar{M} & \bar{g} \\ \bar{\beta}_k e_{k-2}^T & \bar{\gamma} \end{bmatrix},$$

with $e_{k-1}^T W_1 = e_{k-1}^T$. The above idea is repeated resulting in Householder matrices $W_1, W_2, \ldots, W_{k-3}$ that returns $\bar{H}$ to upper Hessenberg form. Defining

$$\bar{W} \quad = \quad \begin{bmatrix} 1 & 0 \\ 0 & W_1 W_2 \cdots W_{k-3} \end{bmatrix},$$

it follows by the construction of the $W_j$ that $e_k^T \bar{W} = e_k^T$ and

$$\bar{W}^T W^T H W \bar{W} e_1 \quad = \quad \theta e_1. \tag{6.2.5}$$

The process of computing a similarity transformation as in equation (6.2.5) is not new. Wilkinson discusses the more general notion of deflating with invariant subspaces in § 20–25, Chapter 9 in [101]. Wilkinson also references the work of Feller and Forsythe [31] who appear to be the first to use elementary Householder transformations for deflation. Problem 7.4.8 of [35] addresses the case when working with upper Hessenberg matrices. What appears to be new is the application to the Arnoldi factorization for converged Ritz values.

Since $\|fw^T\bar{W}\| = \|f\|\,\|\bar{W}^T w\| = \|f\|\,\|w\|$, the size of $\|fw^T\|$ remains the unchanged. Making the updates

$$V \leftarrow VW\bar{W}, \quad H \leftarrow \bar{W}^T W^T HW\bar{W}, \quad w^T \leftarrow w^T\bar{W}$$

we obtain the relation

$$AV = VH + fe_k^T + fw^T. \tag{6.2.6}$$

A deflated Arnoldi factorization is obtained from (6.2.6 ) by discarding the term $fw^T$.

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

**Theorem 6.1** Let an Arnoldi factorization of length $k$ be given by (6.2.6) where $Hs = s\theta$ and $\sqrt{2}|e_k^T s|\,\|f\| \leq \epsilon\|A\|$ for $\epsilon > 0$. Then there exists a matrix $E \in \mathbf{R}^{n\times n}$ such that

$$(A+E)V = VH + fe_k^T, \tag{6.2.7}$$

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

**Proof** Subtract $fw^T$ from both sides of equation (6.2.6). Set $E = -f(Vw)^T$ and then

$$EV = -f(Vw)^T V = -fw^T,$$

and equation (6.2.7) follows. Using Lemma 6.1 it follows that $\|E\| = \|f\|\,\|w\| = \sqrt{2}|e_k^T s|\,\|f\| \leq \epsilon\|A\|$. □

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

Once a converged Ritz value $\theta$ is deflated, the Arnoldi vector corresponding to $\theta$ is locked or purged as described in the previous section. The only difficulty that remains is decoupling the Ritz vector corresponding to the Ritz value $\theta$, or purging, from the trailing factorization when $A$ is nonsymmetric.

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

$$\begin{bmatrix} \theta & \bar{h}^T \\ & \bar{H} \end{bmatrix} \begin{bmatrix} 1 & z^T \\ & I_{k-1} \end{bmatrix} = \begin{bmatrix} 1 & z^T \\ & I_{k-1} \end{bmatrix} \begin{bmatrix} \theta & \\ & \bar{H} \end{bmatrix}. \tag{6.2.8}$$

Solving the linear system

$$(\bar{H}^T - \theta I_{k-1})z = \bar{h}, \tag{6.2.9}$$

determines $z$. Define

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

Post multiplication of equation (6.2.6) by $Z$ results in

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

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

$$AV \begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix} = V \begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix} \bar{H} + fe_{k-1}^T + fw^T \begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix}. \tag{6.2.10}$$

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

$$QR = \begin{bmatrix} z^T \\ I_{k-1} \end{bmatrix}, \tag{6.2.11}$$

where $Q \in \mathbf{R}^{k \times k-1}$ with $Q^T Q = I_{k-1}$ and $R$ is an upper triangular matrix of order $k-1$. Since the last $k-1$ columns of $Z$ are linearly independent, $R$ is nonsingular. Post multiplying equation (6.2.10) by $R^{-1}$ gives

$$AVQ = VQR\bar{H}R^{-1} + \rho_{k-1}^{-1} fe_{k-1}^T + fw^T Q, \tag{6.2.12}$$

where $\rho_{k-1} = e_{k-1}^T R e_{k-1}$. The last term $fw^T Q$ in (6.2.12) is discarded by the deflation scheme and this relation shows that the discarded term is not magnified in norm by the purging procedure. The matrix $R\bar{H}R^{-1}$ remains upper Hessenberg since $R$ is upper triangular. Partitioning $Q$ conformably with the right side of equation (6.2.11) results in

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

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

$$AVQ \;=\; VQR\bar{H}Q_{21} + \rho_{k-1}^{-1} f e_{k-1}^T. \tag{6.2.13}$$

The similarity transformation that produces the new upper Hessenberg matrix does not affect the eigenvectors and thus the Ritz estimates. Since the Ritz estimates are just the residuals of the Ritz pairs which are determined by $A$ and the $\mathcal{R}(V)$, the similarity transformation performed on $H$ through $R$ does not affect the Ritz pairs. Only the basis representation of the $\mathcal{R}(V)$ is modified so that we may decouple and discard an unwanted Ritz pair.

Performing the set of updates

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

defines equation (6.2.13) as an Arnoldi factorization of length $k-1$. Theorem 6.1 implies this is an Arnoldi factorization for a nearby matrix. It is easily verified that $V^T f(e_{k-1}^T + w^T) = 0$ and that $H$ is an upper Hessenberg matrix of order $k-1$.

## 6.3 A Practical Deflating Procedure

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

1. Performing the deflation in real arithmetic when a converged Ritz value has a non-zero imaginary component.
2. Deflation with more than one converged Ritz value.
3. Error Analysis.

Section 6.3.2 presents two algorithms that implement the deflation schemes. The error analysis of the two deflation schemes is presented in the next section.

### 6.3.1 Deflation with Real Arithmetic

Suppose $s = t + iu$ and $\theta = \nu + i\mu$ is an eigenpair of $H$ where $t$ and $u$ are unit vectors in $\mathbf{R}^k$, $H \in \mathbf{R}^{k\times k}$ and $\mu \neq 0$. Thus

$$H \begin{bmatrix} t & u \end{bmatrix} = \begin{bmatrix} t & u \end{bmatrix} \begin{bmatrix} \nu & \mu \\ -\mu & \nu \end{bmatrix} \equiv \begin{bmatrix} t & u \end{bmatrix} C.$$

Factor

$$\begin{bmatrix} t & u \end{bmatrix} = Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{6.3.1}$$

where $Q^TQ = I_k$ and $R$ is an upper triangular matrix. It is easily shown that $t$ and $u$ are linearly independent as vectors in $\mathbf{R}^k$ since $\mu \neq 0$ and the non-singularity of $R$ follows. Performing a similarity transformation with $Q$ on $\begin{bmatrix} t & u \end{bmatrix}$ gives

$$Q^THQ \begin{bmatrix} e_1 & e_2 \end{bmatrix} = \begin{bmatrix} RCR^{-1} \\ 0 \end{bmatrix}.$$

Suppose that $H$ corresponds to an Arnoldi factorization of length $k$ and that $|e_k^Tt| = O(\epsilon) = |e_k^Tu|$. In order to deflate the complex conjugate pair of eigenvalues from the factorization in an implicit manner, we require that $e_k^TQ = e_k^T + q^T$ where $\|q\| = O(\epsilon)$.

We now show that the magnitudes of the last components of $t$ and $u$ are not sufficient to guarantee the required form for $Q$. Suppose that $u = t\cos\phi + r\sin\phi$ where $r$ is a unit vector orthogonal to $t$ and $\phi$ measures the positive angle between $t$ and $u$. Lemma 6.1 allows a Householder $W_1$ matrix such that

$$W_1^T \begin{bmatrix} t & u \end{bmatrix} = \begin{bmatrix} \zeta_1 e_1, \zeta_1 e_1 \cos\phi + W_1^T r \sin\phi \end{bmatrix} \equiv \begin{bmatrix} \zeta_1 & \zeta \\ 0 & \bar{u} \end{bmatrix},$$

where $\zeta_1 = \pm 1$ and the last column and row of $W_1$ and $I_k$ are order $e_k^Tt$ equivalent. To compute the required orthogonal factorization in equation (6.3.1) another Householder matrix $W_2 = \begin{bmatrix} 1 & 0 \\ 0 & \bar{W}_2 \end{bmatrix}$, is needed so that $\bar{W}_2^T\bar{u} = \pm\|\bar{u}\|e_1$. But Lemma 6.1 only results in $e_{k-1}^T\bar{W}_2 = e_{k-1}^T + \bar{w}_2^T$ with $\|\bar{w}_2\| = O(\epsilon)$ if $e_{k-1}^T\bar{u}$ is small relative to $\|\bar{u}\|$. Unfortunately, if $\phi$ is small, $W_1^Tu \approx \zeta_1 e_1$ and $\|\bar{u}\| \approx \phi \approx 0$. Hence we cannot obtain the required form for $Q = W_1W_2$.

Fortunately, when $t$ and $u$ are nearly aligned, $\mu$ may be neglected as the following result demonstrates.

**Lemma 6.2** Let $H(t+iu) = (\nu + i\mu)(t+iu)$ where $t$ and $u$ are unit vectors in $\mathbf{R}^k$, $H \in \mathbf{R}^{k\times k}$ and $\mu \neq 0$. Suppose that $\phi$ measures the positive angle between $t$ and $u$. Then

$$|\mu| \quad \leq \quad \sin\phi \|H\|. \tag{6.3.2}$$

**Proof** Let $u = t\cos\phi + r\sin\phi$ where $r$ is a unit vector orthogonal to $t$ and $\phi$ measures the positive angle between $t$ and $u$. Equating real and imaginary parts of $H(t+iu) = (\nu+i\mu)(t+iu)$ results in $Ht = t\nu - u\mu$ and $Hu = t\mu + u\nu$. The desired estimate follows since

$$2\mu = t^T Hu - u^T Ht = (t^T Hr - r^T Ht)\sin\phi,$$

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

For small $\phi$, $t$ and $u$ are almost parallel eigenvectors of $H$ corresponding to a nearly multiple eigenvalue. Numerically, we set $\mu$ to zero and deflate one copy of $\nu$ from the Arnoldi factorization.

A computable bound on the size of the angle $\phi$ is now determined using only the real and imaginary parts of the eigenvector. The second Householder matrix $W_2$ should not be computed if

$$|e_{k-1}^T \bar{u}| \quad > \quad \|\bar{u}\| |e_k^T u|. \tag{6.3.3}$$

Recall that Lemma 6.1 gives $e_k^T W_1 = e_k^T + w_1^T$ where $w_1^T = \gamma e_k^T t(\zeta_1 e_1^T - t^T)$ and $\gamma = (1 + |e_1^T t|)^{-1}$. Thus

$$e_{k-1}^T \bar{u} = e_k^T W^T u = e_k^T W u = e_k^T u + w^T u,$$

where the symmetry of $W_1$ is used. The estimate

$$\|\bar{u}\| = \|[0, \bar{u}^T]^T\| = \|W_1^T r\| \sin\phi = \sin\phi,$$

follows since $W_1$ is orthonormal and $r$ is a unit vector. Rewriting equation (6.3.3), we obtain

$$\begin{aligned}
\sin\phi \quad &< \quad |\frac{e_k^T u + w^T u}{e_k^T u}|, \\
&= \quad |1 + \frac{w^T u}{e_k^T u}|, \\
&= \quad |1 + \gamma(\zeta_1 e_1^T u - y^T u)\frac{e_k^T t}{e_k^T u}|, \qquad (6.3.4)
\end{aligned}$$

as our computable bound.

Suppose that $HX = XD$ where $X \in \mathbf{R}^{k\times j}$ and $D$ is a quasi-diagonal matrix. The eigenvalues of $H$ are on the diagonal of $D$ if they have zero imaginary component and in blocks of two for the complex conjugate pairs. The columns of $X$ span the eigenspace corresponding to diagonal values of $D$. For the blocks of order two on the diagonal the corresponding complex eigenvector is stored in two consecutive columns of $X$, the first holding the real part, and the second the imaginary part. If we want to block deflate $X$, where the last row is small, from $H$, then we could proceed as follows. Compute the orthogonal factorization $X = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$ via Householder reflectors where $Q^TQ = I_k$ and $R \in \mathbf{R}^{k\times k}$ is upper triangular. Then the last row and column of $Q$ differ from that of $I_k$ with terms on the same order of the entries in the last row of $X$ if the condition number of $R$ is modest. Theorem 6.4 makes this last statement precise. Thus if the columns of $X$ are not almost linearly dependent, an appropriate $Q$ may be determined. Finally, we note that when $H$ is a symmetric tridiagonal matrix, an appropriate $Q$ may always be determined.

### 6.3.2 Algorithms for Deflating Converged Ritz Values

The two procedures presented in this section extend the ideas of § 6.1 to provide deflation of more than one converged Ritz value at a time. The first purges the factorization of the unwanted converged Ritz values. The second locks the Arnoldi vectors corresponding to the desired converged Ritz values. When both deflation algorithms are incorporated within an IRA-iteration, the locked vectors form a basis for an approximate invariant subspace of $A$. This truncated factorization is an approximate partial Schur decomposition. When $A$ is symmetric, the approximate Schur vectors are Ritz vectors and the upper quasi-triangular matrix is the diagonal matrix of Ritz values.

