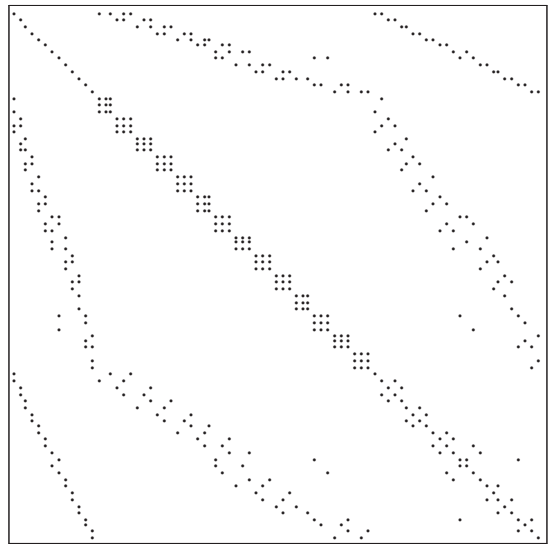


NUMERICAL METHODS FOR LARGE EIGENVALUE PROBLEMS

Second edition

Yousef Saad

Contents

Preface to the Classics Edition

This is a revised edition of a book which appeared close to two decades ago. Someone scrutinizing how the field has evolved in these two decades will make two interesting observations. On the one hand the observer will be struck by the staggering number of new developments in numerical linear algebra during this period. The field has evolved in all directions: theory, algorithms, software, and novel applications. Two decades ago there was essentially no publically available software for large eigenvalue problems. Today one has a flurry to choose from and the activity in software development does not seem to be abating. A number of new algorithms appeared in this period as well. I can mention at the outset the Jacobi-Davidson algorithm and the idea of implicit restarts, both discussed in this book, but there are a few others. The most interesting development to the numerical analyst may be the expansion of the realm of eigenvalue techniques into newer and more challenging applications. Or perhaps, the more correct observation is that these applications were always there, but they were not as widely appreciated or understood by numerical analysts, or were not fully developed due to lack of software.

The second observation to be made when comparing the state of the field now and two decades ago is that at the same time the basic tools used to compute spectra have essentially not changed much: Krylov subspaces are still omnipresent. On the whole, the new methods that have been developed consist of enhancements to these basic methods, sometimes major, in the form of preconditioners, or other variations. One might say that the field has evolved even more from gaining maturity than from the few important developments which took place. This maturity has been brought about by the development of practical algorithms and by software. Therefore, synergetic forces played a major role: new algorithms, enhancements, and software packages were developed which enabled new interest from practitioners, which in turn sparkled demand and additional interest from the algorithm developers.

In light of this observation, I have grouped the 10 chapters of the first edition into three categories. In the first group are those chapters that are of a theoretical nature (Chapters 1, 3, and 9). These have undergone small changes such as correcting errors, improving the style, and adding references.

The second group includes a few chapters that describe basic algorithms or concepts – for example subspace iteration (Chapter 5) or the tools of spectral

approximation (Chapter 4). These have been left unchanged or have received small updates. Chapters 2 and 10 are also in this group which then consists of Chapters 2, 4, 5, and 10.

Chapters in the third group (chapters 6 to 8) underwent the biggest changes. These describe algorithms and their implementations. Chapters 7 and 8 of the first edition contained a mix of topics some of which are less important today, and so some reorganization was needed. I preferred to shorten or reorganize the discussion of some of these topics rather than remove them altogether, because most are not covered in other books. At the same time it was necessary to add a few sections on topics of more recent interest. These include the implicit restart techniques (inlcuded in Chapter 7) and the Jacobi-Davidson method (included as part of Chapter 7 on preconditioning techniques). A section on AMLS (Automatic Multi-Level Substructuring) which has had excellent success in Structural Engineering has also been included with a goal to link it to other known methods.

Problems were left unchanged from the earlier edition, but the *Notes and references* sections ending each chapter were systematically updated. Some notation has also been altered from the previous edition to reflect more common usage. For example, the term "null space" has been substituted to less common term "kernel."

An on-line version of this book, along with a few resources such as tutorials, and MATLAB scripts, is posted on my web site; see:

`http://www.siam.org/books/cl66`

Finally, I am indebted to the National Science Foundation and to the Department of Energy for their support of my research throughout the years.

Yousef Saad
Minneapolis, January 6, 2011

Preface

Matrix eigenvalue problems arise in a large number of disciplines of sciences and engineering. They constitute the basic tool used in designing buildings, bridges, and turbines, that are resistent to vibrations. They allow to model queueing networks, and to analyze stability of electrical networks or fluid flow. They also allow the scientist to understand local physical phenonema or to study bifurcation patterns in dynamical systems. In fact the writing of this book was motivated mostly by the second class of problems.

Several books dealing with numerical methods for solving eigenvalue problems involving symmetric (or Hermitian) matrices have been written and there are a few software packages both public and commercial available. The book by Parlett [148] is an excellent treatise of the problem. Despite a rather strong demand by engineers and scientists there is little written on nonsymmetric problems and even less is available in terms of software. The 1965 book by Wilkinson [222] still constitutes an important reference. Certainly, science has evolved since the writing of Wilkinson's book and so has the computational environment and the demand for solving large matrix problems. Problems are becoming larger and more complicated while at the same time computers are able to deliver ever higher performances. This means in particular that methods that were deemed too demanding yesterday are now in the realm of the achievable. I hope that this book will be a small step in bridging the gap between the literature on what is available in the symmetric case and the nonsymmetric case. Both the Hermitian and the non-Hermitian case are covered, although non-Hermitian problems are given more emphasis.

This book attempts to achieve a good balance between theory and practice. I should comment that the theory is especially important in the nonsymmetric case. In essence what differentiates the Hermitian from the non-Hermitian eigenvalue problem is that in the first case we can always manage to compute an approximation whereas there are nonsymmetric problems that can be arbitrarily difficult to solve and can essentially make any algorithm fail. Stated more rigorously, the eigenvalue of a Hermitian matrix is always well-conditioned whereas this is not true for nonsymmetric matrices. On the practical side, I tried to give a general view of algorithms and tools that have proved efficient. Many of the algorithms described correspond to actual implementations of research software and have been tested on realistic problems. I have tried to convey our experience from the

practice in using these techniques.

As a result of the partial emphasis on theory, there are a few chapters that may be found hard to digest for readers inexperienced with linear algebra. These are Chapter III and to some extent, a small part of Chapter IV. Fortunately, Chapter III is basically independent of the rest of the book. The minimal background needed to use the *algorithmic part* of the book, namely Chapters IV through VIII, is calculus and linear algebra at the undergraduate level. The book has been used twice to teach a special topics course; once in a Mathematics department and once in a Computer Science department. In a quarter period representing roughly 12 weeks of 2.5 hours lecture per week, Chapter I, III, and IV, to VI have been covered without much difficulty. In a semester period, 18 weeks of 2.5 hours lecture weekly, all chapters can be covered with various degrees of depth. Chapters II and X need not be treated in class and can be given as remedial reading.

Finally, I would like to extend my appreciation to a number of people to whom I am indebted. Françoise Chatelin, who was my thesis adviser, introduced me to numerical methods for eigenvalue problems. Her influence on my way of thinking is certainly reflected in this book. Beresford Parlett has been encouraging throughout my career and has always been a real inspiration. Part of the motivation in getting this book completed, rather than 'never finished', is owed to L. E. Scriven from the Chemical Engineering department and to many others in applied sciences who expressed interest in my work. I am indebted to Roland Freund who has read this manuscript with great care and has pointed out numerous mistakes.

Chapter 1

BACKGROUND IN MATRIX THEORY AND LINEAR ALGEBRA

This chapter reviews basic matrix theory and introduces some of the elementary notation used throughout the book. Matrices are objects that represent linear mappings between vector spaces. The notions that will be predominantly used in this book are very intimately related to these linear mappings and it is possible to discuss eigenvalues of linear operators without ever mentioning their matrix representations. However, to the numerical analyst, or the engineer, any theory that would be developed in this manner would be insufficient in that it will not be of much help in developing or understanding computational algorithms. The abstraction of linear mappings on vector spaces does however provide very concise definitions and some important theorems.

1.1 Matrices

When dealing with eigenvalues it is more convenient, if not more relevant, to manipulate complex matrices rather than real matrices. A complex $m \times n$ matrix A is an $m \times n$ array of complex numbers

$$a_{ij}, \;\; i = 1, \ldots, m, \; j = 1, \ldots, n.$$

The set of all $m \times n$ matrices is a complex vector space denoted by $\mathbb{C}^{m \times n}$. The main operations with matrices are the following:

- Addition: $C = A + B$, where A, B and C are matrices of size $m \times n$ and

$$c_{ij} = a_{ij} + b_{ij} \; ,$$

  $i = 1, 2, \ldots m, \, j = 1, 2, \ldots n.$

- Multiplication by a scalar: $C = \alpha A$, where $c_{ij} = \alpha \; a_{ij}$.

- Multiplication by another matrix:

$$C = AB,$$

where $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{n\times p}$, $C \in \mathbb{C}^{m\times p}$, and

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.$$

A notation that is often used is that of column vectors and row vectors. The column vector $a_{.j}$ is the vector consisting of the j-th column of A, i.e., $a_{.j} = (a_{ij})_{i=1,\ldots,m}$. Similarly we will use the notation $a_{i.}$ to denote the i-th row of the matrix A. For example, we may write that

$$A = (a_{.1}, a_{.2}, \ldots, a_{.n}) \ .$$

or

$$A = \begin{pmatrix} a_{1.} \\ a_{2.} \\ . \\ . \\ a_{m.} \end{pmatrix}$$

The *transpose* of a matrix A in $\mathbb{C}^{m\times n}$ is a matrix C in $\mathbb{C}^{n\times m}$ whose elements are defined by $c_{ij} = a_{ji}, i = 1, \ldots, n, \ j = 1, \ldots, m$. The transpose of a matrix A is denoted by A^T. It is more relevant in eigenvalue problems to use the *transpose conjugate* matrix denoted by A^H and defined by

$$A^H = \bar{A}^T = \overline{A^T}$$

in which the bar denotes the (element-wise) complex conjugation.

Finally, we should recall that matrices are strongly related to linear mappings between vector spaces of finite dimension. They are in fact representations of these transformations with respect to two given bases; one for the initial vector space and the other for the image vector space.

1.2 Square Matrices and Eigenvalues

A matrix belonging to $\mathbb{C}^{n\times n}$ is said to be square. Some notions are only defined for square matrices. A square matrix which is very important is the identity matrix

$$I = \{\delta_{ij}\}_{i,j=1,\ldots,n}$$

where δ_{ij} is the Kronecker symbol. The identity matrix satisfies the equality $AI = IA = A$ for every matrix A of size n. The inverse of a matrix, when it exists, is a matrix C such that $CA = AC = I$. The inverse of A is denoted by A^{-1}.

The determinant of a matrix may be defined in several ways. For simplicity we adopt here the following recursive definition. The determinant of a 1×1 matrix (a) is defined as the scalar a. Then the determinant of an $n \times n$ matrix is given by

$$\det(A) = \sum_{j=1}^{n} (-1)^{j+1} a_{1j} \det(A_{1j})$$

where A_{1j} is an $(n-1) \times (n-1)$ matrix obtained by deleting the 1-st row and the $j-th$ column of A. The determinant of a matrix determines whether or not a matrix is singular since A is singular if and only if its determinant is zero. We have the following simple properties:

- $\det(AB) = \det(BA)$,
- $\det(A^T) = \det(A)$,
- $\det(\alpha A) = \alpha^n \det(A)$,
- $\det(\bar{A}) = \overline{\det(A)}$,
- $\det(I) = 1$.

From the above definition of the determinant it can be shown by induction that the function that maps a given complex value λ to the value $p_A(\lambda) = \det(A - \lambda I)$ is a polynomial of degree n (Problem P-1.6). This is referred to as the *characteristic polynomial* of the matrix A.

Definition 1.1 *A complex scalar λ is called an eigenvalue of the square matrix A if there exists a nonzero vector u of $\mathbb{C}^n$ such that $Au = \lambda u$. The vector u is called an eigenvector of A associated with λ. The set of all the eigenvalues of A is referred to as the spectrum of A and is denoted by $\Lambda(A)$.*

An eigenvalue of A is a root of the characteristic polynomial. Indeed λ is an eigenvalue of A iff $\det(A - \lambda I) \equiv p_A(\lambda) = 0$. So there are at most n distinct eigenvalues. The maximum modulus of the eigenvalues is called *spectral radius* and is denoted by $\rho(A)$:

$$\rho(A) = \max_{\lambda \in \Lambda(A)} |\lambda|.$$

The *trace* of a matrix is equal to the sum of all its diagonal elements,

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}.$$

It can be easily shown that this is also equal to the sum of its eigenvalues counted with their multiplicities as roots of the characteristic polynomial.

Proposition 1.1 *If λ is an eigenvalue of A then $\bar{\lambda}$ is an eigenvalue of A^H. An eigenvector v of A^H associated with the eigenvalue $\bar{\lambda}$ is called left eigenvector of A.*

When a distinction is necessary, an eigenvector of A is often called a right eigenvector. Thus the eigenvalue λ and the right and left eigenvectors, u and v, satisfy the relations

$$Au = \lambda u \ , \quad v^H A = \lambda v^H$$

or, equivalently,

$$u^H A^H = \bar{\lambda} u^H \ , \quad A^H v = \bar{\lambda} v \ .$$

1.3 Types of Matrices

The properties of eigenvalues and eigenvectors of square matrices will sometimes depend on special properties of the matrix A. For example, the eigenvalues or eigenvectors of the following types of matrices will all have some special properties.

- *Symmetric matrices:* $A^T = A$;
- *Hermitian matrices:* $A^H = A$;
- *Skew-symmetric matrices:* $A^T = -A$;
- *Skew-Hermitian matrices:* $A^H = -A$;
- *Normal matrices:* $A^H A = AA^H$;
- *Nonnegative matrices:* $a_{ij} \geq 0,\ i, j = 1, \ldots, n$ (similar definition for nonpositive, positive, and negative matrices);
- *Unitary matrices:* $Q \in \mathbb{C}^{n\times n}$ and $Q^H Q = I$.

It is worth noting that a unitary matrix Q is a matrix whose inverse is its transpose conjugate Q^H. Often, a matrix Q such that $Q^H Q$ is diagonal (not necessarily square) is called orthogonal.

1.3.1 Matrices with Special Srtuctures

Some matrices have particular structures that are often convenient for computational purposes and play important roles in numerical analysis. The following list though incomplete, gives an idea of the most important special matrices arising in applications and algorithms. They are mostly defined for square matrices.

- *Diagonal matrices:* $a_{ij} = 0$ for $j \neq i$. Notation for square diagonal matrices:
$$A = \text{diag}\ (a_{11}, a_{22}, \ldots, a_{mm})\,.$$
- *Upper triangular matrices:* $a_{ij} = 0$ for $i > j$.
- *Lower triangular matrices:* $a_{ij} = 0$ for $i < j$.
- *Upper bidiagonal matrices:* $a_{ij} = 0$ for $j \neq i$ or $j \neq i + 1$.
- *Lower bidiagonal matrices:* $a_{ij} = 0$ for $j \neq i$ or $j \neq i - 1$.
- *Tridiagonal matrices:* $a_{ij} = 0$ for any pair i, j such that $|j - i| > 1$. Notation:
$$A = \text{tridiag}\ (a_{i,i-1}, a_{ii}, a_{i,i+1})\,.$$

- *Banded matrices:* there exist two integers m_l and m_u such that $a_{ij} \neq 0$ only if $i - m_l \leq j \leq i + m_u$. The number $m_l + m_u + 1$ is called the bandwidth of A.

- *Upper Hessenberg matrices:* $a_{ij} = 0$ for any pair i, j such that $i > j + 1$. One can define lower Hessenberg matrices similarly.

- *Outer product matrices:* $A = uv^H$, where both u and v are vectors.

- *Permutation matrices:* the columns of A are a permutation of the columns of the identity matrix.

- *Block diagonal matrices:* generalizes the diagonal matrix by replacing each diagonal entry by a matrix. Notation:

$$A = \text{diag } (A_{11}, A_{22}, \ldots, A_{m}) .$$

- *Block tri-diagonal matrices:* generalizes the tri-diagonal matrix by replacing each nonzero entry by a square matrix. Notation:

$$A = \text{tridiag } (A_{i,i-1}, A_{ii}, A_{i,i+1}) .$$

The above properties emphasize structure, i.e., positions of the nonzero elements with respect to the zeros, and assume that there are many zero elements or that the matrix is of low rank. No such assumption is made for, say, orthogonal or symmetric matrices.

1.3.2 Special Matrices

A number of matrices which appear in applications have even more special structures than the ones seen in the previous subsection. These are typically dense matrices, but their entries depend on fewer parameters than n^2.

Thus, *Toeplitz* matrices are matrices whose entries are constant along diagonals. A 5×5 Toeplitz matrix will be as follows:

$$T = \begin{pmatrix} t_0 & t_1 & t_2 & t_3 & t_4 \\ t_{-1} & t_0 & t_1 & t_2 & t_3 \\ t_{-2} & t_{-1} & t_0 & t_1 & t_2 \\ t_{-3} & t_{-2} & t_{-1} & t_0 & t_1 \\ t_{-4} & t_{-3} & t_{-2} & t_{-1} & t_0 \end{pmatrix},$$

where $t_{-4}, t_{-3}, \cdots, t_3, t_4$ are parameters. The entries of A are such that $a_{i,i+k} = t_k$, a constant depending only on k, for $k = -(m-1), -(m-2), \cdots, 0, 1, 2, \cdots, n-1$. Indices $(i, i+k)$ outside the valid range of indices for the matrix are ignored. Such matrices are determined by the $m + n - 1$ values t_k.

Similarly, the entries of *Hankel matrices* are constant along *anti-diagonals*:

$$H = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 & h_5 \\ h_2 & h_3 & h_4 & h_5 & h_6 \\ h_3 & h_4 & h_5 & h_6 & h_7 \\ h_4 & h_5 & h_6 & h_7 & h_8 \\ h_5 & h_6 & h_7 & h_8 & h_9 \end{pmatrix}.$$

The entries of A are such that $a_{i,k+1-i} = h_k$, a constant which depends only on k, for $k = 1, 2, \cdots, m+n-1$. Again, indices $(i, k+1-i)$ falling outside the valid range of indices for A are ignored. Hankel matrices are determined by the $m+n-1$ values h_k.

A special case of Toplitz matrices is that of *Circulant matrices* which are defined by n parameters $\eta_1, \eta_2, \cdots, \eta_n$. In a circulant matrix, the entries in a row are cyclically right-shifted to form next row as is shown in the following 5×5 example:

$$C = \begin{pmatrix} \eta_1 & \eta_2 & \eta_3 & \eta_4 & \eta_5 \\ \eta_5 & \eta_1 & \eta_2 & \eta_3 & \eta_4 \\ \eta_4 & \eta_5 & \eta_1 & \eta_2 & \eta_3 \\ \eta_3 & \eta_4 & \eta_5 & \eta_1 & \eta_2 \\ \eta_2 & \eta_3 & \eta_4 & \eta_5 & \eta_1 \end{pmatrix}$$

An important practical characteristic of these special matrices, is that fast algorithms can often be devised for them. For example, one could hope that a Toeplitz linear system can be solved faster than in the standard $O(n^3)$ operations normally required, perhaps in order n^2 operations. This is indeed the case, see [77] for details.

Circulant matrices are strongly connected to the discrete Fourier transform. The eigenvectors of a circulant matrix of a given size are the columns of the discrete Fourier transform matrix of size n:

$$F_n = (f_{jk}) \quad \text{with} \quad f_{jk} = 1/\sqrt{N}\mathrm{e}^{-2jk\pi \mathrm{i}/n}, \text{for } 0 \le j, k < n.$$

More specifically, it can be shown that a circulant matrix C is of the form

$$C = F_n \mathrm{diag}\ (F_n v) F_n^{-1}$$

where $F_n v$ is the discrete Fourier transform of the vector $v = [\eta_1, \eta_2, \cdots, \eta_n]^T$ (the first column of C). For this reason matrix-vector products with circulant matrices can be performed in $O(n \log_2 n)$ operations via Fast Fourier Transforms (FFTs) instead of the standard $O(n^2)$ operations.

1.4 Vector Inner Products and Norms

We define the Hermitian inner product of the two vectors $x = (x_i)_{i=1,\ldots,m}$ and $y = (y_i)_{i=1,\ldots,m}$ of $\mathbb{C}^m$ as the complex number

$$(x, y) = \sum_{i=1}^{m} x_i \bar{y}_i, \tag{1.1}$$

which is often rewritten in matrix notation as

$$(x, y) = y^H x.$$

A vector norm on $\mathbb{C}^m$ is a real-valued function on $\mathbb{C}^m$, which satisfies the following three conditions,

$$\begin{aligned}
&\|x\| \geq 0 \quad \forall\, x, \quad \text{and} \quad \|x\| = 0 \text{ iff } x = 0;\\
&\|\alpha x\| = |\alpha| \|x\|, \quad \forall\, x \in \mathbb{C}^m, \quad \forall \alpha \in \mathbb{C}\,;\\
&\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in \mathbb{C}^m\,.
\end{aligned}$$

Associated with the inner product (1.1) is the Euclidean norm of a complex vector defined by

$$\|x\|_2 = (x, x)^{1/2}\,.$$

A fundamental additional property in matrix computations is the simple relation

$$(Ax, y) = (x, A^H y) \quad \forall x \in \mathbb{C}^n, y \in \mathbb{C}^m \tag{1.2}$$

the proof of which is straightforward. The following proposition is a consequence of the above equality.

Proposition 1.2 *Unitary matrices preserve the Hermitian inner product, i.e.,*

$$(Qx, Qy) = (x, y)$$

for any unitary matrix Q.

Proof. Indeed $(Qx, Qy) = (x, Q^H Qy) = (x, y)$. □

In particular a unitary matrix preserves the 2-norm metric, i.e., it is isometric with respect to the 2-norm.

The most commonly used vector norms in numerical linear algebra are special cases of the Hölder norms defined as follows for $p \geq 1$

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}\,. \tag{1.3}$$

It can be shown that these do indeed define norms for $p \geq 1$. Note that the limit of $\|x\|_p$ when p tends to infinity exists and is equal to the maximum modulus of the x_i's. This defines a norm denoted by $\|.\|_\infty$. The cases $p = 1$, $p = 2$, and $p = \infty$ lead to the most important norms in practice,

$$\begin{aligned}
\|x\|_1 &= |x_1| + |x_2| + \cdots + |x_n|\\
\|x\|_2 &= \left[|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2 \right]^{1/2}\\
\|x\|_\infty &= \max_{i=1,..,n} |x_i|\,.
\end{aligned}$$

A very important relation satisfied by the 2-norm is the so-called Cauchy-Schwarz inequality:

$$|(x,y)| \leq \|x\|_2 \|y\|_2.$$

This is a special case of the Hölder inequality:

$$|(x,y)| \leq \|x\|_p \|y\|_q,$$

for any pair p, q such that $1/p + 1/q = 1$ and $p \geq 1$.

1.5 Matrix Norms

For a general matrix A in $\mathbb{C}^{m\times n}$ we define a special set of norms of matrices as follows

$$\|A\|_{pq} = \max_{x\in\mathbb{C}^n,\ x\neq 0} \frac{\|Ax\|_p}{\|x\|_q}\ . \tag{1.4}$$

We say that the norms $\|.\|_{pq}$ are induced by the two norms $\|.\|_p$ and $\|.\|_q$. These satisfy the usual properties of norms, i.e.,

$$\begin{aligned}
&\|A\| \geq 0 \quad \forall A \in \mathbb{C}^{m\times n} \quad \text{and} \quad \|A\| = 0 \quad \text{iff} \quad A = 0\ ;\\
&\|\alpha A\| = |\alpha|\|A\|, \forall A \in \mathbb{C}^{m\times n}, \quad \forall \alpha \in \mathbb{C}\ ;\\
&\|A+B\| \leq \|A\| + \|B\|, \quad \forall A, B \in \mathbb{C}^{m\times n}\ .
\end{aligned}$$

Again the most important cases are the ones associated with the cases $p, q = 1, 2, \infty$. The case $q = p$ is of particular interest and the associated norm $\|.\|_{pq}$ is simply denoted by $\|.\|_p$.

A fundamental property of these norms is that

$$\|AB\|_p \leq \|A\|_p \|B\|_p,$$

which is an immediate consequence of the definition (1.4). Matrix norms that satisfy the above property are sometimes called *consistent*. As a result of the above inequality, for example, we have that for any square matrix A, and for any non-negative integer k,

$$\|A^k\|_p \leq \|A\|_p^k\ ,$$

which implies in particular that the matrix A^k converges to zero as k goes to infinity, if *any* of its p-norms is less than 1.

The Frobenius norm of a matrix is defined by

$$\|A\|_F = \left(\sum_{j=1}^{n} \sum_{i=1}^{m} |a_{ij}|^2 \right)^{1/2}\ . \tag{1.5}$$

This can be viewed as the 2-norm of the column (or row) vector in $\mathbb{C}^{m.n}$ consisting of all the columns (resp. rows) of A listed from 1 to n (resp. 1 to m). It can easily be shown that this norm is also consistent, in spite of the fact that is not induced

by a pair of vector norms, i.e., it is not derived from a formula of the form (1.4), see Problem P-1.3. However, it does not satisfy some of the other properties of the p-norms. For example, the Frobenius norm of the identity matrix is not unity. To avoid these difficulties, *we will only use the term matrix norm for a norm that is induced by two norms as in the definition (1.4).* Thus, we will not consider the Frobenius norm to be a proper matrix norm, according to our conventions, even though it is consistent.

It can be shown that the norms of matrices defined above satisfy the following equalities which provide alternative definitions that are easier to use in practice.

$$\|A\|_1 = \max_{j=1,..,n} \sum_{i=1}^{m} |a_{ij}| \; ; \tag{1.6}$$

$$\|A\|_\infty = \max_{i=1,..,m} \sum_{j=1}^{n} |a_{ij}| \; ; \tag{1.7}$$

$$\|A\|_2 = \left[\rho(A^H A)\right]^{1/2} = \left[\rho(AA^H)\right]^{1/2} \; ; \tag{1.8}$$

$$\|A\|_F = \left[\mathrm{tr}(A^H A)\right]^{1/2} = \left[\mathrm{tr}(AA^H)\right]^{1/2} \; . \tag{1.9}$$

It will be shown in Section 5 that the eigenvalues of $A^H A$ are nonnegative. Their square roots are called *singular values* of A and are denoted by $\sigma_i, i = 1, \ldots, n$. Thus, relation (1.8) shows that $\|A\|_2$ is equal to σ_1, the largest singular value of A.

Example 1.1. From the above properties, it is clear that the spectral radius $\rho(A)$ is equal to the 2-norm of a matrix when the matrix is Hermitian. However, it is not a matrix norm in general. For example, the first property of norms is not satisfied, since for

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

we have $\rho(A) = 0$ while $A \neq 0$. The triangle inequality is also not satisfied for the pair A, B where A is defined above and $B = A^T$. Indeed,

$$\rho(A + B) = 1 \quad \text{while} \quad \rho(A) + \rho(B) = 0. \qquad \square$$

1.6 Subspaces

A subspace of $\mathbb{C}^m$ is a subset of $\mathbb{C}^m$ that is also a complex vector space. The set of all linear combinations of a set of vectors $G = \{a_1, a_2, ..., a_q\}$ of $\mathbb{C}^m$ is a vector subspace called the linear span of G,

$$\begin{aligned} \mathrm{span}\{G\} &= \mathrm{span}\,\{a_1, a_2, \ldots, a_q\} \\ &= \left\{ z \in \mathbb{C}^m \mid z = \sum_{i=1}^{q} \alpha_i a_i \; ; \; \{\alpha\}_{i=1,\ldots,q} \in \mathbb{C}^q \right\} . \end{aligned}$$

If the a_i's are linearly independent, then each vector of $\mathrm{span}\{G\}$ admits a unique expression as a linear combination of the a_i's. The set G is then called a basis of the subspace $span\{G\}$.

Given two vector subspaces S_1 and S_2, their sum S is a subspace defined as the set of all vectors that are equal to the sum of a vector of S_1 and a vector of S_2. The intersection of two subspaces is also a subspace. If the intersection of S_1 and S_2 is reduced to $\{0\}$ then the sum of S_1 and S_2 is called their direct sum and is denoted by $S = S_1 \bigoplus S_2$. When S is equal to $\mathbb{C}^m$ then every vector x of $\mathbb{C}^m$ can be decomposed in a unique way as the sum of an element x_1 of S_1 and an element x_2 of S_2. In this situation, we clearly have $\dim\ (S_1) + \dim\ (S_2) = m$. The transformation P that maps x into x_1 is a linear transformation that is *idempotent* ($P^2 = P$). It is called a *projector*, onto S_1 along S_2.

Two important subspaces that are associated with a matrix A of $\mathbb{C}^{m\times n}$ are its *range*, defined by

$$\mathrm{Ran}(A) = \{Ax \mid x \in \mathbb{C}^n\}, \tag{1.10}$$

and its *null space* or *kernel*:

$$\mathrm{Null}(A) = \{x \in \mathbb{C}^n \mid Ax = 0\ \}.$$

The range of A, a subspace of $\mathbb{C}^m$, is clearly equal to the linear *span* of its columns. The *column rank* of a matrix is equal to the dimension of the range of A, i.e., to the number of linearly independent columns. An important property of matrices is that the *column rank* of a matrix is equal to its *row rank*, the number of linearly independent rows of A. This common number is the *rank* of A and it clearly satisfies the inequality

$$\mathrm{rank}(A) \leq \min\{m, n\}. \tag{1.11}$$

A matrix in $\mathbb{C}^{m\times n}$ is of *full rank* when its rank is equal to the smallest of n and m, i.e., when equality is achieved in (1.11).

A fundamental result of linear algebra is stated by the following relation

$$\mathbb{C}^m = \mathrm{Ran}(A) \oplus \mathrm{Null}(A^T)\ . \tag{1.12}$$

The same result applied to the transpose of A yields:

$$\mathbb{C}^n = \mathrm{Ran}(A^T) \oplus \mathrm{Null}(A). \tag{1.13}$$

Taking the dimensions of both sides and recalling that $\dim\ (S_1 \oplus S_2)$ equals $\dim\ (S_1)+\dim\ (S_2)$ shows that $\mathrm{dim}(\mathrm{Ran}(A^T))+\mathrm{dim}(\mathrm{Null}(A)) = n$. However, since

$$\dim\ (\mathrm{Ran}(A^T)) = \dim\ (\mathrm{Ran}(A)) = \mathrm{rank}(A)$$

then (1.13) leads to the following equality

$$\mathrm{rank}(A) + \mathrm{dim}(\mathrm{Null}(A)) = n. \tag{1.14}$$

The dimension of the null-space of A is often called the *nullity* or *co-rank* of A. The above result is therefore often known as the *Rank+Nullity theorem* which states that the rank and nullity of a matrix add up to its number of columns.

A subspace S is said to be *invariant* under a (square) matrix A whenever $AS \subseteq S$. In particular, for any eigenvalue λ of A the subspace $\mathrm{Null}(A - \lambda I)$ is invariant under A. This subspace, which consists of all the eigenvectors of A associated with λ (in addition to the zero-vector), is called the *eigenspace* of A associated with λ.

1.7 Orthogonal Vectors and Subspaces

A set of vectors $G = \{a_1, a_2, \ldots, a_p\}$ is said to be *orthogonal* if

$$(a_i, a_j) = 0 \quad \text{when} \quad i \neq j$$

It is *orthonormal* if in addition every vector of G has a 2-norm equal to unity. Every subspace admits an orthonormal basis which is obtained by taking any basis and "orthonormalizing" it. The orthonormalization can be achieved by an algorithm referred to as the Gram-Schmidt orthogonalization process which we now describe. Given a set of linearly independent vectors $\{x_1, x_2, \ldots, x_p\}$, we first normalize the vector x_1, i.e., we divide it by its 2-norm, to obtain the scaled vector q_1. Then x_2 is orthogonalized against the vector q_1 by subtracting from x_2 a multiple of q_1 to make the resulting vector orthogonal to q_1, i.e.,

$$x_2 \leftarrow x_2 - (x_2, q_1)q_1.$$

The resulting vector is again normalized to yield the second vector q_2. The i-th step of the Gram-Schmidt process consists of orthogonalizing the vector x_i against all previous vectors q_j.

ALGORITHM **1.1 Gram-Schmidt**

1. **Start**: *Compute* $r_{11} := \|x_1\|_2$. *If* $r_{11} = 0$ *stop, else* $q_1 := x_1/r_{11}$.
2. **Loop**: *For* $j = 2, \ldots, p$ *do:*
   (a) *Compute* $r_{ij} := (x_j, q_i)$ *for* $i = 1, 2, \ldots, j-1$,
   (b) $\hat{q} := x_j - \sum_{i=1}^{j-1} r_{ij} q_i$,
   (c) $r_{jj} := \|\hat{q}\|_2$,
   (d) *If* $r_{jj} = 0$ *then stop, else* $q_j := \hat{q}/r_{jj}$.

It is easy to prove that the above algorithm will not break down, i.e., all r steps will be completed, if and only if the family of vectors $x_1, x_2, \ldots, x_p$ is linearly

independent. From 2-(b) and 2-(c) it is clear that at every step of the algorithm the following relation holds:

$$x_j = \sum_{i=1}^{j} r_{ij} q_i \ .$$

If we let $X = [x_1, x_2, \ldots, x_p]$, $Q = [q_1, q_2, \ldots, q_p]$, and if R denotes the $p \times p$ upper-triangular matrix whose nonzero elements are the r_{ij} defined in the algorithm, then the above relation can be written as

$$X = QR \ . \tag{1.15}$$

This is called the QR decomposition of the $n \times p$ matrix X. Thus, from what was said above the QR decomposition of a matrix exists whenever the column vectors of X form a linearly independent set of vectors.

The above algorithm is the standard Gram-Schmidt process. There are other formulations of the same algorithm which are mathematically equivalent but have better numerical properties. The Modified Gram-Schmidt algorithm (MGSA) is one such alternative.

ALGORITHM **1.2 Modified Gram-Schmidt**

1. *Start: define* $r_{11} := \|x_1\|_2$. *If* $r_{11} = 0$ *stop, else* $q_1 := x_1/r_{11}$.
2. *Loop: For* $j = 2, \ldots, p$ *do:*
   (a) *Define* $\hat{q} := x_j$,
   (b) *For* $i = 1, \ldots, j-1$, *do* $\begin{cases} r_{ij} := (\hat{q}, q_i) \\ \hat{q} := \hat{q} - r_{ij} q_i \end{cases}$
   (c) *Compute* $r_{jj} := \|\hat{q}\|_2$,
   (d) *If* $r_{jj} = 0$ *then stop, else* $q_j := \hat{q}/r_{jj}$.

A vector that is orthogonal to all the vectors of a subspace S is said to be orthogonal to that subspace. The set of all the vectors that are orthogonal to S is a vector subspace called the *orthogonal complement* of S and denoted by $S^{\perp}$. The space $\mathbb{C}^n$ is the direct sum of S and its orthogonal complement. The projector onto S along its orthogonal complement is called an *orthogonal projector* onto S. If $V = [v_1, v_2, \ldots, v_p]$ is an orthonormal matrix then $V^H V = I$, i.e., V is orthogonal. However, VV^H is not the identity matrix but represents the orthogonal projector onto $\mathrm{span}\{V\}$, see Section 1 of Chapter 3 for details.

1.8 Canonical Forms of Matrices

In this section we will be concerned with the reduction of square matrices into matrices that have simpler forms, such as diagonal or bidiagonal, or triangular. By reduction we mean a transformation that preserves the eigenvalues of a matrix.

Definition 1.2 *Two matrices A and B are said to be similar if there is a nonsingular matrix X such that*

$$A = XBX^{-1}$$

The mapping $B \rightarrow A$ is called a similarity transformation.

It is clear that *similarity* is an equivalence relation. Similarity transformations preserve the eigenvalues of matrix. An eigenvector u_B of B is transformed into the eigenvector $u_A = Xu_B$ of A. In effect, a similarity transformation amounts to representing the matrix B in a different basis.

We now need to define some terminology.

1. An eigenvalue λ of A is said to have *algebraic multiplicity* μ if it is a root of multiplicity μ of the characteristic polynomial.
2. If an eigenvalue is of algebraic multiplicity one it is said to be *simple*. A nonsimple eigenvalue is said to be *multiple*.
3. An eigenvalue λ of A has *geometric multiplicity* γ if the maximum number of independent eigenvectors associated with it is γ. In other words the geometric multiplicity γ is the dimension of the eigenspace $\mathrm{Null}\,(A - \lambda I)$.
4. A matrix is said to be *derogatory* if the geometric multiplicity of at least one of its eigenvalues is larger than one.
5. An eigenvalue is said to be *semi-simple* if its algebraic multiplicity is equal to its geometric multiplicity. An eigenvalue that is not semi-simple is called *defective* .

We will often denote by $\lambda_1, \lambda_2, \ldots, \lambda_p$, ($p \leq n$), all the *distinct* eigenvalues of A. It is a simple exercise to show that the characteristic polynomials of two similar matrices are identical, see Exercise P-1.7. Therefore, the eigenvalues of two similar matrices are equal and so are their algebraic multiplicities. Moreover if v is an eigenvector of B then Xv is an eigenvector of A and, conversely, if y is an eigenvector of A then $X^{-1}y$ is an eigenvector of B. As a result the number of independent eigenvectors associated with a given eigenvalue is the same for two similar matrices, i.e., their geometric multiplicity is also the same.

The possible desired forms are numerous but they all have the common goal of attempting to simplify the original eigenvalue problem. Here are some possibilities with comments as to their usefulness.

- *Diagonal:* the simplest and certainly most desirable choice but it is not always achievable.
- *Jordan:* this is an upper bidiagonal matrix with ones or zeroes on the super diagonal. Always possible but not numerically trustworthy.
- *Upper triangular:* in practice this is the most reasonable compromise as the similarity from the original matrix to a triangular form can be chosen to be isometric and therefore the transformation can be achieved via a sequence of elementary unitary transformations which are numerically stable.

1.8.1 Reduction to the Diagonal Form

The simplest form in which a matrix can be reduced is undoubtedly the diagonal form but this reduction is, unfortunately, not always possible. A matrix that can be reduced to the diagonal form is called diagonalizable. The following theorem characterizes such matrices.

Theorem 1.1 *A matrix of dimension n is diagonalizable if and only if it has n linearly independent eigenvectors.*

Proof. A matrix A is diagonalizable if and only if there exists a nonsingular matrix X and a diagonal matrix D such that $A = XDX^{-1}$ or equivalently $AX = XD$, where D is a diagonal matrix. This is equivalent to saying that there exist n linearly independent vectors – the n column-vectors of X – such that $Ax_i = d_i x_i$, i.e., each of these column-vectors is an eigenvector of A. □

A matrix that is diagonalizable has only semi-simple eigenvalues. Conversely, if all the eigenvalues of a matrix are semi-simple then there exist n eigenvectors of the matrix A. It can be easily shown that these eigenvectors are linearly independent, see Exercise P-1.1. As a result we have the following proposition.

Proposition 1.3 *A matrix is diagonalizable if and only if all its eigenvalues are semi-simple.*

Since every simple eigenvalue is semi-simple, an immediate corollary of the above result is that when A has n distinct eigenvalues then it is diagonalizable.

1.8.2 The Jordan Canonical Form

From the theoretical viewpoint, one of the most important canonical forms of matrices is the well-known Jordan form. In what follows, the main constructive steps that lead to the Jordan canonical decomposition are outlined. For details, the reader is referred to a standard book on matrix theory or linear algebra.

• For every integer l and each eigenvalue λ_i it is true that

$$\mathrm{Null}(A - \lambda_i I)^{l+1} \supset \mathrm{Null}(A - \lambda_i I)^{l} .$$

• Because we are in a finite dimensional space the above property implies that there is a first integer l_i such that

$$\mathrm{Null}(A - \lambda_i I)^{l_i+1} = \mathrm{Null}(A - \lambda_i I)^{l_i},$$

and in fact $\mathrm{Null}(A - \lambda_i I)^{l} = \mathrm{Null}(A - \lambda_i I)^{l_i}$ for all $l \geq l_i$. The integer l_i is called the index of λ_i.

• The subspace $M_i = \mathrm{Null}(A - \lambda_i I)^{l_i}$ is invariant under A. Moreover, the space $\mathbb{C}^n$ is the direct sum of the subspaces M_i, $i = 1, 2, \ldots, p$. Let $m_i = \dim(M_i)$.

• In each invariant subspace M_i there are γ_i independent eigenvectors, i.e., elements of $\mathrm{Null}(A - \lambda_i I)$, with $\gamma_i \leq m_i$. It turns out that this set of vectors can be completed to form a basis of M_i by adding to it elements of $\mathrm{Null}(A - \lambda_i I)^2$, then elements of $\mathrm{Null}(A - \lambda_i I)^3$, and so on. These elements are generated by starting separately from each eigenvector u, i.e., an element of $\mathrm{Null}(A - \lambda_i I)$, and then seeking an element that satisfies $(A - \lambda_i I)z_1 = u$. Then, more generally we construct z_{i+1} by solving the equation $(A - \lambda_i I)z_{i+1} = z_i$ when possible. The vector z_i belongs to $\mathrm{Null}(A - \lambda_i I)^{i+1}$ and is called a principal vector (sometimes generalized eigenvector). The process is continued until no more principal vectors are found. There are at most l_i principal vectors for each of the γ_i eigenvectors.

• The final step is to represent the original matrix A with respect to the basis made up of the p bases of the invariant subspaces M_i defined in the previous step.

The matrix representation J of A in the new basis described above has the block diagonal structure,

$$X^{-1}AX = J = \begin{pmatrix} J_1 & & & & & \\ & J_2 & & & & \\ & & \ddots & & & \\ & & & J_i & & \\ & & & & \ddots & \\ & & & & & J_p \end{pmatrix}$$

where each J_i corresponds to the subspace M_i associated with the eigenvalue λ_i. It is of size m_i and it has itself the following structure,

$$J_i = \begin{pmatrix} J_{i1} & & & \\ & J_{i2} & & \\ & & \ddots & \\ & & & J_{i\gamma_i} \end{pmatrix} \text{ with } J_{ik} = \begin{pmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \lambda_i & 1 \\ & & & \lambda_i \end{pmatrix}.$$

Each of the blocks J_{ik} corresponds to a different eigenvector associated with the eigenvalue λ_i. Its size is equal to the number of principal vectors found for the eigenvector to which the block is associated and does not exceed l_i.

Theorem 1.2 *Any matrix A can be reduced to a block diagonal matrix consisting of p diagonal blocks, each associated with a distinct eigenvalue. Each diagonal block number i has itself a block diagonal structure consisting of γ_i subblocks, where γ_i is the geometric multiplicity of the eigenvalue λ_i. Each of the subblocks, referred to as a Jordan block, is an upper bidiagonal matrix of size not exceeding l_i, with the constant λ_i on the diagonal and the constant one on the super diagonal.*

We refer to the i-th diagonal block, $i = 1, \ldots, p$ as the i-th Jordan submatrix (sometimes "Jordan Box"). The Jordan submatrix number i starts in column $j_i \equiv m_1 + m_2 + \cdots + m_{i-1} + 1$. From the above form it is not difficult to see that

$M_i = \text{Null}(A - \lambda_i I)^{l_i}$ is merely the span of the columns $j_i, j_i + 1, \ldots, j_{i+1} - 1$ of the matrix X. These vectors are all the eigenvectors and the principal vectors associated with the eigenvalue λ_i.

Since A and J are similar matrices their characteristic polynomials are identical. Hence, it is clear that the algebraic multiplicity of an eigenvalue λ_i is equal to the dimension of M_i:

$$\mu_i = m_i \equiv \dim(M_i) .$$

As a result,

$$\mu_i \geq \gamma_i.$$

Because $\mathbb{C}^n$ is the direct sum of the subspaces $M_i, i = 1, \ldots, p$ each vector x can be written in a unique way as

$$x = x_1 + x_2 + \cdots + x_i + \cdots + x_p,$$

where x_i is a member of the subspace M_i. The linear transformation defined by

$$P_i : x \rightarrow x_i$$

is a projector onto M_i along the direct sum of the subspaces $M_j, j \neq i$. The family of projectors $P_i, i = 1, \ldots, p$ satisfies the following properties,

$$\text{Ran}(P_i) = M_i \tag{1.16}$$

$$P_i P_j = P_j P_i = 0, \text{ if } i \neq j \tag{1.17}$$

$$\sum_{i=1}^{p} P_i = I \tag{1.18}$$

In fact it is easy to see that the above three properties define a decomposition of $\mathbb{C}^n$ into a direct sum of the images of the projectors P_i in a unique way. More precisely, any family of projectors that satisfies the above three properties is uniquely determined and is associated with the decomposition of $\mathbb{C}^n$ into the direct sum of the images of the P_i 's.

It is helpful for the understanding of the Jordan canonical form to determine the matrix representation of the projectors P_i. Consider the matrix $\hat{J}_i$ which is obtained from the Jordan matrix by replacing all the diagonal submatrices by zero blocks except the i^{th} submatrix which is replaced by the identity matrix.

$$\hat{J}_i = \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & I & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix}$$

In other words if each i-th Jordan submatrix starts at the column number j_i, then the columns of $\hat{J}_i$ will be zero columns except columns $j_i, \ldots, j_{i+1} - 1$ which are the corresponding columns of the identity matrix. Let $\hat{P}_i = X \hat{J}_i X^{-1}$. Then it is not difficult to verify that $\hat{P}_i$ is a projector and that,

1. The range of $\hat{P}_i$ is the span of columns $j_i, \ldots, j_{i+1} - 1$ of the matrix X. This is the same subspace as M_i.

2. $\hat{P}_i \hat{P}_j = \hat{P}_j \hat{P}_i = 0$ whenever $i \neq j$

3. $\hat{P}_1 + \hat{P}_2 + \cdots + \hat{P}_p = I$

According to our observation concerning the uniqueness of a family of projectors that satisfy (1.16) - (1.18) this implies that

$$\hat{P}_i = P_i \quad , \quad i = 1, \ldots, p$$

Example 1.2. Let us assume that the eigenvalue λ_i is simple. Then,

$$P_i = X e_i e_i^H X^{-1} \equiv u_i w_i^H,$$

in which we have defined $u_i = X e_i$ and $w_i = X^{-H} e_i$. It is easy to show that u_i and w_i are right and left eigenvectors, respectively, associated with λ_i and normalized so that $w_i^H u_i = 1$. □

Consider now the matrix $\hat{D}_i$ obtained from the Jordan form of A by replacing each Jordan submatrix by a zero matrix except the i-th submatrix which is obtained by zeroing its diagonal elements, i.e.,

$$\hat{D}_i = \begin{pmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & J_i - \lambda_i I & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}$$

Define $D_i = X \hat{D}_i X^{-1}$. Then it is a simple exercise to show by means of the explicit expression for $\hat{P}_i$, that

$$D_i = (A - \lambda_i I) P_i. \tag{1.19}$$

Moreover, $D_i^{l_i} = 0$, i.e., D_i is a *nilpotent matrix* of index l_i. We are now ready to state the following important theorem which can be viewed as an alternative mathematical formulation of Theorem 1.2 on Jordan forms.

Theorem 1.3 *Every square matrix A admits the decomposition*

$$A = \sum_{i=1}^{p} (\lambda_i P_i + D_i) \tag{1.20}$$

where the family of projectors $\{P_i\}_{i=1,\ldots,p}$ satisfies the conditions (1.16), (1.17), and (1.18), and where $D_i = (A - \lambda_i I) P_i$ is a nilpotent operator of index l_i.