Partition a length $m$ Arnoldi factorization as

$$A \begin{bmatrix} V_j & \bar{V}_{m-j} \end{bmatrix} = \begin{bmatrix} V_j & \bar{V}_{m-j} \end{bmatrix} \begin{bmatrix} H_j & M_j \\ 0 & \bar{H}_{m-j} \end{bmatrix} + f_m e_m^T + f w^T, \tag{6.3.5}$$

where $H_j$ and $\bar{H}_{m-j}$ are upper quasi-triangular and unreduced upper Hessenberg matrices, respectively. The matrix $H_j \in \mathbf{R}^{j\times j}$ contains the wanted converged Ritz values of the matrix $H_m$. The columns of $V_j \in \mathbf{R}^{n\times j}$ are the locked Arnoldi vectors that represent an approximate Schur basis for the invariant subspace of interest. The matrix $\bar{H}_{m-j}$ designates the trailing sub-matrix of order $m-j$. Analogously, the last $m-j$ columns of $V_m$ are denoted by $\bar{V}_{m-j}$. We shall refer to the last $m-j$ columns of (6.3.5) as the *active* part of the factorization. Finally, $M_j \in \mathbf{R}^{j \times m-j}$ denotes the sub-matrix in the north–east corner of $H_m$. Figure 6.1 illustrates the matrix product $V_m H_m$ of equation (6.3.5).

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

**Algorithm 6.2**

function $[V_m, H_m, f_m] =$ `Lock` $(V_m, H_m, f_m, X_i, j)$

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

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

1. Compute the orthogonal factorization

$$Q \begin{bmatrix} R_i \\ 0_{m-j-i} \end{bmatrix} = X_i,$$

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

2. Update the factorization

   $\bar{H}_{m-j} \leftarrow Q^T \bar{H}_{m-j} Q$ ; $\bar{V}_{m-j} \leftarrow \bar{V}_{m-j} Q$ ; $M_j \leftarrow M_j Q$ ;

3. Compute an orthogonal matrix $P \in \mathbf{R}^{m-j-i \times m-j-i}$ using Householder matrices that restores $\bar{H}_{m-j-i}$ to upper Hessenberg form ;

4. Update the factorization

   $\bar{H}_{m-j-i} \leftarrow P^T \bar{H}_{m-j-i} P$ ; $\bar{V}_{m-j-i} \leftarrow \bar{V}_{m-j-i} P$ ; $M_{j+i} \leftarrow M_{j+i} P$ ;

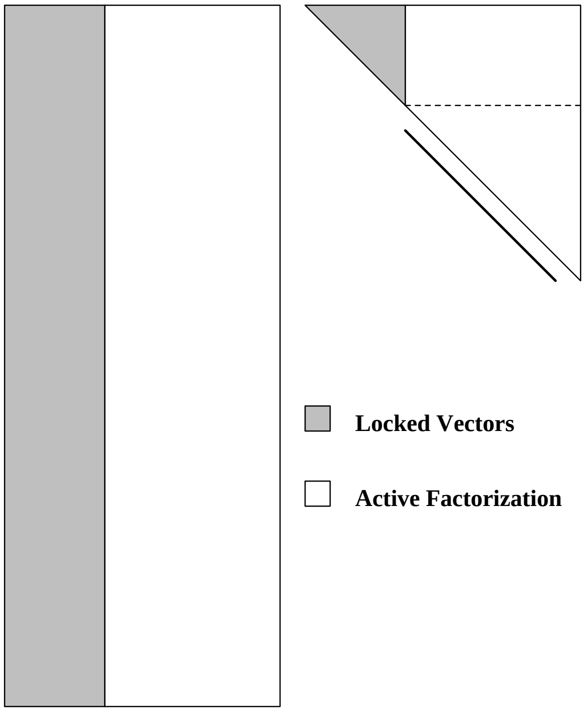


**Figure 6.1** The matrix product $V_m H_m$ of the factorization upon entering Algorithm 6.2 or 6.3. The shaded region corresponds to the converged portion of the factorization.

Line 1 computes an orthogonal basis for the eigenvectors of $\bar{H}_{m-j}$ that correspond to the Ritz estimates that are converged. The matrix of eigenvectors in line 1 satisfies the equation $\bar{H}_{m-j}X_i = X_iD_i$ where $D_i$ is a quasi-diagonal matrix containing the eigenvalues to be locked. From the § 6.3.1, we see that the leading sub-matrix of $Q^T\bar{H}_{m-j}Q$ of order $i$ is upper quasi-triangular. The required relation $e_m^TQ = e_m^T + q^T$, with $\|q\|$ small is guaranteed if the condition number of $R_i$ is modest. Since $i$ is typically a small number, we compute the condition number of $R_i$. The number of vectors to be locked is assumed to be such that the condition number of $R_i$ is small. In particular, if $H_m$ is a symmetric tridiagonal matrix, $Q$ always has the required form. Lines 3–4 return the updated $\bar{H}_{m-j}$ to upper Hessenberg form.

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

**Algorithm 6.3**

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

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

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

Lines 1–3 purge the factorization of the unwanted converged Ritz values contained in the leading portion of $H_m$ ;

1. Solve the Sylvester set of equations,

$$Z\bar{H}_{m-i} - H_iZ \;=\; M_i,$$

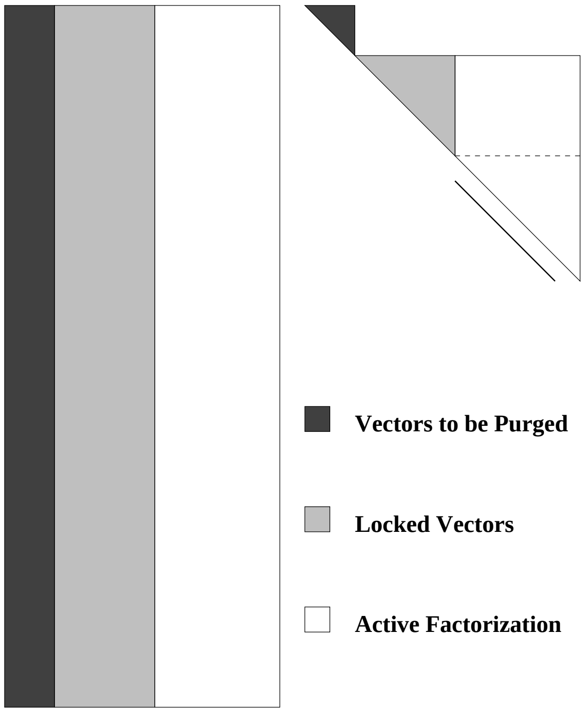


**Figure 6.2** The matrix product $V_m H_m$ of the factorization just prior to discarding in Algorithm 6.3. The darkly shaded regions may now be dropped from the factorization.

for $Z \in \mathbf{R}^{i \times m-i}$ that arise from block diagonalizing $H_m$ ;

$$H_m \begin{bmatrix} I_i & Z \\ & I_{m-i} \end{bmatrix} = \begin{bmatrix} I_i & Z \\ & I_{m-i} \end{bmatrix} \begin{bmatrix} H_i & \\ & \bar{H}_{m-i} \end{bmatrix},$$

2. Compute the orthogonal factorization

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

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

3. Update the factorization and obtain a length $m-i$ factorization ;

$H_{m-i} \leftarrow R_{m-i}\bar{H}_{m-i}\bar{Q}_{m-i}$ ; $V_{m-i} \leftarrow V_m Q$ ; $f_{m-i} \leftarrow \rho^{-1}_{m-i,m-i} f_m$ ;

where $\rho_{m-i,m-i} = e^T_{m-i} R_{m-i} e_{m-i}$ ;

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

The solution of the Sylvester equation at line 1 determines the matrix $Z$ that block diagonalizes the spectrum of $H_m$ into two sub-matrices. The unwanted portion is in the leading corner and the remaining eigenvalues of $H_m$ are in the other block. A solution $Z$ exists when the $H_i$ and $\bar{H}_{m-i}$ do not have a common eigenvalue. If there is an eigenvalue is shared by $H_i$ and $\bar{H}_{m-i}$, then $H_m$ has an eigenvalue of multiplicity greater than one. The remedy is a criterion that determines whether to increase or decrease $i$, the number of Ritz values that require purging. Analysis similar to that in section 6.2 demonstrates that after line 3 the Ritz estimates for the eigenvalues of $H_{m-i}$ are not altered. We also remark that $R_{m-i}$ is nonsingular since the matrix $\begin{bmatrix} Z \\ I_{m-i} \end{bmatrix}$ is of full column rank and that $|\rho^{-1}_{m-i,m-i}| \leq 1$.

## 6.4 Error Analysis

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

For ease of notation $H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$ replaces $H_m \in \mathbf{R}^{m\times m}$ used by procedures `Lock` and `Purge` of § 6.3.2. The sub-matrix $H_{11}$ is of order $i$ and $H_{21}$ is zero except for the sub-diagonal entry of $H$ located in the north-east corner. Analogously, $\hat{H}$ represents $H$ after the similarity transformation performed by `Lock` or `Purge`, partitioned conformably.

### 6.4.1 Locking

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

The matrix pair $(X, D)$ represents an approximate quasi-diagonal form for $H$. The computed columns of $X$ span the right eigenspace corresponding to diagonal blocks of $D$. We assume that $X$ is a non-singular matrix and that each column is a unit vector.

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

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

**Theorem 6.4**

Let $H \in \mathbf{R}^{m\times m}$ be an unreduced upper Hessenberg matrix with distinct eigenvalues. Suppose that $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$ and $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$ are an approximate quasi-diagonal form for $H$ that satisfies $(H + E)X = XD$ where $\|E\| \le \epsilon_1 \kappa(X)\|H\|$. Let $Q_1 R_1 = X_1 \in \mathbf{R}^{m\times j}$ where $Q_1^T Q_1 = I_j$.

Suppose the QR factorization of $X_1$ is computed so that $\hat{Q}\hat{R} = X_1 + \hat{E}$ where $\hat{Q}^T\hat{Q} = I_m$ and $\|\hat{E}\| \leq \epsilon_2\|X_1\|$. Both $\epsilon_1$ and $\epsilon_2$ are small multiples of the machine precision $\epsilon_M$. Let $\epsilon = \max(\epsilon_1, 2\epsilon_2)$ and let $\kappa(R_1) = \|R_1\|\|R_1^{-1}\|$ be the condition number for $R_1$ where

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

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

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

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

$$\|C\| \leq \epsilon(\kappa(X) + \mu)\|H\| + O(\epsilon^2). \tag{6.4.1}$$

A few remarks are in order.

1. If $H$ is symmetric $\hat{H}_{12} = 0$ and $\hat{H}_{11}$ is diagonal. Procedure `Lock` is stable since noted previously, $\kappa(X) = 1$ and $\mu \approx 1$. Parlett [61, pages 85–86] proves Theorem 6.4 for symmetric matrices when locking one approximate eigenvector.

2. If only one column is locked, then $\mu = 1 + O(\epsilon)$ and $\|C\|$ is small relative to $\kappa(X)\|H\|$.

3. If $\kappa(R_1)$ is large, the columns of $X_1$ are nearly dependent. In this case, $\kappa(X)$ will also be large and locking introduces no more error into the computation than already present from computing the quasi-diagonal pair $(X, D)$. The factor of $\mu$ may be minimized by decreasing $j$ the number of columns locked.

4. A conservative strategy locks only one vector at a time. The only real concern is when locking two vectors corresponding to a complex conjugate pair. If the real and imaginary part of the complex eigenvector are nearly aligned, $\mu$ will be large and locking may be unstable. But as § 6.3.1 explains, the complex conjugate pair may be numerically regarded as a double eigenvalue with zero imaginary part. Only one copy is deflated and $\mu \approx 1$.

**Proof**

Partition $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$ and $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$. The $i$ columns of $X_1$ are a basis for the right eigenspace to be locked and $D_1$ contains the corresponding eigenvalues. We assume that the eigenvalues of $D_1$ and $D_2$ are distinct and that $X$ is non-singular. Let $Y^T = \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix}$ denote the inverse of $X$. The rows of $Y_1^T$ span the left eigenspace associated with $X_1$ and $D_1$.

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

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

where $\|\hat{E}\| \leq \epsilon_2 \|X_1\|$. Using Theorem 1.1 of Stewart [89], since $\|R_1^{-1}\|\|\hat{E}\| < \eta < 1$ there exists matrices $W_1 \in \mathbf{R}^{m \times j}$ and $F_1 \in \mathbf{R}^{j \times j}$ such that $(Q_1 + W_1)(R_1 + F_1) = \hat{Q}_1 \hat{R}_1$ where $QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = X_1$ and $(Q_1 + W_1)^T (Q_1 + W_1) = I_j$. Define $F = \begin{bmatrix} F_1 \\ 0 \end{bmatrix}$ and $W = \begin{bmatrix} W_1 & 0 \end{bmatrix}$. The matrices $W$ and $F$ are the perturbations that account for the backward error $\hat{E}$ produced by computation.

Partitioning $W$ conformably with $Q$ gives

$$\begin{aligned}
\hat{Q}^T H \hat{Q} &= \hat{Q}^T X D Y^T \hat{Q} - \hat{Q}^T E \hat{Q}, \\
&= \hat{Q}^T (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) \hat{Q} - \hat{Q}^T E \hat{Q}, \\
&= \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} + \\
&\quad W^T (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} + \\
&\quad \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T) W - \hat{Q}^T E \hat{Q},
\end{aligned} \tag{6.4.2}$$

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

Using these relationships, equation (6.4.2) becomes

$$\hat{Q}^T H \hat{Q} = \begin{bmatrix} R_1 D_1 R_1^{-1} & Q_1^T X D Y^T Q_2 \\ 0 & Q_2^T X_2 D_2 Y_2^T Q_2 \end{bmatrix} + \hat{C}, \tag{6.4.3}$$

$$(6.4.4) \qquad \equiv \hat{H} + \hat{C},$$

where the matrix $\hat{C}$ absorbs the three matrix products involving $W$ or $E$ on the right hand side of equation (6.4.2). We note that if $H$ is symmetric, $Q_1^T X_2 = 0 = Y_1^T Q_2$, $R_1$ is a diagonal matrix and hence $R_1 D_1 R_1^{-1} = D_1$. Thus $\hat{H}$ is also a symmetric matrix. Defining $C = \hat{Q}\hat{C}\hat{Q}^T$ equation (6.4.4) is rewritten as $\hat{Q}^T(H - C)\hat{Q} = \hat{H}$. Since $Q\hat{H} = (X_1 D_1 Y_1^T + X_2 D_2 Y_2^T)Q$ and using the definition of $\hat{C}$ from equation (6.4.2),

$$(6.4.5) \qquad \hat{C} \;=\; W^T Q \hat{H} + Q^T W \hat{H} - \hat{Q}^T E \hat{Q},$$

it follows that $\|C\| \leq 2\|W^T Q\|\|\hat{H}\| + \|E\|$. The result of Theorem 1.1 of Stewart [89] also allows the estimate

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

where $O(\epsilon^3)$ terms are ignored. For modest values of $\mu$, $W$ is numerically orthogonal to $Q$. From equation (6.4.5)

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

where the second inequality uses equation (6.4.4). Since $\hat{\eta} < \eta$, rearranging the last inequality gives $\|C\|(1 - \eta) \leq \eta\|H\|$. Ignoring $O(\eta^2)$ terms $\|C\| \leq \eta\|H\|$. The estimate on the size of $C$ in equation (6.4.1) now follows since $\eta = \epsilon(\kappa(X) + \mu(1 + \epsilon\mu\kappa(X))) \leq \epsilon(\kappa(X) + \mu) + O(\epsilon^2)$. □

### 6.4.2 Purging

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

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

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

Let $\hat{Z}$ be the computed solution to the Sylvester set of equations. In a similar analysis, Bai and Demmel [9] assume that the $QR$ factorization of $S$ is performed exactly and we do also. The major source of error is that arising from computing $\hat{Z}$.

Suppose that $\hat{Q}\hat{R} = \begin{bmatrix} \hat{Z} \\ I \end{bmatrix} \equiv \hat{S}$. Write $\hat{Z} = Z + E$ where $E$ is the error in $\hat{Z}$. If $QR = S$ and $\|R^{-1}\|\|E\| < 1$, then Theorem 1.1 of Stewart [89] gives matrices $W$ and $F$ such that $(Q+W)(R+F) = \hat{Q}\hat{R}$ where $(Q+W)^T(Q+W) = I_m$. The result gives the bound $\|F\| \leq \|R\|\|E\| + O(\|E\|^2)$. Up to first order perturbation terms,

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

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

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

Ignoring second-order terms, we obtain the estimate

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

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

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

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

The previous discussion is summarized in the following result.

**Theorem 6.5** Let $\hat{Z}$ be the computed solution to the Sylvester set of equations, $ZH_{22} - H_{11}Z = H_{12}$, where the eigenvalues of $H_{11}$ and $H_{22}$ are distinct. Let $\hat{Z} = Z + E$ where $E$ is the error in $\hat{Z}$ and suppose that $\|R^{-1}\|\|E\| < 1$ where $QR = \begin{bmatrix} Z \\ I \end{bmatrix}$.

Then there exists a matrix $C$ such that

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

where

$$\|C\| \quad \leq \quad 2\sqrt{1+\|Z\|^2}\,\|E\|\,\|H\|. \tag{6.4.6}$$

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

The two standard approaches [11, 36] for solving Sylvester's equation show that $\|\hat{F}\|_F \leq \epsilon_3(\|H_{11}\|_F + \|H_{22}\|_F)\|\hat{Z}\|_F$ where $\hat{F} \equiv H_{12} - \hat{Z}H_{22} + H_{11}\hat{Z}$ and $\epsilon_3$ is a modest multiple of machine precision. Standard bounds [18, 35] also give $\|Z\|_F \leq \mathrm{sep}^{-1}(H_{11}, H_{22})\|H_{12}\|_F$ where

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

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

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

Varah [94] indicates that if the matrices involved are highly non-normal, the smallest difference between the spectrums of $H_{11}$ and $H_{22}$ may be an over estimate of the actual separation. Recently, Higham [40] gives a detailed error analysis for the solution of Sylvester's equation. The analysis takes into account the special structure of the equations involved. For example, Higham shows that $\|E\|_F \leq \mathrm{sep}^{-1}(H_{11}, H_{22})\|\hat{F}\|_F$ but this may lead to an arbitrarily large estimate of the true forward error. For use in practical error estimation, "LAPACK-style" software is available.

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

## 6.5 Other Deflation Techniques

Wilkinson [101, pages 584–602] has given a comprehensive treatment of various deflation schemes associated with iterative methods. Recently, Saad [78, pages 117–125,180–182] discussed several deflation strategies used with both simultaneous iteration and Arnoldi's method. Algorithm 6.2 is an in place version of one of these schemes [78, page 181]. Saad's version explicitly orthonormalizes the newly converged Ritz vectors against the already computed approximate $j$ Schur vectors. This is the form of locking used by Scott [80]. Instead, procedure `Lock` achieves the same task implicitly through the use of Householder matrices in $\mathbf{R}^{m\times m}$. Thus we are able to orthogonalize vectors in $\mathbf{R}^n$ at a reduced expense since $m \ll n$.

Other deflation strategies include the various Wielandt deflation techniques [78, 101]. We briefly review those that do not require the approximate left eigenvectors of $A$ or complex arithmetic. Denote by $\lambda_1, \ldots, \lambda_j$ the wanted eigenvalues of $A$. The Wielandt and Schur–Wielandt forms of deflation determine a rank $j$ modification of $A$,

$$A_j \;=\; A - U_j S_j U_j^T, \tag{6.5.1}$$

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

Both forms of deflation differ in the choice of $U_j$. The Wielandt variant uses converged Ritz vectors while the Schur–Wielandt uses an approximate Schur basis set vectors. With either form of deflation, the eigenvalues of $A_j$ are $\lambda_i - \sigma_i$ for $i \le j$ and $\lambda_i$ otherwise and both forms leave the Schur vectors unchanged. This motivates Saad to suggest that an approximate Schur basis should be incrementally built as Ritz vectors of $A_j$ converge. Braconnier [16] employs the Wielandt variant and discusses the details of deflating a converged Ritz value that has nonzero imaginary part in real arithmetic.

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

$$A \begin{bmatrix} U_j & V_m \end{bmatrix} \;=\; \begin{bmatrix} U_j & V_m \end{bmatrix} \begin{bmatrix} R_j & M_j \\ 0 & H_m \end{bmatrix} + f_{m+j} e_{m+j}^T, \tag{6.5.2}$$

is a length $m + j$ Arnoldi factorization obtained after locking. Consider any associated roundoff errors as being absorbed in $A$ here. Equate the last $m$ columns of equation (6.5.2) to obtain

$$AV_m \;=\; U_j M_j + V_m H_m + f_{m+j} e_m^T. \tag{6.5.3}$$

Since $U_j$ is orthogonal to $V_m$, it follows that $(I - U_jU_j^T)A(I - U_jU_j^T)V_m = V_mH_m + f_{m+j}e_m^T$. This implies that the Arnoldi factorization (6.5.2) is equivalent to applying Arnoldi's method to the projected matrix $(I - U_jU_j^T)A(I - U_jU_j^T)$ with the first column of $V_m$ as the starting vector. Keeping the locked vectors active in the construction and the IRA update of this Arnoldi factorization assures that the Krylov space generated by $V_m$ remains free of components corresponding to locked Ritz values. The appearance of spurious Ritz values in the subsequent factorization is automatically avoided. Note that when $A$ is symmetric, this is equivalent to the selective orthogonalization [61, pages275–284] scheme proposed by Parlett and Scott.

In contrast to locking, consider the consequences of applying the Schur–Wielandt deflation scheme to construct a new Arnoldi factorization using $V_me_1$ as a starting vector. In the symmetric case with exact arithmetic, the two schemes would be mathematically equivalent. Without these assumptions, there may be considerable differences. From equation (6.5.3), it follows that

$$(A - U_jR_jU_j^T)V_m = A(I - U_jU_j^T)V_m = U_jM_j + V_mH_m + f_{m+j}e_m^T. \tag{6.5.4}$$

From equation (6.5.4) we can use an easy induction to derive the relations

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

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

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

The cost of matrix vector products with $A_j$ increases due to the rank $j$ modifications of $A$ required. Moreover, every time an approximate Schur vector or a Ritz vector converges, the iteration needs to be explicitly restarted with $A_j$. The

[‡]Page 125 of [78]

two deflation techniques introduced in this paper allow the iteration to be implicitly restarted—avoiding the need to build a new factorization from scratch.

Finally, we mention that the idea of deflating a converged Ritz value from a Lanczos iteration is also discussed by Parlett and Nour-Omid [64]. They present an explicit deflation technique by using the QR algorithm with converged Ritz values as shifts. Parlett indicates that this was a primary reason for undertaking the study concerning the forward instability of the QR algorithm [63].

## 6.6 Numerical Results

An IRA-iteration using the two deflation procedures of section 6.3.2 was written in MATLAB, Version 4.2a. An informal description given parameters $k$ and $p$ is given in Table 6.1. The codes are available from the author upon request. A high-quality and robust implementation of the deflation procedures is planned for the Fortran software package ARPACK [49].

In the examples that follow $Q_k$ and $R_k$ denote the approximate Schur factors for an invariant subspace of order $k$ computed by an IRA-iteration. All the experiments used the starting vector equal to `randn(n, 1)` where the seed is set with `randn('seed', 0)` and `n` is the order of the matrix. The shifting strategy uses the unwanted eigenvalues of $H_{k+p}$ that have not converged. An eigenpair $(\theta, s)$ of $H_{k+p}$ is accepted if its Ritz estimate (2.5.1) satisfies,

$$|e_{k+p}^T s| \, \|f_{k+p}\| \;\; \leq \;\; \epsilon|\theta|. \tag{6.6.1}$$

The value of $\epsilon$ is chosen according to the relative accuracy of the Ritz value desired.