Proof. From (1.19), we have

$$AP_i = \lambda_i P_i + D_i \quad i = 1, 2, \ldots, p$$

Summing up the above equalities for $i = 1, 2, \ldots, p$ we get

$$A \sum_{i=1}^{p} P_i = \sum_{i=1}^{p} (\lambda_i P_i + D_i)$$

The proof follows by substituting (1.18) into the left-hand-side. □

The projector P_i is called the *spectral projector* associated with the eigenvalue λ_i. The linear operator D_i is called the *nilpotent* associated with λ_i. The decomposition (1.20) is referred to as the spectral decomposition of A. Additional properties that are easy to prove from the various expressions of P_i and D_i are the following

$$P_i D_j = D_j P_i = \delta_{ij} P_i \tag{1.21}$$

$$AP_i = P_i A = P_i A P_i = \lambda_i P_i + D_i \tag{1.22}$$

$$A^k P_i = P_i A^k = P_i A^k P_i = P_i (\lambda_i I + D_i)^k = (\lambda_i I + D_i)^k P_i \tag{1.23}$$

$$AP_i = [x_{j_i}, \ldots, x_{j_{i+1}-1}] B_i [y_{j_i}, \ldots, y_{j_{i+1}-1}]^H \tag{1.24}$$

where B_i is the i-th Jordan submatrix and where the columns y_j are the columns of the matrix X^{-H}.

Corollary 1.1 *For any matrix norm $\|.\|$, the following relation holds*

$$\lim_{k \to \infty} \|A^k\|^{1/k} = \rho(A) \,. \tag{1.25}$$

Proof. The proof of this corollary is the subject of exercise P-1.8. □

Another way of stating the above corollary is that there is a sequence ϵ_k such that

$$\|A^k\| = (\rho(A) + \epsilon_k)^k$$

where $\lim_{k \to \infty} \epsilon_k = 0$.

1.8.3 The Schur Canonical Form

We will now show that any matrix is unitarily similar to an upper-triangular matrix. The only result needed to prove the following theorem is that any vector of 2-norm one can be completed by $n - 1$ additional vectors to form an orthonormal basis of $\mathbb{C}^n$.

Theorem 1.4 *For any given matrix A there exists a unitary matrix Q such that $Q^H A Q = R$ is upper-triangular.*

Proof. The proof is by induction over the dimension n. The result is trivial for $n = 1$. Let us assume that it is true for $n-1$ and consider any matrix A of size n. The matrix admits at least one eigenvector u that is associated with an eigenvalue λ. We assume without loss of generality that $\|u\|_2 = 1$. We can complete the vector u into an orthonormal set, i.e., we can find an $n \times (n-1)$ matrix V such that the $n \times n$ matrix $U = [u, V]$ is unitary. Then we have $AU = [\lambda u, AV]$ and hence,

$$U^H AU = \begin{bmatrix} u^H \\ V^H \end{bmatrix} \quad [\lambda u, AV] = \begin{pmatrix} \lambda & u^H AV \\ 0 & V^H AV \end{pmatrix} \tag{1.26}$$

We now use our induction hypothesis for the $(n-1)\times(n-1)$ matrix $B = V^H AV$: there exists an $(n-1) \times (n-1)$ unitary matrix Q_1 such that $Q_1^H B Q_1 = R_1$ is upper-triangular. Let us define the $n \times n$ matrix

$$\hat{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & Q_1 \end{pmatrix}$$

and multiply both members of (1.26) by $\hat{Q}_1^H$ from the left and $\hat{Q}_1$ from the right. The resulting matrix is clearly upper triangular and this shows that the result is true for A, with $Q = \hat{Q}_1 U$ which is a unitary $n \times n$ matrix. □

A simpler proof that uses the Jordan canonical form and the QR decomposition is the subject of Exercise P-1.5. Since the matrix R is triangular and similar to A, its diagonal elements are equal to the eigenvalues of A ordered in a certain manner. In fact it is easy to extend the proof of the theorem to show that we can obtain this factorization with *any order* we want for the eigenvalues. One might ask the question as to which order might be best numerically but the answer to the question goes beyond the scope of this book. Despite its simplicity, the above theorem has far reaching consequences some of which will be examined in the next section.

It is important to note that for any $k \leq n$ the subspace spanned by the first k columns of Q is invariant under A. This is because from the Schur decomposition we have, for $1 \leq j \leq k$,

$$Aq_j = \sum_{i=1}^{i=j} r_{ij} q_i \ .$$

In fact, letting $Q_k = [q_1, q_2, \ldots, q_k]$ and R_k be the principal leading submatrix of dimension k of R, the above relation can be rewritten as

$$AQ_k = Q_k R_k$$

which we refer to as the partial Schur decomposition of A. The simplest case of this decomposition is when $k = 1$, in which case q_1 is an eigenvector. The vectors q_i are usually referred to as Schur vectors. Note that the Schur vectors are not unique and in fact they depend on the order chosen for the eigenvalues.

A slight variation on the Schur canonical form is the quasi Schur form, also referred to as the real Schur form. Here, diagonal blocks of size 2×2 are allowed

in the upper triangular matrix R. The reason for this is to avoid complex arithmetic when the original matrix is real. A 2×2 block is associated with each complex conjugate pair of eigenvalues of the matrix.

Example 1.3. Consider the 3×3 matrix

$$A = \begin{pmatrix} 1 & 10 & 0 \\ -1 & 3 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

The matrix A has the pair of complex conjugate eigenvalues

$$2.4069.. \pm i \times 3.2110..$$

and the real eigenvalue $0.1863...$ The standard (complex) Schur form is given by the pair of matrices

$$V = \begin{pmatrix} 0.3381 - 0.8462i & 0.3572 - 0.1071i & 0.1749 \\ 0.3193 - 0.0105i & -0.2263 - 0.6786i & -0.6214 \\ 0.1824 + 0.1852i & -0.2659 - 0.5277i & 0.7637 \end{pmatrix}$$

and

$$S = \begin{pmatrix} 2.4069 + 3.2110i & 4.6073 - 4.7030i & -2.3418 - 5.2330i \\ 0 & 2.4069 - 3.2110i & -2.0251 - 1.2016i \\ 0 & 0 & 0.1863 \end{pmatrix}.$$

It is possible to avoid complex arithmetic by using the quasi-Schur form which consists of the pair of matrices

$$U = \begin{pmatrix} -0.9768 & 0.1236 & 0.1749 \\ -0.0121 & 0.7834 & -0.6214 \\ 0.2138 & 0.6091 & 0.7637 \end{pmatrix}$$

and

$$R = \begin{pmatrix} 1.3129 & -7.7033 & 6.0407 \\ 1.4938 & 3.5008 & -1.3870 \\ 0 & 0 & 0.1863 \end{pmatrix}$$ □

We would like to conclude this section by pointing out that the Schur and the quasi Schur forms of a given matrix are in no way unique. In addition to the dependence on the ordering of the eigenvalues, any column of Q can be multiplied by a complex sign $e^{i\theta}$ and a new corresponding R can be found. For the quasi Schur form there are infinitely many ways of selecting the 2×2 blocks, corresponding to applying arbitrary rotations to the columns of Q associated with these blocks.

1.9 Normal and Hermitian Matrices

In this section we look at the specific properties of normal matrices and Hermitian matrices regarding among other things their spectra and some important optimality properties of their eigenvalues. The most common normal matrices that arise in practice are Hermitian or skew-Hermitian. In fact, symmetric real matrices form a large part of the matrices that arise in practical eigenvalue problems.

1.9.1 Normal Matrices

By definition a matrix is said to be normal if it satisfies the relation

$$A^H A = AA^H. \tag{1.27}$$

An immediate property of normal matrices is stated in the following proposition.

Proposition 1.4 *If a normal matrix is triangular then it is necessarily a diagonal matrix.*

Proof. Assume for example that A is upper-triangular and normal and let us compare the first diagonal element of the left hand side matrix of (1.27) with the corresponding element of the matrix on the right hand side. We obtain that

$$|a_{11}|^2 = \sum_{j=1}^{n} |a_{1j}|^2,$$

which shows that the elements of the first row are zeros except for the diagonal one. The same argument can now be used for the second row, the third row, and so on to the last row, to show that $a_{ij} = 0$ for $i \neq j$. □

As a consequence of this we have the following important result.

Theorem 1.5 *A matrix is normal if and only if it is unitarily similar to a diagonal matrix.*

Proof. It is straightforward to verify that a matrix which is unitarily similar to a diagonal matrix is normal. Let us now show that any normal matrix A is unitarily similar to a diagonal matrix. Let $A = QRQ^H$ be the Schur canonical form of A where we recall that Q is unitary and R is upper-triangular. By the normality of A we have

$$QR^H Q^H QRQ^H = QRQ^H QR^H Q^H$$

or,

$$QR^H RQ^H = QRR^H Q^H$$

Upon multiplication by Q^H on the left and Q on the right this leads to the equality $R^H R = RR^H$ which means that R is normal, and according to the previous proposition this is only possible if R is diagonal. □

Thus, any normal matrix is diagonalizable and admits an orthonormal basis of eigenvectors, namely the column vectors of Q.

Clearly, Hermitian matrices are just a particular case of normal matrices. Since a normal matrix satisfies the relation $A = QDQ^H$, with D diagonal and Q unitary, the eigenvalues of A are the diagonal entries of D. Therefore, if these entries are real it is clear that we will have $A^H = A$. This is restated in the following corollary.

Corollary 1.2 *A normal matrix whose eigenvalues are real is Hermitian.*

As will be seen shortly the converse is also true, in that a Hermitian matrix has real eigenvalues.

An eigenvalue λ of any matrix satisfies the relation

$$\lambda = \frac{(Au, u)}{(u, u)}$$

where u is an associated eigenvector. More generally one might consider the complex scalars,

$$\mu(x) = \frac{(Ax, x)}{(x, x)} \tag{1.28}$$

defined for any nonzero vector in $\mathbb{C}^n$. These ratios are referred to as *Rayleigh quotients* and are important both from theoretical and practical purposes. The set of all possible Rayleigh quotients as x runs over $\mathbb{C}^n$ is called the *field of values* of A. This set is clearly bounded since each $|\mu(x)|$ is bounded by the the 2-norm of A, i.e., $|\mu(x)| \leq \|A\|_2$ for all x.

If a matrix is normal then any vector x in $\mathbb{C}^n$ can be expressed as

$$\sum_{i=1}^{n} \xi_i q_i$$

where the vectors q_i form an orthogonal basis of eigenvectors, and the expression for $\mu(x)$ becomes,

$$\mu(x) = \frac{(Ax, x)}{(x, x)} = \frac{\sum_{k=1}^{n} \lambda_k |\xi_k|^2}{\sum_{k=1}^{n} |\xi_k|^2} \equiv \sum_{k=1}^{n} \beta_k \lambda_k \tag{1.29}$$

where

$$0 \leq \beta_i = \frac{|\xi_i|^2}{\sum_{k=1}^{n} |\xi_k|^2} \leq 1\,, \quad \text{and} \quad \sum_{i=1}^{n} \beta_i = 1$$

From a well-known characterization of convex hulls due to Hausdorff, (Hausdorff's convex hull theorem) this means that the set of all possible Rayleigh quotients as x runs over all of $\mathbb{C}^n$ is equal to the convex hull of the λ_i's. This leads to the following theorem.

Theorem 1.6 *The field of values of a normal matrix is equal to the convex hull of its spectrum.*

The question that arises next is whether or not this is also true for non-normal matrices and the answer is no, i.e., the convex hull of the eigenvalues and the field of values of a non-normal matrix are different in general, see Exercise P-1.10 for an example. As a generic example, one can take any nonsymmetric real matrix that has real eigenvalues only; its field of values will contain imaginary values. It has been shown (Hausdorff) that the field of values of a matrix is a convex set. Since the eigenvalues are members of the field of values, their convex hull is contained in the field of values. This is summarized in the following proposition.

Proposition 1.5 *The field of values of an arbitrary matrix is a convex set which contains the convex hull of its spectrum. It is equal to the convex hull of the spectrum when the matrix in normal.*

1.9.2 Hermitian Matrices

A first and important result on Hermitian matrices is the following.

Theorem 1.7 *The eigenvalues of a Hermitian matrix are real, i.e.,* $\Lambda(A) \subset \mathbb{R}$.

Proof. Let λ be an eigenvalue of A and u an associated eigenvector or 2-norm unity. Then

$$\lambda = (Au, u) = (u, Au) = \overline{(Au, u)} = \overline{\lambda} \qquad \square$$

Moreover, it is not difficult to see that if, in addition, the matrix is real then the eigenvectors can be chosen to be real, see Exercise P-1.16. Since a Hermitian matrix is normal an immediate consequence of Theorem 1.5 is the following result.

Theorem 1.8 *Any Hermitian matrix is unitarily similar to a real diagonal matrix.*

In particular a Hermitian matrix admits a set of orthonormal eigenvectors that form a basis of $\mathbb{C}^n$.

In the proof of Theorem 1.6 we used the fact that the inner products (Au, u) are real. More generally it is clear that any Hermitian matrix is such that (Ax, x) is real for any vector $x \in \mathbb{C}^n$. It turns out that the converse is also true, i.e., it can be shown that if (Az, z) is real for all vectors z in $\mathbb{C}^n$ then the matrix A is Hermitian, see Problem P-1.14.

Eigenvalues of Hermitian matrices can be characterized by optimality properties of the Rayleigh quotients (1.28). The best known of these is the Min-Max principle. Let us order all the eigenvalues of A in descending order:

$$\lambda_1 \geq \lambda_2 \ldots \geq \lambda_n.$$

Here the eigenvalues are not necessarily distinct and they are repeated, each according to its multiplicity. In what follows, we denote by S a generic subspace of $\mathbb{C}^n$. Then we have the following theorem.

Theorem 1.9 (Min-Max theorem) *The eigenvalues of a Hermitian matrix A are characterized by the relation*

$$\lambda_k = \min_{S,\ \dim(S)=n-k+1} \ \max_{x\in S, x\neq 0} \ \frac{(Ax,x)}{(x,x)} \tag{1.30}$$

Proof. Let $\{q_i\}_{i=1,\ldots,n}$ be an orthonormal basis of $\mathbb{C}^n$ consisting of eigenvectors of A associated with $\lambda_1,\ldots,\lambda_n$ respectively. Let S_k be the subspace spanned by the first k of these vectors and denote by $\mu(S)$ the maximum of $(Ax,x)/(x,x)$ over all nonzero vectors of a subspace S. Since the dimension of S_k is k, a well-known theorem of linear algebra shows that its intersection with any subspace S of dimension $n-k+1$ is not reduced to $\{0\}$, i.e., there is vector x in $S\bigcap S_k$. For this $x=\sum_{i=1}^{k}\xi_i q_i$ we have

$$\frac{(Ax,x)}{(x,x)} \ = \ \frac{\sum_{i=1}^{k}\lambda_i|\xi_i|^2}{\sum_{i=1}^{k}|\xi_i|^2} \quad \geq \quad \lambda_k$$

so that $\mu(S) \ \geq \ \lambda_k$.

Consider on the other hand the particular subspace S_0 of dimension $n-k+1$ which is spanned by $q_k,\ldots,q_n$. For each vector x in this subspace we have

$$\frac{(Ax,x)}{(x,x)} = \frac{\sum_{i=k}^{n}\lambda_i|\xi_i|^2}{\sum_{i=k}^{n}|\xi_i|^2} \quad \leq \quad \lambda_k$$

so that $\mu(S_0)\leq\lambda_k$. In other words, as S runs over all $n-k+1$-dimensional subspaces $\mu(S)$ is always $\geq \ \lambda_k$ and there is at least one subspace S_0 for which $\mu(S_0)\leq\lambda_k$ which shows the result. □

This result is attributed to Courant and Fisher, and to Poincaré and Weyl. It is often referred to as Courant-Fisher min-max principle or theorem. As a particular case, the largest eigenvalue of A satisfies

$$\lambda_1 = \max_{x\neq 0} \ \frac{(Ax,x)}{(x,x)} \ . \tag{1.31}$$

Actually, there are four different ways of rewriting the above characterization. The second formulation is

$$\lambda_k = \max_{S,\ \dim(S)=k} \ \min_{x\in S, x\neq 0} \ , \ \frac{(Ax,x)}{(x,x)} \tag{1.32}$$

and the two other ones can be obtained from the above two formulations by simply relabeling the eigenvalues increasingly instead of decreasingly. Thus, with our labeling of the eigenvalues in descending order, (1.32) tells us that the smallest eigenvalue satisfies,

$$\lambda_n = \min_{x\neq 0} \ \frac{(Ax,x)}{(x,x)} \ .$$

with λ_n replaced by λ_1 if the eigenvalues are relabeled increasingly.

In order for all the eigenvalues of a Hermitian matrix to be positive it is necessary and sufficient that

$$(Ax, x) > 0, \quad \forall\, x \in \mathbb{C}^n, \quad x \neq 0.$$

Such a matrix is called *positive definite*. A matrix that satisfies $(Ax, x) \geq 0$ for any x is said to be *positive semi-definite*. In particular the matrix $A^H A$ is semi-positive definite for any rectangular matrix, since

$$(A^H Ax, x) = (Ax, Ax) \;\geq\; 0 \quad \forall\, x.$$

Similarly, AA^H is also a Hermitian semi-positive definite matrix. The square roots of the eigenvalues of $A^H A$ for a general rectangular matrix A are called the *singular values* of A and are denoted by σ_i. In Section 1.5 we have stated without proof that the 2-norm of any matrix A is equal to the largest singular value σ_1 of A. This is now an obvious fact, because

$$\|A\|_2^2 = \max_{x \neq 0} \frac{\|Ax\|_2^2}{\|x\|_2^2} \;=\; \max_{x \neq 0} \frac{(Ax, Ax)}{(x, x)} \;=\; \max_{x \neq 0} \frac{(A^H Ax, x)}{(x, x)} \;=\; \sigma_1^2$$

which results from (1.31).

Another characterization of eigenvalues, known as the Courant characterization, is stated in the next theorem. In contrast with the min-max theorem this property is recursive in nature.

Theorem 1.10 *The eigenvalue λ_i and the corresponding eigenvector q_i of a Hermitian matrix are such that*

$$\lambda_1 = \frac{(Aq_1, q_1)}{(q_1, q_1)} \;=\; \max_{x \in \mathbb{C}^n, x \neq 0} \frac{(Ax, x)}{(x, x)}$$

and for $k > 1$:

$$\lambda_k = \frac{(Aq_k, q_k)}{(q_k, q_k)} \;=\; \max_{x \neq 0, q_1^H x = \ldots = q_{k-1}^H x = 0} \frac{(Ax, x)}{(x, x)}\,. \tag{1.33}$$

In other words, the maximum of the Rayleigh quotient over a subspace that is orthogonal to the first $k-1$ eigenvectors is equal to λ_k and is achieved for the eigenvector q_k associated with λ_k. The proof follows easily from the expansion (1.29) of the Rayleigh quotient.

1.10 Nonnegative Matrices

A nonnegative matrix is a matrix whose entries are nonnegative,

$$a_{ij} \geq 0\,.$$

Nonnegative matrices arise in many applications and play a crucial role in the theory of matrices. They play for example a key role in the analysis of convergence of

iterative methods for partial differential equations. They also arise in economics, queuing theory, chemical engineering, etc..

A matrix is said to be reducible if, there is a permutation matrix P such that PAP^T is block upper-triangular. An important result concerning nonnegative matrices is the following theorem known as the Perron-Frobenius theorem.

Theorem 1.11 *Let A be a real $n \times n$ nonnegative irreducible matrix. Then $\lambda \equiv \rho(A)$, the spectral radius of A, is a simple eigenvalue of A. Moreover, there exists an eigenvector u with positive elements associated with this eigenvalue.*

PROBLEMS

P-1.1 Show that two eigenvectors associated with two distinct eigenvalues are linearly independent. More generally show that a family of eigenvectors associated with distinct eigenvalues forms a linearly independent family.

P-1.2 Show that if λ is any eigenvalue of the matrix AB then it is also an eigenvalue of the matrix BA. Start with the particular case where A and B are square and B is nonsingular then consider the more general case where A, B may be singular or even rectangular (but such that AB and BA are square).

P-1.3 Show that the Frobenius norm is consistent. Can this norm be associated to two vector norms via (1.4)? What is the Frobenius norm of a diagonal matrix? What is the p-norm of a diagonal matrix (for any p)?

P-1.4 Find the Jordan canonical form of the matrix:

$$A = \begin{pmatrix} 1 & 2 & -4 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix}.$$

Same question for the matrix obtained by replacing the element a_{33} by 1.

P-1.5 Give an alternative proof of Theorem 1.4 on the Schur form by starting from the Jordan canonical form. [Hint: write $A = XJX^{-1}$ and use the QR decomposition of X.]

P-1.6 Show from the definition of determinants used in Section (1.2) that the characteristic polynomial is a polynomial of degree n for an $n \times n$ matrix.

P-1.7 Show that the characteristic polynomials of two similar matrices are equal.

P-1.8 Show that

$$\lim_{k \to \infty} \|A^k\|^{1/k} = \rho(A),$$

for any matrix norm. [Hint: use the Jordan canonical form or Theorem 1.3]

P-1.9 Let X be a nonsingular matrix and, for any matrix norm $\|.\|$, define $\|A\|_X = \|AX\|$. Show that this is indeed a matrix norm. Is this matrix norm consistent? Similar questions for $\|XA\|$ and $\|YAX\|$ where Y is also a nonsingular matrix. These norms are not, in general, associated with any vector norms, i.e., they can't be defined by a formula of the form (1.4). Why? What about the particular case $\|A\|' = \|XAX^{-1}\|$?

P-1.10 Find the field of values of the matrix

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

and verify that it is not equal to the convex hull of its eigenvalues.

P-1.11 Show that any matrix can be written as the sum of a Hermitian and a skew-Hermitian matrix (or the sum of a symmetric and a skew-symmetric matrix).

P-1.12 Show that for a skew-Hermitian matrix S, we have

$$\Re e(Sx, x) = 0 \quad \text{for any } x \in \mathbb{C}^n.$$

P-1.13 Given an arbitrary matrix S, show that if $(Sx, x) = 0$ for all x in $\mathbb{C}^n$ then we must have

$$(Sy, z) + (Sz, y) = 0 \quad \forall\, y\ ,\ z \in \mathbb{C}^n.$$

[Hint: expand $(S(y+z), y+z)$].

P-1.14 Using the result of the previous two problems, show that if (Ax, x) is real for all x in $\mathbb{C}^n$, then A must be Hermitian. Would this result be true if we were to replace the assumption by: (Ax, x) *is real for all real* x? Explain.

P-1.15 The definition of a positive definite matrix is that (Ax, x) be real and positive for all real vectors x. Show that this is equivalent to requiring that the Hermitian part of A, namely $\frac{1}{2}(A + A^H)$, be (Hermitian) positive definite.

P-1.16 Let A be a real symmetric matrix and λ an eigenvalue of A. Show that if u is an eigenvector associated with λ then so is $\bar{u}$. As a result, prove that for any eigenvalue of a real symmetric matrix, there is an associated eigenvector which is real.

P-1.17 Show that a Hessenberg matrix H such that $h_{j+1,j} \neq 0, j = 1, 2, \ldots, n-1$ cannot be derogatory.

NOTES AND REFERENCES. A few textbooks can be consulted for additional reading on the material of this Chapter. Since the classic volumes by Golub and Van Loan [77] and Stewart [198] mentioned in the first edition of this book a few more texts have been added to the literature. These include Demmel [44], Trefethen and Bau [213], Datta [40], and the introductory text by Gilbert Strang [208]. Details on matrix eigenvalue problems can be found in Gantmacher's book [69] and Wilkinson [222]. Stewart and Sun's book [205] devotes a separate chapter to matrix norms and contains a wealth of information. Some of the terminology we use is borrowed from Chatelin [22, 23] and Kato [105]. For a good overview of the linear algebra aspects of matrix theory and a complete proof of Jordan's canonical form Halmos' book [86] is highly recommended. ■

Chapter 2

SPARSE MATRICES

The eigenvalue problems that arise in practice often involve very large matrices. The meaning of 'large' is relative and it is changing rapidly with the progress of computer technology. A matrix of size a few tens of thousands can be considered large if one is working on a workstation, while, similarly, a matrix whose size is in the hundreds of millions can be considered large if one is using a high-performance computer. [1]*. Fortunately, many of these matrices are also sparse, i.e., they have very few nonzeros. Again, it is not clear how 'few' nonzeros a matrix must have before it can be called sparse. A commonly used definition due to Wilkinson is to say that a matrix is sparse whenever it is possible to take advantage of the number and location of its nonzero entries. By this definition a tridiagonal matrix is sparse, but so would also be a triangular matrix, which may not be as convincing. It is probably best to leave this notion somewhat vague, since the decision as to whether or not a matrix should be considered sparse is a practical one that is ultimately made by the user.*

2.1 Introduction

The natural idea of taking advantage of the zeros of a matrix and their location has been exploited for a long time. In the simplest situation, such as for banded or tridiagonal matrices, special techniques are straightforward to develop. However, the notion of exploiting sparsity for general sparse matrices, i.e., sparse matrices with irregular structure, has become popular only after the 1960's. The main issue, and the first one to be addressed by sparse matrix technology, is to devise direct solution methods for linear systems, that are economical both in terms of storage and computational effort. These sparse direct solvers allow to handle very large problems that could not be tackled by the usual 'dense' solvers. We will briefly discuss the solution of large sparse linear systems in Section 2.4 of this Chapter.

There are basically two broad types of sparse matrices: *structured* and *unstructured*. A structured sparse matrix is one whose nonzero entries, or square blocks of nonzero entries, form a regular pattern, often along a small number of

[1] In support of this observation is the fact that in the first edition of this book, the numbers I used were 'a few hundreds' and 'in the millions', respectively

diagonals. A matrix with irregularly located entries is said to be irregularly structured. The best example of a regularly structured matrix is that of a matrix that consists only of a few diagonals. Figure 2.2 shows a small irregularly structured sparse matrix associated with the finite element grid problem shown in Figure 2.1.

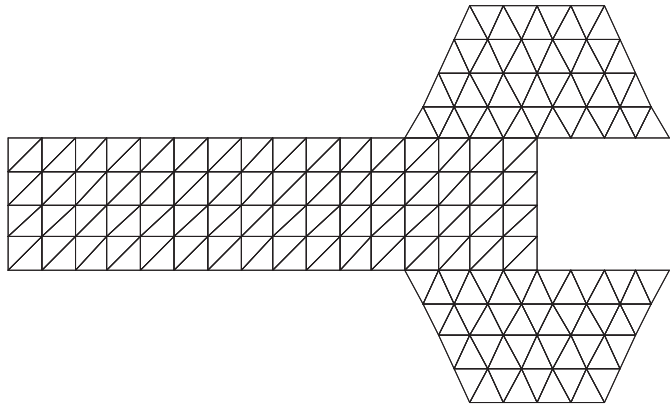


Figure 2.1: A finite element grid model

Although the difference between the two types of matrices may not matter that much for direct solvers, it may be important for eigenvalue methods or iterative methods for solving linear systems. In these methods, one of the essential operations are matrix by vector products. The performance of these operations on supercomputers can differ significantly from one data structure to another. For example, diagonal storage schemes are ideal for vector machines, whereas more general schemes, may suffer on such machines because of the need to use indirect addressing.

In the next section we will discuss some of the storage schemes used for sparse matrices. Then we will see how some of the simplest matrix operations with sparse matrices can be performed. We will then give an overview of sparse linear system solution methods. The last two sections discuss test matrices and a set of tools for working with sparse matrices called SPARSKIT.

2.2 Storage Schemes

In order to take advantage of the large number of zero elements special schemes are required to store sparse matrices. Clearly, the main goal is to represent only the nonzero elements, and be able at the same time to perform the commonly needed matrix operations. In the following we will denote by Nz the total number of

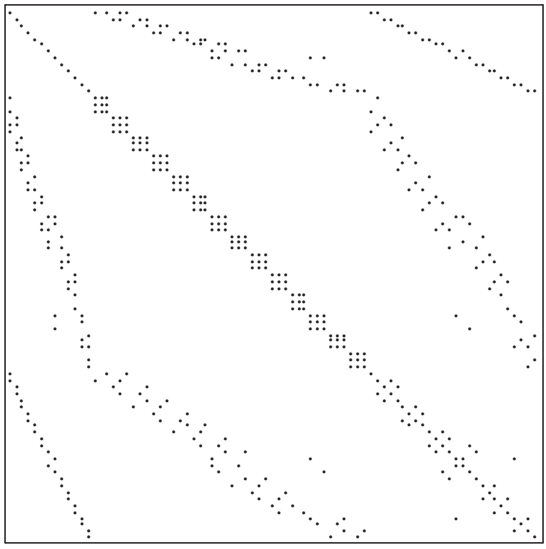


Figure 2.2: Sparse matrix associated with the finite element grid of Figure 2.1

nonzero elements. We describe only the most popular schemes but additional details can be found in the book by Duff, Erisman, and Reid [52].

The simplest storage scheme for sparse matrices is the so-called coordinate format. The data structure consists of three arrays: a real array containing all the real (or complex) values of the nonzero elements of A in any order, an integer array containing their row indices and a second integer array containing their column indices. All three arrays are of length Nz. Thus the matrix

$$A = \begin{pmatrix} 1. & 0. & 0. & 2. & 0. \\ 3. & 4. & 0. & 5. & 0. \\ 6. & 0. & 7. & 8. & 9. \\ 0. & 0. & 10. & 11. & 0. \\ 0. & 0. & 0. & 0. & 12. \end{pmatrix} \tag{2.1}$$

will be represented (for example) by

AA	=	12.	9.	7.	5.	1.	2.	11.	3.	6.	4.	8.	10.
JR	=	5	3	3	2	1	1	4	2	3	2	3	4
JC	=	5	5	3	4	1	4	4	1	1	2	4	3

In the above example we have, on purpose, listed the elements in an arbitrary order. In fact it would have been more natural to list the elements by row or columns. If we listed the elements row-wise, we would notice that the array JC

contains redundant information, and may be replaced by an array that points to the beginning of each row instead. This would entail nonnegligible savings in storage. The new data structure consists of three arrays with the following functions.

- A real array AA contains the real values a_{ij} stored row by row, from row 1 to n. The length of AA is Nz.

- An integer array JA contains the column indices of the elements a_{ij} as stored in the array AA. The length of JA is Nz.

- An integer array IA contains the pointers to the beginning of each row in the arrays AA and JA. Thus, the content of $IA(i)$ is the position in arrays AA and JA where the i-th row starts. The length of IA is $n+1$ with $IA(n+1)$ containing the number $IA(1)+Nz$,i.e., the address in A and JA of the beginning of a fictitious row $n+1$.

For example, the above matrix could be stored as follows.

AA	=	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
JA	=	1	4	1	2	4	1	3	4	5	3	4	5
IA	=	1	3	6	10	12	13						

This format is probably the most commonly used to store general sparse matrices. We will refer to it as the *Compressed Sparse Row* (CSR) format. An advantage of this scheme over the coordinate scheme is that it is often more amenable to perform typical computations. On the other hand the coordinate scheme is attractive because of its simplicity and its flexibility. For this reason it is used as the 'entry' format in software packages such as the Harwell library.

There are a number of variations to the Compressed Sparse Row format. The most obvious variation is to store the columns instead of the rows. The corresponding scheme will be called the *Compressed Sparse Column* (CSC) scheme Another common variation exploits the fact that the diagonal elements of many matrices are usually all nonzero and/or that they are accessed more often than the rest of the elements. As a result they can be stored separately. In fact, what we refer to as the *Modified Sparse Row* (MSR) format, consists of only two arrays: a real array AA and an integer array JA. The first n positions in AA contain the diagonal elements of the matrix, in order. The position $n+1$ of the array AA is not used, or may sometimes be used to carry some other information concerning the matrix. Starting at position $n+2$,the nonzero elements of AA,excluding its diagonal elements, are stored row-wise. Corresponding to each element $AA(k)$ the integer $JA(k)$ is the column index of the element $A(k)$ in the matrix AA. The $n+1$ first positions of JA contain the pointer to the beginning of each row in AA and JA. Thus, for the above example the two arrays will be as follows.

AA	=	1.	4.	7.	11.	12.	*	2.	3.	5.	6.	8.	9.	10.
JA	=	7	8	10	13	14	14	4	1	4	1	4	5	3

The star denotes an unused location. Notice that $JA(n) = JA(n+1) = 14$, indicating that the last row, is a zero row, once the diagonal element has been removed.

There are a number of applications that lead to regularly structured matrices. Among these matrices one can distinguish two different types: block matrices, and diagonally structured matrices. Here we discuss only diagonally structured matrices which are matrices whose nonzero elements are located along a small number of diagonals. To store such matrices we may store the diagonals in a rectangular array $DIAG(1:n, 1:Nd)$ where Nd is the number of diagonals. We also need to know the offsets of each of the diagonals with respect to the main diagonal. These will be stored in an array $IOFF(1:Nd)$. Thus, in position (i,j) of the array $DIAG$ is located the element $a_{i,i+\mathrm{IOFF}(j)}$ of the original matrix, i.e.,

$$DIAG(i,j) \leftarrow a_{i,i+\mathrm{ioff}(j)}.$$

The order in which the diagonals are stored in the columns of $DIAG$ is unimportant in general. If many more operations are performed with the main diagonal there may be a slight advantage in storing it in the first column. Note also that all the diagonals except the main diagonal have fewer than n elements, so there are positions in $DIAG$ that will not be used.

For example the following matrix which has three diagonals

$$A = \begin{pmatrix} 1. & 0. & 2. & 0. & 0. \\ 3. & 4. & 0. & 5. & 0. \\ 0. & 6. & 7. & 0. & 8. \\ 0. & 0. & 9. & 10. & 0. \\ 0. & 0. & 0. & 11. & 12. \end{pmatrix} \tag{2.2}$$

will be represented the two arrays

DIAG =

*	1.	2.
3.	4.	5.
6.	7.	8.
9.	10.	*
11	12.	*

IOFF =

-1	0	2

A more general scheme that has been popular on vector machines is the so-called Ellpack-Itpack format. The assumption in this scheme is that we have at most Nd nonzero elements per row, where Nd is small. Then two rectangular arrays of dimension $n \times Nd$ each are required, one real and one integer. The first, $COEF$,is similar to $DIAG$ and contains the nonzero elements of A. We can store the nonzero elements of each row of the matrix in a row of the array

$COEF(1:n, 1:Nd)$ completing the row by zeros if necessary. Together with $COEF$ we need to store an integer array $JCOEF(1:n, 1:Nd)$ which contains the column positions of each entry in $COEF$. Thus, for the above matrix, we would have,

$$\text{COEF} = \begin{bmatrix} 1. & 2. & 0. \\ 3. & 4. & 5. \\ 6. & 7. & 8. \\ 9. & 10. & 0. \\ 11 & 12. & 0. \end{bmatrix} \qquad \text{JCOEF} = \begin{bmatrix} 1 & 3 & 1 \\ 1 & 2 & 4 \\ 2 & 3 & 5 \\ 3 & 4 & 4 \\ 4 & 5 & 5 \end{bmatrix}.$$

Note that in the above $JCOEF$ array we have put a column number equal to the row number, for the zero elements that have been added to pad the rows of $DIAG$ that correspond to shorter rows in the matrix A. This is somewhat arbitrary, and in fact any integer between 1 and n would be acceptable, except that there may be good reasons for not putting the same integers too often, for performance considerations.

2.3 Basic Sparse Matrix Operations

One of the most important operations required in many of the algorithms for computing eigenvalues of sparse matrices is the matrix-by-vector product. We do not intend to show how these are performed for each of the storage schemes considered earlier, but only for a few important ones.

The following Fortran 8-X segment shows the main loop of the matrix by vector operation for matrices stored in the Compressed Sparse Row stored format.

```
DO I=1, N
   K1 = IA(I)
   K2 = IA(I+1)-1
   Y(I) = DOTPRODUCT(A(K1:K2),X(JA(K1:K2)))
ENDDO
```

Notice that each iteration of the loop computes a different component of the resulting vector. This has the obvious advantage that each of these iterations can be performed independently. If the matrix is stored column-wise, then we would use the following code instead.

```
DO J=1, N
   K1 = IA(J)
   K2 = IA(J+1)-1
   Y(JA(K1:K2)) = Y(JA(K1:K2))+X(J)*A(K1:K2)
ENDDO
```

In each iteration of the loop a multiple of the j-th column is added to the result, which is assumed to have been set initially to zero. Notice now that the outer

loop is no longer parallelizable. Barring the use of a different data structure, the only alternative left to improve parallelization is to attempt to split the vector operation in each inner loop, which has few operations, in general. The point of this comparison is that we may have to change data structures to improve performance when dealing with supercomputers.

We now consider the matrix-vector product in diagonal storage.

```
DO J=1, NDIAG
   JOFF = IOFF(J)
   DO I=1, N
      Y(I) = Y(I) + DIAG(I,J)*X(JOFF+I)
   ENDDO
 ENDDO
```

Here, each of the diagonals is multiplied by the vector x and the result added to the vector y. It is again assumed that the vector y has been filled with zero elements before the start of the loop. From the point of view of parallelization and/or vectorization the above code is probably the one that has the most to offer. On the other hand, its drawback is that it is not general enough.

Another important 'kernel' in sparse matrix computations is that of solving a lower or upper-triangular system. The following segment shows a simple routine for solving a unit lower-triangular system.

```
X(1) = Y(1)
DO K = 2, N
   K1 = IAL(K)
   k2 = IAL(K+1)-1
   X(K)=Y(K)-DOTPRODUCT(AL(K1:K2),X(JAL(K1:K2)))
ENDDO
```

2.4 Sparse Direct Solution Methods

Solution methods for large sparse linear systems of equations are important in eigenvalue calculations mainly because they are needed in the context of the shift-and-invert techniques, described in Chapter 4. In these techniques the matrix that is used in the iteration process is $(A - \sigma I)^{-1}$ or $(A - \sigma B)^{-1}B$ for the generalized eigenvalue problem. In this section we give a brief overview of sparse matrix techniques for solving linear systems. The difficulty here is that we must deal with problems that are not only complex, since complex shifts are likely to occur, but also indefinite. There are two broad classes of methods that are commonly used: direct and iterative. Direct methods are more commonly used in the context of shift-and-invert techniques because of their robustness when dealing with indefinite problems.

Most direct methods for sparse linear systems perform an LU factorization of the original matrix and try to reduce cost by minimizing fill-ins, i.e., non-zero elements introduced during the elimination process in positions which were initially

zeros. Typical codes in this category include MA28, see reference [50], from the Harwell library and the Yale Sparse Matrix Package (YSMP), see reference [192]. For a detailed view of sparse matrix techniques we refer to the book by Duff, Erisman, and Reid [52].

Currently, the most popular iterative methods are the preconditioned conjugate gradient type techniques. In these techniques an approximate factorization $A = LU + E$ of the original matrix is obtained and then the conjugate gradient method is applied to a preconditioned system, a form of which is $U^{-1}L^{-1}Ax = U^{-1}L^{-1}b$. The conjugate gradient method is a projection method related to the Lanczos algorithm, which will be described in Chapter 4. One difficulty with conjugate gradient-type methods is that they are designed for matrices that are positive real, i.e., matrices whose symmetric parts are positive definite, and as a result they will perform well for the types of problems that will arise in the context of shift-and-invert.

2.5 Test Problems

When developing algorithms for sparse matrix computations it is desirable to be able to use test matrices that are well documented and often used by other researchers. There are many different ways in which these test matrices can be useful but their most common use is for comparison purposes.

Two different ways of providing data sets consisting of large sparse matrices for test purposes have been used in the past. The first one is to collect sparse matrices in a well-specificed format, from various applications. This approach has is used in the well-known Harwell-Boeing collection of test matrices. The second approach is to collect subroutines or programs that generate such matrices. This approach is taken in the SPARSKIT package which we briefly describe in the next section.

In the course of the book we will often use two test problems in the examples. These are described in detail next. While these two examples are far from being representative of all the problems that occur they have the advantage of being easy to reproduce. They have also been extensively used in the literature.

2.5.1 Random Walk Problem

The first test problem is issued from a Markov model of a random walk on a triangular grid. It was proposed by G. W. Stewart [201] and has been used in several papers for testing eigenvalue algorithms. The problem models a random walk on a $(k+1) \times (k+1)$ triangular grid as is shown in Figure 2.3.

We label by (i, j) the node of the grid with coordinates (ih, jh) where h is the grid spacing, for $i, j = 0, 1, ...k$. A particle moves randomly on the grid by jumping from a node (i, j) into either of its (at most 4) neighbors. The probability of jumping from node (i, j) to either node $(i-1, j)$ or node $(i, j-1)$ (down

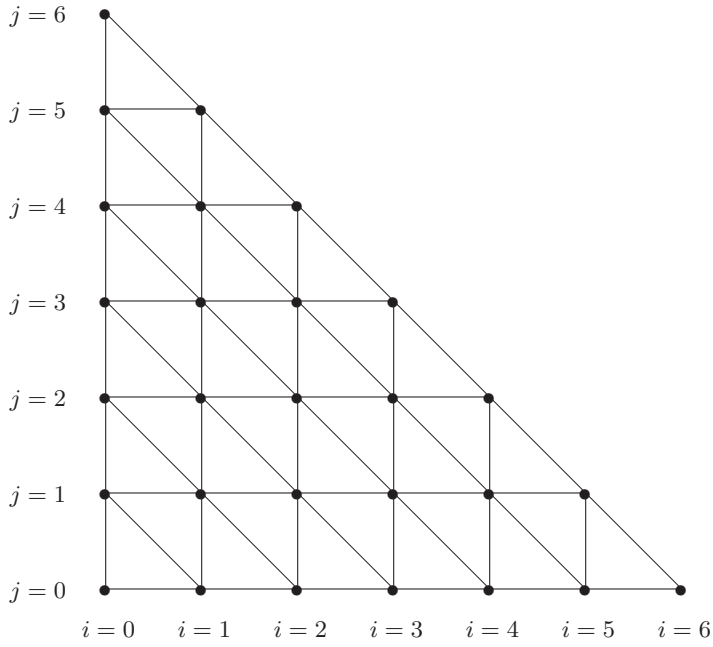


Figure 2.3: Random walk on a triangular grid

transition) is given by

$$\mathrm{pd}(i,j) = \frac{i+j}{2k}$$

this probability being doubled when either i or j is equal to zero. The probability of jumping from node (i,j) to either node $(i{+}1,j)$ or node $(i,j{+}1)$ (up transition) is given by

$$\mathrm{pu}(i,j) = \frac{1}{2} - \mathrm{pd}(i,j).$$

Note that there cannot be an up transition when $i+j=k$, i.e., for nodes on the oblique boundary of the grid. This is reflected by the fact that in this situation $\mathrm{pu}(i,j) = 0$.

The problem is to compute the steady state probability distribution of the chain, i.e., the probabilities that the particle be located in each grid cell after a very long period of time. We number the nodes from the bottom up and from left to right, i.e., in the order,

$$(0,0),(0,1),\ldots,(0,k);(1,0),(1,1),\ldots(1,k-1);.....;(k,0)$$

The matrix P of transition probabilities is the matrix whose generic element $p_{k,q}$ is the probability that the particle jumps from node k to node q. This is a stochastic

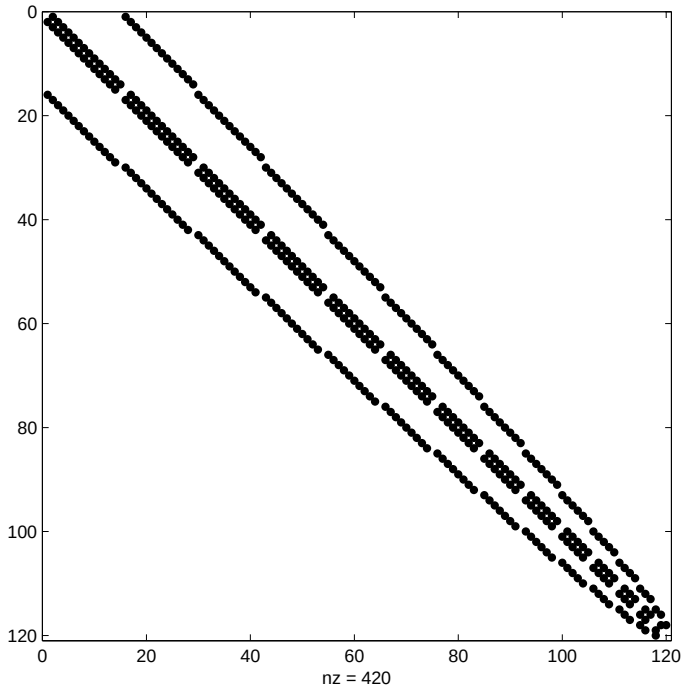


Figure 2.4: Sparsity pattern of the matrix Mark(15).

matrix, i.e., its elements are nonnegative and the sum of elements in the same row is equal to one. The vector $(1, 1, ..., 1)^T$ is an eigenvector of P associated with the eigenvalue unity. As is known the steady state probability distribution vector is the appropriately scaled eigenvector of the transpose of P associated with the eigenvalue one. Note that the number of different states is $\frac{1}{2}(k+1)(k+2)$, which is the dimension of the matrix. We will denote by Mark(k+1) the corresponding matrix. Figure 2.4 shows the sparsity pattern of Mark(15) which is a matrix of dimension $n = 120$ with $nz = 420$ nonzero elements.

2.5.2 Chemical Reactions

The second test example, models concentration waves in reaction and transport interaction of some chemical solutions in a tubular reactor. The concentrations $x(\tau, z), y(\tau, z)$ of two reacting and diffusing components, where $0 \leq z \leq 1$ represents a coordinate along the tube, and τ is the time, are modeled by the

system:

$$\frac{\partial x}{\partial \tau} = \frac{D_x}{L^2}\frac{\partial^2 x}{\partial z^2} + f(x,y), \tag{2.3}$$

$$\frac{\partial y}{\partial \tau} = \frac{D_y}{L^2}\frac{\partial^2 y}{\partial z^2} + g(x,y), \tag{2.4}$$

with the initial condition

$$x(0,z) = x_0(z), \quad y(0,z) = y_0(z), \quad \forall\, z \in [0,1],$$

and the Dirichlet boundary conditions:

$$x(0,\tau) = x(1,\tau) = \bar{x}$$

$$y(0,\tau) = y(1,\tau) = \bar{y}.$$

The linear stability of the above system is traditionally studied around the steady state solution obtained by setting the partial derivatives of x and y with respect to time to be zero. More precisely, the stability of the system is the same as that of the Jacobian of (2.3) - (2.4) evaluated at the steady state solution. In many problems one is primarily interested in the existence of limit cycles, or equivalently the existence of periodic solutions to (2.3), (2.4). This translates into the problem of determining whether the Jacobian of (2.3), (2.4) evaluated at the steady state solution admits a pair of purely imaginary eigenvalues.

We consider in particular the so-called Brusselator wave model in which

$$f(x,y) = A - (B+1)x + x^2 y$$
$$g(x,y) = Bx - x^2 y.$$

Then, the above system admits the trivial stationary solution $\bar{x} = A,\ \bar{y} = B/A$. A stable periodic solution to the system exists if the eigenvalues of largest real parts of the Jacobian of the right-hand side of (2.3), (2.4) is exactly zero. To verify this numerically, we first need to discretize the equations with respect to the variable z and compute the eigenvalues with largest real parts of the resulting discrete Jacobian.

For this example, the exact eigenvalues are known and the problem is analytically solvable. The following set of parameters have been commonly used in previous articles,

$$D_x = 0.008, \quad D_y = \frac{1}{2}D_x = 0.004,$$
$$A = 2, \quad B = 5.45\ .$$

The bifurcation parameter is L. For small L the Jacobian has only eigenvalues with negative real parts. At $L \approx 0.51302$ a purely imaginary eigenvalue appears.