### 6.6.1 Example 1

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

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

where

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

```
1. Initialize an Arnoldi factorization of length k

2. Main Loop

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

**Table 6.1** Formal description of an IRA-iteration

for $1 \leq i, j \leq n$ and $\eta_l \equiv \sqrt{\xi_l}$. The eigenvalues of $C$ are $\xi_l \pm \eta_l i$ where $i = \sqrt{-1}$. Since the eigenvalues of a quasi-diagonal matrix are invariant under orthogonal similarity transformations, using an IRA-iteration on $C$ with a randomly generated starting vector is general. An IRA-iteration was used to compute the $k = 12$ eigenvalues of $C_{450}$ with smallest real part. The number of shifts used was $p = 16$ and the convergence tolerance $\epsilon$ was set equal to $10^{-10}$. With these choices of $k$ and $p$, the iteration stores at most twenty eight Arnoldi vectors.

There are four eigenvalues with multiplicity two. Table 6.2 shows the results attained. Let the diagonal matrix $D_{12}$ denote the eigenvalues of the upper triangular matrix $R_{12}$ computed by the iteration. The diagonal matrix $\Lambda_{12}$ contains the wanted eigenvalues. After twenty four iterations twelve Ritz values converged. But the pair of Ritz values purged at iteration twenty one was a previously locked value which the iteration discarded. This behavior is typical when there are clusters of eigenvalues.

### 6.6.2 Example 2

Consider the eigenvalue problem for the convection–diffusion operator,

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

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

**Table 6.2** Convergence history for Example one

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

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

for $1 \le i, j \le n$ where $\gamma = \rho h/2$. An IRA-iteration was used to compute the $k = 6$ smallest eigenvalues of $L_{625}$ where $\rho = 25$. The number of shifts used was $p = 10$ and the convergence tolerance $\epsilon$ was set equal to $10^{-8}$. With these choices of $k$ and $p$, the iteration stores at most sixteen Lanczos vectors. Let the diagonal matrix $D_6$ denote the eigenvalues of the upper triangular matrix $R_6$ computed by the iteration. The diagonal matrix $\Lambda_6 \in \mathbf{R}^{6\times 6}$ contains the six smallest eigenvalues. We note that there are two eigenvalues with multiplicity two. Table 6.3 shows the results attained. The diagonal matrix $D_6$ approximates $\Lambda_6$. After thirty iterations six Ritz values converged. But the Ritz value purged at iteration twenty four was a previously locked value. The other purged Ritz values are approximations to the eigenvalues of $L_{625}$ larger than $\lambda_6$.

Figure 6.3 gives a graphical interpretation of the expense of an IRA-iteration in terms of matrix vector products when the value of $p$ is increased. For all values of $p$ shown, the results of the iteration were similar to those of Table 6.3. The results presented in Table 6.3 correspond to the value of $p$ that gave the minimum number matrix vector products. For the value of $p = 1$, the iteration converged to the five smallest eigenvalues after nine hundred ninety nine matrix vector products. But the iteration was not able to converge to the second copy of $\lambda_5$. For $p = 2$, the only form of deflation employed was locking. All others values of $p$ shown demonstrated similar behavior to that of Table 6.3.

In order to determine the benefit of the two deflation techniques, experiments were repeated without the use of locking or purging. In addition, all the unwanted Ritz values were used as shifts, converged or not. The first run used the same parameters as given in Table 6.3. After 210 matrix vector products, the iteration converged to six Ritz values. But the second copy of the fifth smallest eigenvalue was not among the final six. The value of $p$ was increased to twenty three with the same results.

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

**Table 6.3** Convergence history for Example two

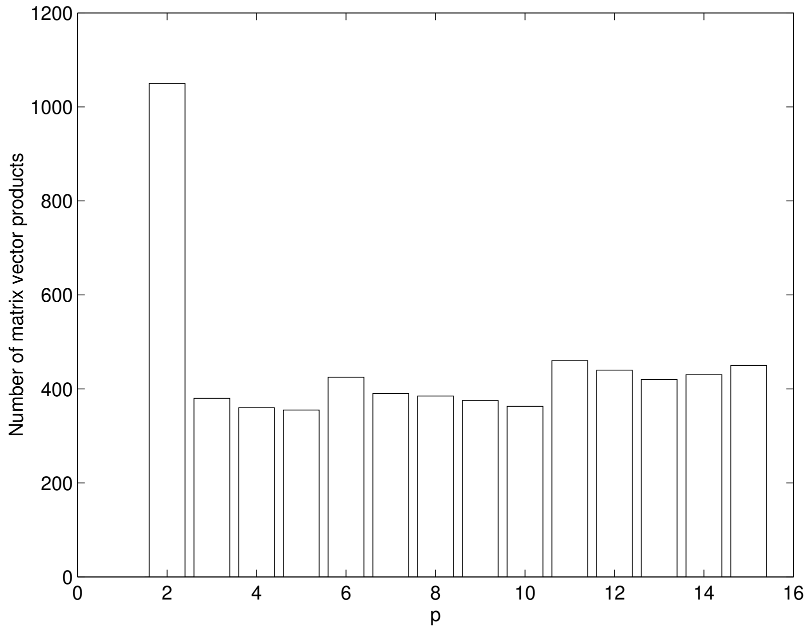


**Figure 6.3** Bar graph of the number of matrix vector products used by an IRA-iteration for Example 2 as a function of $p$.

### 6.6.3 Example 3

The following example shows the behavior of the iteration on a matrix with a very ill conditioned basis of eigenvectors. Define the Clement tridiagonal matrix [41] of order $n+1$

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

The eigenvalues are $\pm n, \pm n-2, \cdots, \pm 1$ and zero if $n$ is even. We note that $B_{n+1} = S_{n+1}A_{n+1}S_{n+1}^{-1}$ where $S_{n+1}^2 = \mathrm{diag}(1, \frac{n}{1}, \frac{n}{1}\frac{n-1}{2}, \cdots, \frac{n!}{n!})$ is a diagonal matrix. Thus the condition number of the basis of eigenvectors for $B_{n+1}$ is $\|S_{n+1}\| \, \|S_{n+1}^{-1}\|$ which implies that the eigenvalue problem for $B_{n+1}$ is quite ill conditioned. An IRA-iteration was used to compute the $k = 4$ largest in magnitude eigenvalues of $B_{1000}$. The number of shifts used was $p = 16$ and the convergence tolerance $\epsilon$ was set equal to $10^{-6}$. With these choices of $k$ and $p$, the iteration stores at most twenty Arnoldi vectors.

Let the diagonal matrix $D_4$ denote the eigenvalues of the upper triangular matrix $R_4$ computed by the iteration. The diagonal matrix $\Lambda_4 \in \mathbf{R}^{4\times 4}$ contains the four largest in magnitude eigenvalues. Table 6.4 shows the results attained.

Although the iteration needed a large number of matrix vector products, the iteration was able to extract accurate Ritz values given the convergence tolerance.

### 6.6.4 Example 4

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

$$\upsilon_1 = 10^{-6}, \upsilon_{i=2:8} = i \cdot 10^{-3}, \upsilon_{9:10} = 1,$$

on the diagonal. Since the eigenvalues of a matrix are invariant under orthogonal similarity transformations, using an IRA-iteration on $T$ with a randomly generated starting vector is general. An IRA-iteration was used to compute an approximation to the smallest eigenvalue. The number of shifts used was $p = 3$ and the convergence tolerance $\epsilon$ was set equal to $10^{-3}$. Table 6.5 shows the results attained.

Another experiment was run with the locking and purging mechanisms turned off. Additionally, all unwanted Ritz values were used as shifts. The same parameters were used as in Table 6.5 but the iteration now consumed forty one matrix vector products. As in the results for Table 6.5, the modified iteration converged to one of the dominant eigenvalues after one iteration. After six iterations, the leading block of $H_4$ split off, having converged to the invariant subspace corresponding to $\upsilon_{9:10}$. But since purging was turned off, the modified iteration had to continue attempting to converge to $\upsilon_1$ using only the lower block of order two in $H_4$. Incidently, if the iteration instead simply discarded the leading portion of the factorization corresponding to $\upsilon_{9:10}$ after the sixth iteration, convergence to $\upsilon_1$ never occurred. Crucial to the success of an IRA-iteration is the ability to deflate converged Ritz values in a stable manner. Both purging and locking allow faster convergence.

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

**Table 6.4** Convergence history for Example three

| IRA-iteration on $T$ | | |
|---|---|---|
| $k = 1$ and $p = 3$ with convergence tolerance is $\epsilon = 10^{-3}$ | | |
| Iteration | Ritz values Locked | Ritz values Purged |
| 1 | 0 | 1 |
| 15 | 1 | 1 |
| Totals | 1 | 2 |
| Number of matrix vector products | | 32 |
| $\\|TQ_1 - Q_1R_1\\|/v_1 \approx 10^{-3}$ | | |
| $\\|Q_1^T T Q_1 - R_1\\|/v_1 \approx 10^{-3}$ | | |
| $\\|Q_1^T Q_1 - I_1\\| \approx 10^{-15}$ | | |
| $\\|R_1 - v_1\\|_\infty / v_1 \approx 10^{-3}$ | | |

**Table 6.5** Convergence history for Example four

# Chapter 7

# Maintaining Orthogonality during an IRA-iteration

Probably the single most important factor governing the robust implementation of an IRA-iteration is that of computing an orthogonal set of Arnoldi vectors defined by the columns of $V_k$ in Algorithm 2.2 of Chapter 2. If Algorithm 2.2 of Chapter 2 is used to compute an Arnoldi factorization, a point is typically reached where the columns of the Arnoldi matrix constructed will no longer be orthogonal to the residual vector. Thus, we require a computational procedure that monitors the possible loss of orthogonality in an inexpensive manner. Additionally, an efficient and stable computational procedure is needed to enforce orthogonality when needed.

The Arnoldi/Lanczos factorizations fell from favor among numerical analysts due to the observed loss of orthogonality soon after their discovery. The work of Paige [57] revived the Lanczos factorization since it explained the significance of the loss of orthogonality that occurred in actual computation. This chapter introduces the additional difficulties associated with nonsymmetric $A$ and reviews the ways in which orthogonality may be enforced. We first explain the loss of orthogonality of an Arnoldi factorization in § 7.1. The significance of the loss of orthogonality during the Lanczos iteration is discussed in § 7.2. The different approaches used to ensure orthogonality are surveyed in § 7.3.

## 7.1 Orthogonalization and the Arnoldi Factorization

Computing the Arnoldi factorization in finite precision gives

$$A\hat{V}_k = \hat{V}_k\hat{H}_k + \hat{f}_k e_k^T + R_k, \qquad (7.1.1)$$

where $R_k \in \mathbf{R}^{n\times k}$ accounts for the roundoff error and hatted quantities are computed analogues of those in Algorithm 2.2. The residual $\hat{f}_k$ is the computed projection of $A\hat{V}_k e_k \equiv A\hat{v}_k$ onto the $\mathcal{R}(\hat{V}_k)$: $\hat{f}_k = (I - \hat{V}_k\hat{V}_k^T)A\hat{v}_k$. Figure 7.1 shows this geometric relationship. From Algorithm 2.2 of Chapter 2, the residual $\hat{f}_k$ associated with the length $k$ Arnoldi factorization becomes the $(k+1)$-th Arnoldi vector.

A forward error analysis shows that $\|R_k\| = O(\epsilon_M)\|A\|$ where $\epsilon_M$ designates the machine precision. Although equation (7.1.1) is an exact relationship it does not follow that $\|\hat{V}_k^T \hat{f}_k\| = \|\hat{f}_k\|\epsilon_k$ where $\epsilon_k \approx \epsilon_M$. A robust implementation computing an Arnoldi factorization has $\hat{V}_k^T \hat{V}_k = I_k + E_k$ where $\|E_k\| \approx \epsilon_M$. Thus, the loss of orthogonality may be studied by analyzing the construction of $\hat{f}_k$ and the resulting vector $\hat{V}_k^T \hat{f}_k$. Numerical difficulties may be expected when $\hat{f}_k$ is nearly in the $\mathcal{R}(\hat{V}_k)$ or equivalently, the angle $\phi$ in Figure 7.1 is small.

## 7.2 Loss of Orthogonality

As mentioned after Algorithm 2.2 of Chapter 2, a three term recurrence may be used to compute the residual vector $\hat{f}_k$ when $A$ is symmetric. Unfortunately, computing in floating point arithmetic removes the possibility of an exact three term recurrence: Since the columns of $\hat{V}_k$ are only approximately orthogonal, the computed $\hat{f}_k$ depends on all the columns of $\hat{V}_k$. The work of Paige [57] was the first to analyze the effects of floating point arithmetic upon the Lanczos factorization. Bai [5] recently analyzed the nonsymmetric Lanczos procedure. Both Paige and Bai demonstrate that a loss of orthogonality is accompanied by a group of Ritz pairs emerging as excellent approximations to eigenpairs of $A$. The Arnoldi factorization lacks a similar result.

If orthogonality is not enforced, as the Lanczos factorization is extended, further copies of the "converged" Ritz values emerge. Determining whether these spurious copies are not actual eigenvalues of $A$ of multiplicity greater than one is not an easy task. Cullum and Willoughby [21] present heuristics that attempt to distinguish the spurious Ritz values from the actual multiple ones.

For symmetric $A$, Simon [81] presents a comprehensive study of the impact orthogonalization methods have on the Lanczos iteration using the three term recurrence. This includes the work of Parlett and Scott [66] on *selective* orthogonalization. The analysis presented by Paige shows that the computed residual vector $\hat{f}_k$ losses the most orthogonality in the direction of the Ritz vectors associated with the Ritz values that are nearly eigenvalues of $A$. Selective orthogonalization is a strategy that seeks to correct the loss of orthogonality in only these "converged" directions. We remark that the locking of Ritz pairs with small Ritz estimates presented in Chapter 6 is also a selective orthogonalization method.

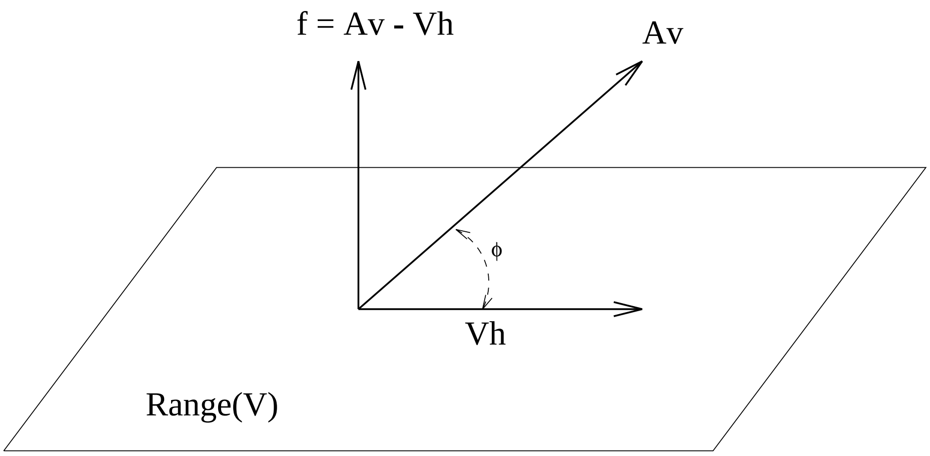


**Figure 7.1** Projecting $Av_k \equiv Av$ onto the column space of $V_k \equiv V$ and its orthogonal compliment.

## 7.3 Practical Implementations

The problem of computing an orthogonal residual vector $\hat{f}_k$ is equivalent to updating the approximate QR factorization of

$$(7.3.1) \qquad \begin{bmatrix} \hat{V}_k & A\hat{v}_k \end{bmatrix} = \begin{bmatrix} \hat{V}_k & \hat{\beta}_{k+1}^{-1}\hat{f}_k \end{bmatrix} \begin{bmatrix} I_k & \hat{h}_{k+1} \\ 0 & \hat{\beta}_{k+1} \end{bmatrix},$$

where we use the notation of Algorithm 2.2 of Chapter 2. The factorization exists as long as $\hat{\beta}_{k+1}$ is not equal to zero. From Figure 7.1 we see that $\hat{\beta}_{k+1} = \|A\hat{v}_k\| \sin\phi$ implying that for small $\phi$ the computed residual $\hat{f}_k$ has probably suffered cancelation. We emphasize that this cancelation is responsible for the loss of orthogonality between $\hat{V}_k$ and $\hat{f}_k$ even though $\|f_k - \hat{f}_k\| = O(\epsilon_M)\|A\hat{v}_k\|$. Theorems 1 and 2 in Hoffmann [42] establish these important relationships.