We discretize the interval $[0,1]$ using $n+1$ points, and define the mesh size $h \equiv 1/n$. The discrete vector is of the form $\begin{pmatrix} x \\ y \end{pmatrix}$ where x and y are n-dimensional

vectors. Denoting by f_h and g_h the corresponding discretized functions f and g,the Jacobian is a 2×2 block matrix in which the diagonal blocks $(1,1)$ and $(2,2)$ are the matrices

$$\frac{1}{h^2}\frac{D_x}{L^2}\ \text{tridiag}\ \{1,-2,1\} + \frac{\partial f_h(x,y)}{\partial x}$$

and

$$\frac{1}{h^2}\frac{D_y}{L^2}\ \text{tridiag}\ \{1,-2,1\} + \frac{\partial g_h(x,y)}{\partial y}$$

respectively, while the blocks $(1,2)$ and $(2,1)$ are

$$\frac{\partial f_h(x,y)}{\partial y} \quad \text{and} \quad \frac{\partial g_h(x,y)}{\partial x}$$

respectively. Note that because the steady state solution is a constant with respect to the variable z,the Jacobians of either f_h or g_h with respect to either x or y are scaled identity matrices. We denote by A the resulting $2n \times 2n$ Jacobian matrix. The matrix A has the following structure

$$A = \begin{pmatrix} \alpha T & \beta I \\ \gamma I & \delta T \end{pmatrix},$$

In which $T = \text{tridiag}\ \{1,-2,1\}$,and α,β,γ, and δ are scalars. The exact eigenvalues of A are readily computable, since there exists a quadratic relation between the eigenvalues of the matrix A and those of the classical difference matrix T.

2.5.3 The Harwell-Boeing Collection

This large collection of test matrices has been gathered over several years by I. Duff (Harwell) and R. Grimes and J. Lewis (Boeing) [53]. The number of matrices in the collection at the time of this writing is 292. The matrices have been contributed by researchers and engineers in many different areas. The sizes of the matrices vary from very small, such as counter example matrices, to very large. One drawback of the collection is that it contains few *non-Hermitian* eigenvalue problems. Many of the eigenvalue problems in the collection are from structural engineering, which are generalized eigenvalue problems. One the other hand the collection provides a data structure which constitutes an excellent medium of exchanging matrices.

The matrices are stored as ASCII files with a very specific format consisting of a 4 or 5 line header and then the data containing the matrix stored in CSC format together with any right-hand sides, initial guesses, or exact solutions.

The collection is available for public distribution from the authors.

2.6 SPARSKIT

SPARSKIT is a package aimed at providing subroutines and utilities for working with general sparse matrices. Its purpose is not as much to solve particular problems involving sparse matrices (linear systems, eigenvalue problems) but rather

to make available the little tools to manipulate and performs simple operations with sparse matrices. For example there are tools for exchanging data structures, e.g., passing from the Compressed Sparse Row format to the diagonal format and vice versa. There are various tools for extracting submatrices or performing other similar manipulations. SPARSKIT also provides matrix generation subroutines as well as basic linear algebra routines for sparse matrices (such as addition, multiplication, etc...).

A short description of the contents of SPARSKIT follows. The package is divided up in six modules, each having a different function. To refer to these six parts we will use the names of the subdirectories where they are held in the package in its current version.

FORMATS This module contains essentially two sets of routines. The first set contained in the file formats.f consists of the routines needed to translate data structures. Translations from the basic Compressed Sparse Row format to any of the other formats supported is provided together with a routine for the reverse transformation. This way one can translate from any of the data structures supported to any other one with two transformation at most. The formats currently supported are the following.

DNS Dense format

BND Linpack Banded format

CSR Compressed Sparse Row format

CSC Compressed Sparse Column format

COO Coordinate format

ELL Ellpack-Itpack generalized diagonal format

DIA Diagonal format

BSR Block Sparse Row format

MSR Modified Compressed Sparse Row format

SSK Symmetric Skyline format

NSK Nonsymmetric Skyline format

JAD The Jagged Diagonal scheme

The second set of routines contains a number of routines, currently 27, called 'unary', to perform simple manipulation functions on sparse matrices, such as extracting a particular diagonal or permuting a matrix, or yet for filtering out small elements. For reasons of space we cannot list these routines here.

BLASSM This module contains a number of routines for doing basic linear algebra with sparse matrices. It is comprised of essentially two sets of routines. Basically, the first one consists of matrix-matrix operations (e.g., multiplication of matrices) and the second consists of matrix-vector operations. The first set allows to perform the following operations with sparse matrices, where A, B, C are sparse matrices, D is a diagonal matrix, and σ is a scalar. $C = AB$,$C = A + B$,$C = A + \sigma B$,$C = A \pm B^T$, $C = A + \sigma B^T$,$A := A + \sigma I$,$C = A + D$.

The second set contains various routines for performing matrix by vector products and solving sparse triangular linear systems in different storage formats.

INOUT This module consists of routines to read and write matrices in the Harwell-Boeing format. For more information on this format and the Harwell-Boeing collection see the reference [53]. It also provides routines for plotting the pattern of the matrix or simply dumping it in a nice format.

INFO There is currently only one subroutine in this module. Its purpose is to provide as many statistics as possible on a matrix with little cost. About 33 lines of information are written. For example, the code analyzes diagonal dominance of the matrix (row and column), its degree of symmetry (structural as well as numerical), its block structure, its diagonal structure, etc,...

MATGEN The set of routines in this module allows one to generate test matrices. For now there are generators for 5 different types of matrices.

1. Five-point and seven point matrices on rectangular regions discretizing a general elliptic partial differential equation.

2. Same as above but provides block matrices (several degrees of freedom per grid point in the PDE).

3. Finite elements matrices for the heat condition problem, using various domains (including user provided ones).

4. Test matrices from the paper by Z. Zlatev, K. Schaumburg, and J. Wasniewski, [228].

5. Markov chain matrices arising from a random walk on a triangular grid. See Section 2.5.1 for details.

UNSUPP As is suggested by its name this module contains various *unsupported* software tools that are not necessarily portable or that do not fit in any of the previous modules. For example software for viewing matrix patterns on some workstations will be found here. For now UNSUPP contains subroutines for visualizing matrices and a preconditioned GMRES package (with a 'robust' preconditioner based on Incomplete LU factorization with controlled fill-in).

2.7 The New Sparse Matrix Repositories

The Harwell-Boeing collection project started in the 1980s [53]. As time went by, matrices of this collection became too small relative to the capabilities of modern computers. As a result several collections were added. The best known of these in the sparse matrix computation communities are the matrix market located in

`http://math.nist.gov/MatrixMarket`

and the Florida collection:

`http://www.cise.ufl.edu/research/sparse/matrices/`

In addition, the old Harwell-Boeing format which was geared toward efficient utilization from fortran 77, became unnecessarily rigid. This format was developed to save memory but with todays capabilities it is not worth it to avoid the coordinate format to save a space. Recall that for square matrices that the requirements for these two schemes ia as follows: `Nz*(1*float + 2*int)` for the COO format versus `Nz*(1*float + 1*int) + n*int` for the CSC/CSC format. Instead one can store matrix data in a file and an information header of arbitrary length can be added. Separate files can be used for the right-hand sides and solutions. This is the essence of the Matrix Market format (MM). The first line of the file header if of the form (for example)

`%%MatrixMarket matrix coordinate real general`

which specifies the format, the class of storage used (here coordinate), the type of data (real) and the type of storage (general, meaning symmetry is not exploited).

These new storage schemes made it necessary to develop additional tools to deal with them. For example, the site

`http://bebop.cs.berkeley.edu/smc/`

offers a package named BeBop for converting matrices between various formats.

2.8 Sparse Matrices in Matlab

Matlab^{TM} is a commercial interactive programming language [2] which was developed initially as an interactive version to the Linpack[46] and Eispack [195] packages, now both replaced by LAPACK. GNU Octave [3], is a rather similar product based on a GNU-community effort, which is also publically available (under the GPL license). Matlab became more common for performing general computations. As its use began to spread in the scientific computing community, there was a need to provide support for sparse matrices. Starting in the mid 1990's this support became available. GNU Octave has also added support for sparse matrices in recent years.

It is possible to generate sparse matrices, solve sparse linear systems, and compute eigenvalues of large sparse matrices with Matlab or Octave. The following descriptions is restricted to Matlab but Octave can be invoked in essentially an identical way. Matlab scripts can be invoked to implement functions. For example, the following few lines of code will generate the sparse matrix related to the

[2] See: `http://www.mathworks.com/`

[3] See: `http://www.gnu.org/software/octave/`

Markov Chain example seen in Section 2.5.1.

```
function [A] = mark(m)
%% [A] = mark(m)
%% generates a Markov chain matrix for a random walk
%% on a triangular ggrid --
%% The matrix is sparse -- and of size n= m*(m+1)/2
%%-------------------------------------------------
      ix = 0;
      cst = 0.5/(m-1) ;
      n = (m*(m+1))/2;
      A = sparse(n,n) ;
%%-------------------- sweep y coordinates;
      for i=1:m
        jmax = m-i+1;
%%-------------------- sweep x coordinates;
        for j=1:jmax,
           ix = ix + 1;
           if (j<jmax)
              pd = cst*(i+j-1) ;
%%-------------------- north move
              jx =  ix + 1;
             jx =  ix + 1;
              A(ix,jx) = pd;
              if (i == 1)
                 A(ix,jx) = A(ix,jx)+pd;
              end
%%-------------------- east move
              jx = ix + jmax;
              A(ix,jx) = pd;
              if (j == 1)
                 A(ix,jx) = A(ix,jx)+pd;
              end
         end
%%--------------------  south move
           pu = 0.5 - cst*(i+j-3) ;
           if ( j>1)
               jx = ix-1;
               A(ix,jx) = pu;
            end
%%-------------------- west move
            if ( i > 1)
               jx = ix - jmax - 1 ;
               A(ix,jx) = pu;
   end
```

```
 end
```

Once Matlab is launched in a directory where this script is available, then we can issue the following commands for example

```
>> A = mark(15);
>> size(A)
ans =
   120   120
>> spy(A);
```

The lines starting with $>>$ are commands typed in and `ans` are responses (if any) from Matlab. The command `spy(A)` generated the plot used in Figure 2.4.

Matlab enables one to compute eigenvalues of full matrices with the `eig` command. In the test shown above the matrice is sparse so `eig(A)` will generate an error. Instead the matrix must first converted to dense format:

```
>> eig(full(A))
ans =
  -1.0000
   1.0000
   0.9042
   0.9714
   0.8571
  -0.9042
  -0.9714
  -0.8571
   .....
```

Only the first 8 eigenvalues are shown but the command generated 120 numbers, all 120 eigenvalies. What if the matrix is too large to be converted to dense format first? Then one can use the `eigs` command which computes a few eigenvalues using some of the methods which will be covered later in this book. Specifically the ARPACK package [118] is invoked.

PROBLEMS

P-2.1 Write a FORTRAN code segment to perform the matrix-vector product for matrices stored in Ellpack-Itpack format.

P-2.2 Write a small subroutine to perform the following operations on a sparse matrix in coordinate format, diagonal format, and in CSR format: a) count the number on nonzero elements in the main diagonal; b) extract the diagonal whose offset is k (which may be negative); c) add a nonzero element in position (i, j) of the matrix (assume that this position may contain a zero or a nonzero element); d) add a given diagonal to the matrix. What is the most convenient storage scheme for each of these operations?

P-2.3 Generate explicitly the matrix $\mathrm{Mark}(4)$. Verify that it is a stochastic matrix. Verify that 1 and -1 are eigenvalues.

NOTES AND REFERENCES. Two good sources of reading on sparse matrix computations are the books by George and Liu [71] and by Duff, Erisman, and Reid [52]. Also of interest are [140] and [159] and the early survey by Duff [49]. A notable recent addition to these is the volume by Davis [43], which deals with sparse direct solution methods and contains a wealth of hepful details for dealing with sparse matrices.

For applications related to eigenvalue problems, see [37] and [13]. For details on Markov Chain modeling see [106, 191, 206].

SPARSKIT [180] is now more than 20 years old. It is written in FORTRAN-77 and as such is somewhat outdated. However, the many routines available therein remain useful, judging from the requests I receive. ■

Chapter 3

PERTURBATION THEORY AND ERROR ANALYSIS

This chapter introduces some elementary spectral theory for linear operators on finite dimensional spaces as well as some elements of perturbation analysis. The main question that perturbation theory addresses is: how does an eigenvalue and its associated eigenvectors, spectral projector, etc.., vary when the original matrix undergoes a small perturbation. This information is important both for theoretical and practical purposes. The spectral theory introduced in this chapter is the main tool used to extend what is known about spectra of matrices to general operators on infinite dimensional spaces. However, it has also some consequences in analyzing the behavior of eigenvalues and eigenvectors of matrices. The material discussed in this chapter is probably the most theoretical of the book. Fortunately, most of it is independent of the rest and may be skipped in a first reading. The notions of condition numbers and some of the results concerning error bounds are crucial in understanding the difficulties that eigenvalue routines may encounter.

3.1 Projectors and their Properties

A projector P is a linear transformation from $\mathbb{C}^n$ to itself which is idempotent, i.e., such that

$$P^2 = P.$$

When P is a projector then so is $(I - P)$ and we have $\mathrm{Null}(P) = \mathrm{Ran}(I - P)$. The two subspaces $\mathrm{Null}(P)$ and $\mathrm{Ran}(P)$ have only the element zero in common. This is because if a vector x is in $\mathrm{Ran}(P)$ then $Px = x$ and if it is also in $\mathrm{Null}(P)$ then $Px = 0$ so that $x = 0$ and the intersection of the two subspaces reduces to $\{0\}$. Moreover, every element of $\mathbb{C}^n$ can be written as $x = Px + (I - P)x$. As a result the space $\mathbb{C}^n$ can be decomposed as the direct sum

$$\mathbb{C}^n = \mathrm{Null}(P) \oplus \mathrm{Ran}(P).$$

Conversely, every pair of subspaces M and S that form a direct sum of $\mathbb{C}^n$ define a unique projector such that $\mathrm{Ran}(P) = M$ and $\mathrm{Null}(P) = S$. The corresponding transformation P is the linear mapping that maps any element x of

$\mathbb{C}^n$ into the component x_1 where x_1 is the M-component in the unique decomposition $x = x_1 + x_2$ associated with the direct sum. In fact, this association is unique in that a projector is uniquely determined by its null space and its range, two subspaces that form a direct sum of $\mathbb{C}^n$.

3.1.1 Orthogonal Projectors

An important particular case is when the subspace S is the orthogonal complement of M,i.e., when

$$\mathrm{Null}(P) = \mathrm{Ran}(P)^{\perp}.$$

In this case the projector P is said to be the *orthogonal projector* onto M. Since $\mathrm{Ran}(P)$ and $\mathrm{Null}(P)$ from a direct sum of $\mathbb{C}^n$,the decomposition $x = Px + (I - P)x$ is unique and the vector Px is uniquely defined by the set of equations

$$Px \in M \quad \text{and} \quad (I - P)x \perp M \tag{3.1}$$

or equivalently,

$$Px \in M \quad \text{and} \quad ((I - P)x, y) = 0 \quad \forall y \in M\ .$$

Proposition 3.1 *A projector is orthogonal if and only if it is Hermitian.*

Proof. As a consequence of the equality

$$(P^H x, y) = (x, Py) \quad \forall x\ ,\ \forall y \tag{3.2}$$

we conclude that

$$\mathrm{Null}(P^H) = \mathrm{Ran}(P)^{\perp} \tag{3.3}$$

$$\mathrm{Null}(P) = \mathrm{Ran}(P^H)^{\perp}\ . \tag{3.4}$$

By definition an orthogonal projector is one for which $\mathrm{Null}(P) = \mathrm{Ran}(P)^{\perp}$. Therefore, by (3.3), if P is Hermitian then it is orthogonal.

To show that the converse is true we first note that P^H is also a projector since $(P^H)^2 = (P^2)^H = P^H$. We then observe that if P is orthogonal then (3.3) implies that $\mathrm{Null}(P) = \mathrm{Null}(P^H)$ while (3.4) implies that $\mathrm{Ran}(P) = \mathrm{Ran}(P^H)$. Since P^H is projector this implies that $P = P^H$,because a projector is uniquely determined by its range and its null space. □

Given *any* unitary $n \times m$ matrix V whose columns form an orthonormal basis of $M = \mathrm{Ran}(P)$,we can represent P by the matrix $P = VV^H$. Indeed, in addition to being idempotent, the linear mapping associated with this matrix satisfies the characterization given above, i.e.,

$$VV^H x \ \in M \quad \text{and} \quad (I - VV^H)x \ \in \ M^{\perp}.$$

It is important to note that this representation of the orthogonal projector P is not unique. In fact any orthonormal basis V will give a different representation of P in the above form. As a consequence for any two orthogonal bases V_1, V_2 of M, we must have $V_1 V_1^H = V_2 V_2^H$,an equality which can also be verified independently, see Exercise P-3.2.

From the above representation it is clear that when P is an orthogonal projector then we have $\|Px\|_2 \leq \|x\|_2$ for any x. As a result the maximum of $\|Px\|_2/\|x\|_2$ for all x in $\mathbb{C}^n$ does not exceed one. On the other hand the value one is reached for any element in $\mathrm{Ran}(P)$ and therefore,

$$\|P\|_2 = 1$$

for any orthogonal projector P.

Recall that the acute angle between two nonzero vectors of $\mathbb{C}^n$ is defined by

$$\cos\theta(x,y) = \frac{|(x,y)|}{\|x\|_2\|y\|_2} \quad 0 \leq \theta(x,y) \leq \frac{\pi}{2} .$$

We define the acute angle between a vector and a subspace S as the smallest acute angle made between x and all vectors y of S,

$$\theta(x,S) = \min_{y\in S} \theta(x,y) . \tag{3.5}$$

An optimality property of orthogonal projectors is the following.

Theorem 3.1 *Let P be an orthogonal projector onto the subspace S. Then given any vector x in $\mathbb{C}^n$ we have,*

$$\min_{y\in S} \|x-y\|_2 = \|x - Px\|_2 , \tag{3.6}$$

or, equivalently,

$$\theta(x,S) = \theta(x,Px) . \tag{3.7}$$

Proof. Let y any vector of S and consider the square of its distance from x. We have,

$$\|x-y\|_2^2 = \|x - Px + (Px - y)\|_2^2 = \|x - Px\|_2^2 + \|(Px - y)\|_2^2 ,$$

because $x-Px$ is orthogonal to S to which $Px-y$ belongs. Therefore, $\|x-y\|_2 \geq \|x - Px\|_2$ for all y in S and this establishes the first result by noticing that the minimum is reached for $y = Px$. The second equality is a simple reformulation of the first. □

It is sometimes important to be able to measure distances between two subspaces. If P_i represents the orthogonal projector onto M_i,for $i = 1, 2$,a natural measure of the distance between M_1 and M_2 is provided by their *gap* defined by:

$$\omega(M_1, M_2) = \max\left\{ \max_{\substack{x\in M_2 \\ \|x\|_2=1}} \|x - P_1 x\|_2 , \max_{\substack{x\in M_1 \\ \|x\|_2=1}} \|x - P_2 x\|_2 \right\} .$$

We can also redefine $\omega(M_1, M_2)$ as

$$\omega(M_1, M_2) = \max\{\|(I - P_1)P_2\|_2 \ , \ \|(I - P_2)P_1\|_2\}$$

and it can even be shown that

$$\omega(M_1, M_2) = \|P_1 - P_2\|_2. \tag{3.8}$$

3.1.2 Oblique Projectors

A projector that is not orthogonal is said to be oblique. It is sometimes useful to have a definition of oblique projectors that resembles that of orthogonal projectors, i.e., a definition similar to (3.1). If we call L the subspace that is the orthogonal complement to $S = \mathrm{Null}(P)$,it is clear that L will have the same dimension as M. Moreover, to say that $(I - P)x$ belongs to $\mathrm{Null}(P)$ is equivalent to saying that it is in the orthogonal complement of L. Therefore, from the definitions seen at the beginning of Section 1, the projector P can be characterized by the defining equation

$$Px \in M \quad \text{and} \quad (I - P)x \perp L \, . \tag{3.9}$$

We say that P is a projector onto M and orthogonal to L or along the orthogonal complement of L. This is illustrated in Figure 3.1.

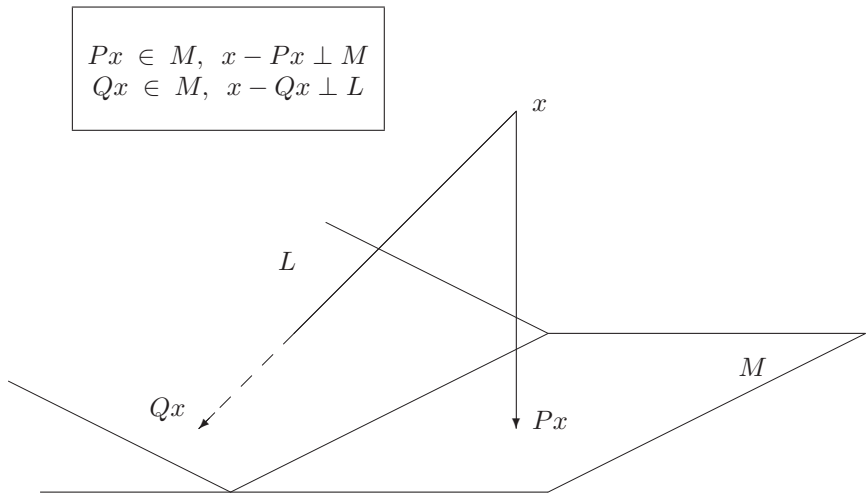


Figure 3.1: Orthogonal and oblique projectors P and Q.

Matrix representations of oblique projectors require two bases: a basis $V = [v_1, \ldots, v_m]$ of the subspace $M = \mathrm{Ran}(P)$ and the other $W = [w_1, \ldots, w_m]$ for

the subspace L,the orthogonal complement of $\mathrm{Null}(P)$. We will say that these two bases are *biorthogonal* if

$$(v_i, w_j) = \delta_{ij} \tag{3.10}$$

Given any pair of biorthogonal bases V, W the projector P can be represented by

$$P = VW^H \tag{3.11}$$

In contrast with orthogonal projectors, the norm of P is larger than one in general. It can in fact be arbitrarily large, which implies that the norms of $P - Q$,for two oblique projectors P and Q, will not, in general, be a good measure of the distance between the two subspaces $\mathrm{Ran}(P)$ and $\mathrm{Ran}(Q)$. On the other hand, it may give an idea on the difference between their rank as is stated in the next theorem.

Theorem 3.2 *Let $\|.\|$ be any matrix norm, and assume that two projectors P and Q are such that $\|P - Q\| < 1$ then*

$$\mathrm{rank}(P) \;=\; \mathrm{rank}(Q) \tag{3.12}$$

Proof. First let us show that $\mathrm{rank}(Q) \leq \mathrm{rank}(P)$. Given a basis $\{x_i\}_{i=1,\ldots,q}$ of $\mathrm{Ran}(Q)$ we consider the family of vectors $G = \{Px_i\}_{i=1,\ldots,q}$ in $\mathrm{Ran}(P)$ and show that it is linearly independent. Assume that

$$\sum_{i=1}^{q} \alpha_i P x_i = 0.$$

Then the vector $y = \sum_{i=1}^{q} \alpha_i x_i$ is such that $Py = 0$ and therefore $(Q - P)y = Qy = y$ and $\|(Q - P)y\| = \|y\|$. Since $\|Q - P\| < 1$ this implies that $y = 0$. As a result the family G is linearly independent and so $\mathrm{rank}(P) \geq q = \mathrm{rank}(Q)$. It can be shown similarly that $\mathrm{rank}(P) \leq \mathrm{rank}(Q)$. □

The above theorem indicates that no norm of $P-Q$ can be less than one if the two subspaces have different dimensions. Moreover, if we have a family of projectors $P(t)$ that depends continuously on t then the rank of $P(t)$ remains constant. In addition, an immediate corollary is that if the gap between two subspaces is less than one then they must have the same dimension.

3.1.3 Resolvent and Spectral Projector

For any given complex z not in the spectrum of a matrix A we define the resolvent operator of A at z as the linear transformation

$$R(A, z) = (A - zI)^{-1} . \tag{3.13}$$

The notation $R(z)$ is often used instead of $R(A, z)$ if there is no ambiguity. This notion can be defined for operators on infinite dimensional spaces in which case

the spectrum is defined as the set of all complex scalars such that the inverse of $(A - zI)$ does not exist, see reference [22, 105] for details.

The resolvent regarded as a function of z admits singularities at the eigenvalues of A. Away from any eigenvalue the resolvent $R(z)$ is analytic with respect to z. Indeed, we can write for any z around an element z_0 not equal to an eigenvalue,

$$\begin{aligned} R(z) \equiv (A - zI)^{-1} &= ((A - z_0 I) - (z - z_0)I)^{-1} \\ &= R(z_0)(I - (z - z_0)R(z_0))^{-1} \end{aligned}$$

The term $(I-(z-z_0)R(z_0))^{-1}$ can be expanded into the Neuman series whenever the spectral radius of $(z - z_0)R(z_0)$ is less than unity. Therefore, the Taylor expansion of $R(z)$ in the open disk $|z - z_0| < 1/\rho(R(z_o))$ exists and takes the form,

$$R(z) = \sum_{k=0}^{\infty} (z - z_0)^k R(z_0)^{k+1}. \tag{3.14}$$

It is important to determine the nature of the singularity of $R(z)$ at the eigenvalues $\lambda_i, i = 1, \ldots, p$. By a simple application of Cramer's rule it is easy to see that these singularities are not essential. In other words, the Laurent expansion of $R(z)$

$$R(z) = \sum_{k=-\infty}^{+\infty} (z - \lambda_i)^k C_k$$

around each pole λ_i has only a finite number of negative powers. Thus, $R(z)$ is a meromorphic function.

The resolvent satisfies the following immediate properties.

First resolvent equality:

$$R(z_1) - R(z_2) = (z_1 - z_2)R(z_1)R(z_2) \tag{3.15}$$

Second resolvent equality:

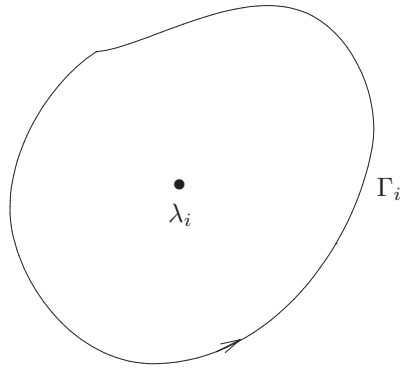
$$R(A_1, z) - R(A_2, z) = R(A_1, z)(A_2 - A_1)R(A_2, z) \tag{3.16}$$

In what follows we will need to integrate the resolvent over Jordan curves in the complex plane. A Jordan curve is a simple closed curve that is piecewise smooth and the integration will always be counter clockwise unless otherwise stated. There is not much difference between integrating complex valued functions with values in $\mathbb{C}$ or in $\mathbb{C}^{n\times n}$. In fact such integrals can be defined over functions taking their values in Banach spaces in the same way.

Consider *any* Jordan curve Γ_i that encloses the eigenvalue λ_i and no other eigenvalue of A,and let

$$P_i = \frac{-1}{2\pi i} \int_{\Gamma_i} R(z)dz \tag{3.17}$$

The above integral is often referred to as the Taylor-Dunford integral.



3.1.4 Relations with the Jordan form

The purpose of this subsection is to show that the operator P_i defined by (3.17) is identical with the spectral projector defined in Chapter 1 by using the Jordan canonical form.

Theorem 3.3 *The linear transformations $P_i,\ i = 1, 2, \ldots, p$, associated with the distinct eigenvalues $\lambda_i, i = 1, \ldots, p$, are such that*

(1) $P_i^2 = P_i$, i.e., each P_i is a projector.

(2) $P_i P_j = P_j P_i = 0$ if $i \neq j$.

(3) $\sum_{i=1}^{p} P_i = I$.

Proof. (1) Let Γ and Γ' two curves enclosing λ_i with Γ' enclosing Γ. Then

$$\begin{aligned}(2i\pi)^2 P_i^2 &= \int_\Gamma \int_{\Gamma'} R(z)R(z')dzdz' \\ &= \int_\Gamma \int_{\Gamma'} \frac{1}{z'-z}(R(z') - R(z))dz'dz\end{aligned}$$

because of the first resolvent equality. We observe that

$$\int_\Gamma \frac{dz}{z'-z} = 0 \quad \text{and} \quad \int_{\Gamma'} \frac{dz'}{z'-z} = 2i\pi,$$

so that

$$\int_\Gamma \int_{\Gamma'} \frac{R(z')}{z'-z}dz'dz \;=\; \int_{\Gamma'} R(z') \left(\int_\Gamma \frac{dz}{z'-z} \right) dz' = 0$$

and,

$$\int_\Gamma \int_{\Gamma'} \frac{R(z)}{z'-z}dz'dz \;=\; \int_\Gamma R(z) \left(\int_{\Gamma'} \frac{dz'}{z'-z} \right) dz \;=\; 2i\pi \int_\Gamma R(z)dz$$

from which we get $P_i^2 = P_i$.

(2) The proof is similar to (1) and is left as an exercise.

(3) Consider

$$P = \frac{-1}{2i\pi} \sum_{i=1}^{p} \int_{\Gamma_i} R(z)dz \ .$$

Since $R(z)$ has no poles outside of the p Jordan curves, we can replace the sum of the integrals by an integral over any curve that contains all of the eigenvalues of A. If we choose this curve to be a circle C of radius r and center the origin, we get

$$P = \frac{-1}{2i\pi} \int_{C} R(z)dz \ .$$

Making the change of variables $t = 1/z$ we find that

$$P = \frac{-1}{2i\pi} \int_{C'_-} (A - (1/t)I)^{-1} \left(-\frac{dt}{t^2}\right) \ = \ \frac{-1}{2i\pi} \int_{C'_+} (tA - I)^{-1} \frac{dt}{t}$$

where C'_- (resp. C'_+) is the circle of center the origin, radius $1/r$ run clock-wise (resp. counter-clockwise). Moreover, because r must be larger than $\rho(A)$ we have $\rho(tA) < 1$ and the inverse of $I - tA$ is expandable into its Neuman series, i.e., the series

$$(I - tA)^{-1} = \sum_{k=0}^{\infty} (tA)^k$$

converges and therefore,

$$P = \frac{1}{2i\pi} \int_{C'_+} \left[\sum_{k=0}^{k=\infty} t^{k-1} A^k \right] dt = I$$

by the residue theorem. □

The above theorem shows that the projectors P_i satisfy the same properties as those of the spectral projector defined in the previous chapter, using the Jordan canonical form. However, to show that these projectors are identical we still need to prove that they have the same range. Note that since A and $R(z)$ commute we get by integration that $AP_i = P_iA$ and this implies that the range of P_i is invariant under A. We must show that this invariant subspace is the invariant subspace M_i associated with the eigenvalue λ_i,as defined in Chapter 1. The next lemma establishes the desired result.

Lemma 3.1 *Let $\hat{M}_i = \mathrm{Ran}(P_i)$ and let $M_i = \mathrm{Null}(A - \lambda_i I)^{l_i}$ be the invariant subspace associated with the eigenvalue λ_i. Then we have $M_i = \hat{M}_i$ for $i = 1, 2, \ldots, p$.*

Proof. We first prove that $M_i \subseteq \hat{M}_i$. This follows from the fact that when $x \in \text{Null}(A - \lambda_i I)^{l_i}$,we can expand $R(z)x$ as follows:

$$\begin{aligned} R(z)x &= (A - zI)^{-1}x \\ &= [(A - \lambda_i I) - (z - \lambda_i)I]^{-1}x \\ &= -\frac{1}{z - \lambda_i}\left[I - (z - \lambda_i)^{-1}(A - \lambda_i I)\right]^{-1}x \\ &= \frac{-1}{z - \lambda_i}\sum_{j=0}^{l_i}(z - \lambda_i)^{-j}(A - \lambda_i I)^j x\ . \end{aligned}$$

The integral of this over Γ_i is simply $-2i\pi x$ by the residue theorem, hence the result.

We now show that $\hat{M}_i \subseteq M_i$. From

$$(z - \lambda_i)R(z) = -I + (A - \lambda_i I)R(z) \tag{3.18}$$

it is easy to see that

$$\frac{-1}{2i\pi}\int_\Gamma (z - \lambda_i)R(z)dz = \frac{-1}{2i\pi}(A - \lambda_i I)\int_\Gamma R(z)dz = (A - \lambda_i I)P_i$$

and more generally,

$$\begin{aligned} \frac{-1}{2i\pi}\int_\Gamma (z - \lambda_i)^k R(z)dz &= \frac{-1}{2i\pi}(A - \lambda_i I)^k \int_\Gamma R(z)dz \\ &= (A - \lambda_i I)^k P_i\ . \end{aligned} \tag{3.19}$$

Notice that the term in the left-hand side of (3.19) is the coefficient A_{-k-1} of the Laurent expansion of $R(z)$ which has no essential singularities. Therefore, there is some integer k after which all the left-hand sides of (3.19) vanish. This proves that for every $x = P_i x$ in $\hat{M}_i$,there exists some l for which $(A - \lambda_i I)^k x = 0, k \geq l$. It follows that x belongs to M_i. □

This finally establishes that the projectors P_i are identical with those defined with the Jordan canonical form and seen in Chapter 1. Each projector P_i is associated with an eigenvalue λ_i. However, it is important to note that more generally one can define a projector associated with a group of eigenvalues, which will be the sum of the individual projectors associated with the different eigenvalues. This can also be defined by an integral similar to (3.17) where Γ is a curve that encloses all the eigenvalues of the group and no other ones. Note that the rank of P thus defined is simply the sum of the algebraic multiplicities of the eigenvalue. In other words, the dimension of the range of such a P would be the sum of the algebraic multiplicities of the distinct eigenvalues enclosed by Γ.

3.1.5 Linear Perturbations of A

In this section we consider the family of matrices defined by

$$A(t) = A + tH$$

where t belongs to the complex plane. We are interested in the behavior of the eigenelements of $A(t)$ when t varies around the origin. Consider first the 'parameterized' resolvent,

$$R(t,z) = (A + tH - zI)^{-1}.$$

Noting that $R(t,z) = R(z)(I + tR(z)H)^{-1}$ it is clear that if the spectral radius of $tR(z)H$ is less than one then $R(t,z)$ will be analytic with respect to t. More precisely,

Proposition 3.2 *The resolvent $R(t,z)$ is analytic with respect to t in the open disk $|t| < \rho^{-1}(HR(z))$.*

We wish to show by integration over a Jordan curve Γ that a similar result holds for the spectral projector $P(t)$,i.e., that $P(t)$ is analytic for t small enough. The result would be true if the resolvent $R(t,z)$ were analytic with respect to t *for each z on Γ_i*. To ensure this we must require that

$$|t| < \inf_{z\in\Gamma} \rho^{-1}(R(z)H)) \; .$$

The question that arises next is whether or not the disk of all t 's defined above is empty. The answer is no as the following proof shows. We have

$$\rho(R(z)H) \leq \|R(z)H\| \leq \|R(z)\|\|H\|.$$

The function $\|R(z)\|$ is continuous with respect to z for $z \in \Gamma$ and therefore it reaches its maximum at some point z_0 of the closed curve Γ and we obtain

$$\rho(R(z)H) \leq \|R(z)H\| \leq \|R(z_0)\|\|H\| \equiv \kappa \; .$$

Hence,

$$\inf_{z\in\Gamma} \rho^{-1}(R(z)H)) \geq \kappa^{-1} \; .$$

Theorem 3.4 *Let Γ be a Jordan curve around one or a few eigenvalues of A and let*

$$\rho_a = \inf_{z\in\Gamma} [\rho(R(z)H)]^{-1} \; .$$

Then $\rho_a > 0$ and the spectral projector

$$P(t) = \frac{-1}{2\pi i} \int_\Gamma R(t,z)dz$$

is analytic in the disk $|t| < \rho_a$.

We have already proved that $\rho_a > 0$. The rest of the proof is straightforward. As an immediate corollary of Theorem 3.4, we know that the rank of $P(t)$ will stay constant as long as t stays in the disk $|t| < \rho_a$.

Corollary 3.1 *The number m of eigenvalues of $A(t)$, counted with their algebraic multiplicities, located inside the curve Γ, is constant provided that $|t| < \rho_a$.*

In fact the condition on t is only a sufficient condition and it may be too restrictive since the real condition required is that $P(t)$ be continuous with respect to t.

While individual eigenvalues may not have an analytic behavior, their average is usually analytic. Consider the average

$$\hat{\lambda}(t) = \frac{1}{m}\sum_{i=1}^{m} \lambda_i(t)$$

of the eigenvalues $\lambda_1(t), \lambda_2(t), \ldots, \lambda_m(t)$ of $A(t)$ that are inside Γ where we assume that the eigenvalues are counted with their multiplicities. Let $B(t)$ be a matrix representation of the restriction of $A(t)$ to the invariant subspace $M(t) = \mathrm{Ran}(P(t))$. Note that since $M(t)$ is invariant under $A(t)$ then $B(t)$ is the matrix representation of the rank m transformation

$$A(t)_{|M(t)} = A(t)P(t)_{|M(t)} = P(t)A(t)_{|M(t)} = P(t)A(t)P(t)_{|M(t)}$$

and we have

$$\begin{aligned} \hat{\lambda}(t) \equiv \frac{1}{m}\mathrm{tr}[B(t)] &= \frac{1}{m}\mathrm{tr}[A(t)P(t)_{|M(t)}] \\ &= \frac{1}{m}\mathrm{tr}[A(t)P(t)] \end{aligned} \tag{3.20}$$

The last equality in the above equation is due to the fact that for any x not in $M(t)$ we have $P(t)x = 0$ and therefore the extension of $A(t)P(t)$ to the whole space can only bring zero eigenvalues in addition to the eigenvalues $\lambda_i(t), i = 1, \ldots, m$.

Theorem 3.5 *The linear transformation $A(t)P(t)$ and its weighted trace $\hat{\lambda}(t)$ are analytic in the disk $|z| < \rho_a$.*

Proof. That $A(t)P(t)$ is analytic is a consequence of the previous theorem. That $\hat{\lambda}(t)$ is analytic, comes from the equivalent expression (3.20) and the fact that the trace of an operator $X(t)$ that is analytic with respect to t is analytic. □

Therefore, a simple eigenvalue $\lambda(t)$ of $A(t)$ not only stays simple around a neighborhood of $t = 0$ but it is also analytic with respect to t. Moreover, the vector $u_i(t) = P_i(t)u_i$ is an eigenvector of $A(t)$ associated with this simple eigenvalue, with $u_i = u_i(0)$ being an eigenvector of A associated with the eigenvalue λ_i. Clearly, the eigenvector $u_i(t)$ is analytic with respect to the variable t. However, the situation is more complex for the case of a multiple eigenvalue. If an eigenvalue is of multiplicity m then after a small perturbation, it will split into at most m distinct small branches $\lambda_i(t)$. These branches taken individually are not analytic in general. On the other hand, their *arithmetic average* is analytic. For this reason it is critical, in practice, to try to recognize groups of eigenvalues that are likely to originate from the splitting of a perturbed multiple eigenvalue.

Example 3.1. That an individual branch of the m branches of eigenvalues $\lambda_i(t)$ is not analytic can be easily illustrated by the example

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} , \; H = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} .$$

The matrix $A(t)$ has the eigenvalues $\pm\sqrt{t}$ which degenerate into the double eigenvalue 0 as $t \rightarrow 0$. The individual eigenvalues are not analytic but their average remains constant and equal to zero. □

In the above example each of the individual eigenvalues behaves like the square root of t around the origin. One may wonder whether this type of behavior can be generalized. The answer is stated in the next proposition.

Proposition 3.3 *Any eigenvalue $\lambda_i(t)$ of $A(t)$ inside the Jordan curve Γ satisfies*

$$|\lambda_i(t) - \lambda_i| = O\left(|t|^{1/l_i}\right)$$

where l_i is the index of λ_i.

Proof. Let $f(z) = (z - \lambda_i)^{l_i}$. We have seen earlier (proof of Lemma 3.1) that $f(A)P_i = 0$. For an eigenvector $u(t)$ of norm unity associated with the eigenvalue $\lambda_i(t)$ we have

$$\begin{aligned} f(A(t))P(t)u(t) &= f(A(t))u(t) = (A(t) - \lambda_i I)^{l_i} u(t) \\ &= (\lambda(t) - \lambda_i)^{l_i} u(t) \; . \end{aligned}$$

Taking the norms of both members of the above equation and using the fact that $f(A)P_i = 0$ we get

$$\begin{aligned} |\lambda_i(t) - \lambda_i|^{l_i} &= \|f(A(t))P(t)u(t)\| \\ &\leq \|f(A(t))P(t)\| = \|f(A(t))P(t) - f(A)P_i\| \; . \end{aligned}$$

Since $f(A) = f(A(0))$,$P_i = P(0)$ and $P(t), f(A(t))$ are analytic the right-hand-side in the above inequality is $O(t)$ and therefore

$$|\lambda_i(t) - \lambda_i|^{l_i} = O\left(|t|\right)$$

which shows the result. □

Example 3.2. A standard illustration of the above result is provided by taking A to be a Jordan block and H to be the rank one matrix $H = e_n e_1^T$:

$$A = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & 1 & \\ & & & 0 & 1 \\ & & & & 0 \end{pmatrix} \quad H = \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ 1 & & & & 0 \end{pmatrix} .$$

The matrix A has nonzero elements only in positions $(i, i+1)$ where they are equal to one. The matrix H has its elements equal to zero except for the element in position $(n, 1)$ which is equal to one. For $t = 0$ the matrix $A + tH$ admits only the eigenvalue $\lambda = 0$. The characteristic polynomial of $A + tH$ is equal to

$$p_t(z) = \det(A + tH - zI) = (-1)^n(z^n - t)$$

and its roots are $\lambda_j(t) = t^{1/n}e^{\frac{2ij\pi}{n}}$ $j = 1, \ldots, n$. Thus, if $n = 20$ then for a perturbation on A of the order of 10^{-16}, a reasonable number if double precision arithmetic is used, the eigenvalue will be perturbed by as much as $0.158..$ □

3.2 A-Posteriori Error Bounds

In this section we consider the problem of predicting the error made on an eigenvalue/eigenvector pair from some a posteriori knowledge on their approximations. The simplest criterion used to determine the accuracy of an approximate eigenpair $\tilde{\lambda}, \tilde{u}$, is to compute the norm of the so called residual vector

$$r = A\tilde{u} - \tilde{\lambda}\tilde{u}.$$

The aim is to derive error bounds that relate some norm of r,typically its 2-norm, to the errors on the eigenpair. Such error bounds are referred to a posteriori error bounds. Such bounds may help determine how accurate the approximations provided by some algorithm may be. This information can in turn be helpful in choosing a stopping criterion in iterative algorithms, in order to ensure that the answer delivered by the numerical method is within a desired tolerance.

3.2.1 General Error Bounds

In the non-Hermitian case there does not exist any 'a posteriori' error bounds in the strict sense of the definition. The error bounds that exist are in general weaker and not as easy to use as those known in the Hermitian case. The first error bound which we consider is known as the Bauer-Fike theorem. We recall that the condition number of a matrix X relative to the p-norm is defined by $\mathrm{Cond}_p(X) = \|X\|_p\|X^{-1}\|_p$.

Theorem 3.6 (Bauer-Fike) *Let $\tilde{\lambda}, \tilde{u}$ be an approximate eigenpair of A with residual vector $r = A\tilde{u} - \tilde{\lambda}\tilde{u}$, where $\tilde{u}$ is of 2-norm unity. Moreover, assume that the matrix A is diagonalizable and let X be the matrix that transforms it into diagonal form. Then, there exists an eigenvalue λ of A such that*

$$|\lambda - \tilde{\lambda}| \leq \mathrm{Cond}_2(X)\|r\|_2 \,.$$

Proof. If $\tilde{\lambda} \in \Lambda(A)$ the result is true. Assume that $\tilde{\lambda}$ is not an eigenvalue. From $A = XDX^{-1}$,where D is the diagonal of eigenvalues and since we assume that $\lambda \notin \Lambda(A)$, we can write

$$\tilde{u} = (A - \tilde{\lambda}I)^{-1}r = X(D - \tilde{\lambda}I)^{-1}X^{-1}r$$

and hence

$$\begin{aligned} 1 &= \|X(D-\tilde{\lambda}I)^{-1}X^{-1}r\|_2 \\ &\leq \|X\|_2\|X^{-1}\|_2\|(D-\tilde{\lambda}I)^{-1}\|_2\ \|r\|_2\ . \end{aligned} \tag{3.21}$$

The matrix $(D-\tilde{\lambda}I)^{-1}$ is a diagonal matrix and as a result its 2-norm is the maximum of the absolute values of its diagonal entries. Therefore,

$$1 \leq \mathrm{Cond}_2(X)\|r\|_2 \max_{\lambda_i \in \Lambda(A)} |\lambda_i - \tilde{\lambda}|^{-1}$$

from which the result follows. □

In case the matrix is not diagonalizable then the previous result can be generalized as follows.

Theorem 3.7 *Let $\tilde{\lambda}, \tilde{u}$ an approximate eigenpair with residual vector $r = A\tilde{u} - \tilde{\lambda}\tilde{u}$, where $\tilde{u}$ has unit 2-norm. Let X be the matrix that transforms A into its Jordan canonical form, $A = XJX^{-1}$. Then, there exists an eigenvalue λ of A such that*

$$\frac{|\lambda - \tilde{\lambda}|^l}{1 + |\lambda - \tilde{\lambda}| + \cdots + |\lambda - \tilde{\lambda}|^{l-1}} \ \leq\ \mathrm{Cond}_2(X)\|r\|_2$$

where l is the index of λ.

Proof. The proof starts as in the previous case but here the diagonal matrix D is replaced by the Jordan matrix J. Because the matrix $(J-\tilde{\lambda}I)$ is block diagonal its 2-norm is the maximum of the 2-norms of each block (a consequence of the alternative formulation for 2-norms seen in Chapter 1). For each of these blocks we have

$$(J_i - \tilde{\lambda}I)^{-1} = ((\lambda_i - \tilde{\lambda})I + E)^{-1}$$

where E is the nilpotent matrix having ones in positions $(i, i+1)$ and zeros elsewhere. Therefore,

$$(J_i - \tilde{\lambda}I)^{-1} = \sum_{j=1}^{l_i} (\lambda_i - \tilde{\lambda})^{-j}(-E)^{j-1}$$

and as a result, setting $\delta_i = |\lambda_i - \tilde{\lambda}|$ and noting that $\|E\|_2 = 1$,we get

$$\|(J_i - \tilde{\lambda}I)^{-1}\|_2 \ \leq\ \sum_{j=1}^{l_i} |\lambda_i - \tilde{\lambda}|^{-j}\|E\|_2^{j-1} \ =\ \sum_{j=1}^{l_i} \delta_i^{-j} \ =\ \delta_i^{-l_i} \sum_{j=0}^{l_i-1} \delta_i^j.$$

The analogue of (3.21) is

$$1 \leq \mathrm{Cond}_2(X)\|(J-\tilde{\lambda}I)^{-1}\|_2\|r\|_2. \tag{3.22}$$

Since,

$$\|(J - \tilde{\lambda}I)^{-1}\|_2 \;=\; \max_{i=1,\ldots,p} \|(J_i - \tilde{\lambda}I)^{-1}\|_2 \;\leq\; \max_{i=1,\ldots,p} \delta_i^{-l} \sum_{j=0}^{l_i-1} \delta_i^j$$

we get

$$\min_{i=1,\ldots,p} \left\{ \frac{\delta_i^{l_i}}{\sum_{j=0}^{l_i-1} \delta_i^j} \right\} \leq\; \mathrm{Cond}_2(X)\|r\|_2$$

which is essentially the desired result. □

Corollary 3.2 (Kahan, Parlett, and Jiang, 1980). *Under the same assumptions as those of theorem 3.7, there exists an eigenvalue λ of A such that*

$$\frac{|\lambda - \tilde{\lambda}|^l}{(1 + |\lambda - \tilde{\lambda}|)^{l-1}} \;\leq\; \mathrm{Cond}_2(X)\|r\|_2$$

where l is the index of λ.

Proof. Follows immediately from the previous theorem and the inequality,

$$\sum_{j=0}^{l-1} \delta_i^j \;\leq\; (1 + \delta_i)^{l-1}.$$ □

For an alternative proof see [101]. Unfortunately, the bounds of the type shown in the previous two theorems are not practical because of the presence of the condition number of X. The second result even requires the knowledge of the index of λ_i,which is not numerically viable. The situation is much improved in the particular case where A is Hermitian because in this case $\mathrm{Cond}_2(X) = 1$. This is taken up next.

3.2.2 The Hermitian Case

In the Hermitian case, Theorem 3.6 leads to the following corollary.

Corollary 3.3 *Let $\tilde{\lambda}, \tilde{u}$ be an approximate eigenpair of a Hermitian matrix A,with $\|u\|_2 = 1$ and let r be the corresponding residual vector. Then there exists an eigenvalue of A such that*

$$|\lambda - \tilde{\lambda}| \;\leq\; \|r\|_2 \,. \tag{3.23}$$

This is a remarkable result because it constitutes a simple yet general error bound. On the other hand it is not sharp as the next a posteriori error bound, due to Kato and Temple [104, 210], shows. We start by proving a lemma that will be used to prove Kato-Temple's theorem. In the next results it is assumed that the approximate eigenvalue $\tilde{\lambda}$ is the Rayleigh quotient of the approximate eigenvector.

Lemma 3.2 *Let $\tilde{u}$ be an approximate eigenvector of norm unity of A,and $\tilde{\lambda} = (A\tilde{u}, \tilde{u})$. Let (α, β) be an interval that contains $\tilde{\lambda}$ and no eigenvalue of A. Then*

$$(\beta - \tilde{\lambda})(\tilde{\lambda} - \alpha) \;\leq\; \|r\|_2^2.$$

Proof. This lemma uses the observation that the residual vector r is orthogonal to $\tilde{u}$. Then we have

$$\begin{aligned}
&((A - \alpha I)\tilde{u}, (A - \beta I)\tilde{u}) \\
&\qquad = ((A - \tilde{\lambda} I)\tilde{u} + (\tilde{\lambda} - \alpha I)\tilde{u}, ((A - \tilde{\lambda} I)\tilde{u} + (\tilde{\lambda} - \beta I)\tilde{u}) \\
&\qquad = \|r\|_2^2 + (\tilde{\lambda} - \alpha I)(\tilde{\lambda} - \beta I),
\end{aligned}$$

because of the orthogonality property mentioned above. On the other hand, one can expand $\tilde{u}$ in the orthogonal eigenbasis of A as

$$\tilde{u} = \xi_1 u_1 + \xi_2 u_2 + \cdots + \xi_n u_n$$

to transform the left hand side of the expression into

$$((A - \alpha I)\tilde{u}, (A - \beta I)\tilde{u}) = \sum_{i=1}^{n} |\xi_i|^2 \; (\lambda_i - \alpha)(\lambda_i - \beta) \; .$$

Each term in the above sum is nonnegative because of the assumptions on α and β. Therefore $\|r\|_2^2 + (\beta - \tilde{\lambda})(\tilde{\lambda} - \alpha) \geq 0$ which is the desired result. □

Theorem 3.8 (Kato and Temple [104, 210]) *Let $\tilde{u}$ be an approximate eigenvector of norm unity of A,and $\tilde{\lambda} = (A\tilde{u}, \tilde{u})$. Assume that we know an interval (a, b) that contains $\tilde{\lambda}$ and one and only one eigenvalue λ of A. Then*

$$-\frac{\|r\|_2^2}{\tilde{\lambda} - a} \;\leq\; \tilde{\lambda} - \lambda \;\leq\; \frac{\|r\|_2^2}{b - \tilde{\lambda}} \; .$$

Proof. Let λ be the closest eigenvalue to $\tilde{\lambda}$. In the case where λ is located at left of $\tilde{\lambda}$ then take $\alpha = \lambda$ and $\beta = b$ in the lemma to get

$$0 \leq \tilde{\lambda} - \lambda \;\leq\; \frac{\|r\|_2^2}{b - \tilde{\lambda}} \; .$$

In the opposite case where $\lambda > \tilde{\lambda}$,use $\alpha = a$ and $\beta = \lambda$ to get

$$0 \leq \lambda - \tilde{\lambda} \leq \frac{\|r\|_2^2}{\tilde{\lambda} - a} \; .$$

This completes the proof. □

A simplification of Kato-Temple's theorem consists of using a particular interval that is symmetric about the approximation $\tilde{\lambda}$, as is stated in the next corollary.

Corollary 3.4 *Let $\tilde{u}$ be an approximate eigenvector of norm unity of A,and $\tilde{\lambda} = (A\tilde{u}, \tilde{u})$. Let λ be the eigenvalue closest to $\tilde{\lambda}$ and δ the distance from $\tilde{\lambda}$ to the rest of the spectrum, i.e.,*

$$\delta = \min_i \{|\lambda_i - \tilde{\lambda}|,\ \lambda_i \neq \lambda\}.$$

Then,

$$|\tilde{\lambda} - \lambda| \leq \frac{\|r\|_2^2}{\delta}\ . \tag{3.24}$$

Proof. This is a particular case of the previous theorem with $a = \tilde{\lambda} - \delta$ and $b = \tilde{\lambda} + \delta$. □

It is also possible to show a similar result for the angle between the exact and approximate eigenvectors.

Theorem 3.9 *Let $\tilde{u}$ be an approximate eigenvector of norm unity of A,$\tilde{\lambda} = (A\tilde{u}, \tilde{u})$ and $r = (A - \tilde{\lambda}I)\tilde{u}$. Let λ be the eigenvalue closest to $\tilde{\lambda}$ and δ the distance from $\tilde{\lambda}$ to the rest of the spectrum, i.e., $\delta = \min_i\{|\lambda_i - \tilde{\lambda}|, \lambda_i \neq \lambda\}$. Then, if u is an eigenvector of A associated with λ we have*

$$\sin\theta(\tilde{u}, u)\ \leq\ \frac{\|r\|_2}{\delta}\ . \tag{3.25}$$

Proof. Let us write the approximate eigenvector $\tilde{u}$ as $\tilde{u} = u\cos\theta + z\sin\theta$ where z is a vector orthogonal to u. We have

$$\begin{aligned}(A - \tilde{\lambda}I)\tilde{u} &= \cos\theta\ (A - \tilde{\lambda}I)u + \sin\theta\ (A - \tilde{\lambda}I)z \\ &= \cos\theta\ (\lambda - \tilde{\lambda}I)u + \sin\theta\ (A - \tilde{\lambda}I)z\ .\end{aligned}$$

The two vectors on the right hand side are orthogonal to each other because,

$$(u, (A - \tilde{\lambda}I)z) = ((A - \tilde{\lambda}I)u, z) = (\lambda - \tilde{\lambda})(u, z) = 0\ .$$

Therefore,

$$\|r\|_2^2 = \|(A - \tilde{\lambda}I)\tilde{u}\|^2 = \sin^2\theta\ \|(A - \tilde{\lambda}I)z\|_2^2 + \cos^2\theta\ |\lambda - \tilde{\lambda}|^2\ .$$

Hence,

$$\sin^2\theta\ \|(A - \tilde{\lambda}I)z\|_2^2 \leq \|r\|_2^2\ .$$

The proof follows by observing that since z is orthogonal to u then $\|(A-\tilde{\lambda}I)z\|_2$ is larger than the smallest eigenvalue of $A-\tilde{\lambda}I$ restricted to the subspace orthogonal to u,which is precisely δ. □

Although the above bounds for the Hermitian case are sharp they are still not computable since δ involves a distance from the 'next closest' eigenvalue of A to $\tilde{\lambda}$ which is not readily available. In order to be able to use these bounds in practical situations one must provide a lower bound for the distance δ. One might

simply approximate δ by $\tilde{\lambda} - \tilde{\lambda}_j$ where $\tilde{\lambda}_j$ is some approximation to the next closest eigenvalue to $\tilde{\lambda}$. The result would no longer be an actual upper bound on the error but rather an 'estimate' of the error. This may not be safe however. To ensure that the computed error bound used is rigorous it is preferable to exploit the simpler inequality provided by Corollary 3.3 in order to find a lower bound for the distance δ,for example

$$\begin{aligned} \delta = |\tilde{\lambda} - \lambda_j| &\geq |(\tilde{\lambda} - \tilde{\lambda}_j) + (\lambda_j - \tilde{\lambda}_j)| \\ &\geq |\tilde{\lambda} - \tilde{\lambda}_j| - |\lambda_j - \tilde{\lambda}_j| \\ &\geq |\tilde{\lambda} - \tilde{\lambda}_j| - \|r_j\|_2. \end{aligned}$$

where $\|r_j\|_2$ is the residual norm associated with the eigenvalue λ_j. Now the above lower bound of δ is computable. In order for the resulting error bound to have a meaning, $\|r_j\|_2$ must be small enough to ensure that there are no other potential eigenvalues λ_k that might be closer to λ than is λ_j. The above error bounds when used cautiously can be quite useful.

Example 3.3. Let

$$A = \begin{pmatrix} 1.0 & 2.0 & & & \\ 2.0 & 1.0 & 2.0 & & \\ & 2.0 & 1.0 & 2.0 & \\ & & 2.0 & 1.0 & 2.0 \\ & & & 2.0 & 1.0 \end{pmatrix}.$$

The eigenvalues of A are $\{3, -1, 1, 1 - 2\sqrt{3}, 1 + 2\sqrt{3}\}$

An eigenvector associated with the eigenvalue $\lambda = 3$ is

$$u = \begin{pmatrix} -0.5 \\ -0.5 \\ 0.0 \\ 0.5 \\ 0.5 \end{pmatrix}.$$

Consider the vector

$$\tilde{u} = \begin{pmatrix} -0.49 \\ -0.5 \\ 0.0 \\ 0.5 \\ 0.5 \end{pmatrix}.$$

The Rayleigh quotient of $\tilde{u}$ with respect to A is $\tilde{\lambda} = 2.9998...$ The closest eigenvalue is $\lambda = 3.0$ and the corresponding actual error is 2.02×10^{-4}. The residual norm is found to be

$$\|(A - \tilde{\lambda} I)\tilde{u}\|_2 \approx 0.0284 .$$

The distance δ here is

$$\delta = |2.9998 - 4.464101...| \approx 1.46643 .$$

So the error bound for the eigenvalue 2.9998 found is

$$\frac{(0.0284..)^2}{1.4643} \approx 5.5177 \times 10^{-4}.$$

For the eigenvector, the angle between the exact and approximate eigenvector is such that $\cos\theta = 0.999962$,giving an angle $\theta \approx 0.0087$ and the sine of the angle is approximately $\sin\theta \approx 0.0087$. The error as estimated by (3.9) is

$$\sin\theta \leq \frac{0.0284}{1.4643} \approx 0.01939$$

which is about twice as large as the actual error. □

We now consider a slightly more realistic situation. There are instances in which the off-diagonal elements of a matrix are small. Then the diagonal elements can be considered approximations to the eigenvalues of A and the question is how good an accuracy can one expect? We illustrate this with an example.

Example 3.4. Let

$$A = \begin{pmatrix} 1.00 & 0.0055 & 0.10 & 0.10 & 0.00 \\ 0.0055 & 2.00 & -0.05 & 0.00 & -0.10 \\ 0.10 & -0.05 & 3.00 & 0.10 & 0.05 \\ 0.10 & 0.00 & 0.10 & 4.00 & 0.00 \\ 0.00 & -0.10 & 0.05 & 0.00 & 5.00 \end{pmatrix}.$$

The eigenvalues of A rounded to 6 digits are

$$\Lambda(A) = \{0.99195, 1.99443, 2.99507, 4.01386, 5.00466\}\ .$$

A natural question is how accurate is each of the diagonal elements of A as an approximate eigenvalue? We assume that we know nothing about the exact spectrum. We can take as approximate eigenvectors the e_i's, $i = 1, \cdots, 5$ and the corresponding residual norms are

$$0.141528\ ; 0.1119386\ ; 0.1581139\ ; 0.1414214\ ; 0.1118034$$

respectively. The simplest residual bound (3.23) tells us that

$$\begin{array}{ll} |\lambda - 1.0| \leq 0.141528; & |\lambda - 2.0| \leq 0.111939; \\ |\lambda - 3.0| \leq 0.158114; & |\lambda - 4.0| \leq 0.141421;\ . \\ |\lambda - 5.0| \leq 0.111803. & \end{array}$$

The intervals defined above are all disjoint. As a result, we can get a reasonable idea of δ_i the distance of each of the approximations from the eigenvalues not in the interval. For example,

$$\delta_1 \equiv |a_{11} - \lambda_2| \geq |1 - (2.0 - 0.1119386)| \approx 0.88806$$

and

$$\begin{aligned}\delta_2 &= \min\{|a_{22}-\lambda_3|, |a_{22}-\lambda_1|\} \\ &\geq \min\{|2.0-(3.0-0.15811)|, |2.0-(1.0+0.14153)|\} \\ &= 0.8419...\end{aligned}$$

We find similarly $\delta_3 \geq 0.8585$,$\delta_4 \geq 0.8419$, and $\delta_5 \geq 0.8586$.

We now get from the bounds (3.24) the following inequalities,

$$\begin{array}{ll} |\lambda - 1.0| \leq 0.0226; & |\lambda - 2.0| \leq 0.0149; \\ |\lambda - 3.0| \leq 0.0291; & |\lambda - 4.0| \leq 0.0238; \\ |\lambda - 5.0| \leq 0.0146. & \end{array}$$

whereas the actual errors are

$$\begin{array}{lll} |\lambda - 1.0| \approx 0.0080; & |\lambda - 2.0| \approx 0.0056; & |\lambda - 3.0| \approx 0.0049; \\ |\lambda - 4.0| \approx 0.0139; & |\lambda - 5.0| \approx 0.0047. & \end{array}$$

□

3.2.3 The Kahan-Parlett-Jiang Theorem

We now return to the general non-Hermitian case. The results seen for the Hermitian case in the previous section can be very useful in practical situations. For example they can help develop efficient stopping criteria in iterative algorithms. In contrast, those seen in Section 3.2.1 for the general non-Hermitian case are not too easy to exploit in practice. The question that one might ask is whether or not any residual bounds can be established that will provide information similar to that provided in the Hermitian case. There does not seem to exist any such result in the literature. A result established by Kahan, Parlett and Jiang [101], which we now discuss, seems to be the best compromise between generality and sharpness. However, the theorem is of a different type. It does not guarantee the existence of, say, an eigenvalue in a given interval whose size depends on the residual norm. It only gives us the size of the smallest perturbation that must be applied to the original data (the matrix), in order to transform the approximate eigenpair into an exact one (for the perturbed problem).

To explain the nature of the theorem we begin with a very simple result which can be regarded as a one-sided version of the one proved by Kahan, Parlett, and Jiang in that it only considers the right eigenvalue – eigenvector pair instead of the eigen-triplet consisting of the eigenvalue and the right and left eigenvectors.

Proposition 3.4 *Let a square matrix A and a unit vector u be given. For any scalar γ define the residual vector,*

$$r = Au - \gamma u,$$

and let $\mathcal{E} = \{E : (A-E)u = \gamma u\}$. Then

$$\min_{E \in \mathcal{E}} \|E\|_2 = \|r\|_2 \,. \tag{3.26}$$

Proof. From the assumptions we see that each E is in $\mathcal{E}$ if and only if it satisfies the equality

$$Eu = r\ . \tag{3.27}$$

Since $\|u\|_2 = 1$ the above equation implies that for any such E

$$\|E\|_2 \geq \|r\|_2,$$

which in turn implies that

$$\min_{E\in\mathcal{E}} \|E\|_2 \geq \|r\|_2. \tag{3.28}$$

Now consider the matrix $E_0 = ru^H$ which is a member of $\mathcal{E}$ since it satisfies (3.27). The 2-norm of E_0 is such that

$$\|E_0\|_2^2 = \sigma_{max}\{ru^H ur^H\} = \sigma_{max}\{rr^H\} = \|r\|_2^2.$$

As a result the minimum in the left hand side of (3.28) is reached for $E = E_0$ and the value of the minimum is equal to $\|r\|_2$. □

We now state a simple version of the Kahan-Parlett-Jiang theorem [101].

Theorem 3.10 (Kahan, Parlett, and Jiang) *Let a square matrix A and two unit vectors u, w with $(u, w) \neq 0$ be given. For any scalar γ define the residual vectors,*

$$r = Au - \gamma u \qquad s = A^H w - \bar{\gamma} w$$

and let $\mathcal{E} = \{E : (A - E)u = \gamma u; (A - E)^H w = \bar{\gamma} w\}$. Then

$$\min_{E\in\mathcal{E}} \|E\|_2 = \max\left\{\|r\|_2, \|s\|_2\right\}. \tag{3.29}$$

Proof. We proceed in the same way as for the proof of the simpler result of the previous proposition. The two conditions that a matrix E must satisfy in order to belong to $\mathcal{E}$ translate into

$$Eu = r \quad \text{and} \quad E^H w = s. \tag{3.30}$$

By the same argument used in the proof of Proposition 3.4 any such E satisfies

$$\|E\|_2 \geq \|r\|_2 \quad \text{and} \quad \|E\|_2 \geq \|s\|_2. \tag{3.31}$$

which proves the inequality

$$\min_{E\in\mathcal{E}} \|E\|_2 \geq \max\{\|r\|_2, \|s\|_2\}. \tag{3.32}$$

We now define,

$$\begin{aligned} \delta &= s^H u = w^H r \\ x &= r - \delta\, w \\ y &= s - \bar{\delta}\, u \end{aligned} \tag{3.33}$$

and consider the particular set of matrices of the form

$$E(\beta) = ru^H + ws^H - \delta\, wu^H - \beta\, xy^H \tag{3.34}$$

where β is a parameter. It is easy to verify that these matrices satisfy the constraints (3.30) for any β.

We distinguish two different cases depending on whether $\|s\|_2$ is larger or smaller than $\|r\|_2$. When $\|s\|_2 > \|r\|_2$ we rewrite $E(\beta)$ in the form

$$E(\beta) = x(u - \beta\, y)^H + ws^H \tag{3.35}$$

and select β in such a way that

$$s^H(u - \beta\, y) = 0 \tag{3.36}$$

which leads to

$$\beta = \frac{\delta}{\|s\|_2^2 - |\delta|^2}.$$

We note that the above expression is not valid when $\|s\|_2 = |\delta|$,which occurs only when $y = 0$. In this situation $E(\beta) = ru^H$ for any β, and the following special treatment is necessary. As in the proof of the previous proposition $E(\beta) = \|r\|_2$. On the other hand we have

$$\|s\|_2 = |\delta| = |w^H r| \leq \|r\|_2$$

which shows that $\max\{\|r\|_2, \|s\|_2\} = \|r\|_2$ and establishes the result that the minimum in the theorem is reached for $E(\beta)$ in this very special case.

Going back to the general case where $\|s\|_2 \neq |\delta|$, with the above choice of β the two vectors x and w in the range of $E(\beta)$ as defined by (3.35) are orthogonal and similarly, the vectors $u - \beta y$ and s are also orthogonal. In this situation the norm of $E(\beta)$ is equal to [See problem P-2.14]:

$$\|E(\beta)\|_2 = \max\{\|s\|_2, \|x\|_2 \|\|u^H - \beta\, y\|_2\}.$$

Because of the orthogonality of x and w,we have

$$\|x\|_2^2 = \|r\|_2^2 - |\delta|^2\ .$$

Similarly, exploiting the orthogonality of the pair u, y,and using the definition of β we get

$$\begin{aligned} \|u - \beta\, y\|_2^2 &= 1 + \beta^2 \|y\|_2^2 \\ &= 1 + \beta^2 [\|s\|_2^2 - |\delta|^2] \\ &= \frac{\|s\|_2^2}{\|s\|_2^2 - |\delta|^2}\ . \end{aligned}$$

The above results yield

$$\|E(\beta)\|_2^2 = \max\left\{\|s\|_2^2, \|s\|_2^2 \frac{\|r\|_2^2 - |\delta|^2}{\|s\|_2^2 - |\delta|^2}\right\} = \|s\|_2^2.$$

This shows from (3.32) that the equality (3.29) is satisfied for the case when $\|s\|_2 > \|r\|_2$.

To prove the result for the case $\|s\|_2 < \|r\|_2$,we proceed in the same manner, writing this time $E(\beta)$ as

$$E(\beta) = ru^H + (\alpha w - \beta\, x)y^H$$

and choosing β such that $u^H(w - \beta\, x) = 0$. A special treatment will also be necessary for the case where $\|r\|_2 = |\delta|$ which only occurs when $x = 0$. □

The actual result proved by Kahan, Parlett and Jiang is essentially a block version of the above theorem and includes results with other norms, such as the Frobenius norm.

Example 3.5. Consider the matrix,

$$A = \begin{pmatrix} 1.0 & 2.1 & & & \\ 1.9 & 1.0 & 2.1 & & \\ & 1.9 & 1.0 & 2.1 & \\ & & 1.9 & 1.0 & 2.1 \\ & & & 1.9 & 1.0 \end{pmatrix}.$$

which is obtained by perturbing the symmetric tridiagonal matrix of Example 3.3. Consider the pair

$$\gamma = 3.0, \qquad v = \begin{pmatrix} -0.5 \\ -0.5 \\ 0.0 \\ 0.5 \\ 0.5 \end{pmatrix}.$$

Then we have

$$\|r\|_2 = \|(A - \gamma I)u\|_2 \approx 0.1414,$$

which tells us, using the one-sided result (Proposition 3.4), that we need to perturb A by a matrix E of norm 0.1414 to make the pair γ, v an exact eigenpair of A.

Consider now v as defined above and

$$w = \alpha\ (0.6, 0.6, 0.0, 0.4, 0.4)^T\ ,$$

where α is chosen to normalize w to so that its 2-norm is unity. Then, still with $\gamma = 3$,we find

$$\|r\|_2 \approx 0.1414\ , \quad \|s\|_2 \approx 0.5004\ .$$

As a result of the theorem, we now need a perturbation E whose 2-norm is roughly 0.5004 to make the triplet γ, v, w an exact eigentriplet of A,a much stricter requirement than with the one-sided result. □

The outcome of the above example was to be expected. If one of the left of right approximate eigen-pair, for example the left pair (γ, v), is a poor approximation, then it will take a larger perturbation on A to make the triplet γ, v, w exact,

than it would to make the pair γ, u exact. Whether one needs to use the one-sided or the two-sided result depends on whether one is interested in the left and right eigenvectors simultaneously or in the right (or left) eigenvector only.

3.3 Conditioning of Eigen-problems

When solving a linear system $Ax = b$,an important question that arises is how sensitive is the solution x to small variations of the initial data, namely to the matrix A and the right-hand side b. A measure of this sensitivity is called the condition number of A defined by

$$\mathrm{Cond}(A) = \|A\|\|A^{-1}\|$$

relative to some norm.

For the eigenvalue problem we raise a similar question but we must now define similar measures for the eigenvalues as well as for the eigenvectors and the invariant subspaces.

3.3.1 Conditioning of Eigenvalues

Let us assume that λ is a simple eigenvalue and consider the family of matrices $A(t) = A + tE$. We know from the previous sections that there exists a branch of eigenvalues $\lambda(t)$ of $A(t)$ that is analytic with respect to t, when t belongs to a small enough disk centered at the origin. It is natural to call conditioning of the eigenvalue λ of A relative to the perturbation E the modulus of the derivative of $\lambda(t)$ at the origin $t = 0$. Let us write

$$A(t)u(t) = \lambda(t)u(t) \tag{3.37}$$

and take the inner product of both members with a left eigenvector w of A associated with λ to get

$$((A + tE)u(t), w) = \lambda(t)(u(t), w)$$

or,

$$\begin{aligned} \lambda(t)(u(t), w) &= (Au(t), w) + t(Eu(t), w) \\ &= (u(t), A^H w) + t(Eu(t), w) \\ &= \lambda(u(t), w) + t(Eu(t), w). \end{aligned}$$

Hence,

$$\frac{\lambda(t) - \lambda}{t}(u(t), w) = (Eu(t), w)$$

and therefore by taking the limit at $t = 0$,

$$\lambda'(0) = \frac{(Eu, w)}{(u, w)}$$

Here we should recall that the left and right eigenvectors associated with a simple eigenvalue cannot be orthogonal to each other. The actual conditioning of an eigenvalue, given a perturbation "in the direction of E"is the modulus of the above quantity. In practical situations, one often does not know the actual perturbation E but only its magnitude, e.g., as measured by some matrix norm $\|E\|$. Using the Cauchy-Schwarz inequality and the 2-norm, we can derive the following upper bound,

$$|\lambda'(0)| \leq \frac{\|Eu\|_2\|w\|_2}{|(u,w)|} \leq \|E\|_2 \frac{\|u\|_2\|w\|_2}{|(u,w)|}$$

In other words the actual condition number of the eigenvalue λ is bounded from above by the norm of E divided by the cosine of the acute angle between the left and the right eigenvectors associated with λ. Hence the following definition.

Definition 3.1 *The condition number of a simple eigenvalue λ of an arbitrary matrix A is defined by*

$$\mathrm{Cond}(\lambda) = \frac{1}{\cos\theta(u,w)}$$

in which u and w are the right and left eigenvectors, respectively, associated with λ.

Example 3.6. Consider the matrix

$$A = \begin{pmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{pmatrix}.$$

The eigenvalues of A are $\{1, 2, 3\}$. The right and left eigenvectors of A associated with the eigenvalue $\lambda_1 = 1$ are approximately

$$u = \begin{pmatrix} 0.3162 \\ -0.9487 \\ 0.0 \end{pmatrix} \quad \text{and} \quad w = \begin{pmatrix} 0.6810 \\ 0.2253 \\ 0.6967 \end{pmatrix} \tag{3.38}$$

and the corresponding condition number is approximately

$$\mathrm{Cond}(\lambda_1) \approx 603.64$$

A perturbation of order 0.01 may cause perturbations of magnitude up to 6. Perturbing a_{11} to -149.01 yields the spectrum:

$$\{0.2287, 3.2878, 2.4735\}. \qquad \square$$

For Hermitian, or more generally normal, matrices every simple eigenvalue is well-conditioned, since $\mathrm{Cond}(\lambda) = 1$. On the other hand the condition number of a non-normal matrix can be excessively high, in fact arbitrarily high.

Example 3.7. As an example simply consider the matrix

$$\begin{pmatrix} \lambda_1 & -1 & & & \\ & \lambda_2 & -1 & & \\ & & . & . & \\ & & & . & -1 \\ & & & & \lambda_n \end{pmatrix} \tag{3.39}$$

with $\lambda_1 = 0$ and $\lambda_i = 1/(i-1)$ for $i > 1$. A right eigenvector associated with the eigenvalue λ_1 is the vector e_1. A left eigenvector is the vector w whose i-th component is equal to $(i-1)!$ for $i = 1, \ldots, n$. A little calculation shows that the condition number of λ_1 satisfies

$$(n-1)! \leq \mathrm{Cond}(\lambda_1) \leq (n-1)! \ \sqrt{n}.$$

Thus, this example shows that the condition number can be quite large even for modestly sized matrices. □

An important comment should be made concerning the above example. The eigenvalues of A are explicitly known in terms of the diagonal entries of the matrix, whenever the structure of A stays the same. One may wonder whether it is sensible to discuss the concept of condition number in such cases. For example, if we perturb the (1,1) entry by 0.1 we know exactly that the eigenvalue λ_1 will be perturbed likewise. Is the notion of condition number useless in such situations? The answer is no. First, the argument is only true if perturbations are applied in specific positions of the matrix, namely its upper triangular part. If perturbations take place elsewhere then some or all of the eigenvalues of the perturbed matrix may not be explicitly known. Second, one can think of applying an orthogonal similarity transformation to A. If Q is orthogonal then the eigenvalues of the matrix $B = Q^H A Q$ have the same condition number as those of the original matrix A,(see Problem P-3.15). The resulting matrix B may be dense and the dependence of its eigenvalues with respect to its entries is no longer explicit.

3.3.2 Conditioning of Eigenvectors

To properly define the condition number of an eigenvector we need to use the notion of *reduced resolvent*. Although the resolvent operator $R(z)$ has a singularity at an eigenvalue λ it can still be defined on the restriction to the invariant subspace $\mathrm{Null}(P)$. More precisely, consider the restriction of the mapping $A - \lambda I$ to the subspace $(I-P)\mathbb{C}^n = \mathrm{Null}(P)$,where P is the spectral projector associated with the eigenvalue λ. This mapping is invertible because if x is an element of $\mathrm{Null}(P)$ then $(A - \lambda I)x = 0$,i.e., x is in $\mathrm{Null}(A - \lambda I)$ which is included in $\mathrm{Ran}(P)$ and this is only possible when $x = 0$. We will call reduced resolvent at λ the inverse of this linear mapping and we will denote it by $S(\lambda)$. Thus,

$$S(\lambda) = \left[(A - \lambda I)_{|\mathrm{Null}(P)} \right]^{-1}.$$

The reduced resolvent satisfies the relation,

$$S(\lambda)(A-\lambda I)x = S(\lambda)(A-\lambda I)(I-P)x = (I-P)x \quad \forall\ x \tag{3.40}$$

which can be viewed as an alternative definition of $S(\lambda)$.

We now consider a simple eigenvalue λ of a matrix A with an associated eigenvector u, and write that a pair $\lambda(t), u(t)$ is an eigenpair of the matrix $A+tE$,

$$(A+tE)u(t) = \lambda(t)u(t)\ . \tag{3.41}$$

Subtracting $Au = \lambda u$ from both sides we have,

$$A(u(t)-u) + tEu(t) = \lambda(t)u(t) - \lambda u = \lambda(u(t)-u) + (\lambda(t)-\lambda)u(t)$$

or,

$$(A-\lambda I)(u(t)-u) + tEu(t) = (\lambda(t)-\lambda)u(t)\ .$$

We then multiply both sides by the projector $I-P$ to obtain

$$\begin{aligned}(I-P)(A-\lambda I)(u(t)-u) \quad &+ \quad t(I-P)Eu(t) \\ &= \quad (\lambda(t)-\lambda)(I-P)u(t) \\ &= \quad (\lambda(t)-\lambda)(I-P)(u(t)-u)\end{aligned}$$

The last equality holds because $(I-P)u = 0$ since u is in $\mathrm{Ran}(P)$. Hence,

$$\begin{aligned}&(A-\lambda I)(I-P)(u(t)-u) = \\ &\quad (I-P)\left[-tEu(t) + (\lambda(t)-\lambda)(u(t)-u)\right].\end{aligned}$$

We now multiply both sides by $S(\lambda)$ and use (3.40) to get

$$\begin{aligned}&(I-P)(u(t)-u) = \\ &\quad S(\lambda)(I-P)\left[-tEu(t) + (\lambda(t)-\lambda)(u(t)-u)\right]\end{aligned} \tag{3.42}$$

In the above development we have not scaled $u(t)$ in any way. We now do so by requiring that its projection onto the eigenvector u be exactly u,i.e., $Pu(t) = u$ for all t. With this scaling, we have

$$(I-P)(u(t)-u) = u(t)-u.$$

As a result, equality (3.42) becomes

$$u(t)-u = S(\lambda)\left[-t(I-P)Eu(t) + (\lambda(t)-\lambda)(u(t)-u),\right]$$

from which we finally get, after dividing by t and taking the limit,

$$u'(0) = -S(\lambda)(I-P)Eu\ . \tag{3.43}$$

Using the same argument as before, we arrive at the following general definition of the condition number of an eigenvector.

Definition 3.2 *The condition number of an eigenvector u associated with an eigenvalue λ of an arbitrary matrix A is defined by*

$$\mathrm{Cond}(u) = \|S(\lambda)(I - P)\|_2. \tag{3.44}$$

in which $S(\lambda)$ is the reduced resolvent of A at λ.

In the case where the matrix A is Hermitian it is easy to verify that the condition number simplifies to the following

$$\mathrm{Cond}(u) = \frac{1}{dist[\lambda, \Lambda(A) - \{\lambda\}]}\ . \tag{3.45}$$

In the general non-Hermitian case, it is difficult to assess the size of $\mathrm{Cond}(u)$.

To better understand the nature of the operator $S(\lambda)(I-P)$, consider its spectral expansion in the particular case where A is diagonalizable and the eigenvalue λ_i of interest is simple.

$$S(\lambda_i)(I - P_i) = \sum_{\substack{j=1 \\ j\neq i}}^{p} \frac{1}{\lambda_j - \lambda_i} P_j$$

Since we can write each projector as a sum of outer product matrices $P_j = \sum_{k=1}^{\mu_i} u_k w_k^H$ where the left and right eigenvectors u_k and w_k are normalized such that $(u_j, w_j) = 1$,the expression (2.9) can be rewritten as

$$u'(0) = \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{1}{\lambda_j - \lambda_i} u_j w_j^H E u_i \ = \ \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{w_j^H E u_i}{\lambda_j - \lambda_i} u_j$$

which is the standard expression developed in Wilkinson's book [222].

What the above expression reveals is that when eigenvalues get close to one another then the eigenvectors are not too well defined. This is predictable since a multiple eigenvalue has typically several independent eigenvectors associated with it, and we can rotate the eigenvector arbitrarily in the eigenspace while keeping it an eigenvector of A. As an eigenvalue gets close to being multiple, the condition number for its associated eigenvector deteriorates. In fact one question that follows naturally is whether or not one can define the notion of condition number for eigenvectors associated with multiple eigenvalues. The above observation suggests that a more realistic alternative is to try to analyze the sensitivity of the invariant subspace. This is taken up in the next section.

Example 3.8. Consider the matrix seen in example 3.6

$$A = \begin{pmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{pmatrix}\ .$$

The matrix is diagonalizable since it has three distinct eigenvalues and

$$A = X \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} X^{-1} .$$

One way to compute the reduced resolvent associated with $\lambda_1 = 1$ is to replace in the above equality the diagonal matrix D by the 'inverse' of $D - \lambda_1 I$ obtained by inverting the nonzero entries $(2,2)$ and $(3,3)$ and placing a zero in entry $(1,1)$,i.e.,

$$S(\lambda_1) = X \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} X^{-1} = \begin{pmatrix} -118.5 & -39.5 & -122.5 \\ 316.5 & 105.5 & 325.5 \\ 13.5 & 4.5 & 14.5 \end{pmatrix} .$$

We find that the 2-norm of $\|S(\lambda_1)\|_2$ is $\|S(\lambda_1)\|_2 = 498.27$. Thus, a perturbation of order 0.01 may cause changes of magnitude up to 4.98 on the eigenvector. This turns out to be a pessimistic overestimate. If we perturb a_{11} from -149.00 to -149.01 the eigenvector u_1 associated with λ_1 is perturbed from $u_1 = (-1/3, 1, 0)^T$ to $\tilde{u}_1 = (-0.3170, 1, -0.0174)^T$. A clue as to why we have a poor estimate is provided by looking at the norms of X and X^{-1}.

$$\|X\|_2 = 1.709 \quad \text{and} \quad \|X^{-1}\|_2 = 754.100 ,$$

which reveals that the eigenvectors are poorly conditioned. □

3.3.3 Conditioning of Invariant Subspaces

Often one is interested in the invariant subspace rather than the individual eigenvectors associated with a given eigenvalue. In these situations the condition number for eigenvectors as defined before is not sufficient. We would like to have an idea on how the whole subspace behaves under a given perturbation.

We start with the simple case where the multiplicity of the eigenvalue under consideration is one, and we define some notation. Referring to (3.41), let $Q(t)$ be the orthogonal projector onto the invariant subspace associated with the simple eigenvalue $\lambda(t)$ and $Q(0) \equiv Q$ be the orthogonal projector onto the invariant subspace of A associated with λ. The orthogonal projector Q onto the invariant subspace associated with λ has different properties from those of the spectral projector. For example A and Q do not commute. All we can say is that

$$AQ = QAQ \text{ or } (I - Q)AQ = 0 ,$$

leading to

$$\begin{aligned} (I - Q)A &= (I - Q)A(I - Q) \\ (I - Q)(A - \lambda I) &= (I - Q)(A - \lambda I)(I - Q) \end{aligned} \tag{3.46}$$

Note that the linear operator $(A - \lambda I)$ when restricted to the range of $I - Q$ is invertible. This is because if $(A - \lambda I)x = 0$ then x belongs to $\mathrm{Ran}(Q)$ whose

intersection with $\mathrm{Ran}(I-Q)$ is reduced to $\{0\}$. We denote by $S^+(\lambda)$ the inverse of $(A-\lambda I)$ restricted to $\mathrm{Ran}(I-Q)$. Note that although both $S(\lambda)$ and $S^+(\lambda)$ are inverses of $(A-\lambda I)$ restricted to complements of $\mathrm{Null}(A-\lambda I)$,these inverses are quite different.

Starting from (3.41), we subtract λu from each side to get,

$$(A-\lambda I)u(t) = -tEu(t) + (\lambda(t)-\lambda)u(t).$$

Now multiply both sides by the orthogonal projector $I-Q$,

$$(I-Q)(A-\lambda I)u(t) = -t(I-Q)Eu(t) + (\lambda(t)-\lambda)(I-Q)u(t).$$

to obtain from (3.46),

$$\begin{aligned}[(I-Q)(A-\lambda I)(I-Q)](I-Q)u(t)\\ = -t(I-Q)Eu(t) + (\lambda(t)-\lambda)(I-Q)u(t).\end{aligned}$$

Therefore,

$$(I-Q)u(t) = S^+(\lambda)\left[-t(I-Q)Eu(t) + (\lambda(t)-\lambda)(I-Q)u(t)\right].$$

We now write the vector $u(t)$ as $u(t) = Q(t)x$ for an arbitrary vector x,

$$\begin{aligned}(I-Q)Q(t)x &= S^+(\lambda)\left[-t(I-Q)EQ(t)x+\right.\\ &\qquad \left.(\lambda(t)-\lambda)(I-Q)Q(t)x\right].\end{aligned}$$

The above equation yields an estimate of the norm of $(I-Q)Q(t)$, which is the sine of the angle between the invariant subspaces $M = \mathrm{Ran}(Q)$ and $M(t) = \mathrm{Ran}(Q(t))$.

Proposition 3.5 *Assume that λ is a simple eigenvalue of A. When the matrix A is perturbed by the matrix tE,then the sine of the angle between the invariant subspaces M and $M(t)$ of A and $A+tE$ associated with the eigenvalues λ and $\lambda(t)$ is approximately,*

$$\sin\theta(M, M(t)) \approx |t|\|S^+(\lambda)(I-Q)EQ(t)\|$$

the approximation being of second order with respect to t.

Thus, we can define the condition number for invariant subspaces as being the (spectral) norm of $S^+(\lambda)$.

The more interesting situation is when the invariant subspace is associated with a multiple eigenvalue. What was just done for one-dimensional invariant subspaces can be generalized to multiple-dimensional invariant subspaces. The notion of condition numbers here will require some knowledge about generalized solutions to Sylvester's equations. A Sylvester equation is a matrix equation of the form

$$AX - XR = B, \tag{3.47}$$

where A is $n \times n$,X and B are $n \times r$ and R is $r \times r$. The important observation which we would like to exploit is that (3.47) is nothing but a linear system of equations with $n\,r$ unknowns. It can be shown that the mapping $X \rightarrow AX - XR$ is invertible under the simple condition that the spectra of A and R have no point in common.

We now proceed in a similar manner as for simple eigenvalues and write,

$$\begin{aligned} AU &= UR \\ (A + tE)U(t) &= U(t)R(t), \end{aligned}$$

in which U and $U(t)$ are $n \times r$ unitary matrices and R and $R(t)$ are $r \times r$ upper triangular. Subtracting $U(t)R$ from the second equation we obtain

$$AU(t) - U(t)R = -tEU(t) + U(t)(R(t) - R).$$

Multiplying both sides by $I - Q$ and using again the relation (3.46),

$$\begin{aligned} (I - Q)A(I - Q)U(t) - (I - Q)U(t)R \\ = (I - Q)[-tEU(t) + U(t)(R(t) - R)]. \end{aligned}$$

Observe that the operator

$$X \rightarrow (I - Q)A(I - Q)X - XR,$$

is invertible because the eigenvalues of $(I - Q)A(I - Q)$ and those of R form disjoint sets. Therefore, we can define its inverse which we call $S^+(\lambda)$, and we have

$$(I - Q)U(t) = S^+(\lambda)\,[t(I - Q)EU(t) + (I - Q)U(t)(R(t) - R)]\ .$$

As a result, up to lower order terms, the sine of the angle between the two subspaces is $|t|\|S^+(\lambda)(I-Q)EU(t)\|$,a result that constitutes a direct generalization of the previous theorem.

3.4 Localization Theorems

In some situations one wishes to have a rough idea of where the eigenvalues lie in the complex plane, by directly exploiting some knowledge on the entries of the matrix A. We already know a simple localization result that uses any matrix norm, since we have

$$|\lambda_i| \ \leq \ \|A\|$$

i.e., any eigenvalue belongs to the disc centered at the origin and of radius $\|A\|$. A more precise localization result is provided by Gerschgorin's theorem.

Theorem 3.11 (Gerschgorin [73]) *Any eigenvalue λ of a matrix A is located in one of the closed discs of the complex plane centered at a_{ii} and having the radius*

$$\sum_{\substack{j=1 \\ j \neq i}}^{j=n} |a_{ij}|\ .$$

In other words,

$$\forall\, \lambda \in \Lambda(A), \quad \exists\, i \quad \text{such that} \quad |\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{j=n} |a_{ij}| \,. \tag{3.48}$$

Proof. The proof is by contradiction. Assume that (3.48) does not hold. Then there is an eigenvalue λ such that for $i = 1, 2, \ldots, n$ we have

$$|\lambda - a_{ii}| > \sum_{j=1, j\neq i}^{j=n} |a_{ij}| \,. \tag{3.49}$$

We can write $A - \lambda I = D - \lambda I + H$,where $D = \text{diag}\,\{a_{ii}\}$ and H is the matrix obtained from A by replacing its diagonal elements by zeros. Since $D - \lambda$ is invertible we have

$$A - \lambda I = (D - \lambda I)(I + (D - \lambda I)^{-1} H) \,. \tag{3.50}$$

The elements in row i of the matrix $C = (D - \lambda I)^{-1} H$ are $c_{ij} = a_{ij}/(a_{ii} - \lambda)$ for $j \neq i$ and $c_{ii} = 0$, and so the sum of their moduli are less than unity by (3.49). Hence

$$\rho((D - \lambda I)^{-1} H) \leq \|(D - \lambda I)^{-1} H\|_\infty < 1$$

and as a result the matrix $I + C = (I + (D - \lambda I)^{-1} H)$ is nonsingular. Therefore, from (3.50) $(A - \lambda I)$ would also be nonsingular which is a contradiction. □

Since the result also holds for the transpose of A,we can formulate a version of the theorem based on column sums instead of row sums,

$$\forall\, \lambda \in \Lambda(A), \quad \exists\, j \quad \text{such that} \quad |\lambda - a_{jj}| \leq \sum_{\substack{i=1 \\ i\neq j}}^{i=n} |a_{ij}| \,. \tag{3.51}$$

The discs defined in the theorem are called Gerschgorin discs. There are n Gerschgorin discs and their union contains the spectrum of A. The above results can be especially useful when the matrix is almost diagonal, as is often the case when an algorithm is used to diagonalize a matrix and the process is nearing convergence. However, in order to better exploit the theorem, we need to show the following additional result.

Theorem 3.12 . *Suppose that there are m Gerschgorin discs whose union S is disjoint from all other discs. Then S contains exactly m eigenvalues, (counted with their multiplicities).*

Proof. Let $A(t) = D + tH$ where $0 \leq t \leq 1$, and D, H are defined in the proof of Gerschgorin's theorem. Initially when $t = 0$ all eigenvalues of $A(t)$ are at the discs of radius 0,centered at a_{ii}. By a continuity argument, as t increases to 1, the branches of eigenvalues $\lambda_i(t)$ will stay in their respective discs as long as these discs stay disjoint. This is because the image of the connected interval [0,1] by $\lambda_i(t)$ must be connected. More generally, if the union of m of the discs are disjoint from the other discs, the union $S(t)$ of the corresponding discs as t varies, will contain m eigenvalues. □

An important particular case is that when one disc is disjoint from the others then it must contain exactly one eigenvalue.

There are other ways of estimating the error of a_{ii} regarded as an eigenvalue of A. For example, if we take as approximate eigenvector the i-th column of the identity matrix we get the following result from a direct application of Kato-Temple's theorem in the Hermitian case.

Proposition 3.6 *Let i be any integer between 1 and n and let λ be the eigenvalue of A closest to a_{ii},and μ the next closest eigenvalue to a_{ii}. Then if we call ϵ_i the 2-norm of the $(n-1)$-vector obtained from the $i-th$ column of A by deleting the entry a_{ii} we have*

$$|\lambda - a_{ii}| \leq \frac{\epsilon_i^2}{|\mu - a_{ii}|} .$$

Proof. The proof is a direct application of Kato-Temple's theorem. □

Thus, in the Hermitian case, the Gerschgorin bounds are not tight in general since the error is of the order of the square of the vector of the off-diagonal elements in a row (or column), whereas Gerschgorin's result will provide an error estimate of the same order as the 1-norm of the same vector (in the ideal situation when the discs are disjoint). However, we note that the isolated application of the above proposition in practice may not be too useful since we may not have an estimate of $|\mu - a_{ii}|$. A simpler, though less powerful, bound is $|\lambda - a_{ii}| \leq \epsilon_i$. These types of results are quite different in nature from those of Gerschgorin's theorem. They simply tell us how accurate an approximation a diagonal element can be when regarded as an approximate eigenvalue. It is an isolated result and does not tell us anything on the other eigenvalues. Gerschgorin's result on the other hand is a global result, in that it tells where *all* the eigenvalues are located, as a group. This distinction between the two types of results, namely the (local) a-posteriori error bounds on the one hand, and the global localizations results such as Gerschgorin's theorem on the other, is often misunderstood.

3.5 Pseudo-eigenvalues

As was seen in earlier sections, eigenvalues can be very sensitive to perturbations for non-normal matrices. Roughly speeking pseudo-eigenvalues are sets of eigenvalues of all perturbed matrices within a radius of the original matrix. These sets can be very large when the eigenvalue is ill-conditioned.

One can define the spectrum of a matrix A as the set of values z for which the resolvent $R(z)$ has infinite 2-norm. The formal definition of the pseudo-spectrum is derived from this by using a parameter ϵ.

Definition 3.3 *Let $A \in \mathbb{C}^{n\times n}$. For $\epsilon > 0$ the ϵ-pseudospectrum of A is the set*

$$\Lambda_\epsilon(A) = \{z \in \mathbb{C} \mid \|R(z)\|_2 > \epsilon^{-1}\}. \tag{3.52}$$

Note that one can also say $\Lambda_\epsilon(A) = \{z \in \mathbb{C} \mid [\|R(z)\|_2]^{-1} < \epsilon\}$. Therefore, recalling that for a given matrix B, we have $(\|B^{-1}\|_2)^{-1} = \sigma_{min}(B)$, the smallest singular value of B, we can restate the above definition as

$$\Lambda_\epsilon(A) = \{z \in \mathbb{C} \mid \sigma_{min}(A - zI) < \epsilon\}. \tag{3.53}$$

There is an interesting connection with perturbation theory via Proposition 3.4. First, it is easy to see that

$$z \in \Lambda_\epsilon(A) \quad \text{iff} \quad \exists\, v \in \mathbb{C}^n, \|(A - zI)^{-1}v\|_2 > \epsilon^{-1} \text{ and } \|v\|_2 = 1. \tag{3.54}$$

If we define the vector $t = (A - zI)^{-1}v$ then clearly,

$$\frac{\|(A - zI)t\|_2}{\|t\|_2} = \frac{\|v\|_2}{\|t\|_2} < \epsilon.$$

Setting $w = t/\|t\|_2$, we arrive at this characterization of the pseudo-spectrum:

$$z \in \Lambda_\epsilon(A) \quad \text{iff} \quad \exists\, w \in \mathbb{C}^n, \|(A - zI)w\|_2 < \epsilon \text{ and } \|w\|_2 = 1. \tag{3.55}$$

The following proposition summarizes the above equivalences and adds one more.