There are several ways to compute a residual $\hat{f}$ that is numerically orthogonal to the columns of $\hat{V}_k$. Hoffmann [42] analyzes in detail iterative algorithms for computing the QR factorization of a matrix using Gram–Schmidt methods. In the special case where $\hat{V}_k$ is a single vector, an unpublished result of Kahan found in Parlett [61, pages 105–109], shows that orthogonality to working precision is accomplished with at most one step of re-orthogonalization. Their decision to perform a re-orthogonalization is based on whether the cosine of the angle between the computed projection $\hat{f}_1$ and $A\hat{v}_1$ is less than some prescribed tolerance. This leads Saad [78, page 177] along with Reichel and Gragg [67, page 372] to conjecture that at most one step of re-orthogonalization suffices for the more general result of orthogonalizing one vector against a group of others. Although widely believed to be true, there exists no proof for how many re-orthogonalizations are required for the more general case of orthogonalizing a vector against a group of others. For example, Björck [14] states that Hoffmann's analysis proves that at most one re-orthogonalization suffices for the more general case but no proof is offered. Hoffmann's extensive experimentation never revealed the need for a second orthogonalization but that this would always be true was never rigorously justified.

The decision to perform another step of orthogonalization for the more general case required by Algorithm 2.2 is essentially the same as for the two vector case. If the ratio $\|A\hat{v}_k\|/\|\hat{f}_k\| = \sin\phi$ is less than a prescribed tolerance $\eta$ then a re-orthogonalization of $\hat{f}_k$ against all the columns of $\hat{V}_k$ is performed. Performing the first re-orthogonalization step for the Arnoldi factorization in equation (7.1.1) results in

$$A\hat{V}_k \;=\; \hat{V}_k(\hat{H}_k + g_k e_k^T) + (\hat{f}_k - \hat{V}_k g_k)e_k^T + R_k, \tag{7.3.2}$$

where $g_k = \hat{V}_k^T \hat{f}_k$. The goal is to force $\|\hat{V}_k^T(\hat{f}_k - \hat{V}_k g_k)\| = O(\epsilon_M)\|\hat{f}_k - \hat{V}_k g_k\|$. We remark that the eigenvalues of $\hat{H}_k + g_k e_k^T$ now approximate those of $A$. The next section considers determining whether this process needs to be repeated.

### 7.3.1 DGKS Analysis and Method

Daniel, Gragg, Kaufman, and Stewart [22] present a numerically stable algorithm for updating the factorization of equation (7.3.1). Their formulation is summarized by Algorithm 7.1. For clarity, the subscripts are dropped and the algorithm is used at every step $j$ of Algorithm 2.2 to compute a numerically orthogonal residual vector $f_j$.

**Algorithm 7.1**

1.1 $h \leftarrow 0$ ;

1.2 $f \leftarrow Av$ ;

1.3 Begin loop ;

  2.1 $w \leftarrow f$ ;

  2.2 $g \leftarrow V^T w$ ;

2.3 $h \leftarrow h + g$ ;

2.4 $f \leftarrow w - Vg$ ;

1.4 Repeat loop until $\|f\| > \eta\|w\|$ ;

The loop in Algorithm 7.1 is entered a second time if the sine of the angle between $f$ and $Av$ is less than or equal to $\eta$. The parameter $\eta$ is chosen to satisfy $0 < \eta < 1$. Larger values of $\eta$ will result in more work while smaller values result in a relaxing of the orthogonality between $V$ and the final $f$. Further iterations of the loop are only required if the cosine of the angle between successive approximate residual vectors is less than or equal to $\eta$. Intuitively, after the second pass through the loop, termination depends upon successive approximate residual vectors being nearly aligned. Analysis by Daniel et al. [22] shows that Algorithm 7.1 eventually terminates given some mild assumptions on the model of floating point arithmetic used.

### 7.3.2 Classical and Modified Gram-Schmidt Orthogonalization

Algorithm 7.1 is an implementation of iterative *classical* Gram–Schmidt (CGS) orthogonalization. It is well known that CGS orthogonalization is not a stable algorithm for computing the QR factorization of a matrix. On the other hand, a simple rearrangement of the CGS process, the *modified* Gram–Schmidt algorithm (MGS) is conditionally stable. If we denote the $j$-th column of $\hat{V}_k$ by $\hat{v}_j$, the MGS and CGS algorithms for computing $\hat{f}_k$ are mathematically equivalent to

$$\hat{f}_k \leftarrow (I_n - \hat{v}_k\hat{v}_k^T)\cdots(I_n - \hat{v}_1\hat{v}_1^T)A\hat{v}_k, \tag{7.3.3}$$

$$\hat{f}_k \leftarrow (I_n - \hat{V}_k\hat{V}_k^T)A\hat{v}_k, \tag{7.3.4}$$

respectively. In exact arithmetic, both variants are the same. However, as Björck [13] showed, both may compute drastically different residual vectors in floating point arithmetic.

Using MGS orthogonalization to compute the QR factorization of equation (7.3.1) results in

$$\|\hat{Q}_{k+1}^T\hat{Q}_{k+1} - I_{k+1}\| \approx \kappa(\hat{R}_{k+1})\epsilon_M,$$

where $\hat{Q}_{k+1} = \begin{bmatrix} \hat{V}_{k+1} & \hat{\beta}_{k+1}^{-1}\hat{f}_k \end{bmatrix}$ and $\hat{R}_k = \begin{bmatrix} I_k & \hat{h}_{k+1} \\ 0 & \hat{\beta}_{k+1} \end{bmatrix}$, and the condition number of $\begin{bmatrix} \hat{V}_k & A\hat{v}_k \end{bmatrix}$ is approximated by $\kappa(\hat{R}_{k+1}) \equiv \|\hat{R}_{k+1}^{-1}\|\|\hat{R}_{k+1}\|$. Thus, a small $\hat{\beta}_{k+1}$ gives a large condition number and hence MGS may not be stable.

However, Hoffmann's [42] analysis shows that the iterative versions of CGS and MGS orthogonalization, i.e. performing re-orthogonalizations to ensure orthogonality, are stable. From equation (7.3.4), the main computation of CGS orthogonalization involves the matrix–vector products $\hat{h}_{k+1} = \hat{V}_k^T \hat{w}_k$ and $\hat{w}_k - \hat{V}_k \hat{h}_{k+1}$ where $\hat{w}_k A \hat{v}_k$. Hence, iterative CGS orthogonalization is better suited for vector and parallel computing because of the matrix vector products. Instead, iterative MGS orthogonalization involves a recurrence of vector–vector operations $\hat{\gamma}_j = \hat{v}_j^T \hat{w}_k$ and $\hat{w}_k - \hat{v}_j \hat{\gamma}_j$ to compute the residual.

### 7.3.3 Using Householder Transformations

Another alternative that must be mentioned is that of employing Householder transformations as introduced by Walker [95]. Walker presents an algorithm for computing a sequence of Householder matrices $P_1, \ldots, P_k$ so that a length $k$ Arnoldi factorization is constructed for $P_k \cdots P_1 A P_1 \cdots P_k$. Saad [78, page 177] compares the cost of the two Gram–Schmidt variants with Walker's Householder approach. For modest values of $k$, the Householder approach requires about twice as many floating point operations as the iterative Gram–Schmidt one if no re-orthogonalizations are required—an unlikely occurrence. The Householder and the iterative Gram–Schmidt orthogonalizations methods for computing a length $k$ Arnoldi factorization are roughly the same when every column of $\hat{V}_k$ requires a re-orthogonalization. In addition, Walker considers the efficient implementation of the Householder approach on a parallel machine. Further study is needed to determine the comparative numerical behavior as well as the efficiency of the competing orthogonalization algorithms.

### 7.3.4 ARPACK Software

The ARPACK [49] software currently uses CGS with possible re-orthogonalization at each step. This remains feasible within an IRA-iteration since storage requirements for the Arnoldi basis vectors may be fixed in advance of the iteration. The implementation is efficient since the level 2 BLAS [26] are employed. The matrix–vector multiplications often allow the underlying architecture of the computer to be more efficiently utilized. Parallel and vector computers exemplify this behavior.

The actual choice for the parameter $\eta$ is as follows. When $A$ is symmetric, the value of $\eta = .5 \equiv \sin \pi/6$ results in a good compromise between maintaining an orthogonal set of Lanczos vectors without an unnecessary amount of re-orthogonalizations. For nonsymmetric $A$, the value of $\eta = 1/\sqrt{2} \equiv \sin \pi/2$ achieved the same goal. Work is underway to better understand the selection of $\eta$ and its impact upon the numerical orthogonality of the Arnoldi vectors.

# Chapter 8

# Some Practical Aspects for the Convergence of an IRA-iteration

The determination of the parameters $k$ and $p$ needed during an IRA-iteration requires further analysis as mentioned at the end of § 4.2 of Chapter 4. The value of $k$ is typically the number of eigenvalues of $A$ requiring approximation. At present, there is no a-priori analysis to guide the selection of $p$ relative to $k$. Increasing $p$ relative to $k$ usually decreases the required number of matrix vector products with $A$ needed by Algorithm 4.2 but it also increases the work and storage required to maintain the orthogonal Arnoldi basis vectors. The optimal cross-over value of $p$ depends upon $A$'s spectral properties and the underlying computer system.

One of the goals of this chapter is to present some heuristics and formal analysis that help in selecting of $p$ relative to $k$. A connection was made between an implicitly shifted QR-iteration and the IRA-iteration in Chapters 3 and 4. There is also a well known connection between simultaneous, or subspace, iteration and the QR-iteration. Subspace iteration is an extension of the simple power method applied to a starting matrix consisting of linearly independent vectors. When the columns of the starting matrix are orthonormal, subspace iteration is also referred to as orthogonal iteration. Thus, we may then make use of the practical knowledge known about orthogonal/subspace iteration methods.

Simple orthogonal iteration is introduced in § 8.1. A more elaborate version, shifted orthogonal iteration is the subject of § 8.2. Comparing orthogonal iteration and an IRA-iteration is considered in § 8.3 including an adaptive procedure for preventing stagnation and accelerating the convergence of the iteration. An implicitly shifted orthogonal iteration algorithm, analogous to the IRA-iteration, is introduced in the final section.

## 8.1 Orthogonal Iteration

Suppose that $AQ = QR$ is a real Schur decomposition where we partition $Q = \begin{bmatrix} Q_k & \bar{Q}_{n-k} \end{bmatrix}$ and the eigenvalues are ordered in descending order of magnitude along the quasi-diagonal of $R$. If $|\lambda_k| > |\lambda_{k+1}|$ then $D_k(A) = \mathcal{R}(Q_k)$ is said to be the *dominant* invariant subspace of dimension $k$ for $A$.

Simple orthogonal iteration is defined by the following procedure:

**Algorithm 8.1** (Simple Orthogonal Iteration)

1.1 Initialize : $U_k^{(1)} \leftarrow \begin{bmatrix} e_1 & e_2 & \cdots & e_k \end{bmatrix}$ ;

1.2 For $j = 1, 2, \ldots$ ;

  2.1 $W_k^{(j)} = AU_k^{(j)}$ ;

  2.2 Compute the QR factorization $U_k^{(j+1)} R_k^{(j+1)} = W_k^{(j)}$ ;

1.3 End $j$.

Golub and Van Loan [35, page 354] show that if

$$\mathcal{D}_k(A^T)^{\perp} \cap \operatorname{Span}\{e_j\}_{j=1}^{k} \;=\; \{0\}, \tag{8.1.1}$$

then $\mathcal{R}(U_k^{(j)}) \to \mathcal{D}_k(A)$ as $j \to \infty$ and rate of convergence is proportional to $|\lambda_{k+1}/\lambda_k|$. Thus, $(U_k^{(j)})^T AU_k^{(j)} \equiv T_k^{(j)}$ is converging to $R_k = Q_k^T AQ_k$. The geometrical interpretation of the subspace condition in equation (8.1.1) is that a vector in $\operatorname{Span}\{e_j\}_{j=1}^{k}$ must have a nonzero component in the direction of some vector in $\mathcal{D}_k(A)$. Since $AQ = QR$ implies that $A^TQ = QR^T$, we may equate the last $n-k$ columns to obtain that $\mathcal{D}_k(A^T)^{\perp} = \mathcal{R}(\bar{Q}_{n-k})$.

As Golub and Van Loan [35, page 355] also observe, the QR-iteration is orthogonal iteration in disguise. Consider the identities

$$T_n^{(j)} = (U_n^{(j)})^T AU_n^{(j)} = (U_n^{(j)})^T W_n^{(j)} = (U_n^{(j)})^T U_n^{(j+1)} R_n^{(j+1)}, \tag{8.1.2}$$

and

$$\begin{aligned} T_n^{(j+1)} &= (U_n^{(j+1)})^T AU_n^{(j+1)} \\ &= (U_n^{(j+1)})^T AU_n^{(j)} (U_n^{(j)})^T U_n^{(j+1)} \\ &= R_n^{(j+1)} (U_n^{(j)})^T U_n^{(j+1)}. \end{aligned} \tag{8.1.3}$$

The identity in equation (8.1.2) computes the QR factorization of $R_n^{(j)}$ while the second in equation (8.1.3) multiplies the factors in reverse order to get $R_n^{(j+1)}$— successive orthogonal iterations define a QR step with shift zero ! This is also implied by Theorem 3.2. We remark that if $A$ is first reduced to upper Hessenberg form $H$, and Algorithm 8.1 is used with $A$ replaced with $H$, then $H^{(j)} = T_n^{(j)}$ and $(U_n^{(j)})^T U_n^{(j+1)} = Q^{(j)}$ where zero shifts are used.

## 8.2 Shifted Orthogonal Iteration

The following extension of orthogonal iteration allows a set of shifts to be applied. This allows the possibility of converging to another invariant subspace of $A$ besides the dominant one. We present the algorithm first and then discuss its many features at some length.

**Algorithm 8.2** (Shifted Orthogonal Iteration)
function $[U_k, T_k] =$ `orthit`$(A, k, p)$

Output: $AU_k - U_k T_k = F_k$ where the residual matrix $F_k$ is small in norm, $U_k^T U_k = I_k$, and $U_k^T F_k = 0$ and $T_k$ is upper quasi-triangular.

1.1 Initialize : $U_{k+p}^{(1)} \leftarrow \begin{bmatrix} e_1 & e_2 & \cdots & e_{k+p} \end{bmatrix}$ ;

1.2 For $j = 1, 2, \ldots$

  2.1 $W_{k+p}^{(j)} \leftarrow \psi_{m_j}^{(j)}(A) U_{k+p}^{(j)}$ where $\mathcal{P}_{m_j}^{(j)}(\lambda) \equiv (\lambda - \tau_1^{(j)}) \cdots (\lambda - \tau_m^{(j)})$ ;

  2.2 Compute the QR factorization : $Q_{k+p}^{(j)} R_{k+p}^{(j)} = W_{k+p}^{(j)}$ ;