Proposition 3.7 *The following five properties are equivalent:*

(i) $z \in \Lambda_\epsilon(A)$.

(ii) $\sigma_{min}(A - zI) < \epsilon$.

(iii) There exist a unit vector v such that $\|(A - zI)^{-1}v\|_2 > \epsilon^{-1}$.

(iv) There exist a unit vector w such that $\|(A - zI)w\|_2 < \epsilon$.

(v) There exist a matrix E, with $\|E\|_2 \le \epsilon$ such that z is an eigenvalue of $A-E$.

Proof. For the equivalences of (i), (ii), (iii), and (iv) see equations (3.53–3.55). To show that (iv) $\leftrightarrow$ (v) we invoke Proposition 3.4 with $r = (A - zI)w$. If (iv) holds then the proposition implies that there is a matrix E with $\|E\|_2 = \|r\| < \epsilon$ and $z \in \Lambda(A-E)$. If (v) holds then there is a unit vector w such that $(A-E)w = zw$. Hence, $(A - zI)w = Ew$ and $\|(A - zI)w\|_2 = \|Ew\|_2 \le \|E\|_2\|w\|_2 < \epsilon$ showing that (iv) holds. □

Pseudo-spectra are useful to analyze *transient* behavior of operators when the usual *asymptotic* analysis fails. Consider for example the powers of a given non-normal matrix G. Sequences of the form $w_k = G^k w_0$ are commonly studied when considering so-called stationary iterative methods for solving linear sytems. As is well-known these methods amount to an iteration of the form

$$x_{k+1} = Gx_k + f, \tag{3.56}$$

starting with some initial guess x_0. The above is a fixed point iteration which attempts to reach the fixed point for which $x_* = Gx_* + f$. The error $\|x_k - x_*\|_2$ at the k-th step, satisfies:

$$\|x_k - x_*\|_2 = \|G^k(x_k - x_0)\|_2 \leq \|G^k\|_2 \|x_k - x_0\|_2. \tag{3.57}$$

As a result the scheme will converge when $\rho(G) < 1$. We have seen in Chapter 1 (Corollary 1.1) that for any matrix norm $\|G^k\|^{1/k}$ tends to $\rho(G)$, so we may infer that asymptotically, the above error behaves like $\|x_k - x_*\|_2 \approx \rho(G)^k \|x_k - x_0\|_2$. While this a reasonable assumption, in practice what is often obversed is a long stage of big increases of the norm of the error before it finally reaches the 'asymptotic' phase where it declines steadily to zero. Such a behavior is only characteristic of non-normal matrices. For normal matrices $\|G^k\|_2$ cannot stray too far away from its approximation $\rho(G)^k$.

In other words, while the spectral radius gives a good idea of the *asymptotic* behavior of G^k, the spectrum of G does not help analyze its *transient* behavior.

This can be understood if one considers the power of G using the Taylor-Dunford integral:

$$G^k = \frac{-1}{2\pi i} \sum_{j=1}^{p} \int_{\Gamma_j} R(z) z^k dz \tag{3.58}$$

where the Jordan curves around each distinct eigenvalue $\lambda_1, \cdots, \lambda_p$, were defined at the end of Section 3.1.3. In the normal case, $R(z)$ can be expanded into a sum of terms of the form

$$R(z) = \sum_{j=1}^{p} \frac{P_j}{\lambda_j - z}$$

and the above expression will simply integrate to $G^k = \sum \lambda_j^k P_j$. In the non-normal case, the situation is more complicated because the expansion of G will now involve a nilpotent, see Theorem 1.3 in Chapter 1. Writing $G - zI = \sum(\lambda_j - zI)P_j + D_j$, it is possible to expand $R(z)$:

$$(G - zI)^{-1} = \sum_{j=1}^{p} \frac{P_j}{\lambda_j - z} \left(\sum_{l=0}^{k} \frac{(-1)^l}{(\lambda_j - z)^l} D_j^l \right) . \tag{3.59}$$

Due to the nilpotence of the operators D_j each inner sum is limited to the power $l_j - 1$ where l_j is the index of λ_j. Substituting (3.59) into the integral (3.58), and writing

$$z^k = (\lambda_j + (z - \lambda_j))^k = \sum_{m=0}^{k} \binom{k}{m} \lambda_j^{k-m} (z - \lambda_j)^m$$

one arrives at the following expression:

$$G^k = \sum_{j=1}^{p} \sum_{l=0}^{\min(k,l_i-1)} \frac{k!}{l!(k-l)!} \lambda_j^{k-l} P_j D_j^l. \tag{3.60}$$

If the dominant eigenvalue is λ_1 then clearly the dominating term will be the one corresponding to $l = 0$ in the sum associated with $j = 1$. This term is $\lambda_1^k P_1$. In other words asomptically, the sum will behave like $\lambda_1^k P_1$. However, it is clear that the intermediate terms, i.e., those of the first few powers, can grow to become very large, due to binomial coefficient, as well as the nilpotent D_j^l. This behavior is typically indicated by large pseudo-spectra. Indeed, the expansion (3.59) suggests that $\|R(z)\|_2$ can be large when z is in a big region surrounding λ_1 in the highly non-normal case.

PROBLEMS

P-3.1 If P is a projector onto M along S then P^H is a projector onto $S^\perp$ along $M^\perp$. [Hint: see proof of Proposition 3.1].

P-3.2 Show that for two orthogonal bases V_1, V_2 of the same subspace M of $\mathbb{C}^n$ we have $V_1 V_1^H x = V_2 V_2^H x \ \forall x$.

P-3.3 What are the eigenvalues of a projector? What about its eigenvectors?

P-3.4 Let P be a projector and $V = [v_1, v_2, \cdots, v_m]$ a basis of $\mathrm{Ran}(P)$. Why does there always exist a basis $W = [w_1, w_2, \cdots, w_m]$ of $L = \mathrm{Null}(P)^\perp$ such that the two sets form a biorthogonal basis? In general given two subspaces M and S of the same dimension m, is there always a biorthogonal pair V, W such that V is a basis of M and W a basis of S?

P-3.5 Let P be a projector, $V = [v_1, v_2, \cdots, v_m]$ a basis of $\mathrm{Ran}(P)$, and U a matrix the columns of which form a basis of $\mathrm{Null}(P)$. Show that the system U, V forms a basis of $\mathbb{C}^n$. What is the matrix representation of P with respect to this basis?

P-3.6 Show that if two projectors P_1 and P_2 commute then their product $P = P_1 P_2$ is a projector. What are the range and null space of P?

P-3.7 Consider the matrix seen in Example 3.6. We perturb the term a_{33} to -25.01. Give an estimate in the changes of the eigenvalues of the matrix. Use any FORTRAN library or interactive tool to compute the eigenvectors/ eigenvalues of the perturbed matrix.

P-3.8 Let

$$\delta(X, Y) \equiv \max_{x \in X, \|x\|_2 = 1} \mathrm{dist}(u, Y).$$

Show that

$$\omega(M_1, M_2) = \max\{\delta(M_1, M_2), \delta(M_2, M_1)\}\ .$$

P-3.9 Given two subspaces M and S with two orthogonal bases V and W show that the singular values of $V^H W$ are between zero and one. The canonical angles between M and S are defined as the acutes angles whose cosines are the singular values σ_i,i.e., $\cos\theta_i = \sigma_i(V^H W)$. The angles are labeled in descending order. Show that this definition

does not depend on the order of the pair M, S (in other words that the singular values of $W^H V$ are identical with those of $V^H W$).

P-3.10 Show that the largest canonical angle between two subspaces (see previous problem) is $\pi/2$ iff the intersection of M and the orthogonal of S is not reduced to $\{0\}$.

P-3.11 Let P_1, P_2 be two orthogonal projectors with ranges M_1 and M_2 respectively of the same dimension $m \leq n/2$ and let $V_i, i = 1, 2$ be an orthogonal basis of $M_i, i = 1, 2$. Assuming at first that the the columns of the system $[V_1, V_2]$ are linearly independent what is the matrix representation of the projector $P_1 - P_2$ with respect to the basis obtained by completing V_1, V_2 into a basis of $\mathbb{C}^n$? Deduce that the eigenvalues of $P_1 - P_2$ are $\pm \sin\theta_i$,where the θ_i's are the canonical angles between M_1 and M_2 as defined in the previous problems. How can one generalize this result to the case where the columns of $[V_1, V_2]$ are not linearly independent?

P-3.12 Use the previous result to show that

$$\omega(M_1, M_2) = \sin\theta_{max}$$

where θ_{max} is the largest canonical angle between the two subspaces.

P-3.13 Prove the second equality in equation (3.33) of the proof of Theorem 3.10.

P-3.14 Let $E = xp^H + yq^H$ where $x \perp y$ and $p \perp q$. What is the 2-norm of E? [Hint: Compute $E^H E$ and then find the singular values of E.]

P-3.15 Show that the condition number of a simple eigenvalue λ of a matrix A does not change if A is transformed by an orthogonal similarity transformation. Is this true for any similarity transformation? What can be said of the condition number of the corresponding eigenvector?

P-3.16 Consider the matrix obtained from that of example 3.7 in which the elements -1 above the diagonal are replaced by $-\alpha$, where α is a constant. Find bounds similar to those in Example 3.7 for the condition number of the eigenvalue λ_1 of this matrix.

P-3.17 Under the same assumptions as those of Theorem 3.6, establish the improved error

$$\sin\theta(\tilde{u}, u) \leq \sqrt{\frac{\|r\|_2^2 - \epsilon^2}{\delta^2 - \epsilon^2}}$$

in which $\epsilon \equiv |\lambda - \tilde{\lambda}|$. [Hint: Follow proof of theorem 3.6]

NOTES AND REFERENCES. Some of the material in this chapter is based on [105] and [22]. A broader and more detailed view of perturbation analysis for matrix problems is the recent book by Stewart and Sun [205]. The treatment of the equivalence between the projectors as defined from the Jordan canonical form and the one defined from the Dunford integral does not seem to have been discussed earlier in the literature. The results of Section 3.2.3 are simpler versions of those found in [101], which should be consulted for more detail. The notion of condition number for eigenvalue problems is discussed in detail in Wilkinson [222] who seems to be at the origin of the notion of condition numbers for eigenvalues and eigenvectors. The notion of pseudo-spectra and pseudo-eigenvalues has been known for some time in the Russian litterature, see for example, references in the book by S. Godunov [76], where they are termed *spectral portraits*. They were promoted as a tool to replace the common spectra in practical applications in a number of papers by Trefethen au co-workers, see

e.g., [211, 212, 214]. Kato in his seminal treatise [105], also refers to pseudo-eigenvalues and pseudo-eigenvectors but these are defined only in the context of sequences of operators, A_k: The pair w_k, z_k such that $(A_k - z_k I)u_k\|_2 = \epsilon_k$, with $\lim \epsilon_k = 0$ is termed a sequence of pseudo-eigenvalue / pseudo-eigenvector pair. ■

Chapter 4

THE TOOLS OF SPECTRAL APPROXIMATION

Many of the algorithms used to approximate spectra of large matrices consist of a blend of a few basic mathematical or algorithmic tools, such as projection methods, Chebyshev acceleration, deflation, shift-and-invert strategies, to name just a few. We have grouped together these tools and techniques in this chapter. We start with some background on well-known procedures based on single vector iterations. These have historically provided the starting point of many of the more powerful methods. Once an eigenvalue-eigenvector pair is computed by one of the single vector iterations, it is often desired to extract another pair. This is done with the help of a standard technique known as deflation which we discuss in some detail. Finally, we will present the common projection techniques which constitute perhaps the most important of the basic techniques used in approximating eigenvalues and eigenvectors.

4.1 Single Vector Iterations

One of the oldest techniques for solving eigenvalue problems is the so-called power method. Simply described this method consists of generating the sequence of vectors $A^k v_0$ where v_0 is some nonzero initial vector. A few variants of the power method have been developed which consist of iterating with a few simple functions of A. These methods involve a single sequence of vectors and we describe some of them in this section.

4.1.1 The Power Method

The simplest of the single vector iteration techniques consists of generating the sequence of vectors $A^k v_0$ where v_0 is some nonzero initial vector. This sequence of vectors when normalized appropriately, and under reasonably mild conditions, converges to a dominant eigenvector, i.e., an eigenvector associated with the eigenvalue of largest modulus. The most commonly used normalization is to ensure that the largest component of the current iterate is equal to one. This yields the following algorithm.

ALGORITHM **4.1 (The Power Method.)**

- *1. Start: Choose a nonzero initial vector* v_0.
- *2. Iterate: for* $k = 1, 2, \ldots$ *until convergence, compute*

$$v_k = \frac{1}{\alpha_k} A v_{k-1}$$

where α_k *is a component of the vector* Av_{k-1} *which has the maximum modulus.*

The following theorem establishes a convergence result for the above algorithm.

Theorem 4.1 *Assume that there is one and only one eigenvalue* λ_1 *of* A *of largest modulus and that* λ_1 *is semi-simple. Then either the initial vector* v_0 *has no component in the invariant subspace associated with* λ_1 *or the sequence of vectors generated by Algorithm 4.1 converges to an eigenvector associated with* λ_1 *and* α_k *converges to* λ_1.

Proof. Clearly, v_k is nothing but the vector $A^k v_0$ normalized by a certain scalar $\hat{\alpha}_k$ in such a way that its largest component is unity. Let us decompose the initial vector v_0 as

$$v_0 = \sum_{i=1}^{p} P_i v_0 \tag{4.1}$$

where the P_i's are the spectral projectors associated with the distinct eigenvalues $\lambda_i, i = 1, \ldots, p$. Recall from (1.23) of Chapter 1, that $AP_i = P_i(\lambda_i P_i + D_i)$ where D_i is a nilpotent of index l_i, and more generally, by induction we have $A^k P_i = P_i(\lambda_i P_i + D_i)^k$. As a result we obtain,

$$v_k = \frac{1}{\hat{\alpha}_k} A^k \sum_{i=1}^{p} P_i v_0 = \frac{1}{\hat{\alpha}_k} \sum_{i=1}^{p} A^k P_i v_0 = \frac{1}{\hat{\alpha}_k} \sum_{i=1}^{p} P_i(\lambda_i I + D_i)^k v_0 \, .$$

Hence, noting that $D_1 = 0$ because λ_1 is semi-simple,

$$\begin{aligned} v_k &= \frac{1}{\hat{\alpha}_k} \sum_{i=1}^{p} P_i(\lambda_i P_i + D_i)^k v_0 \\ &= \frac{1}{\hat{\alpha}_k} \left(\lambda_1^k P_1 v_0 + \sum_{i=2}^{p} P_i(\lambda_i P_i + D_i)^k v_0 \right) \\ &= \frac{\lambda_1^k}{\hat{\alpha}_k} \left(P_1 v_0 + \sum_{i=2}^{p} \frac{1}{\lambda_1^k} (\lambda_i P_i + D_i)^k P_i v_0 \, . \right) \end{aligned} \tag{4.2}$$

The spectral radius of each operator $(\lambda_i P_i + D_i)/\lambda_1$ is less than one since $|\lambda_i/\lambda_1| < 1$ and therefore, its k-th power will converge to zero. If $P_1 v_0 = 0$ the theorem is

true. Assume that $P_1 v_0 \neq 0$. Then it follows immediately from (4.2) that v_k converges to $P_1 v_0$ normalized so that its largest component is one. That α_k converges to the eigenvalue λ_1 is an immediate consequence of the relation $Av_{k-1} = \alpha_k v_k$ and the fact the sequence of vectors v_k converges. □

The proof suggests that the convergence factor of the method is given by

$$\rho_D = \frac{|\lambda_2|}{|\lambda_1|}$$

where λ_2 is the second largest eigenvalue in modulus. This ratio represents the spectral radius of the linear operator $\frac{1}{\lambda_1} A$ restricted to the subspace that excludes the invariant subspace associated with the dominant eigenvalue. It is a common situation that the eigenvalues λ_1 and λ_2 are very close from one another. As a result convergence may be extremely slow.

Example 4.1. Consider the Markov Chain matrix Mark(10) which has been described in Chapter 2. This is a matrix of size $n = 55$ which has two dominant eigenvalues of equal modulus namely $\lambda = 1$ and $\lambda = -1$. As is to be expected the power method applied directly to A does not converge. To obtain convergence we can for example consider the matrix $I+A$ whose eigenvalues are those of A shifted to the right by one. The eigenvalue $\lambda = 1$ is then transformed into the eigenvalue $\lambda = 2$ which now becomes the (only) dominant eigenvalue. The algorithm then converges and the convergence history is shown in Table 4.1. In the first column of the table we show the iteration number. The results are shown only every 20 steps and at the very last step when convergence has taken place. The second column shows the 2-norm of the difference between two successive iterates, i.e., $\|x_{i+1} - x_i\|_2$ at teration i, while the third column shows the residual norm $\|Ax - \mu(x)x\|_2$, in which $\mu(x)$ is the Rayleigh quotient of x and x is normalized to have a 2-norm unity. The algorithm is stopped as soon at the 2-norm of the difference between two successive iterates becomes less than $\epsilon = 10^{-7}$. Finally, the last column shows the corresponding eigenvalue estimates. Note that what is shown is simply the coefficient α_k, shifted by -1 to get an approximation to the eigenvalue of $\mathrm{Mark}(10)$ instead of $\mathrm{Mark}(10) + I$. The initial vector in the iteration is the vector $x_0 = (1, 1, \ldots, 1)^T$. □

If the eigenvalue is multiple, but semi-simple, then the algorithm provides only one eigenvalue and a corresponding eigenvector. A more serious difficulty is that the algorithm will not converge if the dominant eigenvalue is complex and the original matrix as well as the initial vector are real. This is because for real matrices the complex eigenvalues come in complex pairs and as result there will be (at least) two distinct eigenvalues that will have the largest modulus in the spectrum. Then the theorem will not guarantee convergence. There are remedies to all these difficulties and some of these will be examined later.

Iteration	Norm of diff.	Res. norm	Eigenvalue
20	0.639D-01	0.276D-01	1.02591636
40	0.129D-01	0.513D-02	1.00680780
60	0.192D-02	0.808D-03	1.00102145
80	0.280D-03	0.121D-03	1.00014720
100	0.400D-04	0.174D-04	1.00002078
120	0.562D-05	0.247D-05	1.00000289
140	0.781D-06	0.344D-06	1.00000040
161	0.973D-07	0.430D-07	1.00000005

Table 4.1: Power iteration with $A = \mathrm{Mark}(10) + I$.

4.1.2 The Shifted Power Method

In Example 4.1 we have been lead to use the power method not on the original matrix but on the *shifted* matrix $A + I$. One observation is that we could also have iterated with a matrix of the form $B(\sigma) = A + \sigma I$ for any positive σ and the choice $\sigma = 1$ is a rather arbitrary choice. There are better choices of the shift as is suggested by the following example.

Example 4.2. Consider the same matrix as in the previous example, in which the shift σ is replaced by $\sigma = 0.1$. The new convergence history is shown in Table 4.1, and indicates a much faster convergence than before. □

Iteration	Norm of diff.	Res. Norm	Eigenvalue
20	0.273D-01	0.794D-02	1.00524001
40	0.729D-03	0.210D-03	1.00016755
60	0.183D-04	0.509D-05	1.00000446
80	0.437D-06	0.118D-06	1.00000011
88	0.971D-07	0.261D-07	1.00000002

Table 4.1 Power iteration on $A = Mark(10) + 0.1 \times I$.

More generally, when the eigenvalues are real it is not too difficult to find the optimal value of σ, i.e., the shift that maximizes the asymptotic convergence rate, see Problem P-4.5. The scalars σ are called *shifts of origin*. The important property that is used is that shifting does not alter the eigenvectors and that it does change the eigenvalues in a simple known way, it shifts them by σ.

4.1.3 Inverse Iteration

The inverse power method, or inverse iteration, consists simply of iterating with the matrix A^{-1} instead of the original matrix A. In other words, the general iterate

v_k is defined by

$$v_k = \frac{1}{\alpha_k} A^{-1} v_{k-1} \ . \tag{4.3}$$

Fortunately it is not necessary to compute the matrix A^{-1} explicitly as this could be rather expensive for large problems. Instead, all that is needed is to carry out the LU factorization of A prior to starting the vector iteration itself. Subsequently, one must solve an upper and lower triangular system at each step. The vector v_k will now converge to the eigenvector associated with the dominant eigenvalue of A^{-1}. Since the eigenvalues of A and A^{-1} are the inverses of each other while their eigenvectors are identical, the iterates will converges to the eigenvector of A associated with the eigenvalue of smallest modulus. This may or may not be what is desired but in practice the method is often combined with shifts of origin. Indeed, a more common problem in practice is to compute the eigenvalue of A that is closest to a certain scalar σ and the corresponding eigenvector. This is achieved by iterating with the matrix $(A - \sigma I)^{-1}$. Often, σ is referred to as the *shift*. The corresponding algorithm is as follows.

ALGORITHM **4.2 : Inverse Power Method**

1. **Start:** *Compute the LU decomposition $A - \sigma I = LU$ and choose an initial vector v_0.*
2. **Iterate:** *for $k = 1, 2, \ldots,$ until convergence compute*

$$v_k = \frac{1}{\alpha_k}(A - \sigma I)^{-1} v_{k-1} = \frac{1}{\alpha_k} U^{-1} L^{-1} v_{k-1} \tag{4.4}$$

where α_k is a component of the vector $(A - \sigma I)^{-1} v_{k-1}$ which has the maximum modulus.

Note that each of the computations of $y = L^{-1} v_{k-1}$ and then $v = U^{-1} y$ can be performed by a forward and a backward triangular system solve, each of which costs only $O(n^2/2)$ operations when the matrix is dense. The factorization in step 1 is much more expensive whether the matrix is dense or sparse.

If λ_1 is the eigenvalue closest to σ then the eigenvalue of largest modulus of $(A - \sigma I)^{-1}$ will be $1/(\lambda_1 - \sigma)$ and so α_k will converge to this value. An important consideration that makes Algorithm 4.2 quite attractive is its potentially high convergence rate. If λ_1 is the eigenvalue of A closest to the shift σ and λ_2 is the next closet one then the convergence factor is given by

$$\rho_I = \frac{|\lambda_1 - \sigma|}{|\lambda_2 - \sigma|} \tag{4.5}$$

which indicates that the convergence can be very fast if σ is much closer to the desired eigenvalue λ_1 than it is to λ_2.

From the above observations, one can think of changing the shift σ occasionally into a value that is known to be a better approximation of λ_1 than the previous σ. For example, one can replace occasionally σ by the estimated eigenvalue of A

that is derived from the information that α_k converges to $1/(\lambda_1 - \sigma)$, i.e., we can take

$$\sigma_{new} = \sigma_{old} + \frac{1}{\alpha_k}.$$

Strategies of this sort are often referred to as shift-and-invert techniques.

Another possibility, which may be very efficient in the Hermitian case, is to take the new shift to be the Rayleigh quotient of the latest approximate eigenvector v_k. One must remember however, that the LU factorization is expensive so it is desirable to keep such shift changes to a minimum. At one extreme where the shift is never changed, we obtain the simple inverse power method represented by Algorithm 4.2. At the other extreme, one can also change the shift at every step. The algorithm corresponding to this case is called Rayleigh Quotient Iteration (RQI) and has been extensively studied for Hermitian matrices.

ALGORITHM **4.3 Rayleigh Quotient Iteration**

1. **Start:** *Choose an initial vector* v_0 *such that* $\|v_0\|_2 = 1$.
2. **Iterate:** *for* $k = 1, 2, \ldots,$ *until convergence compute*

$$\begin{aligned} \sigma_k &= (Av_{k-1}, v_{k-1}) , \\ v_k &= \frac{1}{\alpha_k}(A - \sigma_k I)^{-1} v_{k-1}, \end{aligned}$$

where α_k *is chosen so that the 2-norm of the vector* v_k *is one.*

It is known that this process is globally convergent for Hermitian matrices, in the sense that α_k converges and the vector v_k either converges to an eigenvector or alternates between two eigenvectors. Moreover, in the first case α_k converges cubically towards an eigenvalue, see Parlett [148]. In the case where v_k oscillates, between two eigenvectors, then α_k converges towards the mid-point of the corresponding eigenvalues. In the non-Hermitian case, the convergence can be at most quadratic and there are no known global convergence results except in the normal case. This algorithm is not much used in practice despite these nice properties, because of the high cost of the frequent factorizations.

4.2 Deflation Techniques

Suppose that we have computed the eigenvalue λ_1 of largest modulus and its corresponding eigenvector u_1 by some simple algorithm, say algorithm (A), which always delivers the eigenvalue of largest modulus of the input matrix, along with an eigenvector. For example, algorithm (A) can simply be one of the single vector iterations described in the previous section. It is assumed that the vector u_1 is normalized so that $\|u_1\|_2 = 1$. The problem is to compute the next eigenvalue λ_2 of A. An old technique for achieving this is what is commonly called a deflation procedure. Typically, a rank one modification is applied to the original matrix so as to displace the eigenvalue λ_1, while keeping all other eigenvalues unchanged.

The rank one modification is chosen so that the eigenvalue λ_2 becomes the one with largest modulus of the modified matrix and therefore, algorithm (A) can now be applied to the new matrix to extract the pair λ_2, u_2.

4.2.1 Wielandt Deflation with One Vector

In the general procedure known as Wielandt's deflation only the knowledge of the right eigenvector is required. The deflated matrix is of the form

$$A_1 \;=\; A - \sigma u_1 v^H \tag{4.6}$$

where v is an arbitrary vector such that $v^H u_1 = 1$, and σ is an appropriate shift. It can be shown that the eigenvalues of A_1 are the same as those of A except for the eigenvalue λ_1 which is transformed into the eigenvalue $\lambda_1 - \sigma$.

Theorem 4.2 (Wielandt) *The spectrum of A_1 as defined by (4.6) is given by*

$$\Lambda(A_1) = \{\lambda_1 - \sigma, \lambda_2, \lambda_3, \ldots, \lambda_p\} .$$

Proof. For $i \neq 1$ the left eigenvectors of A satisfy

$$(A^H - \bar{\sigma} v u_1^H) w_i = \lambda_i w_i$$

because w_i is orthogonal to u_1. On the other hand for $i = 1$, we have $A_1 u_1 = (\lambda_1 - \sigma) u_1$. □

The above proof reveals that the left eigenvectors $w_2, \ldots, w_p$ are preserved by the deflation process. Similarly, the right eigenvector u_1 is preserved. It is also important to see what becomes of the other right eigenvectors. For each i, we seek a right eigenvector of A_1 in the form of $\hat{u}_i = u_i - \gamma_i u_1$. We have,

$$\begin{aligned} A_1 \hat{u}_i &= (A - \sigma u_1 v^H)(u_i - \gamma_i u_1) \\ &= \lambda_i u_i - [\gamma_i \lambda_1 + \sigma v^H u_i - \sigma \gamma_i] u_1. \end{aligned} \tag{4.7}$$

Taking $\gamma_1 = 0$ shows, as is already indicated by the proposition, that any eigenvector associated with the eigenvalue λ_1 remains an eigenvector of A_1, associated with the eigenvalue $\lambda_1 - \sigma$. For $i \neq 1$, it is possible to select γ_i so that the vector $\hat{u}_i$ is an eigenvector of A_1 associated with the eigenvalue λ_i,

$$\gamma_i(v) \equiv \frac{v^H u_i}{1 - (\lambda_1 - \lambda_i)/\sigma} . \tag{4.8}$$

Observe that the above expression is not defined when the denominator vanishes. However, it is known in this case that the eigenvalue $\lambda_i = \lambda_1 - \sigma$ is already an eigenvalue of A_1, i.e., the eigenvalue $\lambda_1 - \sigma$ becomes multiple, and we only know one eigenvector namely u_1.

There are infinitely many different ways of choosing the vector v. One of the most common choices is to take $v = w_1$ the left eigenvector. This is referred to as

Hotelling's deflation. It has the advantage of . preserving both the left and right eigenvectors of A as is seen from the fact that $\gamma_i = 0$ in this situation. Another simple choice is to take $v = u_1$. In the next section we will consider these different possibilities and try to make a rational choice between them.

Example 4.3. As a test we consider again the matrix Mark(10) seen is Example 4.1. For u_1 we use the vector computed from the shifted power method with shift 0.1. If we take v to be a random vector and x_0 to be a random vector, then the algorithm converges in 135 steps and yields $\lambda_2 \approx 0.93715016$. The stopping criterion is identical with the one used in Example 4.1. If we take $v = u_1$ or $v = (1, 1, \ldots, 1)^T$, then the algorithm converges in 127 steps. □

4.2.2 Optimality in Wieldant's Deflation

An interesting question that we wish to answer is: among all the possible choices of v, which one is likely to yield the best possible condition number for the next eigenvalue λ_2 to be computed? This is certainly a desirable goal in practice. We will distinguish the eigenvalues and eigenvectors associated with the matrix A_1 from those of A by denoting them with a tilde. The condition number of the next eigenvalue $\tilde{\lambda}_2$ to be computed is, by definition,

$$\mathrm{Cond}(\tilde{\lambda}_2) = \frac{\|\tilde{u}_2\|_2 \|\tilde{w}_2\|_2}{|(\tilde{u}_2, \tilde{w}_2)|}$$

where $\tilde{u}_2, \tilde{w}_2$ are the right and left eigenvectors of A_1 associated with the eigenvalue $\tilde{\lambda}_2$. From what we have seen before, we know that $\tilde{w}_2 = w_2$ while $\tilde{u}_2 = u_2 - \gamma_2(v)u_1$ where $\gamma_2(v)$ is given by (4.8). Assuming that $\|w_2\|_2 = 1$ we get,

$$\mathrm{Cond}(\tilde{\lambda}_2) = \frac{\|u_2 - \gamma_2(v)u_1\|_2}{|(u_2, w_2)|} \tag{4.9}$$

where we have used the fact that $(u_1, w_2) = 0$. It is then clear from (4.9) that the condition number of λ_2 is minimized whenever

$$\gamma_2(v) = u_1^H u_2 \equiv \cos\theta(u_1, u_2)\ . \tag{4.10}$$

Substituting this result in (4.8) we obtain the equivalent condition

$$v^H u_2 \ = \left(1 - \frac{\lambda_1 - \lambda_2}{\sigma}\right)\ u_1^H u_2\ , \tag{4.11}$$

to which we add the normalization condition,

$$v^H u_1 \ = \ 1. \tag{4.12}$$

There are still infinitely many vectors v that satisfy the above two conditions. However, we can seek a vector v which is spanned by two specific vectors. There are two natural possibilities; we can either take v in the span of (u_1, w_1) or in the

span of (u_1, u_2). The second choice does not seem natural since the eigenvector u_2 is not assumed to be known; it is precisely what we are trying to compute. However, it will illustrate an interesting point, namely that the choice $v = u_1$ may be nearly optimal in realistic situations. Thus, we will now consider the case $v \in \text{span}\{u_1, u_2\}$. The other interesting case, namely $v \in \text{span}\{u_1, w_1\}$, is left as an exercise, see Exercise P-4.3.

We can write v as $v = \alpha u_1 + \beta z$ in which z is obtained by orthonormalizing u_2 against u_1, i.e., $z = \hat{z}/\|\hat{z}\|_2$, $\hat{z} = u_2 - u_1^H u_2 u_1$. From (4.12) we immediately get $\alpha = 1$ and from (4.11) we obtain

$$\beta = -\frac{\lambda_1 - \lambda_2}{\sigma} \frac{u_1^H u_2}{z^H u_2} ,$$

which leads to the expression for the optimal v,

$$v_{opt} = u_1 - \frac{\lambda_1 - \lambda_2}{\sigma} \cot\theta(u_1, u_2) z . \tag{4.13}$$

We also get that

$$\text{Cond}(\tilde{\lambda}_2) = \text{Cond}(\lambda_2) \sin\theta(u_1, u_2) . \tag{4.14}$$

Interestingly enough, when $(\lambda_2 - \lambda_1)$ is small with respect to σ or when θ is close to $\pi/2$, the choice $v = u_1$ is nearly optimal.

This particular choice has an interesting additional property: *it preserves the Schur vectors*.

Proposition 4.1 *Let u_1 be an eigenvector of A of norm 1, associated with the eigenvalue λ_1 and let $A_1 \equiv A - \sigma u_1 u_1^H$. Then the eigenvalues of A_1 are $\tilde{\lambda}_1 = \lambda_1 - \sigma$ and $\tilde{\lambda}_j = \lambda_j, j = 2, 3 \ldots, n$. Moreover, the Schur vectors associated with $\tilde{\lambda}_j, j = 1, 2, 3 \ldots, n$ are identical with those of A.*

Proof. Let $AU = UR$ be the Schur factorization of A, where R is upper triangular and U is orthonormal. Then we have

$$A_1 U = [A - \sigma u_1 u_1^H]U = UR - \sigma u_1 e_1^H = U[R - \sigma e_1 e_1^H] .$$

The result follows immediately. □

Example 4.4. We take again as a test example the matrix $\text{Mark}(10)$ seen is Example 4.1 and Example 4.3. We use the approximate eigenvectors u_1 and u_2 as computed from Example 4.3. We then compute the left eigenvector $\hat{w}_2$ using again the power method on the deflated and transposed matrix $A^H - \sigma u_1^H v$. This is done fpur times: first with $v = w_1 = (1, 1, ..., 1)^T$, then $v = u_1$,

$$v = (1, -1, 1, -1, 1, \ldots, (-1)^n)^T,$$

and finally $v =$ a random vector. The condition numbers obtained for the second eigenvalue for each of these choices are shown in Table 4.2. See Problem P-4.7 for additional facts concerning this example.

v	$\mathrm{Cond}(\lambda_2)$
w_1	1.85153958
u_1	1.85153958
$(1,-1,\ldots)^T$	9.87049400
Random	2.27251031

Table 4.2 Condition numbers of the second eigenvalue for different v's.

As is observed here the best condition numbers are obtained for the first two choices. Note that the vector $(1,1,\ldots,1)$ is a left eigenvector associated with the eigenvalue λ_1. Surprisingly, these best two condition numbers are equal. In fact computing the inner product of u_1 and u_2 we find that it is zero, a result that is probably due to the symmetries in the physical problem. The relation (4.14) indicates that in this situation the two condition numbers are equal to the condition number for the undeflated matrix. □

4.2.3 Deflation with Several Vectors.

Let $q_1, q_2, \ldots q_j$ be a set of Schur vectors associated with the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_j$. We denote by Q_j the matrix of column vectors $q_1, q_2, \ldots q_j$. Thus,

$$Q_j \equiv [q_1, q_2, \ldots, q_j]$$

is an orthonormal matrix whose columns form a basis of the eigenspace associated with the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_j$. We do not assume here that these eigenvalues are real, so the matrix Q_j may be complex. An immediate generalization of Proposition 4.1 is the following.

Proposition 4.2 *Let Σ_j be the $j \times j$ diagonal matrix*

$$\Sigma_j = \mathrm{diag}\ (\sigma_1, \sigma_2, \ldots \sigma_j),$$

and Q_j an $n \times j$ orthogonal matrix consisting of the Schur vectors of A associated with $\lambda_1, \ldots, \lambda_j$. Then the eigenvalues of the matrix

$$A_j \equiv A - Q_j \Sigma_j Q_j^H,$$

are $\tilde{\lambda}_i = \lambda_i - \sigma_i$ for $i \leq j$ and $\tilde{\lambda}_i = \lambda_i$ for $i>j$. Moreover, its associated Schur vectors are identical with those of A.

Proof. Let $AU = UR$ be the Schur factorization of A. We have

$$A_j U = [A - Q_j \Sigma_j Q_j^H]U = UR - Q_j \Sigma_j E_j^H,$$

where $E_j = [e_1, e_2, \ldots e_j]$. Hence

$$A_j U = U[R - E_j \Sigma_j E_j^H]$$

and the result follows. □

Clearly, it is not necessary that Σ_j be a diagonal matrix. We can for example select it to be a triangular matrix. However, it is not clear how to select the non-diagonal entries in such a situation. An alternative technique for deflating with several Schur vectors is described in Exercise P-4.6.

4.2.4 Partial Schur Decomposition.

It is interesting to observe that the preservation of the Schur vectors is analogous to the preservation of the eigenvectors under Hotelling's deflation in the Hermitian case. The previous proposition suggests a simple incremental deflation procedure consisting of building the matrix Q_j one column at a time. Thus, at the j-th step, once the eigenvector $\tilde{u}_{j+1}$ of A_j is computed by the appropriate algorithm (A) we can orthonormalize it against all previous q_i's to get the next Schur vector q_{j+1} which will be appended to q_j to form the new deflation matrix Q_{j+1}. It is a simple exercise to show that the vector q_{j+1} thus computed is a Schur vector associated with the eigenvalue λ_{j+1} and therefore at every stage of the process we have the desired decomposition

$$AQ_j \;=\; Q_j R_j, \tag{4.15}$$

where R_j is some $j \times j$ upper triangular matrix.

More precisely we may consider the following algorithm, in which the successive shifts σ_i are chosen so that for example $\sigma_i = \lambda_i$.

ALGORITHM **4.4 Schur Wielandt Deflation**
For $i = 0, 1, 2, \ldots, j-1$ do:

1. *Define $A_i \equiv A_{i-1} - \sigma_{i-1} q_{i-1} q_{i-1}^H$ (initially define $A_0 \equiv A$) and compute the dominant eigenvalue λ_i of A_i and the corresponding eigenvector $\tilde{u}_i$.*

2. *Orthonormalize $\tilde{u}_i$ against $q_1, q_2, \ldots, q_{i-1}$ to get the vector q_i.*

With the above implementation, we may have to perform most of the computation in complex arithmetic even when A is real. Fortunately, when the matrix A is real, this can be avoided. In this case the Schur form is traditionally replaced by the quasi-Schur form, in which one still seeks for the factorization (4.2) but simply requires that the matrix R_j, be quasi-triangular, i.e. one allows for 2×2 diagonal blocks. In practice, if λ_{j+1} is complex, most algorithms do not compute the complex eigenvector y_{j+1} directly but rather deliver its real and imaginary parts y_R, y_I separately. Thus, the two eigenvectors $y_R \pm i y_I$ associated with the complex pair of conjugate eigenvalues $\lambda_{j+1}, \lambda_{j+2} = \bar{\lambda}_{j+1}$ are obtained at once.

Thinking in terms of bases of the invariant subspace instead of eigenvectors, we observe that the real and imaginary parts of the eigenvector generate the same subspace as the two conjugate eigenvectors and therefore we can work with these two real vectors instead of the (complex) eigenvectors. Hence if a complex pair occurs, all we have to do is orthogonalize the two vectors y_R, y_I against all previous q_i's and pursue the algorithm in the same way. The only difference is that the size of Q_j increases by two instead of just one in these instances.

4.2.5 Practical Deflation Procedures

To summarize, among all the possible deflation procedures we can use to compute the next pair λ_2, u_2, the following ones are the most useful in practice.

1. $v = w_1$ the left eigenvector. This has the disadvantage of requiring the left and right eigenvector. On the other hand both right and left eigenvectors of A_1 are preserved.

2. $v = u_1$ which is often nearly optimal and preserves the Schur vectors.

3. Use a block of Schur vectors instead of a single vector.

From the point of view of the implementation an important consideration is that we never need to form the matrix A_1 explicitly. This is important because in general A_1 will be a full matrix. In many algorithms for eigenvalue calculations, the only operation that is required is an operation of the form $y := A_1 x$. This operation can be performed as follows:

1. Compute the vector $y := Ax$;

2. Compute the scalar $t = \sigma\, v^H x$;

3. Compute $y := y - t\, u_1$.

The above procedure requires only that the vectors u_1, and v be kept in memory along with the matrix A. It is possible to deflate A_1 again into A_2 , and then into A_3 etc. At each step of the process we have

$$A_i = A_{i-1} - \sigma \tilde{u}_i v_i^H \ .$$

Here one only needs to save the vectors $\tilde{u}_i$ and v_i along with the matrix A. However, one should be careful about the usage of deflation in general. It should not be used to compute more than a few eigenvalues and eigenvcectors. This is especially true in the non Hermitian case because of the fact that the matrix A_i will accumulate errors from all previous computations and this could be disastrous if the currently computed eigenvalue is poorly conditioned.

4.3 General Projection Methods

Most eigenvalue algorithms employ in one way or another a projection technique. The projection process can be the body of the method itself or it might simply be used within a more complex algorithm to enhance its efficiency. A simple illustration of the necessity of resorting to a projection technique is when one uses the power method in the situation when the dominant eigenvalue is complex but the matrix A is real. Although the usual sequence $x_{j+1} = \alpha_j A x_j$ where α_j is a normalizing factor, does not converge a simple analysis shows that the subspace spanned by the last two iterates x_{j+1}, x_j will contain converging approximations

to the complex pair of eigenvectors. A simple projection technique onto those vectors will extract the desired eigenvalues and eigenvectors, see Exercise P-4.2 for details.

A projection method consists of approximating the exact eigenvector u, by a vector $\tilde{u}$ belonging to some subspace $\mathcal{K}$ referred to as the subspace of approximants or the right subspace, by imposing the so-called Petrov-Galerkin method that the residual vector of $\tilde{u}$ be orthogonal to some subspace $\mathcal{L}$, referred to as the left subspace. There are two broad classes of projection methods: orthogonal projection methods and oblique projection methods. In an orthogonal projection technique the subspace $\mathcal{L}$ is the same as $\mathcal{K}$. In an oblique projection method $\mathcal{L}$ is different from $\mathcal{K}$ and can be totally unrelated to it.

Not surprisingly, if no vector of the subspace $\mathcal{K}$ comes close to the exact eigenvector u, then it is impossible to get a good approximation $\tilde{u}$ to u from $\mathcal{K}$ and therefore the approximation obtained by any projection process based on $\mathcal{K}$ will be poor. If, on the other hand, there is some vector in $\mathcal{K}$ which is at a small distance ϵ from u then the question is: what accuracy can we expect to obtain? The purpose of this section is to try to answer this question.

4.3.1 Orthogonal Projection Methods

Let A be an $n \times n$ complex matrix and $\mathcal{K}$ be an m-dimensional subspace of $\mathbb{C}^n$. As a notational convention we will denote by the same symbol A the matrix and the linear application in $\mathbb{C}^n$ that it represents. We consider the eigenvalue problem: find u belonging to $\mathbb{C}^n$ and λ belonging to $\mathbb{C}$ such that

$$Au \;=\; \lambda u. \tag{4.16}$$

An orthogonal projection technique onto the subspace $\mathcal{K}$ seeks an approximate eigenpair $\tilde{\lambda}, \tilde{u}$ to the above problem, with $\tilde{\lambda}$ in $\mathbb{C}$ and $\tilde{u}$ in $\mathcal{K}$, such that the following Galerkin condition is satisfied:

$$A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K}\,, \tag{4.17}$$

or, equivalently,

$$(A\tilde{u} - \tilde{\lambda}\tilde{u}, v) \;=\; 0\,, \;\; \forall\, v \in \mathcal{K}. \tag{4.18}$$

Assume that some orthonormal basis $\{v_1, v_2, \ldots, v_m\}$ of $\mathcal{K}$ is available and denote by V the matrix with column vectors $v_1, v_2, \ldots, v_m$. Then we can solve the approximate problem numerically by translating it into this basis. Letting

$$\tilde{u} \;=\; Vy, \tag{4.19}$$

equation (4.19) becomes

$$(AVy - \tilde{\lambda}Vy, v_j) = 0, \quad j = 1, \ldots, m.$$

Therefore, y and $\tilde{\lambda}$ must satisfy

$$B_m y \;=\; \tilde{\lambda} y \tag{4.20}$$

with

$$B_m \;=\; V^H A V.$$

If we denote by A_m the linear transformation of rank m defined by $A_m = \mathcal{P}_{\mathcal{K}} A \mathcal{P}_{\mathcal{K}}$ then we observe that the restriction of this operator to the subspace $\mathcal{K}$ is represented by the matrix B_m with respect to the basis V. The following is a procedure for computing numerically the Galerkin approximations to the eigenvalues/eigenvectors of A known as the Rayleigh-Ritz procedure.

ALGORITHM **4.5 Rayleigh-Ritz Procedure:**

1. *Compute an orthonormal basis* $\{v_i\}_{i=1,\ldots,m}$ *of the subspace* $\mathcal{K}$. *Let* $V = [v_1, v_2, \ldots, v_m]$.
2. *Compute* $B_m = V^H A V$;
3. *Compute the eigenvalues of* B_m *and select the* k *desired ones* $\tilde{\lambda}_i, i = 1, 2, \ldots, k$, *where* $k \leq m$.
4. *Compute the eigenvectors* $y_i, i = 1, \ldots, k$, *of* B_m *associated with* $\tilde{\lambda}_i, i = 1, \ldots, k$, *and the corresponding approximate eigenvectors of* A, $\tilde{u}_i = V y_i$, $i = 1, \ldots, k$.

The above process only requires basic linear algebra computations. The numerical solution of the $m \times m$ eigenvalue problem in steps 3 and 4 can be treated by standard library subroutines such as those in EISPACK. Another important note is that in step 4 one can replace eigenvectors by Schur vectors to get approximate Schur vectors $\tilde{u}_i$ instead of approximate eigenvectors. Schur vectors y_i can be obtained in a numerically stable way and, in general, eigenvectors are more sensitive to rounding errors than are Schur vectors.

We can reformulate orthogonal projection methods in terms of projection operators as follows. Defining $\mathcal{P}_{\mathcal{K}}$ to be the orthogonal projector onto the subspace $\mathcal{K}$, then the Galerkin condition (4.17) can be rewritten as

$$\mathcal{P}_{\mathcal{K}}(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0\ ,\ \tilde{\lambda} \in \mathbb{C}\ ,\ \tilde{u} \in \mathcal{K}$$

or,

$$\mathcal{P}_{\mathcal{K}} A \tilde{u} = \tilde{\lambda}\tilde{u}\ , \quad \tilde{\lambda} \in \mathbb{C}\ ,\ \tilde{u} \in \mathcal{K}\ . \tag{4.21}$$

Note that we have replaced the original problem (4.16) by an eigenvalue problem for the linear transformation $\mathcal{P}_{\mathcal{K}} A_{|\mathcal{K}}$ which is from $\mathcal{K}$ to $\mathcal{K}$. Another formulation of the above equation is

$$\mathcal{P}_{\mathcal{K}} A \mathcal{P}_{\mathcal{K}} \tilde{u} = \tilde{\lambda}\tilde{u}\ ,\ \tilde{\lambda} \in \mathbb{C}\ ,\ \tilde{u} \in \mathbb{C}^n \tag{4.22}$$

which involves the natural extension

$$A_m = \mathcal{P}_{\mathcal{K}} A \mathcal{P}_{\mathcal{K}}$$

of the linear operator $A'_m = \mathcal{P}_{\mathcal{K}} A_{|\mathcal{K}}$ to the whole space. In addition to the eigenvalues and eigenvectors of A'_m, A_m has zero as a trivial eigenvalue with every vector of the orthogonal complement of $\mathcal{K}$, being an eigenvector. Equation (4.21) will be referred to as the Galerkin approximate problem.