  2.3 $W_{k+p}^{(j)} \leftarrow AQ_{k+p}^{(j)}$ ; $B_{k+p}^{(j)} \leftarrow (Q_{k+p}^{(j)})^T W_{k+p}^{(j)}$ ;

  2.4 Compute the real Schur decomposition $B_{k+p}^{(j)} Z_{k+p}^{(j)} = Z_{k+p}^{(j)} T_{k+p}^{(j)}$ with the $k$ wanted eigenvalues in the leading principal matrix $T_k^{(j)}$, of order $k$, in $T_{k+p}^{(j)}$.

  2.5 $U_{k+p}^{(j+1)} \leftarrow Q_{k+p}^{(j)} Z_{k+p}^{(j)}$ ;

  2.6 Determine convergence ; Deflate converged Ritz vectors ; Modify $p$ if desired ;

1.3 End $j$.

We now consider many of the details that will lead to a robust implementation of Algorithm 8.2. As we shall see, many of these details will carry over to a robust implementation of an IRA-iteration. In particular, we consider the two codes, `EA12` by Duff and Scott [28] and `SRRIT`, by Bai and Stewart [10], as model implementations. We remark that `SRRIT` is only set up to compute $A$'s dominant invariant subspace and `EA12` also computes this space as well the invariant subspace corresponding to $A$'s right-most or left-most eigenvalues.

Line 2.1 applies the $m$ shifts $\tau_i^{(j)}$. In order to avoid the use of complex arithmetic, if any shift has a nonzero imaginary part, its complex conjugate is also a shift during the same cycle of iteration. As with the IRA-iteration, there are many choices. One could use an exact shift strategy, applying the unwanted $m = p$ eigenvalues of $T_{k+p}^{(j-1)}$ as shifts during the $j$-th iteration. This leaves an arbitrary choice of during the first iteration. As discussed in [28], the matrix polynomial $\mathcal{P}_m^{(j)}(A)$ is not formed. Rather, the columns of $\mathcal{P}_{m_j}^{(j)}(A)U_{k+p}^{(j)}$ are formed using the recurrence $W_{k+p}^{(j)} \leftarrow (A - \tau_i^{(j)} I)U_{k+p}^{(j)}$ for $i = 1, \ldots, m$. If a Chebyshev polynomial is used, the three term recurrence should be employed [76]. We note that if all the shifts applied are zero, then simple orthogonal iteration is recovered.

The degree $m$ of the polynomial applied should not be chosen too large for otherwise the columns of $W_{k+p}^{(j)}$ will become linearly dependent. However, a small value of $m$ leads to unnecessary orthogonalizations. The important property is that the columns of $W_{k+p}^{(j)}$ of line 2.1 remain numerically linearly independent. Guidelines are provided in [10, 28] for the software determining the degree in an adaptive fashion.

Line 2.5 uses a *Schur–Rayleigh–Ritz*, SRR, step to ensure that $U_k e_i$ converges to the $i$-th Schur vector corresponding to some ordering of the eigenvalues of $A$. Originally introduced by Stewart [88] within the context of simultaneous iteration, $\mathcal{P}_{m_j}^{(j)}(\lambda) = \lambda^{m_j}$, performing a SRR step gives that $U_{k+p}^{(j)} e_i$ converges to the Schur vector associated with the $i$-th largest in magnitude eigenvalue of $A$. Each column of $U_{k+p}^{(j)}$ converges at the rate of $|\lambda_{k+p+1}/\lambda_i|$ where $A$'s eigenvalues are ordered in descending order of magnitude. Thus, the initial columns of $U_{k+p}^{(j)}$ converge faster than the latter ones and increasing the value of $p$ allows a faster rate. We remark that a SRR step does not actually accelerate convergence: The effect is to unscramble the approximations to Schur vectors already present in the column space of $U_{k+p}^{(j)}$. Stewart [88] made this observation and both Chatelin [18, page 253] and Saad [75, page 132] give elegant but elementary proofs.

Both `EA12` and `SRRIT` compute the $l$-the column of the residual matrix $AU_{k+p}^{(j+1)} - U_{k+p}^{(j+1)} T_{k+p}^{(j)}$, where the first $l-1$ columns have already converged. Bai and Stewart further discuss the convergence of `SRRIT` to the invariant subspace corresponding to nearly equimodular eigenvalues. As the columns of $U_{k+p}^{(j)}$ converge, deflation techniques such as locking should be employed.

### 8.2.1 Convergence of Shifted Orthogonal Iteration

Algorithm 8.2 requires a non-negative value of $p$. The last $p$ columns of $U_{k+p}^{(j)}$ are called *guard* vectors. When $\mathcal{P}_{m_j}^{(j)}(\lambda) = \lambda^{m_j}$, increasing the number of guard vectors accelerates the convergence of Algorithm 8.2 to the wanted invariant subspace. However, the number of matrix vector products with $A$ also increases as well the work necessary to maintain the orthogonality of $U_{k+p}^{(j)}$. As with an IRA-iteration, the decision on how to choose $p$ depends upon many factors.

Watkins and Elsner [100, pages 29–35] provide convergence results for a non-stationary, i.e. shifted, subspace iteration which gives an indication of how $p$ might be selected. We present one of their results, which is seen to be a generalization of the Golub and Van Loan [35] one for the non-stationary case, referenced in § 8.1.

**Theorem 8.3** Let $A \in \mathbf{R}^{n\times n}$. Suppose that the eigenvalues of $A$ are all of algebraic multiplicity one and denote by $\lambda_1, \lambda_2, \ldots, \lambda_n$ some ordering of $A$'s eigenvalues. Let a real Schur decomposition $AQ = QR$ be given where $Q = \begin{bmatrix} Q_{k+p} & \bar{Q}_{n-k-p} \end{bmatrix}$ and $Q_{k+p}^T A Q_{k+p} \equiv R_{k+p}$ contains the eigenvalues $\lambda_1, \ldots, \lambda_{k+p}$ where complex conjugate pairs are kept together: $\lambda_i = \bar{\lambda}_j$ implies that $i, j \leq k+p$. Define the matrix $U_{k+p}^{(1)} = \begin{bmatrix} e_1 & e_2 & \cdots & e_{k+p} \end{bmatrix}$. Let $\mathcal{P}_{M_J}(\lambda) = \psi_{m_1}^{(1)}(\lambda) \cdots \psi_{m_J}^{(J)}(\lambda)$ be the product of a sequence polynomials for some positive integer $J$ such that $\mathcal{P}_{M_J}(\lambda_i) \neq 0$ for $i = 1, \ldots, k+p$ and $M_J \equiv m_1 \cdots m_J$ Also assume that if any root of $\psi_{m_i}^{(i)}(\lambda)$ has a nonzero imaginary part, its complex conjugate is also a root.

If $\mathcal{R}(U_{k+p}^{(1)}) \cap \mathcal{R}(\bar{Q}_{n-k-p}) = \{0\}$ and

$$\frac{\max\limits_{k+p+1\leq i\leq n} |\mathcal{P}_{M_J}(\lambda_i)|}{\min\limits_{1\leq i\leq k+p} |\mathcal{P}_{M_J}(\lambda_i)|} \quad \rightarrow \quad 0, \tag{8.2.1}$$

as $J \rightarrow \infty$ then the non-stationary iteration defined by $\mathcal{P}_{M_J}(A)U_{k+p}^{(0)}$ converges in the sense that $\mathcal{R}(U_{k+p}^{(J)}) \rightarrow \mathcal{R}(Q_{k+p})$.

**Proof** See Theorem 5.1 in [100, page 29]. □

The geometrical interpretation of the subspace condition

$$\mathcal{R}(U_{k+p}^{(1)}) \cap \mathcal{R}(\bar{Q}_{n-k-p}) \quad = \quad \{0\}$$

of the theorem is similar to that given in § 8.1 for simple orthogonal iteration. Some vector in Span$\{e_j\}_{j=1}^{k+p}$ must have a nonzero component in the direction of some vector in $\mathcal{R}(Q_{k+p})$.

If $\mathcal{P}_{m_i}^{(i)}(\lambda) = \lambda^{m_i}$ and $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ where $|\lambda_k| > |\lambda_{k+1}|$ then traditional subspace iteration is recovered. The ratio in equation (8.2.1) gives the rate of convergence $|\lambda_{k+p+1}/\lambda_{k+p}|$ for $U_{k+p}^{(j)}$ approaching $A$'s dominant invariant subspace.

For more general shifting strategies, the ratio in equation (8.2.1) gives the global rate of convergence of Algorithm 8.2 to an invariant subspace. Using SRR steps, we formally extend Stewart's result to $U_{k+p}e_l$ converging at the rate of

$$\frac{\max\limits_{k+p+1 \leq i \leq n} |\mathcal{P}_{M_J}(\lambda_i)|}{\min\limits_{1 \leq i \leq l} |\mathcal{P}_{M_J}(\lambda_i)|}. \tag{8.2.2}$$

This convergence rate may be significantly better than the one given by $|\lambda_{k+p+1}/\lambda_l|$ when the interest is in the Schur vectors associated with $A$'s dominant invariant subspace. Since the convergence rate is a complicated function involving the shifts applied, it is not an obvious decision on how to select the optimal value of $p$ that leads to the optimal convergence. Further numerical experimentation is needed.

Again, as noted in § 4.2 of Chapter 4 with an IRA-iteration, the success of Algorithm 8.2 depends upon the quality of the shifting strategy.

## 8.3 Comparing Orthogonal and an IRA-iteration

It is instructive to compare an IRA-iteration with that of Algorithm 8.2. For each of Algorithms 8.2 and 4.2 (an IRA-iteration) of Chapter 4, we have

$$\begin{aligned} AV_{k+p}^{(j+1)} &= V_{k+p}^{(j+1)} H_{k+p}^{(j+1)} + f_{k+p}^{(j+1)} e_{k+p}^T, \\ AQ_{k+p}^{(j)} &= Q_{k+p}^{(j)} B_{k+p}^{(j)} + F_{k+p}^{(j)}, \end{aligned}$$

respectively. We comment that at this point of each algorithm, a polynomial $\psi^{(j)}(\lambda)$ has been applied. Algorithm 8.2 applies the polynomial matrix $\psi^{(j)}(A)$ at the beginning of its iteration to the columns of $U_{k+p}^{(j)}$ while Algorithm 4.2 applies the polynomial during the implicit application of the shifts.

Note that $(V_{k+p}^{(j+1)})^T A V_{k+p}^{(j+1)} = H_{k+p}^{(j+1)}$ and $(Q_{k+p}^{(j)})^T A Q_{k+p}^{(j)} = B$. Both matrices represent the orthogonal projections of $A$ onto two, in general, different column spaces. Suppose the same shifts are applied during each iteration of Algorithms 8.2 and 4.2: $\psi_p^{(i)}(\lambda) = \psi_{m_i}^{(i)}(\lambda)$ for $i = 1, \ldots, j$ where the polynomials $\psi_p^{(i)}(\lambda)$ of degree $p$ were defined in the development leading up to equation (4.2.7) of § 4.2 in Chapter 4. The column spaces are:

$$\begin{aligned}
\mathcal{R}(V_{k+p}^{(j+1)}) &= \mathcal{K}_{k+p}(A, v_1^{(j+1)}) \\
&= \mathcal{K}_{k+p}(A, \mathcal{P}_{jp}(A)v_1^{(1)}), \\
\mathcal{R}(Q_{k+p}^{(j)}) &= \mathcal{R}(\psi_{m_j}^{(j)}(A)U_{k+p}^{(j)}) \\
&= \mathcal{R}(\mathcal{P}_{jp}(A)U_{k+p}^{(1)}).
\end{aligned}$$

Moreover, suppose that $v_1^{(1)} = e_1$ and recall that $U_{k+p}^{(1)}$ represents the first $k+p$ columns of the identity matrix. Algorithm 8.2 computes the leading $k+p$ columns of the QR factorization of $\mathcal{P}_{jp}(A)$. On the other hand, if we assume that the grade of $e_1$ is at least $k+p$, Algorithm 4.2 computes the leading $k+p$ columns of the Krylov matrix $K_{k+p}(A, \mathcal{P}_{jp}(A)e_1)$.

As explained in § 4.2 of Chapter 3, the last $p$ columns of the above Arnoldi factorization are discarded because of the fill-in suffered by $e_{k+p}^T Z^{(p)}$. Extending the ensuing length $k$ Arnoldi factorization to length $k+p$ allows, in general, a different set of Arnoldi vectors to be appended to the last $p$ columns of $V_{k+p}^{(j+1)}$ during each cycle of iteration. This is one of the major differences between Algorithms 4.2 and 8.2. Algorithm 8.2 applies a polynomial in $A$ to the same initial subspace determined by the $\mathcal{R}(U_{k+p}^{(1)})$.

Parlett [62] presents an excellent survey comparing the Lanczos and Subspace iterations for the symmetric eigenvalue problems arising in structural mechanics. The conclusion reached is that the Lanczos iteration is almost always a superior algorithm. The literature is sparse for similar comparisons between Arnoldi's and Subspace iteration for nonsymmetric eigenvalue problems. As Chatelin [18, page 281] notes, the choice between the two nonsymmetric algorithms is not so clear.

### 8.3.1 Adaptive Procedures used within an IRA-iteration

As explained in § 3.2 of Chapter 3, the QR-iteration is a nested sequence of subspace iterations. Since the IRA-iteration is just the leading portion of the QR-iteration, this section gives an indication of how to determine the value of $p$ needed by an IRA-iteration by considering the formal connections with subspace iteration.

The application of shifts during an IRA-iteration is analogous to performing a SRR step. Lines 2.3–2.4 of Algorithm 4.2 effectively apply the SRR step: The first $k$ columns of $Z^{(p)}$ span the wanted invariant subspace of $H_{k+p}^{(j)}$ and the resulting updated Arnoldi factorization

$$AV_{k+p}^{(j)}Z^{(p)} = V_{k+p}^{(j)}Z^{(p)}H_{k+p}^{(j+1)} + f_{k+p}^{(j)}e_{k+p}^{T}Z^{(p)},$$

of Line 2.4 represents the application of the SRR projection. As equation (8.2.2) indicates, the optimal choice of $p$ is a complicated decision. On the one hand, the number of shifts applied should be sufficient so that $\psi_p^{(j)}(A)$ annihilates the unwanted components of $v_1^{(j)}$. On the other hand, since application of the shifts is equivalent to a SRR step, the discussion following Theorem 8.3 indicates that too large of a value of $p$ may slow down convergence. Extensive numerical experiments dictate that the value of $p$ should be slightly decreased during each iteration.

## 8.4 Implicitly Shifted Orthogonal Iteration

The main expense of Algorithm 8.1 is the formation of matrix vector products with $A$ at lines 2.1 and 2.3. The application of the polynomial in $A$ may instead be applied implicitly through $B$ and hence the cost of Algorithm 8.1 may be reduced. We first establish the following result.

**Lemma 8.1** Let $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{k\times k}$ and $U \in \mathbf{R}^{n\times k}$ with $U^TU = I_k$. Let $AU = UB + F$ where $\hat{F} = AU - UB$. If $\mathcal{P}(\lambda) = (\lambda-\tau_1)\cdots(\lambda-\tau_m)$ then

$$\mathcal{P}(A)U = U\mathcal{P}(B) + \tilde{F}, \tag{8.4.1}$$

where $\tilde{F} = \mathcal{P}(A)U - U\mathcal{P}(B)$.

**Proof** The proof is by induction. Consider applying the polynomial of degree one ;

$$\begin{aligned} AU &= UB + F, \\ AU - \tau_1 U &= UB - \tau_1 U + F, \\ (A - \tau_1 I_n)U &= U(B - \tau_1 I_k) + F, \end{aligned}$$

and the base case is established. Suppose that equation (8.4.1) holds for all monic polynomials of degree less than or equal to $m-1$. Defining $\tilde{F} = AU - UB$ it follows that

$$\begin{aligned}(A-\tau_m I_n)\mathcal{P}(A)U &= (A-\tau_m I_n)U\mathcal{P}(B) + (A-\tau_m I_n)F, \\ &= U(B-\tau_m I_k)\mathcal{P}(B) + \tilde{F}\mathcal{P}(B) + (A-\tau_m I_n)F.\end{aligned}$$