The following proposition examines what happens in the particular case when the subspace $\mathcal{K}$ is invariant under A.

Proposition 4.3 *If $\mathcal{K}$ is invariant under A then every approximate eigenvalue / (right) eigenvector pair obtained from the orthogonal projection method onto $\mathcal{K}$ is exact.*

Proof. An approximate eigenpair $\tilde{\lambda}, \tilde{u}$ is defined by

$$\mathcal{P}_{\mathcal{K}}(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0 \,,$$

where $\tilde{u}$ is a nonzero vector in $\mathcal{K}$ and $\tilde{\lambda} \in \mathbb{C}$. If $\mathcal{K}$ is invariant under A then $A\tilde{u}$ belongs to $\mathcal{K}$ and therefore $\mathcal{P}_{\mathcal{K}} A\tilde{u} = A\tilde{u}$. Then the above equation becomes

$$A\tilde{u} - \tilde{\lambda}\tilde{u} = 0 \,,$$

showing that the pair $\tilde{\lambda}, \tilde{u}$ is exact. □

An important quantity for the convergence properties of projection methods is the distance $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ of the exact eigenvector u, supposed of norm 1, from the subspace $\mathcal{K}$. This quantity plays a key role in the analysis of projection methods. First, it is clear that the eigenvector u cannot be well approximated from $\mathcal{K}$ if $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ is not small because we have

$$\|\tilde{u} - u\|_2 \geq \|(I - \mathcal{P}_{\mathcal{K}})u\|_2.$$

The fundamental quantity $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ can also be interpreted as the sine of the acute angle between the eigenvector u and the subspace $\mathcal{K}$. It is also the gap between the space $\mathcal{K}$ and the linear span of u. The following theorem establishes an upper bound for the residual norm of the *exact* eigenpair with respect to the approximate operator A_m, using this angle.

Theorem 4.3 *Let $\gamma = \|\mathcal{P}_{\mathcal{K}} A(I - \mathcal{P}_{\mathcal{K}})\|_2$. Then the residual norms of the pairs $\lambda, \mathcal{P}_{\mathcal{K}} u$ and λ, u for the linear operator A_m satisfy respectively*

$$\|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2 \leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \tag{4.23}$$

$$\|(A_m - \lambda I)u\|_2 \leq \sqrt{\lambda^2 + \gamma^2}\, \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \,. \tag{4.24}$$

Proof. For the first inequality we use the definition of A_m to get

$$\begin{aligned}
\|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2 &= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(u - (I - \mathcal{P}_{\mathcal{K}})u)\|_2 \\
&= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u\|_2 \\
&= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})(I - \mathcal{P}_{\mathcal{K}})u\|_2 \\
&\leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \,.
\end{aligned}$$

As for the second inequality we simply notice that

$$\begin{aligned}(A_m - \lambda I)u &= (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u + (A_m - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u \\ &= (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u - \lambda(I - \mathcal{P}_{\mathcal{K}})u\,.\end{aligned}$$

Using the previous inequality and the fact that the two vectors on the right hand side are orthogonal to each other we get

$$\begin{aligned}\|(A_m - \lambda I)u\|_2^2 &= \|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2^2 + |\lambda|^2\|(I - \mathcal{P}_{\mathcal{K}})u\|_2^2 \\ &\leq (\gamma^2 + |\lambda|^2)\|(I - \mathcal{P}_{\mathcal{K}})u\|_2^2\end{aligned}$$

which completes the proof. □

Note that γ is bounded from above by $\|A\|_2$. A good approximation can therefore be achieved by the projection method in case the distance $\|(I-\mathcal{P}_{\mathcal{K}})u\|_2$ is small, provided the approximate eigenproblem is well conditioned. Unfortunately, in contrast with the Hermitian case the fact that the residual norm is small does not in any way guarantee that the eigenpair is accurate, because of potential difficulties related to the conditioning of the eigenvalue.

If we translate the inequality (4.23) into matrix form by expressing everything in an orthonormal basis V of $\mathcal{K}$, we would write $\mathcal{P}_{\mathcal{K}} = VV^H$ and immediately obtain

$$\|(V^H AV - \lambda I)V^H u\|_2 \leq \gamma\|(I - VV^H)u\|_2,$$

which shows that λ can be considered as an approximate eigenvalue for $B_m = V^H AV$ with residual of the order of $(I - \mathcal{P}_{\mathcal{K}})u$. If we scale the vector $V^H u$ to make it of 2-norm unity, and denote the result by y_u we can rewrite the above equality as

$$\|(V^H AV - \lambda I)y_u\|_2 \leq \gamma\frac{\|(I - \mathcal{P}_{\mathcal{K}})u\|_2}{\|\mathcal{P}_{\mathcal{K}} u\|_2} \equiv \gamma\tan\theta(u, \mathcal{K}).$$

The above inequality gives a more explicit relation between the residual norm and the angle between u and the subspace $\mathcal{K}$.

4.3.2 The Hermitian Case

The approximate eigenvalues computed from orthogonal projection methods in the particular case where the matrix A is Hermitian, satisfy strong optimality properties which follow from the Min-Max principle and the Courant characterization seen in Chapter 1. These properties follow by observing that $(A_m x, x)$ is the same as (Ax, x) when x runs in the subspace $\mathcal{K}$. Thus, if we label the eigenvalues decreasingly, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, we have

$$\begin{aligned}\tilde{\lambda}_1 &= \max_{x\in\mathcal{K}, x\neq 0}\frac{(\mathcal{P}_{\mathcal{K}}\, A\,\mathcal{P}_{\mathcal{K}} x, x)}{(x, x)} = \max_{x\in\mathcal{K}, x\neq 0}\frac{(\mathcal{P}_{\mathcal{K}} Ax, \mathcal{P}_{\mathcal{K}} x)}{(x, x)} \\ &= \max_{x\in\mathcal{K},\ x\neq 0}\frac{(Ax, x)}{(x, x)} \qquad (4.25)\end{aligned}$$

This is because $\mathcal{P}_{\mathcal{K}} x = x$ for any element in $\mathcal{K}$. Similarly, we can show that

$$\tilde{\lambda}_m = \min_{x \in \mathcal{K}, x \neq 0} \frac{(Ax, x)}{(x, x)}.$$

More generally, we have the following result.

Proposition 4.4 *The $i-$th largest approximate eigenvalue of a Hermitian matrix A, obtained from an orthogonal projection method onto a subspace $\mathcal{K}$, satisfies,*

$$\tilde{\lambda}_i = \max_{\substack{S \subseteq \mathcal{K} \\ dim(S) = i}} \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} . \tag{4.26}$$

As an immediate consequence we obtain the following corollary.

Corollary 4.1 *For $i = 1, 2, \ldots, m$ the following inequality holds*

$$\lambda_i \geq \tilde{\lambda}_i . \tag{4.27}$$

Proof. This is because,

$$\tilde{\lambda}_i = \max_{\substack{S \subseteq \mathcal{K} \\ \dim(S) = i}} \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} \leq \max_{\substack{S \subseteq \mathbb{C}^n \\ dim(S) = i}} \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} = \lambda_i .$$

□

A similar argument based on the Courant characterization results in the following theorem.

Theorem 4.4 *The approximate eigenvalue $\tilde{\lambda}_i$ and the corresponding eigenvector $\tilde{u}_i$ are such that*

$$\tilde{\lambda}_1 = \frac{(A\tilde{u}_1, \tilde{u}_1)}{(\tilde{u}_1, \tilde{u}_1)} = \max_{x \in \mathcal{K}, x \neq 0} \frac{(Ax, x)}{(x, x)} .$$

and for $i > 1$:

$$\tilde{\lambda}_i = \frac{(A\tilde{u}_i, \tilde{u}_i)}{(\tilde{u}_i, \tilde{u}_i)} = \max_{\substack{x \in \mathcal{K}, x \neq 0, \\ \tilde{u}_1^H x = \ldots = \tilde{u}_{i-1}^H x = 0}} \frac{(Ax, x)}{(x, x)} \tag{4.28}$$

One may suspect that the general bounds seen earlier for non-Hermitian matrices may be improved for the Hermitian case. This is indeed the case. We begin by proving the following lemma.

Lemma 4.1 *Let A be a Hermitian matrix and u an eigenvector of A associated with the eigenvalue λ. Then the Rayleigh quotient $\mu \equiv \mu_A(\mathcal{P}_{\mathcal{K}} u)$ satisfies the inequality*

$$|\lambda - \mu| \leq \|A - \lambda I\| \frac{\|(I - \mathcal{P}_{\mathcal{K}})u\|_2^2}{\|\mathcal{P}_{\mathcal{K}} u\|_2^2} . \tag{4.29}$$

Proof. From the equality

$$(A-\lambda I)\mathcal{P}_{\mathcal{K}}u = (A-\lambda I)(u-(I-\mathcal{P}_{\mathcal{K}})u) = -(A-\lambda I)(I-\mathcal{P}_{\mathcal{K}})u$$

and the fact that A is Hermitian we get,

$$\begin{aligned} |\lambda-\mu| &= |\frac{((A-\lambda I)\mathcal{P}_{\mathcal{K}}u, \mathcal{P}_{\mathcal{K}}u)}{(\mathcal{P}_{\mathcal{K}}u, \mathcal{P}_{\mathcal{K}}u)}| \\ &= |\frac{((A-\lambda I)(I-\mathcal{P}_{\mathcal{K}})u, (I-\mathcal{P}_{\mathcal{K}})u)}{(\mathcal{P}_{\mathcal{K}}u, \mathcal{P}_{\mathcal{K}}u)}| \quad . \end{aligned}$$

The result follows from a direct application of the Cauchy-Schwarz inequality □

Assuming as usual that the eigenvalues are labeled decreasingly, and letting $\mu_1 = \mu_A(\mathcal{P}_{\mathcal{K}}u_1)$, we can get from (4.25) that

$$0 \leq \lambda_1 - \tilde{\lambda}_1 \leq \lambda_1 - \mu_1 \leq \|A-\lambda_1 I\|_2 \frac{\|(I-\mathcal{P}_{\mathcal{K}})u_1\|_2^2}{\|\mathcal{P}_{\mathcal{K}}u_1\|_2^2} \; .$$

A similar result can be shown for the smallest eigenvalue. We can extend this inequality to the other eigenvalues at the price of a little complication in the equations. In what follows we will denote by $\tilde{Q}_i$ the sum of the spectral projectors associated with the approximate eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{i-1}$. For any given vector x, $(I-\tilde{Q}_i)x$ will be the vector obtained by orthogonalizing x against the first $i-1$ approximate eigenvectors. We consider a candidate vector of the form $(I-\tilde{Q}_i)\mathcal{P}_{\mathcal{K}}u_i$ in an attempt to use an argument similar to the one for the largest eigenvalue. This is a vector obtained by projecting u_i onto the subspace $\mathcal{K}$ and then stripping it off its components in the first $i-1$ approximate eigenvectors.

Lemma 4.2 *Let $\tilde{Q}_i$ be the sum of the spectral projectors associated with the approximate eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{i-1}$ and define $\mu_i = \mu_A(x_i)$, where*

$$x_i = \frac{(I-\tilde{Q}_i)\mathcal{P}_{\mathcal{K}}u_i}{\|(I-\tilde{Q}_i)\mathcal{P}_{\mathcal{K}}u_i\|_2} \quad .$$

Then

$$|\lambda_i - \mu_i| \leq \|A-\lambda_i I\|_2 \; \frac{\|\tilde{Q}_i u_i\|_2^2 + \|(I-\mathcal{P}_{\mathcal{K}})u_i\|_2^2}{\|(I-\tilde{Q}_i)\mathcal{P}_{\mathcal{K}}u_i\|_2^2} \quad . \tag{4.30}$$

Proof. To simplify notation we set $\alpha = 1/\|(I-\tilde{Q}_i)\mathcal{P}_{\mathcal{K}}u_i\|_2$. Then we write,

$$(A-\lambda_i I)x_i = (A-\lambda_i I)(x_i - \alpha u_i) \; ,$$

and proceed as in the previous case to get,

$$|\lambda_i - \mu_i| = |((A-\lambda_i I)x_i, x_i))| = |((A-\lambda_i I)(x_i-\alpha u_i), (x_i - \alpha u_i))| \; .$$

Applying the Cauchy-Schwarz inequality to the above equation, we get

$$|\lambda_i - \mu_i| = \|A - \lambda_i I\|_2 \|x_i - \alpha u_i\|_2^2 \ .$$

We can rewrite $\|x_i - \alpha u_i\|_2^2$ as

$$\begin{aligned} \|x_i - \alpha u_i\|_2^2 &= \alpha^2 \|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i - u_i\|_2^2 \\ &= \alpha^2 \|(I - \tilde{Q}_i)(\mathcal{P}_{\mathcal{K}} u_i - u_i) - \tilde{Q}_i u_i\|_2^2 \ . \end{aligned}$$

Using the orthogonality of the two vectors inside the norm bars, this equality becomes

$$\begin{aligned} \|x_i - \alpha u_i\|_2^2 &= \alpha^2 \left(\|(I - \tilde{Q}_i)(\mathcal{P}_{\mathcal{K}} u_i - u_i)\|_2^2 + \|\tilde{Q}_i u_i\|_2^2 \right) \\ &\leq \alpha^2 \left(\|(I - \mathcal{P}_{\mathcal{K}})u_i\|_2^2 + \|\tilde{Q}_i u_i\|_2^2 \right) \ . \end{aligned}$$

This establishes the desired result. □

The vector x_i has been constructed in such a way that it is orthogonal to all previous approximate eigenvectors $\tilde{u}_1, \ldots, \tilde{u}_{i-1}$. We can therefore exploit the Courant characterization (4.28) to prove the following result.

Theorem 4.5 *Let $\tilde{Q}_i$ be the sum of the spectral projectors associated with the approximate eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{i-1}$. Then the error between the i-th exact and approximate eigenvalues λ_i and $\tilde{\lambda}_i$ is such that*

$$0 \leq \lambda_i - \tilde{\lambda}_i \leq \|A - \lambda_i I\|_2 \frac{\|\tilde{Q}_i u_i\|_2^2 + \|(I - \mathcal{P}_{\mathcal{K}})u_i\|_2^2}{\|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i\|_2^2} \ . \tag{4.31}$$

Proof. By (4.28) and the fact that x_i belongs to $\mathcal{K}$ and is orthogonal to the first $i-1$ approximate eigenvectors we immediately get

$$0 \leq \lambda_i - \tilde{\lambda}_i \leq \lambda_i - \mu_i.$$

The result follows from the previous lemma. □

We point out that the above result is valid for $i = 1$, provided we define $\tilde{Q}_1 = 0$. The quantities $\|\tilde{Q}_i u_i\|_2$ represent the cosines of the acute angle between u_i and the span of the previous approximate eigenvectors. In the ideal situation this should be zero. In addition, we should mention that the error bound is semi-a-priori, since it will require the knowledge of previous eigenvectors in order to get an idea of the quantity $\|\tilde{Q}_i u_i\|_2$.

We now turn our attention to the eigenvectors.

Theorem 4.6 *Let $\gamma = \|\mathcal{P}_{\mathcal{K}} A(I - \mathcal{P}_{\mathcal{K}})\|_2$, and consider any eigenvalue λ of A with associated eigenvector u. Let $\tilde{\lambda}$ be the approximate eigenvalue closest to λ and δ the distance between λ and the set of approximate eigenvalues other than $\tilde{\lambda}$. Then there exists an approximate eigenvector $\tilde{u}$ associated with $\tilde{\lambda}$ such that*

$$\sin\left[\theta(u, \tilde{u})\right] \leq \sqrt{1 + \frac{\gamma^2}{\delta^2}} \ \sin\left[\theta(u, \mathcal{K})\right] \tag{4.32}$$

Proof.

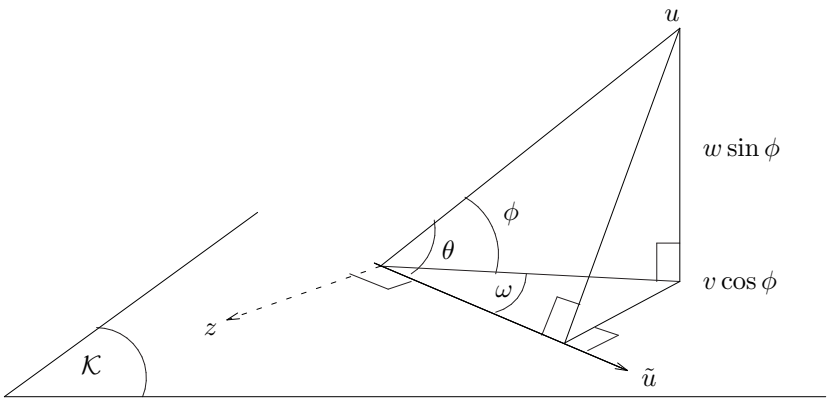


Figure 4.1: Projections of the eigenvector u onto $\mathcal{K}$ and then onto $\tilde{u}$.

Let us define the two vectors

$$v = \frac{\mathcal{P}_{\mathcal{K}} u}{\|\mathcal{P}_{\mathcal{K}} u\|_2} \quad \text{and} \quad w = \frac{(I - \mathcal{P}_{\mathcal{K}})u}{\|(I - \mathcal{P}_{\mathcal{K}})u\|_2} \tag{4.33}$$

and denote by ϕ the angle between u and $\mathcal{P}_{\mathcal{K}} u$, as defined by $\cos \phi = \|\mathcal{P}_{\mathcal{K}} u\|_2$. Then, clearly

$$u = v \cos \phi + w \sin \phi,$$

which, upon multiplying both sides by $(A - \lambda I)$ leads to

$$(A - \lambda I)v \ \cos \phi + (A - \lambda I)w \ \sin \phi = 0 \ .$$

We now project both sides onto $\mathcal{K}$, and take the norms of the resulting vector to obtain

$$\|\mathcal{P}_{\mathcal{K}}(A - \lambda I)v\|_2 \ \cos \phi = \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)w\|_2 \ \sin \phi \ . \tag{4.34}$$

For the-right-hand side note that

$$\begin{aligned} \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)w\|_2 &= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})w\|_2 \\ &= \|\mathcal{P}_{\mathcal{K}} A(I - \mathcal{P}_{\mathcal{K}})w\|_2 \leq \gamma \ . \end{aligned} \tag{4.35}$$

For the left-hand-side, we decompose v further as

$$v = \tilde{u} \ \cos \omega + z \ \sin \omega,$$

in which $\tilde{u}$ is a unit vector from the eigenspace associated with $\tilde{\lambda}$, z is a unit vector in $\mathcal{K}$ that is orthogonal to $\tilde{u}$, and ω is the acute angle between v and $\tilde{u}$. We then obtain,

$$\begin{aligned} \mathcal{P}_{\mathcal{K}}(A - \lambda I)v &= \mathcal{P}_{\mathcal{K}}(A - \lambda I)[\cos \omega \tilde{u} + \sin \omega z] \\ &= \tilde{u}(\tilde{\lambda} - \lambda) \cos \omega + \mathcal{P}_{\mathcal{K}}(A - \lambda I)z \sin \omega. \end{aligned} \tag{4.36}$$

The eigenvalues of the restriction of $\mathcal{P}_{\mathcal{K}}(A-\lambda I)$ to the orthogonal of $\tilde{u}$ are $\tilde{\lambda}_j - \lambda$, for $j = 1, 2, \ldots m$, and $\tilde{\lambda}_j \neq \tilde{\lambda}$. Therefore, since z is orthogonal to $\tilde{u}$, we have

$$\|\mathcal{P}_{\mathcal{K}}(A - \lambda I)z\|_2 \geq \delta > 0. \tag{4.37}$$

The two vectors in the right hand side of (4.36) are orthogonal and by (4.37),

$$\begin{aligned} \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)v\|_2^2 &= |\tilde{\lambda} - \lambda|^2 \cos^2\omega + \sin^2\omega \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)z\|_2^2 \\ &\geq \delta^2 \sin^2\omega \end{aligned} \tag{4.38}$$

To complete the proof we refer to Figure 4.1. The projection of u onto $\tilde{u}$ is the projection onto $\tilde{u}$ of the projection of u onto $\mathcal{K}$. Its length is $\cos\phi\cos\omega$ and as a result the sine of the angle θ between u and $\tilde{u}$ is given by

$$\begin{aligned} \sin^2\theta &= 1 - \cos^2\phi \cos^2\omega \\ &= 1 - \cos^2\phi (1 - \sin^2\omega) \\ &= \sin^2\phi + \sin^2\omega \cos^2\phi . \end{aligned} \tag{4.39}$$

Combining (4.34), (4.35), (4.38) we obtain that

$$\sin\omega \cos\phi \leq \frac{\gamma}{\delta} \sin\phi$$

which together with (4.39) yields the desired result. □

This is a rather remarkable result given that it is so general. It tells us among other things that the only condition we need in order to guarantee that a projection method will deliver a good approximation in the Hermitian case is that the angle between the exact eigenvector and the subspace $\mathcal{K}$ be sufficiently small.

As a consequence of the above result we can establish bounds on eigenvalues that are somewhat simpler than those of Theorem 4.5. This results from the following proposition.

Proposition 4.5 *The eigenvalues λ and $\tilde{\lambda}$ in Theorem 4.6 are such that*

$$|\lambda - \tilde{\lambda}| \leq \|A - \lambda I\|_2 \sin^2\theta(u, \tilde{u}) . \tag{4.40}$$

Proof. We start with the simple observation that $\tilde{\lambda} - \lambda = ((A - \lambda I)\tilde{u}, \tilde{u})$. Letting $\alpha = (u, \tilde{u}) = \cos\theta(u, \tilde{u})$ we can write

$$\tilde{\lambda} - \lambda = ((A - \lambda I)(\tilde{u} - \alpha u), \tilde{u}) = ((A - \lambda I)(\tilde{u} - \alpha u), \tilde{u} - \alpha u)$$

The result follows immediatly by taking absolute values, exploiting the Cauchy-Schwarz inequality, and observing that $\|\tilde{u} - \alpha u\|_2 = \sin\theta(u, \tilde{u})$. □

4.3.3 Oblique Projection Methods

In an oblique projection method we are given two subspaces $\mathcal{L}$ and $\mathcal{K}$ and seek an approximation $\tilde{u} \in \mathcal{K}$ and an element $\tilde{\lambda}$ of $\mathbb{C}$ that satisfy the Petrov-Galerkin condition,

$$((A - \tilde{\lambda} I)\tilde{u}, v) = 0 \quad \forall\, v \in \mathcal{L}\,. \tag{4.41}$$

The subspace $\mathcal{K}$ will be referred to as the right subspace and $\mathcal{L}$ as the left subspace. A procedure similar to the Rayleigh-Ritz procedure can be devised by again translating in matrix form the approximate eigenvector $\tilde{u}$ in some basis and expressing the Petrov-Galerkin condition (4.41). This time we will need two bases, one which we denote by V for the subspace $\mathcal{K}$ and the other, denoted by W, for the subspace $\mathcal{L}$. We assume that these two bases are biorthogonal, i.e., that $(v_i, w_j) = \delta_{ij}$, or

$$W^H V \;=\; I$$

where I is the identity matrix. Then, writing $\tilde{u} = Vy$ as before, the above Petrov-Galerkin condition yields the same approximate problem as (4.20) except that the matrix B_m is now defined by

$$B_m \;=\; W^H A V.$$

We should however emphasize that in order for a biorthogonal pair V, W to exist the following additional assumption for $\mathcal{L}$ and $\mathcal{K}$ must hold.

For any two bases V and W of $\mathcal{K}$ and $\mathcal{L}$ respectively,

$$\det(W^H V) \neq 0 \quad . \tag{4.42}$$

In order to interpret the above condition in terms of operators we will define the oblique projector $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ onto $\mathcal{K}$ and orthogonal to $\mathcal{L}$. For any given vector x in $\mathbb{C}^n$, the vector $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x$ is defined by

$$\begin{cases} \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x \in \mathcal{K} \\ x - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x \perp \mathcal{L}. \end{cases}$$

Note that the vector $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x$ is uniquely defined under the assumption that no vector of the subspace $\mathcal{L}$ is orthogonal to $\mathcal{K}$. This fundamental assumption can be seen to be equivalent to assumption (4.42). When it holds the Petrov-Galerin condition (4.18) can be rewritten as

$$\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A\tilde{u} - \tilde{\lambda}\tilde{u}) \;=\; 0 \tag{4.43}$$

or

$$\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\tilde{u} \;=\; \tilde{\lambda}\tilde{u}\,.$$

Thus, the eigenvalues of the matrix A are approximated by those of $A' = \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A_{|\mathcal{K}}$. We can define an extension A_m of A'_m analogous to the one defined in the previous section, in many different ways. For example introducing $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ before the

occurrences of $\tilde{u}$ in the above equation would lead to $A_m = \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$. In order to be able to utilize the distance $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ in a-priori error bounds a more useful extension is

$$A_m \;=\; \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A \mathcal{P}_{\mathcal{K}} \;.$$

With this notation, it is trivial to extend the proof of Proposition 4.3 to the oblique projection case. In other words, when $\mathcal{K}$ is invariant, then no matter which left subspace $\mathcal{L}$ we choose, the oblique projection method will always extract exact eigenpairs.

We can establish the following theorem which generalizes Theorem 4.3 seen for the orthogonal projection case.

Theorem 4.7 *Let $\gamma = \|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})\|_2$. Then the following two inequalities hold:*

$$\|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2 \leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \tag{4.44}$$

$$\|(A_m - \lambda I)u\|_2 \leq \sqrt{|\lambda|^2 + \gamma^2}\, \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \;. \tag{4.45}$$

Proof. For the first inequality, since the vector $\mathcal{P}_{\mathcal{K}} y$ belongs to $\mathcal{K}$ we have $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} \mathcal{P}_{\mathcal{K}} = \mathcal{P}_{\mathcal{K}}$ and therefore

$$\begin{aligned}
(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u &= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A - \lambda I)\mathcal{P}_{\mathcal{K}} u \\
&= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A - \lambda I)(\mathcal{P}_{\mathcal{K}} u - u) \\
&= -\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u \;.
\end{aligned}$$

Since $(I - \mathcal{P}_{\mathcal{K}})$ is a projector we now have

$$(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u = -\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})(I - \mathcal{P}_{\mathcal{K}})u.$$

Taking Euclidean norms of both sides and using the Cauchy-Schwarz inequality we immediately obtain the first result.

For the second inequality, we write

$$\begin{aligned}
(A_m - \lambda I)u &= (A_m - \lambda I)\left[\mathcal{P}_{\mathcal{K}} u + (I - \mathcal{P}_{\mathcal{K}})u\right] \\
&= (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u + (A_m - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u \;.
\end{aligned}$$

Noticing that $A_m(I - \mathcal{P}_{\mathcal{K}}) = 0$ this becomes

$$(A_m - \lambda I)u = (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u - \lambda(I - \mathcal{P}_{\mathcal{K}})u \;.$$

Using the orthogonality of the two terms in the right hand side, and taking the Euclidean norms we get the second result. □

In the particular case of orthogonal projection methods, $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ is identical with $\mathcal{P}_{\mathcal{K}}$, and we have $\|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}\|_2 = 1$. Moreover, the term γ can then be bounded from above by $\|A\|_2$. It may seem that since we obtain very similar error bounds for both the orthogonal and the oblique projection methods, we are likely to obtain similar errors when we use the same subspace. This is not the case in general.

One reason is that the scalar γ can no longer be bounded by $\|A\|_2$ since we have $\|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}\|_2 \geq 1$ and $\|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}\|_2$ is unknown in general. In fact the constant γ can be quite large. Another reason which was pointed out earlier is that residual norm does not provide enough information. The approximate problem can have a much worse condition number if non-orthogonal transformations are used, which may lead to poorer results. This however is only based on intuition as there are no rigorous results in this direction.

The question arises as to whether there is any need for oblique projection methods since dealing with oblique projectors may be numerically unsafe. Methods based on oblique projectors can offer some advantages. In particular they may allow to compute approximations to left as well as right eigenvectors simultaneously. There are methods based on oblique projection techniques that require also far less storage than similar orthogonal projections methods. This will be illustrated in Chapter 4.

4.4 Chebyshev Polynomials

Chebyshev polynomials are crucial in the study of the Lanczos algorithm and more generally of iterative methods in numerical linear algebra, such as the conjugate gradient method. They are useful both in theory, when studying convergence, and in practice, as a means of accelerating single vector iterations or projection processes.

4.4.1 Real Chebyshev Polynomials

The Chebyshev polynomial of the first kind of degree k is defined by

$$C_k(t) = \cos[k\, \cos^{-1}(t)] \quad \text{for} \quad -1 \leq t \leq 1\,. \tag{4.46}$$

That this is a polynomial with respect to t can be easily shown by induction from the trigonometric relation

$$\cos[(k+1)\theta] + \cos[(k-1)\theta] = 2\cos\theta\cos k\theta,$$

and the fact that $C_1(t) = t, C_0(t) = 1$. Incidentally, this also shows the important three-term recurrence relation

$$C_{k+1}(t) = 2\,tC_k(t) - C_{k-1}(t)\,.$$

It is important to extend the definition (4.46) to cases where $|t| > 1$ which is done with the following formula,

$$C_k(t) = \cosh\,[k\cosh^{-1}(t)], \quad |t| \geq 1\,. \tag{4.47}$$

This is readily seen by passing to complex variables and using the definition $\cos\theta = (e^{i\theta} + e^{-i\theta})/2$. As a result of (4.47) we can derive the expression,

$$C_k(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2-1}\right)^k + \left(t + \sqrt{t^2-1}\right)^{-k}\right]\,, \tag{4.48}$$

which is valid for $|t| \geq 1$ but can also be extended to the case $|t|<1$. As a result, one may use the following approximation for large values of k

$$C_k(t) \quad \begin{matrix} \geq \\ \sim \end{matrix} \quad \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^k \quad \text{for} \quad |t| \geq 1 \,. \tag{4.49}$$

In what follows we denote by $\mathbb{P}_k$ the set of all polynomials of degree k. An important result from approximation theory, which we state without proof, is the following theorem.

Theorem 4.8 *Let $[\alpha, \beta]$ be a non-empty interval in $\mathbb{R}$ and let γ be any real scalar such with $\gamma \geq \beta$. Then the minimum*

$$\min_{p \in \mathbb{P}_k, p(\gamma)=1} \max_{t \in [\alpha,\beta]} |p(t)|$$

is reached by the polynomial

$$\hat{C}_k(t) \equiv \frac{C_k\left(1 + 2\frac{t-\beta}{\beta-\alpha}\right)}{C_k\left(1 + 2\frac{\gamma-\beta}{\beta-\alpha}\right)} \,.$$

For a proof see [26]. The maximum of C_k for t in $[-1, 1]$ is 1 and as a corollary we have

$$\min_{p \in \mathbb{P}_k,\ p(\gamma)=1} \max_{t \in [\alpha,\beta]} |p(t)| = \frac{1}{|C_k(1 + 2\frac{\gamma-\beta}{\beta-\alpha})|} = \frac{1}{|C_k(2\frac{\gamma-\mu}{\beta-\alpha})|} \,.$$

in which $\mu \equiv (\alpha + \beta)/2$ is the middle of the interval. Clearly, the results can be slightly modified to hold for the case where $\gamma \leq \alpha$, i.e., when γ is to the left of the interval.

4.4.2 Complex Chebyshev Polynomials

The standard definition given in the previous section for Chebyshev polynomials of the first kind, see equation (4.46), extends without difficulty to complex variables. First, as was seen before, when t is real and $|t| > 1$ we can use the alternative definition, $C_k(t) = \cosh[k \cosh^{-1}(t)], \quad 1 \leq |t|$. More generally, one can unify these definitions by switching to complex variables and writing

$$C_k(z) = \cosh(k\zeta), \quad \text{where} \quad \cosh(\zeta) = z \,.$$

Defining the variable $w = e^{\zeta}$, the above formula is equivalent to

$$C_k(z) = \frac{1}{2}[w^k + w^{-k}] \quad \text{where} \quad z = \frac{1}{2}[w + w^{-1}]. \tag{4.50}$$

We will use the above definition for Chebyshev polynomials in $\mathbb{C}$. Note that the equation $\frac{1}{2}(w + w^{-1}) = z$ has two solutions w which are inverses of each other, and as a result the value of $C_k(z)$ does not depend on which of these solutions is

chosen. It can be verified directly that the C_k's defined by the above equations are indeed polynomials in the z variable and that they satisfy the three term recurrence

$$C_{k+1}(z) = 2\,zC_k(z) - C_{k-1}(z), \tag{4.51}$$

with $C_0(z) \equiv 1$ and $C_1(z) \equiv z$.

As is now explained, Chebyshev polynomials are intimately related to ellipses in the complex plane. Let C_ρ be the circle of center the origin and radius ρ. Then the so-called Joukowski mapping

$$J(w) = \frac{1}{2}[w + w^{-1}]$$

transforms C_ρ into an ellipse of center the origin, foci $-1, 1$ and major semi-axis $\frac{1}{2}[\rho + \rho^{-1}]$ and minor semi-axis $\frac{1}{2}|\rho - \rho^{-1}|$. This is illustrated in Figure 4.2.

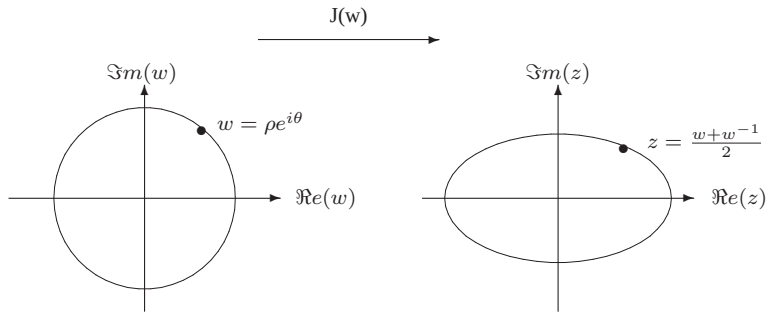


Figure 4.2: The Joukowski mapping transforms a circle into an ellipse in the complex plane.

There are two circles which have the same image by the mapping $J(w)$, one with the radius ρ and the other with the radius ρ^{-1}. So it suffices to consider those circles with $\rho \geq 1$. Note that the case $\rho = 1$ is a degenerate case in which the ellipse $E(0, 1, -1)$ reduces the interval $[-1, 1]$ traveled through twice.

One important question we now ask is whether or not a min-max result similar to the one of Theorem 4.8 holds for the complex case. Here the maximum of $|p(z)|$ is taken over the ellipse boundary and γ is some point not enclosed by the ellipse. A 1963 paper by Clayton [29] was generally believed for quite some time to have established the result, at least for the special case where the ellipse has real foci and γ is real. It was recently shown by Fischer and Freund that in fact Clayton's result was incorrect in general [60]. On the other hand, Chebyshev polynomials are asymptotically optimal and in practice that is all that is needed.

To show the asymptotic optimality, we start by stating a lemma due to Zarantonello, which deals with the particular case where the ellipse reduces to a circle. This particular case is important in itself.

Lemma 4.3 (Zarantonello) *Let $C(0,\rho)$ be a circle of center the origin and radius ρ and let γ a point of $\mathbb{C}$ not enclosed by $C(0,\rho)$. Then,*

$$\min_{p\in\mathbb{P}_k,\ p(\gamma)=1} \max_{z\ \in\ C(0,\rho)} |p(z)| \ = \ \left(\frac{\rho}{|\gamma|}\right)^k \ , \tag{4.52}$$

the minimum being achieved for the polynomial $(z/\gamma)^k$.

Proof. See reference [162] for a proof. □

Note that by changing variables, shifting and rescaling the polynomial, we also get for any circle centered at c and for any scalar γ such that $|\gamma| > \rho$,

$$\min_{p\in\mathbb{P}_k\ \ p(\gamma)=1} \max_{z\ \in\ C(c,\rho)} |p(z)| \ = \ \left(\frac{\rho}{|\gamma - c|}\right)^k$$

We now consider the general case of an ellipse centered at the origin, with foci $1, -1$ and semi-major axis a, which can be considered as mapped by J from the circle $C(0,\rho)$, with the convention that $\rho \geq 1$. We denote by E_ρ such an ellipse.

Theorem 4.9 *Consider the ellipse E_ρ mapped from $C(0,\rho)$ by the mapping J and let γ any point in the complex plane not enclosed by it. Then*

$$\frac{\rho^k}{|w_\gamma|^k} \leq \min_{p\in\mathbb{P}_k\ p(\gamma)=1} \max_{z\ \in\ E_\rho} |p(z)| \leq \frac{\rho^k + \rho^{-k}}{|w_\gamma^k + w_\gamma^{-k}|} \tag{4.53}$$

in which w_γ is the dominant root of the equation $J(w) = \gamma$.

Proof. We start by showing the second inequality. Any polynomial p of degree k satisfying the constraint $p(\gamma) = 1$ can be written as,

$$p(z) = \frac{\sum_{j=0}^k \xi_j z^j}{\sum_{j=0}^k \xi_j \gamma^j} \quad .$$

A point z on the ellipse is transformed by J from a certain w in $C(0,\rho)$. Similarly, let w_γ be one of the two inverse transforms of γ by the mapping, namely the one with largest modulus. Then, p can be rewritten as

$$p(z) = \frac{\sum_{j=0}^k \xi_j (w^j + w^{-j})}{\sum_{j=0}^k \xi_j (w_\gamma^j + w_\gamma^{-j})} \ . \tag{4.54}$$

Consider the particular polynomial obtained by setting $\xi_k = 1$ and $\xi_j = 0$ for $j \neq k$,

$$p^*(z) = \frac{w^k + w^{-k}}{w_\gamma^k + w_\gamma^{-k}}$$

which is a scaled Chebyshev polynomial of the first kind of degree k in the variable z. It is not too difficult to see that the maximum modulus of this polynomial is reached in particular when $w = \rho e^{i\theta}$ is real, i.e., when $w = \rho$. Thus,

$$\max_{z \in E_\rho} |p^*(z)| = \frac{\rho^k + \rho^{-k}}{|w_\gamma^k + w_\gamma^{-k}|}$$

which proves the second inequality.

To prove the left inequality, we rewrite (4.54) as

$$p(z) = \left(\frac{w^{-k}}{w_\gamma^{-k}}\right) \quad \frac{\sum_{j=0}^k \xi_j (w^{k+j} + w^{k-j})}{\sum_{j=0}^k \xi_j (w_\gamma^{k+j} + w_\gamma^{k-j})}$$

and take the modulus of $p(z)$,

$$|p(z)| = \frac{\rho^{-k}}{|w_\gamma|^{-k}} \left| \frac{\sum_{j=0}^k \xi_j (w^{k+j} + w^{k-j})}{\sum_{j=0}^k \xi_j (w_\gamma^{k+j} + w_\gamma^{k-j})} \right| .$$

The polynomial of degree $2k$ in w inside the large modulus bars in the right-hand-side is such that its value at w_γ is one. By Lemma 4.3, the modulus of this polynomial over the circle $C(0, \rho)$ is not less than $(\rho/|w_\gamma|)^{2k}$, i.e., for any polynomial, satisfying the constraint $p(\gamma) = 1$ we have,

$$\max_{z \in\ E_\rho} |p(z)| \geq \frac{\rho^{-k}}{|w_\gamma|^{-k}} \frac{\rho^{2k}}{|w_\gamma|^{2k}} = \frac{\rho^k}{|w_\gamma|^k}.$$

This proves that the minimum over all such polynomials of the maximum modulus on the ellipse E_ρ is $\geq (\rho/|w_\gamma|)^k$. □

The difference between the left and right bounds in (4.53) tends to zero as k increases to infinity. Thus, the important point made by the theorem is that, for large k, the Chebyshev polynomial

$$p^*(z) = \frac{w^k + w^{-k}}{w_\gamma^k + w_\gamma^{-k}}, \quad \text{where} \quad z = \frac{w + w^{-1}}{2}$$

is close to the optimal polynomial. In other words these polynomials are *asymptotically* optimal.

For a more general ellipse centered at c, and with focal distance d, a simple change of variables shows that the near-best polynomial is given by

$$C_k \left(\frac{z - c}{d} \right).$$

We should point out that an alternative result, which is more complete, has been proven by Fischer and Freund in [59].

PROBLEMS

P-4.1 What are the eigenvalues and eigenvectors of $(A - \sigma I)^{-1}$. What are all the shifts σ that will lead to a convergence towards a given eigenvalue λ?

P-4.2 Consider a real nonsymmetric matrix A. The purpose of this exercise is to develop a generalization of the power method that can handle the case where the dominant eigenvalue is complex (i.e., we have a complex conjugate pair of dominant eigenvalues). Show that by a projection process onto two successive iterates of the power method one can achieve convergence towards the dominant pair of eigenvalues [Consider the diagonalizable case only]. Without giving a proof, state what the rate of convergence toward the pair of complex conjugate eigenvectors should be. Develop a simple version of a corresponding algorithm and then a variation of the algorithm that orthonormalizes two successive iterates at every step, i.e., starting with a vector x of 2-norm unity, the iterates are as follows,

$$x_{new} := \frac{\hat{x}}{\|\hat{x}\|_2} \quad \text{where} \quad \hat{x} := Ax_{old} - (Ax_{old}, x_{old})x_{old} \ .$$

Does the orthogonalization have to be done at every step?

P-4.3 By following a development similar to that subsection 4.2, find the v vector for Wielandt deflation, which minimizes the condition number for A_1, among all vectors in the span of u_1, w_1. Show again that the choice $v = u_1$ is nearly optimal when $\lambda_1 - \lambda_2$ is small relative to σ.

P-4.4 Consider the generalized eigenvalue problem $Ax = \lambda Bx$. How can one generalize the power method? The shifted power method? and the shift-and-invert power method?

P-4.5 Assume that all the eigenvalues of a matrix A are real and that one uses the shifted power method for computing the largest, i.e., the rightmost eigenvalue of a given matrix. What are all the admissible shifts, i.e., those that will lead to convergence toward the rightmost eigenvalue? Among all the admissible choices which one leads to the best convergence rate?

P-4.6 Consider a deflation technique which would compute the eigenvalues of the matrix

$$A_1 = (I - Q_j Q_j^H)A$$

in which $Q_j = [q_1, q_2, \ldots, q_j]$ are previously computed Schur vectors. What are the eigenvalues of the deflated matrix A_1? Show that an eigenvector of A_1 is a Schur vector for A. The advantage of this technique is that there is no need to select shifts σ_j. What are the disadvantages if any?

P-4.7 Show that in example 4.4 any linear combination of the vectors u_1 and w_1 is in fact optimal.

P-4.8 Nothing was said about the left eigenvector $\tilde{w}_1$ of the deflated matrix A_1 in Section 4.2. Assuming that the matrix A is diagonalizable find an eigenvector $\tilde{w}_1$ of A_1 associated with the eigenvalue $\lambda_1 - \sigma$. [$Hint$: Express the eigenvector in the basis of the left eigenvectors of A.] How can this be generalized to the situation where A is not diagonalizable?

P-4.9 Assume that the basis V of the subspace $\mathcal{K}$ used in an orthogonal projection process is not orthogonal. What matrix problem do we obtain if we translate the Galerkin

conditions using this basis. Same question for the oblique projection technique, i.e., assuming that V, W does not form a bi-orthogonal pair. Ignoring the cost of the small m-dimensional problems, how do the computational costs compare? What if we include the cost of the orthonormalization (by modified Gram-Schmidt) for the approach which uses orthogonal bases (Assuming that the basis V is obtained from orthonormalizing a set of m basis vectors).

P-4.10 Let A be Hermitian and let $\tilde{u}_i, \tilde{u}_j$ two Ritz eigenvectors associated with two different eigenvalues $\tilde{\lambda}_i, \tilde{\lambda}_j$ respectively. Show that $(A\tilde{u}_i, \tilde{u}_j) = \tilde{\lambda}_j \delta_{ij}$.

P-4.11 Prove from the definition (4.50) that the C_k's are indeed polynomials in z and that they satisfy the three-term recurrence (4.51).

NOTES AND REFERENCES. Much of the material on projection methods presented in this chapter is based on the papers [171, 168] and the section on deflation procedures is from [176] and some well-known results in Wilkinson [222]. Suggested additional reading on projection methods are Chatelin [22] and Krasnoselskii et al. [110]. A good discussion of Chebyshev polynomials in the complex plane is given in the book by Rivlin [162]. Deflation for non Hermitian eigenvalue problems is not that much used in the literature. I found Schur-Wielandt and related deflation procedures (based on Schur vectors rather than eigenvectors) to be essential in the design of robust eigenvalue algorithms. Theorem (4.6) has been extended to the nonsymmetric case by Stewart [203]. ■

Chapter 5

SUBSPACE ITERATION

Among the best known methods for solving large sparse eigenvalue problems, the subspace iteration algorithm is undoubtedly the simplest. This method can be viewed as a block generalization of the power method. Although the method is not competitive with other projections methods to be covered in later chapters, it still is one of the most important methods used in structural engineering. It also constitutes a good illustration of the material covered in the previous chapter.

5.1 Simple Subspace Iteration

The original version of subspace iteration was introduced by Bauer under the name of *Treppeniteration* (staircase iteration). Bauer's method consists of starting with an initial system of m vectors forming an $n \times m$ matrix $X_0 = [x_1, \ldots, x_m]$ and computing the matrix

$$X_k = A^k X_0. \tag{5.1}$$

for a certain power k. If we normalized the column vectors separately in the same manner as for the power method, then in typical cases each of these vectors will converge to the same eigenvector associated with the dominant eigenvalue. Thus the system X_k will progressively loose its linear independence. The idea of Bauer's method is to reestablish linear independence for these vectors by a process such as the LR or the QR factorization. Thus, if we use the more common QR option, we get the following algorithm.

ALGORITHM **5.1 Simple Subspace Iteration**

1. **Start:** *Choose an initial system of vectors* $X_0 = [x_1, \ldots, x_m]$.
2. **Iterate:** *Until convergence do,*
   (a) *Compute* $X_k := AX_{k-1}$
   (b) *Compute* $X_k = QR$ *the QR factorization of* X_k*, and set* $X_k := Q$.

This algorithm can be viewed as a direct generalization of the power method seen in the previous Chapter. Step 2-(b) is a normalization process that is much similar to the normalization used in the power method, and just as for the power method there are many possible normalizations that can be used. An important observation is that the subspace spanned by the vectors X_k is the same as that spanned by $A^k X_0$. Since the cost of 2-(b) can be high, it is natural to orthonormalize as infrequently as possible, i.e. to perform several steps at once before performing an orthogonalization. This leads to the following modification.

ALGORITHM **5.2 Multiple Step Subspace Iteration**

1. **Start:** *Choose an initial system of vectors* $X = [x_1, \ldots, x_m]$. *Choose an iteration parameter* $iter$.
2. **Iterate:** *Until convergence do:*
   (a) *Compute* $Z := A^{iter} X$.
   (b) *Orthonormalize* Z. *Copy resulting matrix onto* X.
   (c) *Select a new* $iter$.

We would like to make a few comments concerning the choice of the parameter $iter$. The best $iter$ will depend on the convergence rate. If $iter$ is too large then the vectors of Z in 2-(a) may become nearly linear dependent and the orthogonalization in 2-(b) may cause some difficulties. Typically an estimation on the speed of convergence is used to determine $iter$. Then $iter$ is defined in such a way that, for example, the fastest converging vector, which is the first one, will have converged to within a certain factor, e.g., the square root of the machine epsilon, i.e., the largest number ϵ that causes rounding to yield $1 + \epsilon == 1$ on a given computer.

Under a few assumptions the column vectors of X_k will converge "in direction" to the Schur vectors associated with the m dominant eigenvalues $\lambda_1, \ldots, \lambda_m$. To formalize this peculiar notion of convergence, a form of which was seen in the context of the power method, we will say that a sequence of vectors x_k converges *essentially* to a vector x if there exists a sequence of signs $e^{i\theta_k}$ such that the sequence $e^{i\theta_k} x_k$ converges to x.