Finally, the result on the residual follows since

$$\begin{aligned}\tilde{F}\mathcal{P}(B) + (A-\tau_m I_n)F &= (AU-UB)\mathcal{P}(B) + (A-\tau_m I_n)(\mathcal{P}(A)U - U\mathcal{P}(B)), \\ &= -UB\mathcal{P}(B) + (A-\tau_m I_n)\mathcal{P}(A)U + \tau_m U\mathcal{P}(B), \\ &= (A-\tau_m I_n)\mathcal{P}(A)U - U(B-\tau_m I_k)\mathcal{P}(B).\end{aligned}$$

$\square$

Although $m$ matrix vector products during each cycle of the iteration with $A$ may be avoided, the error in using $\mathcal{P}(B)$ is $F = \mathcal{P}(A)U - U\mathcal{P}(B)$. As the range of $U$ improves as an approximation to an invariant subspace of $A$, the error $F$ is accordingly reduced. If $AV = VT$, where $T$ is upper triangular, then a simple calculation shows that $\mathcal{P}(A)V = V\mathcal{P}(T)$ and hence the residual

$$\mathcal{P}(A)V - V\mathcal{P}(B) = V(\mathcal{P}(T) - \mathcal{P}(B)) = 0,$$

since $\mathcal{P}(B) = V^T\mathcal{P}(A)V$.

Computing the orthogonal factorization $\mathcal{P}(B) = QR$ we obtain,

$$\mathcal{P}(A)U = UQR + F. \tag{8.4.2}$$

Post-multiplying equation (8.4.2) by $Q$ results in

$$\mathcal{P}(A)(UQ) = (UQ)Q^T\mathcal{P}(B)Q + FQ, \tag{8.4.3}$$

since $RQ = Q^T\mathcal{P}(B)Q$. Thus, $m$ QR steps are performed with the set of shifts $\{\tau_i\}_{i=1}^m$. Note that post-multiplication with the orthogonal $Q$ in equation (8.4.3) does not change the size of the error $F$. Lines 2.1—2.3 of Algorithm 8.1 may be replaced to obtain the following procedure:

**Algorithm 8.4**

2.1 Compute the QR factorization : $Q_{k+p}^{(j)}R_{k+p}^{(j)} \leftarrow \mathcal{P}_{m_j}^{(j)}(B)$ where

$$\mathcal{P}_{m_j}^{(j)}(\lambda) \equiv (\lambda - \tau_1^{(j)}) \cdots (\lambda - \tau_m^{(j)}) \; ;$$

2.2 $W_{k+p}^{(j)} \leftarrow A U_{k+p}^{(j)} Q_{k+p}^{(j)}$ .

An interesting observation is comparing the application of shifts in Algorithm 4.2 with the above implicit application of shifts. Algorithm 4.2 discards the last $p$ columns due to the fill-in that occurs, in contrast to the above implicit application of shifts. ( See Figures 4.1— 4.3 of Chapter 4 for an illustration of the fill-in. )

The convergence properties and numerical behavior of the above implicitly shifted orthogonal iteration requires further investigation. For example, what is the convergence of rate of Algorithm 8.4 when the interest is in $A$'s dominant invariant subspace, i.e. using zero shifts ? If the convergence rate is competitive with Algorithm 8.2, then a significant savings in computational effort may be realized by avoiding $m$ matrix-vector products during each iteration cycle.

# Chapter 9

# Thesis Summary and Future work

This dissertation has examined Sorensen's implicitly re-started Arnoldi iteration. After an introduction to the goals and subject of the thesis in Chapter one, the second and third chapters established the connection that an IRA-iteration is mathematically equivalent to building only the leading portion of a QR-iteration of a matrix. The practical QR algorithm was considered in some detail since the major goal of this thesis is to present numerical techniques that result in a robust implementation of an IRA-iteration. Chapter 4 both investigated and surveyed the various ways in which to re-start an Arnoldi factorization. It was shown that the IRA-iteration uses the same mechanism as the implicitly shifted QR algorithm and thus enjoys its many stability properties. Chapter 5 examined the possible loss of forward stability that an IRA-iteration undergoes and considered its impact upon the Ritz values. A fundamental connection between the algorithms used to re-order a Schur decomposition and an IRA-iteration was also made. The forward instability of QR algorithm was shown to be responsible for the occasional failure of the implicit re-starting technique. A sensitivity analysis was also presented for the orthogonal reduction of a matrix to upper Hessenberg form. Thus, the forward instability of an IRA-iteration was seen to have a geometric interpretation: Small components of the starting vector that are in unwanted invariant subspaces are possibly amplified during the iteration.

Deflation techniques for an IRA-iteration were the subject of Chapter 6. The first technique, Locking, allows an orthogonal change of basis for an Arnoldi factorization which results in a partial Schur decomposition containing the converged Ritz values. The corresponding Ritz value is deflated in an implicit manner. The second technique, Purging, allows implicit removal of unwanted converged Ritz values from the Arnoldi iteration. Both deflation techniques are accomplished by working with matrices in the projected Krylov space which for large eigenvalue problems is a fraction of the order of the matrix from which estimates are sought. Since both deflation techniques are implicitly applied to the Arnoldi factorization, the need for explicit re-starting associated with all other deflation strategies is avoided. Both techniques were carefully examined with respect to numerical stability and computational results were presented. Convergence of the Arnoldi iteration is improved and a reduction in computational effort is realized. The numerical examples demonstrate how the deflation techniques remove the requirement for a block Arnoldi/Lanczos method to compute approximations to multiple or clustered eigenvalues.

The final two chapters surveyed and presented formal analysis for the practical issues associated with maintaining orthogonality of the Arnoldi vectors and choosing $p$, the number of shifts to apply. In addition, two simultaneous iteration algorithms were introduced that require further investigation.

## 9.1 Future Work

There remain several areas that require further research. The future goal is to better understand all of the practical issues that will lead to optimal convergence of an IRA-iteration.

1. Robust stopping criteria; especially for nonsymmetric eigenvalue problems. The discussion of § 2.5 in Chapter 2 gave an indication of the importance of the better understanding needed, especially the impact of the non-normality of $A$.

2. Practical convergence aspects/theory. Although Chapter 8 established a connection between an IRA-iteration and shifted orthogonal iteration, more work is required in order to determine near optimal adaptive selection of $p$ relative to $k$.

3. Reliability of an IRA-iteration. When successful, Algorithm 4.2 computes an approximate invariant subspace of $A$ of dimension $k$. However, there is no guarantee that this is the wanted invariant subspace. For example, suppose the wanted invariant subspace has an eigenvalue of multiplicity greater than one. Does an IRA-iteration correctly resolve this multiplicity ? We remark that all numerical methods for computing a few eigenvalues for a nonsymmetric matrix $A$ face this dilemma.

4. Further investigation is needed to establish a direct connection between the forward instability of an IRA-iteration and the sensitivity of reducing a matrix to upper Hessenberg form via orthogonal transformations. Theorem 5.3 gives a

geometrical interpretation of forward instability but a link with the Parlett and Le [63] condition would be interesting.

5. The generalized eigenvalue problem $Ax = Bx\lambda$. This dissertation concentrated on the case where $B = I$. When $B$ is not the identity matrix, either $A$, $B$, or a linear combination of the two must be factored. For symmetric $A$, the work of Ericsson and Ruhe [30] considers the spectral transformation Lanczos method which was further extended by Nour–Omid, Parlett, Ericsson and Jensen [56]. The ARPACK [49] software implements the techniques described in the latter study. Saad [78] discusses the many difficulties that arise for the nonsymmetric generalized eigenvalue problem. The recent work of Meerbergen and Spence [52] discusses the special but important case of $A$ nonsymmetric and $B$ symmetric positive semi-definite

6. Preconditioning techniques for an IRA-iteration. The analysis and techniques presented in this dissertation also serve to establish the viability of computing approximations to selected portions of $A$'s spectrum using a preconditioner that only needs matrix vector products. The motivation for using preconditioning for eigenvalue problems is to allow faster and more robust convergence to selected portions of $A$'s spectrum that are of interest. It is often observed that the wanted eigenvalues are not those that the Arnoldi iteration naturally converges towards. We first clarify the concept of preconditioning for eigenvalue problems. A preconditioner $\mathcal{F}$ is a transformation on $A$ that results in the matrix $\mathcal{F}(A)$. A good preconditioner results if the Arnoldi/Lanczos iterations on $\mathcal{F}(A)$ converge most rapidly towards the wanted eigenvalues of $A$ under the transformation.

   Among the most powerful preconditioners employed are those factoring and solving linear systems with $A$. An important example is the shift and invert or spectral transformation defined by $\mathcal{F}(\lambda) = (\lambda - \sigma)^{-1}$. The transformation has the affect of transforming the eigenvalues of $A$ closet to $\sigma$ into large and well separated ones for $\mathcal{F}(A)$. The eigenvectors of $\mathcal{F}(A)$ are the same as those of $A$ and the eigenvalues are related through the transformation. Saad [78] discusses shift and invert Arnoldi method for nonsymmetric eigenvalue problems. Ruhe introduces and examines the use of rational preconditioners in the series of papers [69, 70, 71]. The work of Meerbergen and Roose [51] presents an excellent overview of preconditioning for the nonsymmetric eigenvalue problem.

The primary drawback in using rational preconditioning is that linear systems involving $A$ require solution. This may prove quite inefficient and prohibitive in many eigenproblems. Although the order of $A$ is often the culprit, moderately sized eigenvalue problems may involve dense matrices that are expensive both to store and factor. This thesis demonstrates that it is often possible to converge to the extremal portions of the spectrum of $A$ using only matrix vector products or employing a polynomial preconditioner. In these situations, the expense of factoring and solving linear systems with $A$ is avoided. The decision in whether to use only polynomial preconditioning involves a tradeoff between the number of matrix vector products versus the number of matrix factorizations and linear systems solutions that are required, respectively, for solution of the eigen-problem. Further work is required in better understanding all these issues as well as the impact of other shifting strategies besides the exact one considered in this thesis. In particular, the use of an IRA-iteration for computing approximations to the interior eigenvalues of $A$ needs to be carefully examined.

7. An evaluation of software for solving large sparse nonsymmetric eigenvalue problems. The last few years has seen a vigorous research effort in numerical methods for large scale nonsymmetric eigenvalue problems. This effort is starting to be realized in high quality software. However, a review and survey of the current software and the algorithms implemented is needed. The motivation for undertaking this study is to begin the critical review necessary to compare and test the underlying algorithms used in the various software approaches and to better understand where improvements are needed. The software approaches needing review include:

    - The block nonsymmetric Lanczos algorithm [6].
    - The block Arnoldi algorithm [80].
    - The rational Krylov algorithm of Ruhe [69, 70, 71].
    - The ARPACK software package [49].
    - The simultaneous iteration algorithm of Stewart and Jennings [91, 92].
    - The two subspace iteration codes `EA12` of Duff and Scott [28], and `SRRIT` of Bai and Stewart [10].

    Other important issue include comparing Algorithms 4.2 and 4.7 of Chapter 4. Finally, a study comparing the performance of the codes in terms of storage requirements, execution times, and accuracy, and considering their suitability for solving large-scale industrial problems is underway [48].

# Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA., 1992.

[2] Anon. *Harwell Subroutine Library. A catalogue of subroutines (Release 11)*. Theoretical Studies Department, AEA Industrial Technology, Oxfordfordshire, England, 1993.

[3] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[4] J. Baglama, D. Calvetti, and L. Reichel. Iterative methods for the computation of a few eigenvalues of a large symmetric matrix. Technical report, Department of Computer Science, Stanford University, 1995.

[5] Z. Bai. Error analysis of the Lanczos algorithm for nonsymmetric eigenvalue problem. *Mathematics of Computation*, 62:209–226, 1994.

[6] Z. Bai. A spectral transformation block Lanczos algorithm for solving sparse non-hermitian eigenproblems. In J. G. Lewis, editor, *Proceedings of the Fifth SIAM Conference Applied Linear Algebra*, pages 307–311, Philadelphia, 1994. SIAM.

[7] Z. Bai, J. Demmel, and A. Mckenney. On computing condition numbers for the nonsymmetric eigenproblem. *ACM Transactions on Mathematical Software*, 19(2):202–223, June 1993. LAPACK Working Note.

[8] Z. Bai and J. W. Demmel. On a block implementation of Hessenberg multishift QR iteration. *International Journal of High Speed Computation*, 1:97–112, 1989.

[9] Z. Bai and J. W. Demmel. On swapping diagonal blocks in real Schur form. *Linear Algebra and Its Applications*, 186:73–95, 1993.

[10] Z. Bai and G. W. Stewart. SRRIT — A FORTRAN subroutine to calculate the dominant invariant subspace of a nonsymmetric matrix. Technical Report 2908, Department of Computer Science, University of Maryland, 1992. Submitted to ACM Transactions on Mathematical Software.

[11] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*, 15:820–826, 1972.

[12] M. Bennani and T. Braconnier. Stopping criteria for eigensolvers. Technical report, November 1994. Submitted to Jour. Num. Lin. Alg. Appl.

[13] Å. Björck. Solving linear least squares problems by Gram–Schmidt orthogonalization. *BIT*, 7:1–21, 1967.

[14] Å. Björck. Numerics of Gram–Schmidt orthogonalization. *Linear Algebra and Its Applications*, 197,198:297–348, 1994.

[15] A. Bojanczyk and P. Van Dooren. Reordering diagonal blocks in the Schur form. In *Linear Algebra for Large Scale and Real Time Applications*, NATO ASI Series, pages 351–352. Kluwer Academic Publishers, 1993.

[16] T. Braconnier. The Arnoldi–Tchebycheff algorithm for solving large nonsymmetric eigenproblems. Technical Report TR/PA/93/25, CERFACS, Toulouse, France, 1993.

[17] D. Calvetti, L. Reichel, and D. C. Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *ETNA*, 2:1–21, March 1994.

[18] F. Chatelin. *Eigenvalues of Matrices*. Wiley, 1993.

[19] F. Chatelin and V. Fraysée. Qualitative computing: elements of a theory for finite-precision computation. Technical report, CERFACS and THOMSON–CSF, June 1993. Lecture Notes for the Commett European Course, June 8–10, Orsay, France.

[20] 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 *Proceedings of the 1974 IEEE Conference on Decision and Control*, pages 505–509, New York, 1974.

[21] J. Cullum and R. A. Wilboughby. *Lanczos algorithms for large symmetric eigenvalue computations*, volume 1 Theory. Birkhäuser, Boston, MA., 1985.

[22] J. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization. *Mathematics of Computation*, 30:772–795, 1976.

[23] J. W. Demmel. Computing stable eigendecompositions of matrices. *Linear Algebra and Its Applications*, 79:163–193, 1986.

[24] James Demmel. *Numerical Linear Algebra*, volume 1 of *Berkeley Mathematics Lecture Notes*. Center for Pure and Applied Mathematics, Department of Mathematics, University of California, Berkeley, California, 1993.

[25] J. Dongarra, S. Hammarling, and J. Wilkinson. Numerical considerations in computing invariant subspaces. *SIAM Journal on Matrix Analysis and Applications*, 13(1):145–161, January 1992.

[26] J.J. Dongarra, J. DuCroz, S. Hammerling, and R. Hanson. An extendend set of fortran basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 14:1–17, 1988.

[27] A. A. Dubrulle. The multishift QR algorithm: Is it worth the trouble. Palo Alto Center Report G320–3558, IBM Corporation, May 1992. Revised.

[28] I. S. Duff and J. A. Scott. Computing selected eigenvalues of large sparse unsymmetric matrices using subspace iteration. *ACM Transactions on Mathematical Software*, 19(2):137–159, June 1993.

[29] T. Ericsson. On the eigenvalues and eigenvectors of Hessenberg matrices. Technical Report 10, Chalmers University of Technology and University of Göteborg, June 1990. Accepted for publication in Linear Algebra and Its Applications. Presented at the Cornelius Lanczos International Centenary Conference, Raleigh NC, 1993.

[30] T. Ericsson and A. Ruhe. The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems. *Mathematics of Computation*, 35:1251–1268, October 1980.

[31] W. Feller and G.E. Forsythe. New matrix transformations for obtaining characteristic vectors. *Quarterly of Applied Mathematics*, 8(4):325–331, January 1951.

[32] J. G. F. Francis. The QR transformation—part 1. *The Computer Journal*, 4:265–271, October 1961.

[33] J. G. F. Francis. The QR transformation—part 2. *The Computer Journal*, 4:332–345, January 1962.

[34] S. Godet-Thobie. *Eigenvalues of large highly nonnormal matrices.* PhD thesis, University Paris IX, Dauphine, Paris, France, 1993. C.E.R.F.A.C.S. Report Ref.: TH/PA/93/06.

[35] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins, second edition, 1989.

[36] G. H. Golub, S. Nash, and C. F. Van Loan. A Hessenberg–Schur method for the problem $AX + XB = C$. *IEEE Transactions on Automatic Control*, AC-24:909–913, 1979.

[37] G. H. Golub and R. Underwood. The block Lanczos method for computing eigenvalues. In J. R. Rice, editor, *Mathematical Software III*, pages 361–377, 1977.

[38] G. H. Golub and J. H. Wilkinson. Ill-conditioned eigensystems and the computation of the Jordan canonical form. *SIAM Review*, 18(4):578–619, October 1976.

[39] R. G. Grimes, J. G. Lewis, and H. D. Simon. A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 15(1):228–272, January 1994.

[40] N. J. Higham. Perturbation theory and backward error for $AX - XB = C$. *BIT*, 33:124–136, 1993.

[41] N. J. Higham. The Test Matrix Toolbox for Matlab. Numerical Analysis Report No. 237, University of Manchester, England, December 1993.

[42] W. Hoffmann. Iterative algorithms for Gram–Schmidt orthogonalization. *Computing*, 41:335–348, 1989.

[43] Z. Jia. *Some Numerical Methods for Large Unsymmetric Eigenproblems*. PhD thesis, Bielefeld University, Bielefeld, Germany, Feburary 1994.

[44] W. Karush. An iterative method for finding characteristics vectors of a symmetric matrix. *Pacific Journal of Mathematics*, 1:233–248, 1951.

[45] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *USSR Comput. Math. Phys.*, 3:637–657, 1961.

[46] C. Lanzcos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, October 1950. Research Paper 2133.

[47] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice–Hall, 1974.

[48] R. B. Lehoucq and J. A. Scott. An evaluation of software for solving large sparse unsymmetric eigenvalue problems. Technical report, 1995. In preparation.

[49] R. B. Lehoucq, D. C. Sorensen, and P. Vu. ARPACK: *An implementation of the Implicitly Re-started Arnoldi Iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix*. Available from netlib@ornl.gov under the directory scalapack.

[50] T.A. Manteuffel. Adaptive procedure for estimating parameters for the nonsymmetric Tchebychev iteration. *Numer. Math*, 31:183–208, 1978.

[51] Karl Meerbergen and Dirk Roose. Preconditioners for computing rightmost eigenvalues of large sparse nonsymmetric matrices. Technical Report TW206, Katholieke Universitet Leuven, Leuven, Belguim, November 1994. Submitted IMA Journal Numerical Analysis.

[52] Karl Meerbergen and Alastair Spence. Implicitly restarted Arnoldi with purification for the shift–invert transformation. Technical Report TW225, Katholieke Universitet Leuven, Leuven, Belguim, April 1995. Submitted to Mathematics of Computation.

[53] G. S. Miminis and C. C. Paige. Implicit shifting in the QR and related algorithms. *SIAM Journal on Matrix Analysis and Applications*, 12(2):385–400, 1991.

[54] R. B. Morgan. On restarting the Arnoldi method for large scale eigenvalue problems. *Mathematics of Computation*, 1995. Submitted for publication.

[55] N. M. Nachtigal. *A Look-Ahead Variant of the Lanczos Algorithm and its Application to the Quasi-Minimal Residual Method for Non-Hermitian Linear Systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, August 1991. Numerical Analysis Report 91-3.

[56] B. Nour-Omid, B. N. Parlett, and Thomas Ericsson Paul S. Jensen. How to implement the spectral transformation. *Mathematics of Computation*, 48(178):663–673, April 1987.

[57] C. C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices*. PhD thesis, University of London, London, England, 1971.

[58] C. C. Paige, B. N. Parlett, and H. A. Van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numerical Linear Algebra with Applications*, 2(2):115–134, 1995.

[59] B. N. Parlett. Canonical decomposition of Hessenberg matrices. *Mathematics of Computation*, 21:223–227, 1966.

[60] B. N. Parlett. Global convergence of the basic QR algorithm on Hessenberg matrices. *Mathematics of Computation*, 22:803–817, 1968.

[61] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, 1980.

[62] B. N. Parlett. The state-of-the-art in extracting eigenvalues and eigenvectors in structural mechanics. In A. K. Noor and J. T. Oden, editors, *State-of-the-art surveys in computational mechanics*, pages 201–218, New York, NY, 1989. ASME.

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

[64] B. N. Parlett and B. Nour-Omid. The use of a refined error bound when updating eigenvalues of tridiagonals. *Linear Algebra and Its Applications*, 68:179–219, 1984.

[65] B. N. Parlett and W. G. Poole. A geometric theory for the QR, LU, and power iterations. *SIAM Journal on Numerical Analysis*, 10(2):389–412, April 1973.

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

[67] L. Reichel and W. B. Gragg. Algorithm 686: FORTRAN subroutines for updating the QR decomposition. *ACM Transactions on Mathematical Software*, 16(4):369–377, December 1990.

[68] A. Ruhe. An algorithm for numerical determination of the structure of a general matrix. *BIT*, 10:196–216, 1970.

[69] A. Ruhe. Rational Krylov sequence methods for eigenvalue computations. *Linear Algebra and Its Applications*, 58:391–405, 1984.

[70] A. Ruhe. Rational Krylov sequence methods for eigenvalue computations, II: Matrix pairs. *Linear Algebra and Its Applications*, 197,198:283–295, 1994.

[71] A. Ruhe. Rational Krylov sequence methods for eigenvalue computations, III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.

[72] H. Rutihauser. Solution of eigenvalue problems with the LR-transformation. *National Bureau of Standards Applied Mathematics Service*, 49:47–81, 1958.

[73] Y. Saad. On the rates of convergence of the Lanczos and the block Lanczos methods. *SIAM Journal of Numerical Analysis*, 17(5):687–706, October 1980.

[74] Y. Saad. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and Its Applications*, 34:269–295, 1980.

[75] Y. Saad. Projection methods for solving large sparse eigenvalue problems. In B. Kågström and A. Ruhe, editors, *Matrix Pencil Proceedings*, volume 973 of *Lecture Notes in Mathematics*, pages 121–144, Berlin, 1982. Springer–Verlag.

[76] Y. Saad. Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems. *Mathematics of Computation*, 42:567–588, 1984.

[77] Y. Saad. Least squares polynomials in the complex plane and their use for solving sparse nonsymmetric systems. *SIAM Journal on Numerical Analysis*, 24:155–169, 1987.

[78] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Halsted Press, 1992.

[79] M. Sadkane. A block Arnoldi–chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices. Technical Report TR/PA/91/46, CERFACS, 1991.

[80] J. A. Scott. An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices. Technical Report RAL-93-097, Rutherford Appleton Laboratory, 1993.

[81] H. Simon. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Linear Algebra and Its Applications*, 61:101–131, 1984.

[82] B. T. Simth, J. M. Boyle, J. J. Dongarra B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler. *EISPACK Guide.* Springer–Verlag, Berlin, second edition, 1976. Volume 6 of Lecture Notes in Computer Science.

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

[84] G. W. Stewart. Incorporating orgin shifts into the QR algorithm for symmetric tridiagonal matrices. *Communications of the Association for Computing Machinery*, 13:365–367, 1970.

[85] G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, October 1973.

[86] G. W. Stewart. *Introduction to Matrix Computations.* Academic Press, San Diego, California, 1973.

[87] G. W. Stewart. ALGORITHM 506: HQR3 and EXCHANG: Fortran subroutines for calculating and ordering the eigenvalues of a real upper Hessenberg matrix [F2]. *ACM Transactions on Mathematical Software*, 2(3):275–280, 1976.

[88] G. W. Stewart. Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices. *Numerische Mathematik*, 25:123–136, 1976.

[89] G. W. Stewart. Perturbation bounds for the QR factorization of a matrix. *SIAM Journal on Numerical Analysis*, 14:509–518, 1977.

[90] G. W. Stewart and Ji guang Sun. *Matrix Perturbation Theory*. Academic Press, San Diego, California, 1990.

[91] W.J. Stewart and A. Jennings. ALGORITHM 570: LOPSI a simultaneous iteration method for real matrices [F2]. *ACM Transactions on Mathematical Software*, 7(2):230–232, June 1981.

[92] W.J. Stewart and A. Jennings. A simultaneous iteration algorithm for real matrices. *ACM Transactions on Mathematical Software*, 7(2):184–198, June 1981.

[93] K.-C. Toh and L. N. Trefethen. Calculation of psuedospectra by the Arnoldi iteration. *SIAM Journal on Scientific Computing*, 1994. To appear.

[94] J. M. Varah. On the separation of two matrices. *SIAM Journal on Numerical Analysis*, 16(2):216–222, April 1979.

[95] H. F. Walker. Implmentation of the GMRES method using Householder transformations. *SIAM Journal on Scientific and Statistical Computing*, 9(1):152–163, January 1988.

[96] D. S. Watkins. On the transmission of shifts and shift blurring in the QR algorithm. *Linear Algebra and Its Applications*. To Appear.

[97] D. S. Watkins. Understanding the QR algorithm. *SIAM Review*, 24(4):427–439, October 1982.

[98] D. S. Watkins. *Fundamentals of Matrix Computations*. Wiley, 1991.

[99] D. S. Watkins. Forward stability and transmission of shifts in the QR algorithm. *SIAM Journal on Matrix Analysis and Applications*, 16(2):469–487, April 1995.

[100] D. S. Watkins and L. Elsner. Convergence of algorithms of decomposition type for the eigenvalue problem. *Linear Algebra and Its Applications*, 143:19–47, 1991.

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

[102] J. H. Wilkinson. Note on matrices with very ill-conditioned eigenproblem. *Numerische Mathematik*, 19:176–178, 1972.