Theorem 5.1 *Let* $\lambda_1, \ldots, \lambda_m$ *be the* m *dominant eigenvalues of* A *labeled in decreasing order of magnitude and assume that* $|\lambda_i| > |\lambda_{i+1}|, 1 \leq i \leq m$. *Let* $Q = [q_1, q_2, \ldots, q_m]$ *be the Schur vectors associated with* $\lambda_j, j = 1, \ldots, m$ *and* P_i *be the spectral projector associated with the eigenvalues* $\lambda_1, \ldots, \lambda_i$. *Assume that*

$$\text{rank}\,(P_i[x_1, x_2, \ldots, x_i]) = i, \quad \textit{for} \quad i = 1, 2, \ldots, m\,.$$

Then the i*-th column of* X_k *converges essentially to* q_i, *for* $i = 1, 2, \cdots, m$.

Proof. Let the initial system X_0 be decomposed as

$$X_0 = P_m X_0 + (I - P_m)X_0 = QG_1 + WG_2 \tag{5.2}$$

where W is an $n \times (n-m)$ matrix whose column vectors form some basis of the invariant basis $(I - P_m)\mathbb{C}^n$ and G_2 is a certain $(n-m) \times m$ matrix. We know that there exists an $m \times m$ upper triangular matrix R_1 and an $(n-m) \times (n-m)$ matrix R_2 such that

$$AQ = QR_1 \ , \quad AW = WR_2 \ . \tag{5.3}$$

The column vectors of X_k are obtained by orthonormalizing the system $Z_k = A^k X_0$. By assumption, the system of column vectors $P_m X_0$ is nonsingular and therefore G_1 is nonsingular. Applying (5.3) we get

$$\begin{aligned} A^k X_0 &= A^k[QG_1 + WG_2] \\ &= QR_1^k G_1 + WR_2^k G_2 \\ &= [Q + WR_2^k G_2 G_1^{-1} R_1^{-k}]R_1^k G_1 \end{aligned}$$

The term $E_k \equiv WR_2^k G_2 G_1^{-1} R^{-k}$ tends to zero because the spectral radius of R_1^{-1} is equal to $1/|\lambda_m|$ while that of R_2 is $|\lambda_{m+1}|$. Hence,

$$A^k X_0 G_1^{-1} = [Q + E_k]R_1^k$$

with $\lim_{k\to\infty} E_k = 0$. Using the QR decomposition of the matrix $Q + E_k$,

$$Q + E_k = Q^{(k)} R^{(k)},$$

we obtain

$$A^k X_0 G_1^{-1} = Q^{(k)} R^{(k)} R_1^k.$$

Since E_k converges to zero, it is clear that $R^{(k)}$ converges to the identity matrix while $Q^{(k)}$ converges to Q, and because the QR decomposition of a matrix is unique up to scaling constants, we have established that the Q matrix in the QR decomposition of the matrix $A^k X_0 G_1^{-1}$ *converges essentially* to Q. Notice that the span of $A^k X_0 G_1^{-1}$ is identical with that of X_k. As a result the orthogonal projector $\mathcal{P}_m^{(k)}$ onto $span\{X_k\}$ will converge to the orthogonal projector $\mathcal{P}_m$ onto $span\{Q\}$.

In what follows we denote by $[X]_j$ the matrix of the first j vector columns of X. To complete the proof, we need to show that each column converges to the corresponding column vector of Q. To this end we observe that the above proof extends to the case where we consider only the first j columns of X_k, i.e., the j first columns of X_k converge to a matrix that spans the same subspace as $[Q]_j$. In other words, if we let $\mathcal{P}_j$ be the orthogonal projector on $span\{[Q]_j\}$ and $\mathcal{P}_j^{(k)}$ the orthogonal projector on $span\{[X_k]_j\}$ then we have $\mathcal{P}_j^{(k)} \to \mathcal{P}_j$ for $j = 1, 2, \ldots, m$. The proof is now by induction. When $j = 1$, we have the obvious result that the first column of X_k converges essentially to q_1. Assume

that the columns 1 through i of X_k converge essentially to $q_1, \ldots, q_i$. Consider the last column $x_{i+1}^{(k)}$ of $[X_k]_{i+1}$, which we express as

$$x_{i+1}^{(k)} = \mathcal{P}_{i+1}^{(k)} x_{i+1}^{(k)} = \mathcal{P}_i^{(k)} x_{i+1}^{(k)} + (\mathcal{P}_{i+1}^{(k)} - \mathcal{P}_i^{(k)}) x_{i+1}^{(k)} \ .$$

The first term in the right hand side is equal to zero because by construction $x_{i+1}^{(k)}$ is orthogonal to the first i columns of $[X_k]_{i+1}$. Hence,

$$x_{i+1}^{(k)} = (\mathcal{P}_{i+1}^{(k)} - \mathcal{P}_i^{(k)}) x_{i+1}^{(k)}$$

and by the above convergence results on the projectors $\mathcal{P}_j^{(k)}$ we see that $\mathcal{P}_{i+1}^{(k)} - \mathcal{P}_i^{(k)}$ converges to the orthogonal projector onto the span of the single vector q_{i+1}. This is because

$$\mathcal{P}_{i+1} - \mathcal{P}_i = Q_{i+1} Q_{i+1}^H - Q_i Q_i^H = q_{i+1} q_{i+1}^H \ .$$

Therefore we may write $x_{i+1}^{(k)} = q_{i+1} q_{i+1}^H x_{i+1}^{(k)} + \epsilon_k$ where ϵ_k converges to zero. Since the vector $x_{i+1}^{(k)}$ is of norm unity, its orthogonal projection onto q_{i+1} will essentially converge to q_{i+1}. □

The proof indicates that the convergence of each column vector to the corresponding Schur vector is governed by the convergence factor $|\lambda_{i+1}/\lambda_i|$. In addition, we have also proved that each orthogonal projector $\mathcal{P}_i^{(k)}$ onto the first i columns of X_k converges under the assumptions of the theorem.

5.2 Subspace Iteration with Projection

In the subspace iteration with projection method the column vectors obtained from the previous algorithm are not directly used as approximations to the Schur vectors. Instead they are employed in a Rayleigh-Ritz process to get better approximations. In fact as was seen before, the Rayleigh-Ritz approximations are optimal in some sense in the Hermitian case and as a result it is sensible to use a projection process whenever possible. This algorithm with projection is as follows.

ALGORITHM **5.3 Subspace Iteration with Projection**

1. **Start:** *Choose an initial system of vectors* $X = [x_0, \ldots, x_m]$ *and an initial iteration parameter* $iter$.
2. **Iterate:** *Until convergence do:*
   (a) *Compute* $\hat{Z} = A^{iter} X_{old}$.
   (b) *Orthonormalize* $\hat{Z}$ *into* Z.
   (c) *Compute* $B = Z^H A Z$ *and use the QR algorithm to compute the Schur vectors* $Y = [y_1, \ldots, y_m]$ *of* B.
   (d) *Compute* $X_{new} = ZY$.

(e) Test for convergence and select a new iteration parameter $iter$.

There are many implementation details which are omitted for the sake of clarity. Note that there is another version of the algorithm which uses eigenvectors instead of Schur vectors (in Step 2-(c)). These two versions are obviously equivalent when A is Hermitian.

Let S_k be the subspace spanned by X_k and let us denote by $\mathcal{P}_k$ the *orthogonal projector* onto the subspace S_k. Assume that the eigenvalues are ordered in decreasing order of magnitude and that,

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \cdots \geq |\lambda_m| > |\lambda_{m+1}| \geq \cdots \geq |\lambda_n| \ .$$

Again u_i denotes an eigenvector of A of norm unity associated with λ_i. The spectral projector associated with the invariant subspace associated with $\lambda_1, \ldots, \lambda_m$ will be denoted by P. We will now prove the following theorem.

Theorem 5.2 *Let $S_0 = span\{x_1, x_2, \ldots, x_m\}$ and assume that S_0 is such that the vectors $\{Px_i\}_{i=1,\ldots,m}$ are linearly independent. Then for each eigenvector u_i of A, $i = 1, \ldots, m$, there exists a unique vector s_i in the subspace S_0 such that $Ps_i = u_i$. Moreover, the following inequality is satisfied*

$$\|(I - \mathcal{P}_k)u_i\|_2 \leq \|u_i - s_i\|_2 \left(\left| \frac{\lambda_{m+1}}{\lambda_i} \right| + \epsilon_k \right)^k , \tag{5.4}$$

where ϵ_k tends to zero as k tends to infinity.

Proof. By their assumed linear independence, the vectors Px_j, form a basis of the invariant subspace $P\mathbb{C}^n$ and so the vector u_i, which is a member of this subspace, can be written as

$$u_i = \sum_{j=1}^{m} \eta_j P x_j = P \sum_{j=1}^{m} \eta_j x_j \equiv P s_i.$$

The vector s_i is such that

$$s_i = u_i + w, \tag{5.5}$$

where $w = (I - P)s_i$. Next consider the vector y of S_k defined by $y = (\frac{1}{\lambda_i})^k A^k s_i$. We have from (5.5) that

$$y - u_i = \left(\frac{1}{\lambda_i} \right)^k A^k w \ . \tag{5.6}$$

Denoting by W the invariant subspace corresponding to the eigenvalues λ_{m+1}, $\ldots$, λ_n, and noticing that w is in W, we clearly have

$$y - u_i = \left(\frac{1}{\lambda_i} \right)^k [A_{|W}]^k w.$$

Hence,

$$\|u_i - y\|_2 \leq \left\| \left[\frac{1}{\lambda_i} A_{|W} \right]^k \right\|_2 \|w\|_2 \,. \tag{5.7}$$

Since the eigenvalues of $A_{|W}$ are $\lambda_{m+1}, \lambda_{m+2}, \ldots, \lambda_n$ the spectral radius of $[\frac{1}{\lambda_i} A_{|W}]$ is simply $|\lambda_{m+1}/\lambda_i|$ and from Corollary 1.1 of Chapter 1, we have,

$$\left\| \left[\frac{1}{\lambda_i} A_{|W} \right]^k \right\|_2 = \left[\left| \frac{\lambda_{m+1}}{\lambda_i} \right| + \epsilon_k \right]^k, \tag{5.8}$$

where ϵ_k tends to zero as $k \to \infty$. Using the fact that

$$\|(I - \mathcal{P}_k) u_i\|_2 = \min_{y \in S_k} \|y - u_i\|_2$$

together with inequality (5.7) and equality (5.8) yields the desired result (5.4). □

We can be a little more specific about the sequence ϵ_k of the theorem by using the inequality

$$\|B^k\|_2 \leq \alpha \rho^k k^{\eta - 1}, \tag{5.9}$$

where B is any matrix, ρ its spectral radius, η the dimension of its largest Jordan block, and α some constant *independent* on k, see Exercise P-5.6 as well as Householder's book [91]. Without loss of generality we assume that $\alpha \geq 1$.

Initially, consider the case where A is diagonalizable. Then $\eta = 1$, and by replacing (5.9) in (5.8) we observe that (5.4) simplifies into

$$\|(I - \mathcal{P}_k) u_i\|_2 \leq \alpha \|u_i - s_i\|_2 \left| \frac{\lambda_{m+1}}{\lambda_i} \right|^k \,. \tag{5.10}$$

Still in the diagonalizable case, it is possible to get a more explicit result by expanding the vector s_i in the eigenbasis of A as

$$s_i = u_i + \sum_{i=m+1}^{n} \xi_j u_j.$$

Letting $\beta = \sum_{i=m+1}^{n} |\xi_j|$, we can reproduce the proof of the above theorem to obtain

$$\|(I - \mathcal{P}_k) u_i\|_2 \leq \alpha \beta \left| \frac{\lambda_{m+1}}{\lambda_i} \right|^k \,. \tag{5.11}$$

When A is not diagonalizable, then from comparing (5.9) and (5.8) we can bound ϵ_k from above as follows:

$$\epsilon_k \leq \left| \frac{\lambda_{m+1}}{\lambda_i} \right| (\alpha^{1/k} k^{(\eta-1)/k} - 1)$$

which confirms that ϵ_k tends to zero as k tends to infinity.

Finally, concerning the assumptions of the theorem, it can be easily seen that the condition that $\{Px_j\}_{j=1,\ldots,r}$ form an independent system of vectors is equivalent to the condition that

$$\det[U^H S_0] \neq 0,$$

in which U is any basis of the invariant subspace $P\mathbb{C}^n$. This condition constitutes a generalization of a similar condition required for the convergence of the power method.

5.3 Practical Implementations

There are a number of implementation details that enhance the performance of the simple methods described above. The first of these is the use of locking, a form of deflation, which exploits the inequal convergence rates of the different eigenvectors. In addition, the method is rarely used without some form of acceleration. Similarly to the power method the simplest form of acceleration, is to shift the matrix to optimize the convergence rate for the eigenvalue being computed. However, there are more elaborate techniques which will be briefly discussed later.

5.3.1 Locking

Because of the different rates of convergence of each of the approximate eigenvalues computed by the subspace iteration, it is a common practice to extract them one at a time and perform a form of deflation. Thus, as soon as the first eigenvector has converged there is no need to continue to multiply it by A in the subsequent iterations. Indeed we can freeze this vector and work only with the vectors $q_2, \ldots, ...q_m$. However, we will still need to perform the subsequent orthogonalizations with respect to the frozen vector q_1 whenever such orthogonalizations are needed. The term used for this strategy is *locking*. It was introduced by Jennings and Stewart [97]. Note that acceleration techniques and other improvements to the basic subspace iteration desribed in Section 5.3 can easily be combined with locking.

The following algorithm describes a practical subspace iteration with deflation (locking) for computing the nev dominant eigenvalues.

ALGORITHM **5.4 Subspace Iteration with Projection and Deflation**

1. **Start:** *Choose an initial system of vectors* $X := [x_0, \ldots, x_m]$ *and an initial iteration parameter* $iter$. *Set* $j := 1$.
2. **Eigenvalue loop:** *While* $j \leq nev$ *do:*
   (a) *Compute* $\hat{Z} = [q_1, q_2, \ldots, q_{j-1}, A^{iter}X]$.
   (b) *Orthonormalize the column vectors of* $\hat{Z}$ *(starting at column* j*) into* Z.
   (c) *Update* $B = Z^H AZ$ *and compute the Schur vectors* $Y = [y_j, \ldots, y_m]$ *of* B *associated with the eigenvalues* $\lambda_j, \ldots, \lambda_m$.

(d) *Test the eigenvalues $\lambda_j, \ldots, \lambda_m$ for convergence. Let i_{conv} the number of newly converged eigenvalues. Append the i_{conv} corresponding Schur vectors to $Q = [q_1, ..., q_{j-1}]$ and set $j := j + i_{conv}$.*

(e) *Compute $X := Z[y_j, y_{j+1}, \ldots, y_m]$.*

(f) *Compute a new iteration parameter $iter$.*

Example 5.1. Consider the matrix Mark(10) described in Chapter 2 and used in the test examples of Chapter 4. We tested a version of the algorithm just described to compute the three dominant eigenvalues of Mark(10). In this test we took $m = 10$ and started with an initial set of vectors obtained from orthogonalizing $v, Av, ..., A^m v$, in which v is a random vector. Table 5.1 shows the results. Each horizontal line separates an outer loop of the algorithm (corresponding to step (2) in algorithm 5.4). Thus, the algorithm starts with $iter = 5$ and in the first iteration (requiring 63 matrix-vector products) no new eigenvalue has converged. We will need three more outer iterations (requiring each 113 matrix-vector products) to achieve convergence for the two dominant eigenvalues $-1, 1$. Another outer iteration is needed to compute the third eigenvalue. Note that each projection costs 13 additional matrix by vector products, 10 for computing the C matrix and 3 for the residual vectors. □

Mat-vec's	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
63	0.1000349211D+01	0.0	0.820D-02
	-0.9981891280D+00	0.0	0.953D-02
	-0.9325298611D+00	0.0	0.810D-02
176	-0.1000012613D+01	0.0	0.140D-03
	0.9999994313D+00	0.0	0.668D-04
	0.9371856730D+00	0.0	0.322D-03
289	-0.1000000294D+01	0.0	0.335D-05
	0.1000000164D+01	0.0	0.178D-05
	0.9371499768D+00	0.0	0.177D-04
402	-0.1000000001D+01	0.0	0.484D-07
	0.1000000001D+01	0.0	0.447D-07
	0.9371501017D+00	0.0	0.102D-05
495	-0.1000000001D+01	0.0	0.482D-07
	0.1000000000D+01	0.0	0.446D-07
	0.9371501543D+00	0.0	0.252D-07

Table 5.1: Convergence of subspace iteration with projection for computing the three dominant eigenvalues of $A = \mathrm{Mark}(10)$.

5.3.2 Linear Shifts

Similarly to the power method, there are advantages in working with the shifted matrix $A - \sigma I$ instead of A, where σ is a carefully chosen shift. In fact since the eigenvalues are computed one at a time, the situation is very similar to that of the power method. Thus, when the spectrum is real, and the eigenvalues are ordered decreasingly, the best possible σ is

$$\sigma = \frac{1}{2}(\lambda_{m+1} + \lambda_n)$$

which will put the middle of the unwanted part of the spectrum at the origin. Note that when deflation is used this is independent of the eigenvalue being computed. In addition, we note one important difference with the power method, namely that eigenvalue estimates are now readily available. In fact, it is common practice to take $m > nev$, the number of eigenvalues to be computed, in order to be able to obtain valuable estimates dynamically. These estimates can be used in various ways to accelerate convergence, such as when selecting shifts as indicated above, or when using some of the more sophisticated preconditioning techniques mentioned in the next section.

5.3.3 Preconditioning

Preconditioning is especially important for subspace iteration, since the unpreconditioned iteration may be unacceptably slow in some cases. Although we will cover preconditioning in more detail in Chapter 8, we would like to mention here the main ideas used to precondition the subspace iteration.

- Shift-and-invert. This consists of working with the matrix $(A - \sigma I)^{-1}$ instead of A. The eigenvalues near σ will converge fast.
- Polynomial acceleration. The standard method used is to replace the power A^{iter} in the usual subspace iteration algorithm by a polynomial $T_m[(A - \sigma I)/\rho]$ in which T_m is the Chebyshev polynomial of the first kind of degree m.

With either type of preconditioning subspace iteration may be a reasonably efficient method that has the advantage of being easy to code and understand. Some of the methods to be seen in the next Chapter are often preferred however, because they tend to be more economical.

PROBLEMS

P-5.1 In Bauer's original Treppeniteration, the linear independence of the vectors in $A^k X_0$ are preserved by performing its LU decomposition. Thus,

$$\hat{X} = A^k X \quad , \hat{X} = L_k U_k, \quad X := L_k,$$

in which L_k is an $n \times m$ matrix with its upper $m \times m$ corner being a unit lower triangular matrix, and U_k is an $m \times m$ upper triangular matrix. Extend the main convergence theorem of the corresponding algorithm, for this case.

P-5.2 Assume that the matrix A is real and the eigenvalues λ_m, λ_{m+1} forms a complex conjugate pair. If subspace iteration with deflation (Algorithm 5.4) is used, there will be a difficulty when computing the last eigenvalue. Provide a few possible modifications to the algorithm to cope with this case.

P-5.3 Write a modification of Algorithm 5.4 which incorporates a dynamic shifting strategy. Assume that the eigenvalues are real and consider both the case where the rightmost or the leftmost eigenvalues are wanted.

P-5.4 Let A be a matrix whose eigenvalues are real and assume that the subspace iteration algorithm (with projection) is used to compute some of the eigenvalues with largest real parts of A. The question addressed here is how to get the best possible iteration parameter $iter$. We would like to choose $iter$ in such a way that in the worst case, the vectors of X will loose a factor of $\sqrt{\epsilon}$ in their linear dependence, in which ϵ is the machine accuracy. How can we estimate such an iteration parameter $iter$ from quantities derived from the algorithm? You may assume that m is sufficiently large compared with nev (how large should it be?).

P-5.5 Generalize the result of the previous exercise to the case where the eigenvalues are not necessarily real.

P-5.6 Using the Jordan Canonical form, show that for any matrix B,

$$\|B^k\|_2 \leq \alpha \rho^k k^{\eta-1}, \tag{5.12}$$

where ρ is the spectral radius of B, η the dimension of its largest Jordan block, and α some constant.

P-5.7 Implement a subspace iteration with projection to compute the eigenvalues with largest modulus of a large sparse matrix. Implement locking and linear shifts.

NOTES AND REFERENCES. An early reference on Bauer's Treppeniteration, in addition to the original paper by Bauer [4], is Householder's book [91]. See also the paper by Rutishauser [167] and by Clint and Jennings [31] as well as the book by Bathé and Wilson [3] which all specialize to symmetric matrices. A computer code for the symmetric real case was published in Wilkinson and Reinsch's handbook [223] but unlike most other codes in the handbook, never became part of the Eispack library. Later, some work was done to develop computer codes for the non-Hermitian case. Thus, a 'lop-sided' version of Bauer's treppeniteration based on orthogonal projection method rather than oblique projection was introduced by Jennings and Stewart [96] and a computer code was also made available [97]. However, the corresponding method did not incorporate Chebyshev acceleration, which turned out to be so useful in the Hermitian case. Chebyshev acceleration was later incorporated in [173] and some theory was proposed in [171]. G. W. Stewart [200, 201] initiated the idea of using Schur vectors as opposed to eigenvectors in subspace iteration. The motivation is that Schur vectors are easier to handle numerically. A convergence theory of Subspace Iteration was proposed in [200]. The convergence results of Section 5.2 follow the paper [171] and a modification due to Chatelin (private communication). ■

Chapter 6

KRYLOV SUBSPACE METHODS

This chapter will examine one of the most important classes of methods available for computing eigenvalues and eigenvectors of large matrices. These techniques are based on projections methods, both orthogonal and oblique, onto Krylov subpaces, i.e., subspaces spanned by the iterates of the simple power method. What may appear to be a trivial extension of a very slow algorithm turns out to be one of the most successful methods for extracting eigenvalues of large matrices, especially in the Hermitian case.

6.1 Krylov Subspaces

An important class of techniques known as *Krylov subspace methods* extracts approximations from a subspace of the form

$$\mathcal{K}_m \equiv \text{span}\,\{v, Av, A^2v, ...A^{m-1}v\} \tag{6.1}$$

referred to as a Krylov subspace. If there is a possibility of ambiguity, $\mathcal{K}_m$ is denoted by $\mathcal{K}_m(A, v)$. In contrast with subspace iteration, the dimension of the subspace of approximants increases by one at each step of the approximation process. A few well-known of these *Krylov subspace methods* are:

(1) The Hermitian Lanczos algorithm;

(2) Arnoldi's method and its variations;

(3) The nonhermitian Lanczos algorithm.

There are also block extensions of each of these methods termed *Block Krylov Subspace methods*, which we will discuss only briefly. Arnoldi's method and Lanczos' method are orthogonal projection methods while the nonsymmetric Lanczos algorithm is an oblique projection method. Before we pursue with the analysis of these methods, we would like to emphasize an important distinction between *implementation* of a method and *the method itself*. There are several distinct implementations of Arnoldi's method, which are all mathematically equivalent. For

example the articles [55, 169, 216] all propose some different versions of the same mathematical process.

In this section we start by establishing a few elementary properties of Krylov subspaces, many of which need no proof. Recall that the minimal polynomial of a vector v is the nonzero monic polynomial p of lowest degree such that $p(A)v = 0$.

Proposition 6.1 *The Krylov subspace $\mathcal{K}_m$ is the subspace of all vectors in $\mathbb{C}^n$ which can be written as $x = p(A)v$, where p is a polynomial of degree not exceeding $m-1$.*

Proposition 6.2 *Let μ be the degree of the minimal polynomial of v. Then $\mathcal{K}_\mu$ is invariant under A and $\mathcal{K}_m = \mathcal{K}_\mu$ for all $m \geq \mu$.*

The degree of the minimal polynomial of v is often referred to as the *grade* of v with respect to A. Clearly, the grade of v does not exceed n.

Proposition 6.3 *The Krylov subspace $\mathcal{K}_m$ is of dimension m if and only if the grade of v with respect to A is larger than $m-1$.*

Proof. The vectors $v, Av, \ldots A^{m-1}v$ form a basis of $\mathcal{K}_m$ if and only if for any complex $m-$tuple $\alpha_i, i = 0, \ldots, m-1$, where at least one α_i is nonzero, the linear combination $\sum_{i=0}^{m-1} \alpha_i A^i v$ is nonzero. This condition is equivalent to the condition that there be no polynomial of degree $\leq m-1$ for which $p(A)v = 0$. This proves the result. □

Proposition 6.4 *Let Q_m be any projector onto $\mathcal{K}_m$ and let A_m be the section of A to $\mathcal{K}_m$, that is, $A_m = Q_m A_{|\mathcal{K}_m}$. Then for any polynomial q of degree not exceeding $m-1$, we have $q(A)v = q(A_m)v$, and for any polynomial of degree $\leq m$, we have $Q_m q(A)v = q(A_m)v$.*

Proof. We will first prove that $q(A)v = q(A_m)v$ for any polynomial q of degree $\leq m-1$. It suffices to prove the property for the monic polynomials $q_i(t) \equiv t^i$, $i = 0, \ldots m-1$. The proof is by induction. The property is clearly true for the polynomial $q_0(t) \equiv 1$. Assume that it is true for $q_i(t) \equiv t^i$:

$$q_i(A)v = q_i(A_m)v.$$

Multiplying the above equation by A on both sides we get

$$q_{i+1}(A)v = Aq_i(A_m)v.$$

If $i+1 \leq m-1$ the vector on the left hand-side belongs to $\mathcal{K}_m$ and therefore if we multiply the above equation on both sides by Q_m we get

$$q_{i+1}(A)v = Q_m A q_i(A_m)v.$$

Looking at the right hand side we observe that $q_i(A_m)v$ belongs to $\mathcal{K}_m$. Hence

$$q_{i+1}(A)v = Q_m A_{|\mathcal{K}_m} q_i(A_m)v = q_{i+1}(A_m)v,$$

which proves that the property is true for $i+1$ provided $i+1 \leq m-1$. For the case $i+1 = m$ it remains only to show that $Q_m q_m(A)v = q_m(A_m)v$, which follows from $q_{m-1}(A)v = q_{m-1}(A_m)v$ by simply multiplying both sides by $Q_m A$. □

An interesting characterization of *orthogonal* Krylov projection methods can be formulated in terms of the characteristic polynomial of the approximate problem. In the orthogonal projection case, we define the characteristic polynomial of the approximate problem as that of the matrix $V_m^H A V_m$ where V_m is a matrix whose column vectors form an orthonormal basis of $\mathcal{K}_m$. It is a simple exercise to show that this definition is independent of the choice of V_m, the basis of the Krylov subspace.

Theorem 6.1 *Let $\bar{p}_m$ be the characteristic polynomial of the approximate problem resulting from an orthogonal projection method onto the Krylov subspace K_m. Then $\bar{p}_m$ minimizes the norm $\|p(A)v\|_2$ over all monic polynomials p of degree m.*

Proof. We denote by $\mathcal{P}_m$ the orthogonal projector onto $\mathcal{K}_m$ and A_m the corresponding section of A. By Cayley Hamilton's theorem we have $\bar{p}_m(A_m) = 0$ and therefore

$$(\bar{p}_m(A_m)v, w) = 0, \ \forall\, w \in \mathcal{K}_m \ . \tag{6.2}$$

By the previous proposition $\bar{p}_m(A_m)v = \mathcal{P}_m \bar{p}_m(A)v$. Hence (6.2) becomes

$$(\mathcal{P}_m \bar{p}_m(A)v, w) = 0, \ \forall\, w \in \mathcal{K}_m,$$

or, since orthogonal projectors are self adjoint,

$$(\bar{p}_m(A)v, \mathcal{P}_m w) = 0 = (\bar{p}_m(A)v, w) \ \ \forall\, w \in \mathcal{K}_m,$$

which is equivalent to

$$(\bar{p}_m(A)v, A^j v) = 0 \ , \ j = 0, \ldots m-1.$$

Writing $\bar{p}_m(t) = t^m - q(t)$, where q is of degree $\leq m-1$, we obtain

$$(A^m v - q(A)v, A^j v) = 0 \ , \ j = 0, \ldots m-1.$$

In the above system of equations we recognize the normal equations for minimizing the Euclidean norm of $A^m v - s(A)v$ over all polynomials s of degree $\leq m-1$. The proof is complete. □

The above characteristic property is not intended to be used for computational purposes. It is useful for establishing mathematical equivalences between seemingly different methods. Thus, a method developed by Erdelyi in 1965 [55] is based on precisely minimizing $\|p(A)v\|_2$ over monic polynomials of some degree and is therefore mathematically equivalent to any orthogonal projection method on a Krylov subspace. Another such method was proposed by Manteuffel [126, 127]

for the purpose of estimating acceleration parameters when solving linear systems by Chebyshev method. His method named the Generalized Power Method, was essentially Erdelyi's method with a special initial vector.

An important point is that this characteristic property seems to be the only known optimality property that is satisfied by the approximation process in the nonsymmetric case. Other optimality properties, such as the mini-max theorem which are fundamental both in theory and in practice for symmetric problems are no longer valid. This results in some significant difficulties in understanding and analyzing these methods for nonsymmetric eigenvalue problems.

6.2 Arnoldi's Method

Arnoldi's method is an orthogonal projection method onto $\mathcal{K}_m$ for general non-Hermitian matrices. The procedure was introduced in 1951 as a means of reducing a dense matrix into Hessenberg form. Arnoldi introduced this method precisely in this manner and he hinted that the process could give good approximations to some eigenvalues if stopped before completion. It was later discovered that this strategy yields a good technique for approximating eigenvalues of large sparse matrices. We first describe the method without much regard to rounding errors, and then give a few implementation details.

6.2.1 The Basic Algorithm

The procedure introduced by Arnoldi in 1951 starts by building an orthogonal basis of the Krylov subspace $\mathcal{K}_m$. In exact arithmetic, one variant of the algorithm is as follows.

ALGORITHM **6.1 Arnoldi**

1. **Start:** *Choose a vector v_1 of norm 1.*

2. **Iterate:** *for $j = 1, 2, \ldots, m$ compute:*

$$h_{ij} = (Av_j, v_i), \quad i = 1, 2, \ldots, j, \tag{6.3}$$

$$w_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i, \tag{6.4}$$

$$h_{j+1,j} = \|w_j\|_2 , \quad \textit{if } h_{j+1,j} = 0 \textit{ stop} \tag{6.5}$$

$$v_{j+1} = w_j / h_{j+1,j}. \tag{6.6}$$

The algorithm will stop if the vector w_j computed in (6.4) vanishes. We will come back to this case shortly. We now prove a few simple but important properties of the algorithm. A first observation is that the algorithm is a form of classical Gram-Schmidt orthogonalization process whereby a new vector (Av_j) is formed and then orthogonalized against all previous v_i's. As the next proposition shows, the resulting basis is a basis of the Krylov subspave $\mathcal{K}_m$.

Proposition 6.5 *The vectors $v_1, v_2, \ldots, v_m$ form an orthonormal basis of the subspace $\mathcal{K}_m = \mathrm{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$.*

Proof. The vectors $v_j, i = 1, 2, \ldots m$ are orthonormal by construction. That they span $\mathcal{K}_m$ follows from the fact that each vector v_j is of the form $q_{j-1}(A)v_1$ where q_{j-1} is a polynomial of degree $j-1$. This can be shown by induction on j as follows. Clearly, the result is true when $j = 1$, since $v_1 = q_0(A)v_1$ with $q_0(t) \equiv 1$. Assume that the result is true for all integers $\leq j$ and consider v_{j+1}. We have

$$h_{j+1}v_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij}v_i = Aq_{j-1}(A)v_1 - \sum_{i=1}^{j} h_{ij}q_{i-1}(A)v_1 \qquad (6.7)$$

which shows that v_{j+1} can be expressed as $q_j(A)v_1$ where q_j is of degree j and completes the proof. □

Proposition 6.6 *Denote by V_m the $n \times m$ matrix with column vectors $v_1, \ldots, v_m$ and by H_m the $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations hold:*

$$AV_m = V_m H_m + h_{m+1,m}v_{m+1}e_m^H, \qquad (6.8)$$

$$V_m^H AV_m = H_m\ . \qquad (6.9)$$

Proof. The relation (6.8) follows from the following equality which is readily derived from (6.6) and (6.4):

$$Av_j = \sum_{i=1}^{j+1} h_{ij}v_i, \quad j = 1, 2, \ldots, m\ . \qquad (6.10)$$

Relation (6.9) follows by multiplying both sides of (6.8) by V_m^H and making use of the orthonormality of $\{v_1, \ldots, v_m\}$. □

The situation is illustrated in Figure 6.1. As was noted earlier the algorithm may break down in case the norm of w_j vanishes at a certain step j. In this situation the vector v_{j+1} cannot be computed and the algorithm stops. There remains to determine the conditions under which this situation occurs.

Proposition 6.7 *Arnoldi's algorithm breaks down at step j (i.e., $w_j = 0$ in (6.4)) if and only if the minimal polynomial of v_1 is of degree j. Moreover, in this case the subspace $\mathcal{K}_j$ is invariant and the approximate eigenvalues and eigenvectors are exact.*

Proof. If the degree of the minimal polynomial is j, then w_j must be equal to zero. Indeed, otherwise v_{j+1} can be defined and as a result $\mathcal{K}_{j+1}$ would be of dimension $j+1$, and from Proposition 6.3, this would mean that $\mu \geq j+1$,

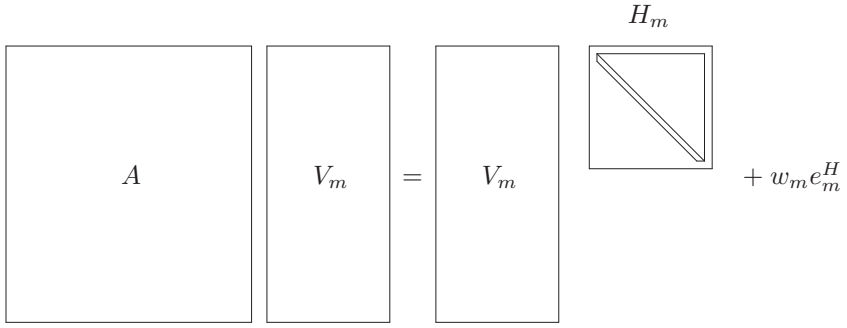


Figure 6.1: The action of A on V_m gives $V_m H_m$ plus a rank one matrix.

which is not true. To prove the converse, assume that $w_j = 0$. Then the degree μ of the minimal polynomial of v_1 is such that $\mu \leq j$. Moreover, we cannot have $\mu < j$ otherwise by the previous proof the vector w_μ would be zero and the algorithm would have stopped at the earlier step number μ. The rest of the result follows from Proposition 4.3 seen in Chapter 4. □

The approximate eigenvalues $\lambda_i^{(m)}$ provided by the projection process onto $\mathcal{K}_m$ are the eigenvalues of the Hessenberg matrix H_m. The Ritz approximate eigenvector associated with $\lambda_i^{(m)}$ is defined by $u_i^{(m)} = V_m y_i^{(m)}$ where $y_i^{(m)}$ is an eigenvector associated with the eigenvalue $\lambda_i^{(m)}$. A number of the Ritz eigenvalues, typically a small fraction of m, will usually constitute good approximations of corresponding eigenvalues λ_i of A and the quality of the approximation will usually improve as m increases. We will examine these 'convergence' properties in detail in later sections. The original algorithm consists of increasing m until all desired eigenvalues of A are found. This is costly both in terms of computation and storage. For storage, we need to keep m vectors of length n plus an $m \times m$ Hessenberg matrix, a total of approximately $nm + m^2/2$. Considering the computational cost of the j-th step, we need to multiply v_j by A, at the cost of $2 \times Nz$, where Nz is number of nonzero elements in A, and then orthogonalize the result against j vectors at the cost of $4(j+1)n$, which increases with the step number j.

On the practical side it is crucial to be able to estimate the residual norm inexpensively as the algorithm progresses. This turns out to be quite easy to do for Arnoldi's method and, in fact, for all the Krylov subspace methods described in this chapter. The result is given in the next proposition.

Proposition 6.8 *Let $y_i^{(m)}$ be an eigenvector of H_m associated with the eigenvalue $\lambda_i^{(m)}$ and $u_i^{(m)}$ the Ritz approximate eigenvector $u_i^{(m)} = V_m y_i^{(m)}$. Then,*

$$(A - \lambda_i^{(m)} I) u_i^{(m)} = h_{m+1,m}\, e_m^H y_i^{(m)} v_{m+1}$$

and, therefore,

$$\|(A - \lambda_i^{(m)} I)u_i^{(m)}\|_2 = h_{m+1,m}|e_m^H y_i^{(m)}| \ .$$

Proof. This follows from multiplying both sides of (6.8) by $y_i^{(m)}$:

$$\begin{aligned} AV_m y_i^{(m)} &= V_m H_m y_i^{(m)} + h_{m+1,m}\, e_m^H y_i^{(m)} v_{m+1} \\ &= \lambda_i^{(m)} V_m y_i^{(m)} + h_{m+1,m} e_m^H y_i^{(m)} v_{m+1} \ . \end{aligned}$$

Hence,

$$AV_m y_i^{(m)} - \lambda_i^{(m)} V_m y_i^{(m)} = h_{m+1,m}\, e_m^H y_i^{(m)}\ v_{m+1} \ . \qquad \square$$

In simpler terms, the proposition states that the residual norm is equal to the last component of the eigenvector $y_i^{(m)}$ multiplied by $h_{m+1,m}$. In practice, the residual norms, although not always indicative of actual errors, are quite helpful in deriving stopping procedures.

6.2.2 Practical Implementations

The description of the Arnoldi process given earlier assumed exact arithmetic. In reality, much is to be gained by using the Modified Gram-Schmidt or the Householder algorithm in place of the standard Gram-Schmidt algorithm. With the modified Gram-Schmidt alternative the algorithm takes the following form.

ALGORITHM **6.2 Arnoldi - Modified Gram-Schmidt**

1. **Start.** *Choose a vector* v_1 *of norm 1.*
2. **Iterate.** *For* $j = 1, 2, \ldots, m$ *do:*
   (a) $w := Av_j$*;*
   (b) *For* $i = 1, 2, \ldots, j$ *do:*

$$\begin{aligned} h_{ij} &= (w, v_i), \\ w &:= w - h_{ij} v_i; \end{aligned}$$

   (c) $h_{j+1,j} = \|w\|_2$ *;*
   (d) $v_{j+1} = w/h_{j+1,j}$ *.*

There is no difference in exact arithmetic between this algorithm and Algorithm 6.1. Although this formulation is numerically superior to the standard Gram Schmidt formulation, we do not mean to imply that the above Modified Gram-Schmidt is sufficient for all cases. In fact there are two alternatives that are implemented to guard against large cancellations during the orthogonalization process.

The first alternative is to resort to double orthogonalization. Whenever the final vector obtained at the end of the second loop in the above algorithm has been computed, a test is performed to compare its norm with the norm of the initial w (which is $\|Av_j\|_2$). If the reduction falls below a certain threshold, an indication that sever cancellation might have occurred, a second orthogonalization is made. It is known from a result by Kahan that additional orthogonalizations are superfluous (see for example Parlett [148]).

The second alternative is to resort to a different technique altogether. In fact one of the most reliable orthogonalization techniques, from the numerical point of view, is the Householder algorithm. This has been implemented for the Arnoldi process by Walker [220]. We do not describe the Householder algorithm here but we would like to compare the cost of each of the three versions.

In the table shown below, GS stands for Gram-Schmidt, MGS for Modified Gram-Schmidt, MGSR for Modified Gram-Schmidt with Reorthogonalization, and HO for Householder.

	GS	MGS	MGSR	HO
Flops	m^2n	m^2n	$2m^2n$	$2m^2n - \frac{2}{3}m^3$
Storage	$(m+1)n$	$(m+1)n$	$(m+1)n$	$(m+1)n - \frac{1}{2}m^2$

A few comments are in order. First, the number of operations shown for MGSR are for the worst case situation when a second orthogonalization is needed every time. This is unlikely to take place and in practice the actual number of operations is much more likely to be close to that of the simple MGS. Concerning storage, the little gain in storage requirement in the Householder version comes from the fact that the Householder transformation requires vectors whose length diminishes by 1 at every step of the process. However, this difference is negligible relative to the whole storage requirement given that usually $m \ll n$. Moreover, the implementation to take advantage of this little gain may become rather complicated. In spite of this we do recommend implementing Householder orthogonalization for developing general purpose reliable software packages. A little additional cost in arithmetic may be more than offset by the gains in robustness in these conditions.

Example 6.1. Consider the matrix Mark(10) used in the examples in the previous two Chapters. Table 6.1 shows the convergence of the rightmost eigenvalue obtained by Arnoldi's method. Comparing the results shown in Table 6.1 with those of the examples seen in Chapter 4, it is clear that the convergence is much faster than the power method or the shifted power method. □

As was mentioned earlier the standard implementations of Arnoldi's method are limited by their high storage and computational requirements as m increases. Suppose that we are interested in only one eigenvalue/eigenvector of A, namely

m	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
5	0.9027159373	0.0	0.316D+00
10	0.9987435899	0.0	0.246D-01
15	0.9993848488	0.0	0.689D-02
20	0.9999863880	0.0	0.160D-03
25	1.000000089	0.0	0.135D-05
30	0.9999999991	0.0	0.831D-08

Table 6.1: Convergence of rightmost eigenvalue computed from a simple Arnoldi algorithm for $A = \mathrm{Mark}(10)$.

m	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
10	0.9987435899D+00	0.0	0.246D-01
20	0.9999523324D+00	0.0	0.144D-02
30	0.1000000368D+01	0.0	0.221D-04
40	0.1000000025D+01	0.0	0.508D-06
50	0.9999999996D+00	0.0	0.138D-07

Table 6.2: Convergence of rightmost eigenvalue computed from a restarted Arnoldi procedure for $A = \mathrm{Mark}(10)$.

the eigenvalue of largest real part of A. Then one way to circumvent the difficulty is to *restart* the algorithm. After a run with m Arnoldi vectors, we compute the approximate eigenvector and use it as an initial vector for the next run with Arnoldi's method. This process, which is the simplest of this kind, is iterated to convergence.

ALGORITHM **6.3 Iterative Arnoldi**

1. **Start:** *Choose an initial vector* v_1 *and a dimension* m.
2. **Iterate:** *Perform* m *steps of Arnoldi's algorithm.*
3. **Restart:** *Compute the approximate eigenvector* $u_1^{(m)}$ *associated with the rightmost eigenvalue* $\lambda_1^{(m)}$.
   If satisfied stop, else set $v_1 \equiv u_1^{(m)}$ *and go to 2.*

Example 6.2. Consider the same matrix Mark(10) as above. We now use a restarted Arnoldi procedure for computing the eigenvector associated with the eigenvalue with algebraically largest real part. We use $m = 10$. Comparing

the results of Table 6.2 with those of the previous example indicates a loss in performance, in terms of total number of matrix-vector products. However, the number of vectors used here is 10 as opposed to 50, so the memory requirement is much less. □

6.2.3 Incorporation of Implicit Deflation

We now consider the following implementation which incorporates a deflation process. The previous algorithm is valid only for the case where only one eigenvalue/eigenvector pair must be computed. In case several such pairs must be computed, then there are two possible options. The first, is to take v_1 to be a linear combination of the approximate eigenvectors when we restart. For example, if we need to compute the p rightmost eigenvectors, we may take

$$\hat{v}_1 = \sum_{i=1}^{p} \rho_i \tilde{u}_i,$$

where the eigenvalues are numbered in decreasing order of their real parts. The vector v_1 is then obtained from normalizing $\hat{v}_1$. The simplest choice for the coefficients ρ_i is to take $\rho_i = 1, i = 1, \ldots, p$. There are several drawbacks to this approach, the most important of which being that there is no easy way of choosing the coefficients ρ_i in a systematic manner. The result is that for hard problems, convergence is difficult to achieve.

An alternative is to compute one eigenpair at a time and use deflation. We can use deflation on the matrix A explicitly as was described in Chapter 4. This entails constructing progressively the first k Schur vectors. If a previous orthogonal basis $[u_1, \ldots, u_{k-1}]$ of the invariant subspace has already been computed, then, to compute the eigenvalue λ_k, we work with the matrix $A - U\Sigma U^H$, in which Σ is a diagonal matrix.

Another implementation, which we now describe, is to work with a single basis $v_1, v_2, ..., v_m$ whose first vectors are the Schur vectors that have already converged. Suppose that $k-1$ such vectors have converged and call them $v_1, v_2, ..., v_{k-1}$. Then we start by choosing a vector v_k which is orthogonal to $v_1, ..., v_{k-1}$ and of norm 1. Next we perform $m-k$ steps of an Arnoldi process in which orthogonality of the vector v_j against all previous $v_i's$, including $v_1, ..., v_{k-1}$ is enforced. This generates an orthogonal basis of the subspace

$$\text{span}\{v_1, \ldots, v_{k-1}, v_k, Av_k, \ldots, A^{m-k}v_k\}\ . \tag{6.11}$$

Thus, the dimension of this modified Krylov subspace is constant and equal to m in general. A sketch of this implicit deflation procedure combined with Arnoldi's method is the following.

ALGORITHM **6.4 Deflated Iterative Arnoldi**

A. Start: *Choose an initial vector* v_1 *of norm unity. Set* $k := 1$.

B. Eigenvalue loop:

1. *Arnoldi Iteration. For* $j = k, k+1, ..., m$ *do:*
   - *Compute* $w := Av_j$.
   - *Compute a set of* j *coefficients* h_{ij} *so that* $w := w - \sum_{i=1}^{j} h_{ij} v_i$ *is orthogonal to all previous* v_i*'s,* $i = 1, 2, ..., j$.
   - *Compute* $h_{j+1,j} = \|w\|_2$ *and* $v_{j+1} = w/h_{j+1,j}$.
2. *Compute approximate eigenvector of* A *associated with the eigenvalue* $\tilde{\lambda}_k$ *and its associated residual norm estimate* ρ_k.
3. *Orthonormalize this eigenvector against all previous* v_j*'s to get the approximate Schur vector* $\tilde{u}_k$ *and define* $v_k := \tilde{u}_k$.
4. *If* ρ_k *is small enough then (accept eigenvalue):*
   - *Compute* $h_{i,k} = (Av_k, v_i)\ ,\ i = 1, .., k$,
   - *Set* $k := k + 1$,
   - *If* $k \geq nev$ *then stop else goto B.*
5. *Else go to B-1.*

Note that in the B-loop, the Schur vectors associated with the eigenvalues $\lambda_1, ..., \lambda_{k-1}$ are frozen and so is the corresponding upper triangular matrix corresponding to these vectors. As a new Schur vector has converged, step B.4 computes the k-th column of R associated with this new basis vector. In the subsequent steps, the approximate eigenvalues are the eigenvalues of the $m \times m$ Hessenberg matrix H_m defined in the algorithm and whose $k \times k$ principal submatrix is upper triangular For example when $m = 6$ and after the second Schur vector, $k = 2$, has converged, the matrix H_m will have the form

$$H_m = \begin{pmatrix} * & * & * & * & * & * \\ & * & * & * & * & * \\ & & * & * & * & * \\ & & * & * & * & * \\ & & & * & * & * \\ & & & & * & * \end{pmatrix}. \tag{6.12}$$

Therefore, in the subsequent steps, we will consider only the eigenvalues that are not associated with the 2×2 upper triangular matrix.

It can be shown that, in exact arithmetic, the $(n-k) \times (n-k)$ Hessenberg matrix in the lower (2×2) block is the same matrix that would be obtained from an Arnoldi run applied to the matrix $(I - P_k)A$ in which P_k is the orthogonal projector onto the (approximate) invariant subspace that has already been computed, see Exercise P-6.3. The above algorithm although not competitive with the more

elaborate versions that use some form of preconditioning, will serve as a good model of a deflation process combined with Arnoldi's projection.

Eig.	Mat-Vec's	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
2	60	0.9370509474	0.0	0.870D-03
	69	0.9371549617	0.0	0.175D-04
	78	0.9371501442	0.0	0.313D-06
	87	0.9371501564	0.0	0.490D-08
3	96	0.8112247133	0.0	0.210D-02
	104	0.8097553450	0.0	0.538D-03
	112	0.8096419483	0.0	0.874D-04
	120	0.8095810281	0.0	0.181D-04
	128	0.8095746489	0.0	0.417D-05
	136	0.8095721868	0.0	0.753D-06
	144	0.8095718575	0.0	0.231D-06
	152	0.8095717167	0.0	0.444D-07

Table 6.3: Convergence of the three rightmost eigenvalues computed from a deflated Arnoldi procedure for $A = \mathrm{Mark}(10)$.

Example 6.3. We will use once more the test matrix Mark(10) for illustration. Here we test our restarted and deflated Arnoldi procedure for computing the three eigenvalues with algebraically largest real part. We use $m = 10$ as in the previous example. We do not show the run corresponding to the first eigenvalue since the data is already listed in Table 6.2. The first column shows the eigenvalue being computed. Thus, it takes five outer iterations to compute the first eigenvalue (see example 6.2), 4 outer iterations to compute the second one, and finally 8 outer iterations to get the third one. The convergence towards the last eigenvalue is slower than for the first two. This could be attributed to poorer separation of λ_3 from the other eigenvalues but also to the fact that m has implicitly decreased from $m = 10$ when computing the first eigenvalue to $m = 8$ when computing the third one. □

6.3 The Hermitian Lanczos Algorithm

The Hermitian Lanczos algorithm can be viewed as a simplification of Arnoldi's method for the particular case when the matrix is Hermitian. The principle of the method is therefore the same in that it is a projection technique on a Krylov subspace. However, there are a number of interesting properties that will cause the algorithm to simplify. On the theoretical side there is also much more that can be said on the Lanczos algorithm than there is on Arnoldi's method.

6.3.1 The Algorithm

To introduce the algorithm we start by making the observation stated in the following theorem.

Theorem 6.2 *Assume that Arnoldi's method is applied to a Hermitian matrix A. Then the coefficients h_{ij} generated by the algorithm are real and such that*

$$h_{ij} = 0, \quad \text{for} \quad 1 \leq i < j-1\ , \tag{6.13}$$

$$h_{j,j+1} = h_{j+1,j}\ ,\ j = 1, 2, \ldots, m. \tag{6.14}$$

In other words the matrix H_m obtained from the Arnoldi process is real, tridiagonal, and symmetric.

Proof. The proof is an immediate consequence of the fact that $H_m = V_m^H A V_m$ is a Hermitian matrix which is also a Hessenberg matrix by construction. Therefore, H_m must be a Hermitian tridiagonal matrix. In addition, observe that by its definition the scalar $h_{j+1,j}$ is real and that $h_{jj} = (Av_j, v_j)$ is also real if A is Hermitian. Therefore, since the matrix H_m is of Hessenberg form, it is real, tridiagonal and symmetric. □

The standard notation used to describe the Lanczos algorithm, is obtained by setting

$$\begin{aligned} \alpha_j &\equiv h_{jj}\ , \\ \beta_j &\equiv h_{j-1,j}\ , \end{aligned}$$

which leads to the following form of the Modified Gram Schmidt variant of Arnoldi's method, namely Algorithm 6.2.

ALGORITHM **6.5 The Lanczos Algorithm**

1. **Start:** *Choose an initial vector v_1 of norm unity. Set $\beta_1 \equiv 0, v_0 \equiv 0$.*
2. **Iterate:** *for $j = 1, 2, \ldots, m$ do*

$$w_j := Av_j - \beta_j v_{j-1} \tag{6.15}$$

$$\alpha_j := (w_j, v_j) \tag{6.16}$$

$$w_j := w_j - \alpha_j v_j \tag{6.17}$$

$$\beta_{j+1} := \|w_j\|_2 \tag{6.18}$$

$$v_{j+1} := w_j / \beta_{j+1} \tag{6.19}$$

An important and rather surprising property is that the above simple algorithm guarantees, at least in exact arithmetic, that the vectors $v_i, i = 1, 2, \ldots,$ are orthogonal. In reality, exact orthogonality of these vectors is only observed at the

beginning of the process. Ultimately, the v_i's start losing their global orthogonality very rapidly. There has been much research devoted to finding ways to either recover the orthogonality, or to at least diminish its effects by *partial* or *selective* orthogonalization, see Parlett [148].

The major practical differences with Arnoldi's method are that the matrix H_m is tridiagonal and, more importantly, that we only need to save three vectors, at least if we do not resort to any form of reorthogonalization.

6.3.2 Relation with Orthogonal Polynomials

In exact arithmetic the equation (6.17) in the algorithm takes the form

$$\beta_{j+1}v_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}.$$

This three term recurrence relation is reminiscent of the standard three term recurrence relation of orthogonal polynomials. In fact as we will show in this section, there is indeed a strong relationship between the Lanczos algorithm and orthogonal polynomials. We start by recalling that if the grade of v_1 is $\geq m$ then the subspace $\mathcal{K}_m$ is of dimension m and consists of all vectors of the form $q(A)v_1$ with $\mathrm{degree}(q) \leq m-1$. In this case there is even an isomorphism between $\mathcal{K}_m$ and $\mathbb{P}_{m-1}$, the space of polynomials of degree $\leq m-1$, which is defined by

$$q \in \mathbb{P}_{m-1} \rightarrow x = q(A)v_1 \in \mathcal{K}_m$$

Moreover, we can consider that the subspace $\mathbb{P}_{m-1}$ is provided with the inner product

$$< p, q >_{v_1} = (p(A)v_1, q(A)v_1) \tag{6.20}$$

which is indeed a nondegenerate bilinear form under the assumption that m does not exceed μ, the grade of v_1. Now observe that the vectors v_i are of the form

$$v_i = q_{i-1}(A)v_1$$

and the orthogonality of the v_i's translates into the orthogonality of the polynomials with respect to the inner product (6.20). Moreover, the Lanczos procedure is nothing but the Stieltjes algorithm (see, for example, Gautschi [70]) for computing a sequence of orthogonal polynomials with respect to the inner product (6.20). From Theorem 6.1 the characteristic polynomial of the tridiagonal matrix produced by the Lanczos algorithm minimizes the norm $\|.\|_{v_1}$ over the monic polynomials. It is easy to prove by using a well-known recurrence for determinants of tridiagonal matrix, that the Lanczos recurrence computes the characteristic polynomial of H_m times the initial vector v_1. This is another way of relating the v_i's to the orthogonal polynomials.

6.4 Non-Hermitian Lanczos Algorithm

This is an extension of the algorithm seen in the previous section to the non-Hermitian case. We already know of one such extension namely Arnoldi's procedure which is an orthogonal projection method. However, the non-Hermitian

Lanczos algorithm is an oblique projection technique and is quite different in concept from Arnoldi's method.

6.4.1 The Algorithm

The algorithm proposed by Lanczos for non-Hermitian matrices differs from Arnoldi's method in one essential way: instead of building an orthogonal basis of $\mathcal{K}_m$, it builds a pair of biorthogonal bases for the two subspaces

$$\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$$

and

$$\mathcal{K}_m(A^H, w_1) = \text{span}\{w_1, A^H w_1, \ldots, (A^H)^{m-1}w_1\}.$$

The algorithm to achieve this is as follows.

ALGORITHM **6.6 The non-Hermitian Lanczos Algorithm**

1. **Start:** *Choose two vectors* v_1, w_1 *such that* $(v_1, w_1) = 1$. *Set* $\beta_1 \equiv 0, w_0 = v_0 \equiv 0$.
2. **Iterate:** *for* $j = 1, 2, \ldots, m$ *do*

$$\alpha_j = (Av_j, w_j) \tag{6.21}$$
$$\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1} \tag{6.22}$$
$$\hat{w}_{j+1} = A^H w_j - \bar{\alpha}_j w_j - \delta_j w_{j-1} \tag{6.23}$$
$$\delta_{j+1} = |(\hat{v}_{j+1}, \hat{w}_{j+1})|^{1/2} \tag{6.24}$$
$$\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})/\delta_{j+1} \tag{6.25}$$
$$w_{j+1} = \hat{w}_{j+1}/\overline{\beta_{j+1}} \tag{6.26}$$
$$v_{j+1} = \hat{v}_{j+1}/\delta_{j+1} \tag{6.27}$$

We should point out that there is an infinity of ways of choosing the scalars $\delta_{j+1}, \beta_{j+1}$ in (6.24)–(6.25). These two parameters are scaling factors for the two vectors v_{j+1} and w_{j+1} and can be selected in any manner to ensure that $(v_{j+1}, w_{j+1}) = 1$. As a result of (6.26), (6.27) all that is needed is to choose two scalars $\beta_{j+1}, \delta_{j+1}$ that satisfy the equality

$$\delta_{j+1}\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1}) \tag{6.28}$$

The choice made in the above algorithm attempts to scale the two vectors so that they are divided by two scalars having the same modulus. Thus, if initially v_1 and w_1 have the same norm, all of the subsequent v_i's will have the same norms as the w_i's. As was advocated in [154], on can scale both vectors by their 2-norms. In this case the inner product of v_i and w_i is no longer equal to one but a modified algorithm can be written with these constraints. In this situation a generalized eigenvalue problem $T_m z = \lambda D_m z$ must be solved to compute the Ritz values

where D_m is a diagonal matrix, whose entries are the inner products (v_i, w_i). The modified algorithm is the subject of Exercise P-6.9.

In what follows we will place ourselves in the situation where the pair of scalars $\delta_{j+1}, \beta_{j+1}$ is *any pair that satisfies the relation* (6.28), instead of restricting ourselves to the particular case defined by (6.24) – (6.25). A consequence is that δ_j can be complex and in fact the formula defining $\hat{w}_{j+1}$ in (6.23) should then be modified to

$$\hat{w}_{j+1} = A^H w_j - \bar{\alpha}_j w_j - \bar{\delta}_j w_{j-1} \ .$$

We will denote by T_m the tridiagonal matrix

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & . & . & . & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix} .$$

Note that in the particular case where A is real as well as the initial vectors v_1, w_1, and if (6.24) – (6.25) are used then the δ_j's are real positive and $\beta_j = \pm\delta_j$.

Our first observation from the algorithm is that the vectors v_i belong to $\mathcal{K}_m(A, v_1)$ while the w_j 's are in $\mathcal{K}_m(A^H, w_1)$. In fact we can show the following proposition.

Proposition 6.9 *If the algorithm does not break down before step m then the vectors $v_i, i = 1, \ldots, m$, and $w_j, j = 1, \ldots, m$, form a biorthogonal system, i.e.,*

$$(v_j, w_i) = \delta_{ij} \quad 1 \leq i,\ j \leq m \ .$$

Moreover, $\{v_i\}_{i=1,2,\ldots,m}$ is a basis of $\mathcal{K}_m(A, v_1)$ and $\{w_i\}_{i=1,2,\ldots,m}$ is a basis of $\mathcal{K}_m(A^H, w_1)$ and we have the relations,

$$AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^H, \tag{6.29}$$

$$A^H W_m = W_m T_m^H + \bar{\beta}_{m+1} w_{m+1} e_m^H, \tag{6.30}$$

$$W_m^H A V_m = T_m \ . \tag{6.31}$$

Proof. The biorthogonality of the vectors v_i, w_i will be shown by induction. By assumption $(v_1, w_1) = 1$. Assume now that the vectors $v_1, \ldots v_j$ and $w_1, \ldots w_j$ are biorthogonal, and let us establish that the vectors $v_1, \ldots v_{j+1}$ and $w_1, \ldots w_{j+1}$ are biorthogonal.

We show first that $(v_{j+1}, w_i) = 0$ for $i \leq j$. When $i = j$ we have

$$(v_{j+1}, w_j) = \delta_{j+1}^{-1}[(Av_j, w_j) - \alpha_j(v_j, w_j) - \beta_j(v_{j-1}, w_j)] \ .$$

The last inner product in the above expression vanishes by the induction hypothesis. The two other terms cancel each other by the definition of α_j and the fact that $(v_j, w_j) = 1$. Consider now

$$(v_{j+1}, w_{j-1}) = \delta_{j+1}^{-1}[(Av_j, w_{j-1}) - \alpha_j(v_j, w_{j-1}) - \beta_j(v_{j-1}, w_{j-1})] \ .$$

Again from the induction hypothesis the middle term in the right hand side vanishes. The first term can be rewritten as

$$\begin{aligned}(Av_j, w_{j-1}) &= (v_j, A^H w_{j-1}) \\ &= (v_j, \bar{\beta}_j w_j + \bar{\alpha}_{j-1} w_{j-1} + \bar{\delta}_{j-1} w_{j-2}) \\ &= \beta_j (v_j, w_j) + \alpha_{j-1}(v_j, w_{j-1}) + \delta_{j-1}(v_j, w_{j-2}) \\ &= \beta_j\end{aligned}$$

and as a result,

$$(v_{j+1}, w_{j-1}) = \delta_{j+1}^{-1}[(Av_j, w_{j-1}) - \beta_j (v_{j-1}, w_{j-1})] = 0 \,.$$

More generally, consider an inner product (v_{j+1}, w_i) with $i < j-1$,

$$\begin{aligned}(v_{j+1}, w_i) &= \delta_{j+1}^{-1}[(Av_j, w_i) - \alpha_j (v_j, w_i) - \beta_j (v_{j-1}, w_i)] \\ &= \delta_{j+1}^{-1}[(v_j, A^H w_i) - \alpha_j (v_j, w_i) - \beta_j (v_{j-1}, w_i)] \\ &= \delta_{j+1}^{-1}[(v_j, \bar{\beta}_{i+1} w_{i+1} + \bar{\alpha}_i w_i + \bar{\delta}_i w_{i-1}) - \alpha_j (v_j, w_i) \\ &\quad -\beta_j (v_{j-1}, w_i)] \,.\end{aligned}$$

By the induction hypothesis, all of the inner products in the above expression vanish. We can show in the same way that $(v_i, w_{j+1}) = 0$ for $i \leq j$. Finally, we have by construction $(v_{j+1}, w_{j+1}) = 1$. This completes the induction proof.

The proof of the other matrix relations is identical with the proof of the similar relations in Arnoldi's method. □

The relation (6.31) is key to understanding the nature of the method. From what we have seen in Chapter 4 on general projection methods, the matrix T_m is exactly the projection of A obtained from an oblique projection process onto $\mathcal{K}_m(A, v_1)$ and orthogonally to $\mathcal{K}_m(A^H, w_1)$. The approximate eigenvalues $\lambda_i^{(m)}$ provided by this projection process are the eigenvalues of the tridiagonal matrix T_m. A Ritz approximate eigenvector of A associated with $\lambda_i^{(m)}$ is defined by $u_i^{(m)} = V_m y_i^{(m)}$ where $y_i^{(m)}$ is an eigenvector associated with the eigenvalue $\lambda_i^{(m)}$ of T_m. Similarly to Arnoldi's method, a number of the Ritz eigenvalues, typically a small fraction of m, will constitute good approximations of corresponding eigenvalues λ_i of A and the quality of the approximation will improve as m increases.

We should mention that the result of Proposition 6.8, which gives a simple and inexpensive way to compute residual norms can readily be extended as follows:

$$(A - \lambda_i^{(m)} I) u_i^{(m)} = \delta_{m+1} e_m^H y_i^{(m)} v_{m+1} \tag{6.32}$$

and, as a result $\|(A - \lambda_i^{(m)} I) u_i^{(m)}\|_2 = |\delta_{m+1} e_m^H y_i^{(m)}| \,.$

An interesting new feature here is that the operators A and A^H play a dual role in that we perform similar operations with them. We can therefore expect that if we get good approximate eigenvectors for A we should in general get as

good approximations for the eigenvectors of A^H. In fact we can also view the non-Hermitian Lanczos procedure as a method for approximating eigenvalues and eigenvectors of the matrix A^H by a projection method onto $L_m = \mathcal{K}(A^H, w_1)$ and orthogonally to $\mathcal{K}_m(A, v_1)$. As a consequence, the left and right eigenvectors of A will both be well approximated by the process. In contrast Arnoldi's method only computes approximations to the right eigenvectors. The approximations to the left eigenvectors are of the form $W_m z_i^{(m)}$ where $z_i^{(m)}$ is a left eigenvector of T_m associated with the eigenvalue $\lambda_i^{(m)}$. This constitutes one of the major differences between the two methods. There are applications where both left and right eigenvectors are needed. In addition, when estimating errors and condition numbers of the computed eigenpair it might be crucial that both the left and the right eigenvectors be available.

From the practical point of view, another big difference between the non-Hermitian Lanczos procedure and the Arnoldi methods is that we now only need to save a few vectors in memory to execute the algorithm if no reorthogonalization is performed. More precisely, we need 6 vectors of length n plus some storage for the tridiagonal matrix, no matter how large m is. This is clearly a significant advantage.

On the other hand there are more risks of breakdown with the non-Hermitian Lanczos method. The algorithm will break down whenever $(\hat{v}_{j+1}, \hat{w}_{j+1}) = 0$ which can be shown to be equivalent to the existence of a vector in $\mathcal{K}_m(A, v_1)$ that is orthogonal to the subspace $\mathcal{K}_m(A^H, w_1)$. In fact this was seen to be a necessary and sufficient condition for the oblique projector onto $\mathcal{K}_m(A, v_1)$ orthogonally to $\mathcal{K}_m(A^H, w_1)$ not to exist. In the case of Arnoldi's method a breakdown is actually a favorable situation since we are guaranteed to obtain exact eigenvalues in this case as was seen before. The same is true in the case of the Lanczos algorithm when either $\hat{v}_{j+1} = 0$ or $\hat{w}_{j+1} = 0$. However, when $\hat{v}_{j+1} \neq 0$ and $\hat{w}_{j+1} \neq 0$ then this is non-longer true. In fact the serious problem is not as much caused by the exact occurrence of this phenomenon which Wilkinson [222] calls *serious breakdown,* , as it is its near occurrence. A look at the algorithm indicates that we may have to scale the Lanczos vectors by small quantities when this happens and the consequence after a number of steps may be serious. This is further discussed in the next subsection.

Since the subspace from which the approximations are taken is identical with that of Arnoldi's method, we have the same bounds for the distance $\|(I - \mathcal{P}_m)u_i\|_2$. However, this does not mean in any way that the approximations obtained by the two methods are likely to be of similar quality. One of the weaknesses of the method is that it relies on oblique projectors which may suffer from poor numerical properties. Moreover, the theoretical bounds shown in Chapter 4 do indicate that the norm of the projector may play a significant role. The method has been used successfully by Cullum and Willoughby [34, 33] to compute eigenvalues of very large matrices. We will discuss these implementations in the next section.

6.4.2 Practical Implementations

There are various ways of improving the standard non-Hermitian Lanczos algorithm which we now discuss briefly. A major focus of researchers in this area is to find ways of circumventing the potential breakdowns or 'near breakdowns' in the algorithm. Other approaches do not attempt to deal with the breakdown but rather try to live with it. We will weigh the pros and cons of both approaches after we describe the various existing scenarios.

Look-Ahead Lanczos Algorithms. As was already mentioned, a problem with the Lanczos algorithm is the potential of breakdown in the normalization steps (6.26) and (6.27). Such a break down will occur whenever

$$(\hat{v}_{j+1}, \hat{w}_{j+1}) = 0, \tag{6.33}$$

which can arise in two different situations. Either one of the two vectors $\hat{v}_{j+1}$ or $\hat{w}_{j+1}$ vanishes or they are both nonzero but their inner product is zero. In the first case, we have again the 'lucky breakdown' scenario which we have seen in the case of Hermitian matrices. Thus, if $\hat{v}_{j+1} = 0$ then $\mathrm{span}\{V_j\}$ is invariant and all approximate eigenvalues and associated right eigenvectors will be exact, while if $\hat{w}_{j+1} = 0$ then $\mathrm{span}\{W_j\}$ will be invariant and the approximate eigenvalues and associated left eigenvectors will be exact. The second case, when neither of the two vectors is zero but their inner product is zero is termed *serious breakdown* by Wilkinson (see [222], p. 389). Fortunately, there are some cures, that will allow one to continue the algorithm in most cases. The corresponding modifications of the algorithm are often put under the denomination *Look-Ahead Lanczos* algorithms . There are also rare cases of 'incurable' breakdowns which will not be discussed here (see [154] and [209]). The main idea of Look-Ahead variants of the Lanczos algorithm is that even though the pair v_{j+1}, w_{j+1} cannot be defined it is often the case that the pair v_{j+2}, w_{j+2} can be defined. The algorithm can then be pursued from that iterate as before until a new breakdown is encountered. If the pair v_{j+2}, w_{j+2} cannot be defined then one can try the pair v_{j+3}, w_{j+3} and so on.

To be more precise on why this is possible, we need to go back to the connection with orthogonal polynomials mentioned earlier for the Hermitian case. We can extend the relationship to the non-Hermitian case by defining the bilinear form on the subspace $\mathbb{P}_{m-1}$

$$< p, q >= (p(A)v_1, q(A^H)w_1). \tag{6.34}$$

Unfortunately, this can constitute an 'indefinite inner product' since $< p, p >$ can now be zero or even negative. We note that there is a polynomial p_j of degree j such that $\hat{v}_{j+1} = p_j(A)v_1$ and in fact the same polynomial intervenes in the equivalent expression of w_{j+1}. More precisely, there is a scalar γ_j such that $\hat{w}_{j+1} = \gamma_j p_j(A^H)v_1$. Similarly to the Hermitian case the non-Hermitian Lanczos algorithm attempts to compute a sequence of polynomials that are orthogonal

with respect to the indefinite inner product defined above. If we define the moment matrix

$$M_k = \{< x^{i-1}, x^{j-1} >\}_{i,j=1\ldots k}$$

then this process is mathematically equivalent to finding a factorization

$$M_k = L_k U_k$$

of the moment matrix M_k, in which U_k is upper triangular and L_k is lower triangular. Note that this matrix is a Hankel matrix, i.e., a_{ij} is constant for $i + j = constant$.

Because

$$< p_j, p_j >= \bar{\gamma}_j (p_j(A)v_1, p_j(A^H)w_1)$$

we observe that there is a serious breakdown at step j if and only if the indefinite norm of the polynomial p_j at step j vanishes. The main idea of the Look-Ahead Lanczos algorithms is that if we skip this polynomial it may still be possible to compute p_{j+1} and continue to generate the sequence. To explain this simply, we consider

$$q_j(x) = xp_{j-1} \quad \text{and} \quad q_{j+1}(x) = x^2 p_{j-1}(x) \ .$$

It is easy to verify that both q_j and q_{j+1} are orthogonal to the polynomials p_1, $\cdots$, p_{j-2}. We can, for example, define (somewhat arbitrarily) $p_j = q_j$, and get p_{j+1} by orthogonalizing q_{j+1} against p_{j-1} and p_j. It is clear that the resulting polynomial will then be orthogonal against all polynomials of degree $\leq j$, see Exercise P-6.11. Therefore we can continue the algorithm from step $j + 1$ in the same manner. Exercise P-6.11 generalizes this to the case where we need to skip k polynomials rather than just one. This simplistic description gives the main mechanism that lies behind the different versions of Look-Ahead Lanczos algorithms proposed in the literature. In the Parlett-Taylor-Liu implementation [154], it is observed that the reason for the break down of the algorithm is that the pivots encountered during the LU factorization of the moment matrix M_k vanish. Divisions by zero are then avoided by *implicitly* performing a pivot with a 2×2 matrix rather than a using a 1×1 pivot.

The drawback of Look-Ahead implementations is the nonnegligible added complexity. In addition to the difficulty of deciding when to consider that one has a near break-down situation, one must cope with the fact that the matrix T_m is no longer tridiagonal. It is easy to see that whenever a step is skipped, we introduce a 'bump', as it it termed in [154], above the superdiagonal element. This further complicates the issue of the computation of the eigenvalues of the Ritz values.

The Issue of Reorthogonalization. Just as in the Hermitian case, the vectors w_j and v_i will tend to lose their bi-orthogonality. Techniques that perform some form of 'partial' or 'selective' reorthogonalization can be developed for non-Hermitian Lanczos algorithm as well. One difficulty here is that selective orthogonalization, which typically requires eigenvectors, will suffer from the fact

that eigenvectors may be inaccurate. Another problem is that we now have to keep two sets of vectors, typically in secondary storage, instead of only one.

An alternative to reorthogonalization is to live with the loss of orthogonality. Although the theory is not as well understood in the non-Hermitian case as it is in the Hermitian case, it has been observed that despite the loss of orthogonality, convergence is still observed in general, at the price of a few practical difficulties. More precisely, a converged eigenvalue may appear several times, and monitoring extraneous eigenvalues becomes important. Cullum and Willoughby [36] suggest precisely such a technique based on a few heuristics. The technique is based on a comparison of the eigenvalues of the successive tridiagonal matrices T_k.

6.5 Block Krylov Methods

In many circumstances it is desirable to work with a block of vectors instead of a single vector. For example, in out-of core finite-element codes it is a good strategy to exploit the presence of a block of the matrix A in fast memory, as much as possible. This can easily done with a method such as the subspace iteration for example, but not the usual Arnoldi/Lanczos algorithms. In essence, the block Arnoldi method is to the Arnoldi method what the subspace iteration is to the usual power method. Thus, the block Arnoldi can be viewed as an acceleration of the subspace iteration method. There are several possible implementations of the algorithm three of which are described next.

ALGORITHM **6.7 Block Arnoldi**

1. Start: *Choose a unitary matrix V_1 of dimension $n \times r$.*

2. Iterate: *for $j = 1, 2, \ldots, m$ compute:*

$$H_{ij} = V_i^H A V_j \quad i = 1, 2, \ldots, j, \tag{6.35}$$

$$W_j = AV_j - \sum_{i=1}^{j} V_i H_{ij} \ , \tag{6.36}$$

$$W_j = V_{j+1} H_{j+1,j} \quad \textit{Q-R decomposition of } W_j. \tag{6.37}$$

The above algorithm is a straightforward block analogue of Algorithm 6.1. By construction, the blocks constructed by the algorithm will be orthogonal blocks that are orthogonal to each other. In what follows we denote by I_k the $k \times k$ identity matrix and use the following notation

$$\begin{aligned} U_m &= [V_1, V_2, \ldots, V_m] \ , \\ H_m &= (H_{ij})_{1 \le i,j \le m}, \quad H_{ij} \equiv 0, \quad i > j+1 \ , \\ E_m &= \text{matrix of the last } r \text{ columns of } I_{nr}. \end{aligned}$$

Then, the analogue of the relation (6.8) is

$$AU_m = U_m H_m + V_{m+1} H_{m+1,m} E_m^H.$$

Thus, we obtain a relation analogous to the one we had before except that the matrix H_m is no longer Hessenberg but band-Hessenberg, in that we have $r-1$ additional diagonals below the subdiagonal.

A second version of the algorithm would consist of using a modified block Gram-Schmidt procedure instead of the simple Gram-Schmidt procedure used above. This leads to a block generalization of Algorithm 6.2, the Modified Gram-Schmidt version of Arnoldi's method.

ALGORITHM **6.8 Block Arnoldi – MGS version**

1. **Start:** *Choose a unitary matrix V_1 of size $n \times r$.*
2. **Iterate:** *For $j = 1, 2, \ldots, m$ do:*
   - *Compute* $W_j := AV_j$
   - *For* $i = 1, 2, \ldots, j$ *do:*

     $$H_{ij} := V_i^H W_j$$
     $$W_j := W_j - V_j H_{ij}.$$

   - *Compute the Q-R decomposition* $W_j = V_{j+1} H_{j+1,j}$

Again, in practice the above algorithm is more viable than its predecessor. Finally, a third version, developed by A. Ruhe, see reference [164], for the symmetric case (Block Lanczos algorithm), yields an algorithm that is quite similar to the original Arnoldi algorithm.

ALGORITHM **6.9 Block Arnoldi - Ruhe's variant**

1. **Start:** *Choose r initial orthonormal vectors $\{v_i\}_{i=1,\ldots,r}$.*
2. **Iterate:** *for $j = r, r+1, , \ldots, m \times r$ do:*
   (a) *Set $k := j - r + 1$;*
   (b) *Compute $w := Av_k$;*
   (c) *For $i = 1, 2, \ldots, j$ do*
      - $h_{i,k} := (w, v_i)$
      - $w := w - h_{i,k} v_i$
   (d) *Compute $h_{j+1,k} := \|w\|_2$ and $v_{j+1} := w/h_{j+1,k}$.*

Observe that the particular case $r = 1$ coincides with the usual Arnoldi process. That the two algorithms 6.8 and 6.9 are mathematically equivalent is straightforward to show. The advantage of the above algorithm, is its simplicity. On the other hand a slight disadvantage is that we give up some potential for parallelism. In the original version the columns of the matrix AV_j can be computed in parallel whereas in the new algorithm, we must compute them in sequence.

Generally speaking, the block methods are of great practical value in some applications but they are not as well studied from the theoretical point of view. One of the reasons is possibly the lack of any convincing analogue of the relationship with orthogonal polynomials established in Subsection 6.3.2 for the single vector Lanczos algorithm. We have not covered the block versions of the two Lanczos algorithms (Hermitian and non-Hermitian) but these generalizations are straightforward.

6.6 Convergence of the Lanczos Process

In this section we examine the convergence properties of the Hermitian Lanczos algorithm, from a theoretical point of view. Well-known results from approximation theory will be used to derive a convergence analysis of the method. In particular Chebyshev polynomials play an important role and we refer the readers to the end of Chapter 4 for some background on these polynomials.

6.6.1 Distance between $\mathcal{K}_m$ and an Eigenvector

In the following we will assume that the eigenvalues of the Hermitian matrix A are labeled in decreasing order, i.e.,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \ ,$$

and that the approximate eigenvalues are labeled similarly. We will now state the main result of this section, starting with the following lemma.

Lemma 6.1 *Let P_i be the spectral projector associated with the eigenvalue λ_i. Then, if $P_i v_1 \neq 0$, we have*

$$\tan\theta(u_i, \mathcal{K}_m) = \min_{p\in\mathbb{P}_{m-1},\ p(\lambda_i)=1} \|p(A)y_i\|_2 \ \tan\theta(u_i, v_1) \tag{6.38}$$

in which

$$y_i = \begin{cases} \frac{(I-P_i)v_1}{\|(I-P_i)v_1\|_2} & \text{if} \quad (I-P_i)v_1 \neq 0 \ , \\ 0 & \text{otherwise.} \end{cases}$$

Proof. The subspace $\mathcal{K}_m$ consists of all vectors of the form $x = q(A)v_1$ where q is any polynomial of degree $\leq m-1$. We have the orthogonal decomposition

$$x = q(A)v_1 = q(A)P_i v_1 + q(A)(I-P_i)v_1$$

and the angle between x and u_i is defined by

$$\begin{aligned} \tan\theta(x, u_i) &= \frac{\|q(A)(I-P_i)v_1\|_2}{\|q(A)P_i v_1\|_2} \\ &= \frac{\|q(A)y_i\|_2}{|q(\lambda_i)|}\frac{\|(I-P_i)v_1\|_2}{\|P_i v_1\|_2} \ . \end{aligned}$$

If we let $p(\lambda) \equiv q(\lambda)/q(\lambda_i)$ we get

$$\tan\theta(x, u_i) = \|p(A)y_i\|_2 \ \tan\theta(v_1, u_i)$$

which shows the result by taking the minimum over all x's in $\mathcal{K}_m$. □

Theorem 6.3 *Let the eigenvalues λ_i of A be ordered decreasingly. Then the angle $\theta(u_i, \mathcal{K}_m)$ between the exact eigenvector u_i associated with λ_i and the $m-th$ Krylov subspace $\mathcal{K}_m$ satisfies the inequality,*

$$\tan\theta(u_i, \mathcal{K}_m) \leq \frac{\kappa_i}{C_{m-i}(1+2\gamma_i)} \ \tan\theta(v_1, u_i)\ , \tag{6.39}$$

where

$$\kappa_1 = 1,\ \kappa_i = \prod_{j=1}^{i-1} \frac{\lambda_j - \lambda_n}{\lambda_j - \lambda_i} \quad \textit{for}\ \ i > 1 \tag{6.40}$$

and,

$$\gamma_i = \frac{\lambda_i - \lambda_{i+1}}{\lambda_{i+1} - \lambda_n}\ . \tag{6.41}$$

Proof. To prove the theorem for the case $i = 1$ we start by expanding the vector y_i defined in the previous lemma in the eigenbasis $\{u_j\}$ as

$$y_1 = \sum_{j=2}^{n} \alpha_j u_j$$

where the α_j's are such that $\sum_{j=2}^{n} |\alpha_j|^2 = 1$. From this we get,

$$\|p(A)y_1\|_2^2 = \sum_{j=2}^{n} |p(\lambda_j)\alpha_j|^2 \leq \max_{j=2,\ldots,n} |p(\lambda_j)|^2 \leq \max_{\lambda\in[\lambda_n,\lambda_2]} |p(\lambda)|^2\ .$$

The result follows by a direct use of theorem 4.8 stated in Chapter 4. For the general case ($i \neq 1$), we can use the upper bound obtained by restricting the polynomials to be of the form

$$p(\lambda) = \frac{(\lambda_1 - \lambda)\cdots(\lambda_{i-1} - \lambda)}{(\lambda_1 - \lambda_i)\cdots(\lambda_{i-1} - \lambda_i)}\, q(\lambda)$$

where q is now any polynomial of degree $k - i$ such that $q(\lambda_i) = 1$. Proceeding as for the case $i = 1$, we arrive at the inequality,

$$\begin{aligned} \|p(A)y_i\|_2 &\leq \max_{\lambda\in[\lambda_{i+1},\lambda_n]} \left| \prod_{j=1}^{i-1} \frac{\lambda_j - \lambda}{\lambda_j - \lambda_i} q(\lambda) \right| \\ &\leq \prod_{j=1}^{i-1} \frac{\lambda_j - \lambda_n}{\lambda_j - \lambda_i} \max_{\lambda\in[\lambda_{i+1},\lambda_n]} |q(\lambda)|\ . \end{aligned}$$

The result follows by minimizing this expression over all polynomials q satisfying thr constraint $q(\lambda_i) = 1$. □

6.6.2 Convergence of the Eigenvalues

We now turn our attention to the approximate eigenvalues. The following error bounds concerning the approximate eigenvalues $\lambda_i^{(m)}$ actually show that these converge to the corresponding eigenvalues of A if exact arithmetic were used.

Theorem 6.4 *The difference between the $i-$th exact and approximate eigenvalues λ_i and $\lambda_i^{(m)}$ satisfies the double inequality,*

$$0 \leq \lambda_i - \lambda_i^{(m)} \leq (\lambda_1 - \lambda_n) \left(\frac{\kappa_i^{(m)} \tan \theta(v_1, u_i)}{C_{m-i}(1 + 2\gamma_i)} \right)^2 \tag{6.42}$$

where γ_i is defined in the previous theorem and $\kappa_i^{(m)}$ is given by

$$\kappa_1^{(m)} \equiv 1, \quad \kappa_i^{(m)} = \prod_{j=1}^{i-1} \frac{\lambda_j^{(m)} - \lambda_n}{\lambda_j^{(m)} - \lambda_i}, \quad i > 1 \, .$$

Proof. We prove the result only for the case $i = 1$. The first inequality is one of the properties proved for general projection methods when applied to Hermitian matrices. For the second, we note that

$$\lambda_1^{(m)} = \max_{x \neq 0, x \in \mathcal{K}_{m-1}} (Ax, x)/(x, x)$$

and hence,

$$\lambda_1 - \lambda_1^{(m)} = \min_{x \neq 0 \in \mathcal{K}_{m-1}} ((\lambda_1 I - A)x, x)/(x, x) \, .$$

Remembering that $\mathcal{K}_{m-1}$ is the set of all vectors of the form $q(A)v_1$ where q runs in the space $\mathbb{P}_{m-1}$ of polynomials of degree not exceeding $m - 1$ this becomes

$$\lambda_1 - \lambda_1^{(m)} = \min_{0 \neq q \in \mathbb{P}_{m-1}} \frac{((\lambda_1 - A)q(A)v_1, q(A)v_1)}{(q(A)v_1, q(A)v_1)} \, . \tag{6.43}$$

Expanding the initial vector v_1 in an orthonormal eigenbasis $\{u_j\}$ as

$$v_1 = \sum_{j=1}^{n} \alpha_j u_j$$

we find that

$$\lambda_1 - \lambda_1^{(m)} = \min_{0 \neq q \in \mathbb{P}_{m-1}} \frac{\sum_{j=2}^{n} (\lambda_1 - \lambda_j) |\alpha_j q(\lambda_j)|^2}{\sum_{j=1}^{n} |\alpha_j q(\lambda_j)|^2}$$

from which we obtain the upper bound

$$\begin{aligned}
\lambda_1 - \lambda_1^{(m)} &\leq (\lambda_1 - \lambda_n) \min_{0 \neq q \in \mathbb{P}_{m-1}} \frac{\sum_{j=2}^{n} |\alpha_j q(\lambda_j)|^2}{\sum_{j=1}^{n} |\alpha_j q(\lambda_j)|^2} \\
&\leq (\lambda_1 - \lambda_n) \min_{0 \neq q \in \mathbb{P}_{m-1}} \frac{\sum_{j=2}^{n} |\alpha_j q(\lambda_j)|^2}{|\alpha_1 q(\lambda_1)|^2} \\
&\leq (\lambda_1 - \lambda_n) \min_{0 \neq q \in \mathbb{P}_{m-1}} \max_{j=2,3,\ldots,n} \frac{|q(\lambda_j)|^2}{|q(\lambda_1)|^2} \frac{\sum_{j=2}^{n} |\alpha_j|^2}{|\alpha_1|^2}
\end{aligned}$$

Defining $p(\lambda) = q(\lambda)/q(\lambda_1)$ and observing that the set of all p's when q runs in the space $\mathbb{P}_{m-1}$ is the set of all polynomials of degree not exceeding $m-1$ satisfying the constraint $p(\lambda_1) = 1$, we obtain

$$\lambda_1 - \lambda_1^{(m)} \leq (\lambda_1 - \lambda_n) \min_{p \in \mathbb{P}_{m-1}, p(\lambda_1)=1} \max_{\lambda \in [\lambda_n, \lambda_2]} |p(\lambda)|^2 \tan^2 \theta(u_1, v_1) \; .$$

The result follows by expressing the min-max quantity in the above expression using Chebyshev polynomials according to Theorem 4.8.

The general case $i > 1$ can be proved by using the Courant-Fisher characterization of $\lambda_i^{(m)}$. The i-th eigenvalue is the maximum of the Rayleigh quotient over the subspace of $\mathcal{K}_m$ that is orthogonal to the first $i-1$ approximate eigenvectors. This subspace can be shown to be the same as the subspace of all vectors of the form $q(A)v_1$ where q is a polynomial of degree not exceeding $m-1$ such that $q(\lambda_1^{(m)}) = q(\lambda_2^{(m)}) = \cdots = q(\lambda_{i-1}^{(m)}) = 0$. □

6.6.3 Convergence of the Eigenvectors

To get a bound for the angle between the exact and approximate eigenvectors produced by the Lanczos algorithm, we exploit the general result of Theorem 4.6 seen in Chapter 4. The theorem tells us that for any eigenpair λ_i, u_i of A there is an approximate eigenpair $\tilde{\lambda}, \tilde{u}_i$ such that,

$$\sin\left[\theta(u_i, \tilde{u}_i)\right] \leq \sqrt{1 + \frac{\gamma^2}{\delta_i^2}} \sin\left[\theta(u_i, \mathcal{K}_m)\right] \tag{6.44}$$

were δ_i is the distance between λ_i and the set of approximate eigenvalues other than $\tilde{\lambda}_i$ and $\gamma = \|\mathcal{P}_m A(I - \mathcal{P}_m)\|_2$. We notice that in the present situation we have

$$\begin{aligned}
(I - \mathcal{P}_m)A\mathcal{P}_m &= (I - V_m V_m^H)AV_m V_m^H \\
&= (I - V_m V_m^H)(V_m H_m + \beta_{m+1} v_{m+1} e_m^H) V_m^H \\
&= \beta_{m+1} v_{m+1} v_m^H \; ,
\end{aligned}$$

in which we used the relation (6.8). As a result

$$\gamma = \|\mathcal{P}_m A(I - \mathcal{P}_m)\|_2 = \|(I - \mathcal{P}_m)A\mathcal{P}_m\|_2 = \beta_{m+1} \; .$$

Since the angle between u_i and the Krylov subspace has been majorized in Theorem 6.3, a bound on the angle $\theta(u_i, \tilde{u}_i)$ can be readily obtained by combining these two results. For example, we can write

$$\begin{aligned}\sin\left[\theta(u_i,\tilde{u}_i)\right] &\leq \sqrt{1+\beta_{m+1}^2/\delta_i^2}\,\sin\left[\theta(u_i,\mathcal{K}_m)\right]\\ &\leq \sqrt{1+\beta_{m+1}^2/\delta_i^2}\,\tan\left[\theta(u_i,\mathcal{K}_m)\right]\\ &\leq \frac{\kappa_i\sqrt{1+\beta_{m+1}^2\,/\,\delta_i^2}}{C_{m-i}(1+2\gamma_i)}\,\tan\theta(v_1,u_i)\end{aligned}$$

where the constants κ_i and γ_i are defined in Theorem 6.3.

6.7 Convergence of the Arnoldi Process

In this section we will analyze the speed of convergence of an approximate eigenvalue/ eigenvector obtained by Arnoldi's method to the exact pair. This will be done by considering the distance of a particular eigenvector u_i from the subspace $\mathcal{K}_m$. We will assume for simplicity that A is diagonalizable and define

$$\epsilon_i^{(m)} \equiv \min_{p\in\mathbb{P}^*_{m-1}} \max_{\lambda\in\Lambda(A)-\lambda_i} |p(\lambda)|, \tag{6.45}$$

where $\mathbb{P}^*_{m-1}$ represents the set of all polynomials of degree not exceeding $m-1$ such that $p(\lambda_i)=1$. The following lemma relates the distance $\|(I-\mathcal{P}_m)u_i\|_2$ to the above quantity.

Lemma 6.2 *Assume that A is diagonalizable and that the initial vector v_1 in Arnoldi's method has the expansion $v_1=\sum_{k=1}^{k=n}\alpha_k u_k$ with respect to the eigenbasis $\{u_k\}_{k=1,\ldots,n}$ in which $\|u_k\|_2=1, k=1,2,\ldots,n$ and $\alpha_i\neq 0$. Then the following inequality holds:*

$$\|(I-\mathcal{P}_m)u_i\|_2 \leq \xi_i\epsilon_i^{(m)}$$

where

$$\xi_i=\sum_{\substack{k=1\\k\neq i}}^{n}\frac{|\alpha_j|}{|\alpha_i|}.$$

Proof. From the relation between $\mathcal{K}_m$ and $\mathbb{P}_{m-1}$ we have

$$\begin{aligned}\|(I-\mathcal{P}_m)\alpha_i u_i\|_2 &= \min_{q\in\mathbb{P}_{m-1}}\|\alpha_i u_i - q(A)v_1\|_2\\ &\leq \min_{q\in\mathbb{P}_{m-1},\ q(\lambda_i)=1}\|\alpha_i u_i - q(A)v_1\|_2,\end{aligned}$$

and therefore, calling p the polynomial realizing the minimum on the right-hand-side

$$\|(I-\mathcal{P}_m)\alpha_i u_i\|_2 \leq \|\sum_{\substack{j=1\\j\neq i}}^{n}\alpha_j p(\lambda_j)u_j\|_2 \leq \max_{j\neq i}|p(\lambda_j)|\sum_{\substack{j=1\\j\neq i}}^{n}|\alpha_j|$$

which follows by using the triangle inequality and the fact that the component in the eigenvector u_1 is zero. The result is then established by dividing both members by $|\alpha_i|$. □

The question has been therefore converted into that of estimating the quantity (6.45) on which we will now focus.

Once an estimate for $\epsilon_1^{(m)}$ is available, the above lemma can be invoked to give an idea on the residual of the exact eigenpair with respect to the approximate (projected) problem. See also [204, th. 3.10] for an alternative approach which exploits the spectral decomposition.

What is left to do is to estimate $\epsilon_1^{(m)}$. For the sake of notational convenience, we will consider estimating $\epsilon_1^{(m+1)}$ instead of $\epsilon_1^{(m)}$. The underlying problem is one of approximation theory. For any continuous function f defined on a compact set Ω, denote the uniform norm:

$$\|f\|_\infty = \max_{z \in \Omega} |f(z)| . \tag{6.46}$$

The set Ω will later be taken to be the spectrum of A excluding λ_1. Estimating $\epsilon_1^{(m+1)}$ amounts to finding an upper bound for the distance, in the sense of the inf-norm just defined, between the function $f(z) = 1$ and polynomials of degree $\leq m$ of the form $p(z) = (z - \lambda_1)q(z)$, or, equivalently:

$$\epsilon_1^{(m+1)} = \min_{q \in \mathbb{P}_{m-1}} \|1 - (z - \lambda_1)q(z)\|_\infty .$$

We recall that a subspace S of continuous functions on Ω, generated by k functions $\phi_1, \cdots, \phi_k$ satisfies the Haar condition if each function in S has at most $k-1$ distinct roots. This means that any linear combination of the ϕ_i's vanishes iff it has k distinct roots in Ω. Let f be a continuous function and let p^* be the best uniform approximation of f over Ω. The difference $f - p^*$ reaches its maximum modulus at a number of *extremal points*. The characteristic property [162] of the best approximation states the following.

Theorem 6.5 *Let f be a continuous function and S a k-dimensional subspace of the space of continuous functions on Ω, which satisfies the Haar condition. Then $p^* \in S$ is the best uniform approximation of f over Ω, iff there exist r extremal points $z_i, i = 1, \cdots, r$ in Ω, and positive numbers $\mu_1, \cdots, \mu_r$, with $k+1 \leq r \leq 2k+1$ such that*

$$\sum_{i=1}^{r} \mu_i \overline{[f(z_i) - p^*(z_i)]} \phi(z_i) = 0 \quad \forall\, \phi \in S. \tag{6.47}$$

One important point here is that the number of extremal points is only known to be between $k+1$ and $2k+1$ in the general complex case. That r must be $\geq k+1$ is a consequence of the Haar condition and can be readily verified. When Ω is real, then $r = k+1$. The fact that r is only known to be $\leq 2k+1$ in the complex case, comes from Caratheodory's characterization of convex hulls which expresses a

point in $co(\Omega)$, the convex hull of Ω, as a convex combination of $k+1$ points of Ω in real spaces and $2k+1$ points of Ω in complex spaces.

We will now translate the above result for our situation. Let $\Omega = \Lambda(A)\backslash\{\lambda_1\}$ and $S = \mathrm{span}\{\phi_j(z)\}_{j=1,\cdots,m}$ where $\phi_j(z) = (z-\lambda_1)z^{j-1}$. Then, the dimension of S is m and therefore the theorem states that there are r eigenvalues from the set Ω, with $m+1 \le r \le 2m+1$ such that

$$\sum_{k=1}^{r} \mu_k \overline{[1-(\lambda_{k+1}-\lambda_1)q^*(\lambda_{k+1})]}\phi_j(\lambda_{k+1}) = 0 \quad j = 1,\ldots,m\ .$$

Although we do not know how many extremal points there are we can still express the best polynomial by selecting any set of m extremal points. Assume without loss of generality that these points are labeled from 2 to $m+1$. Let $p^*(z) = (z-\lambda_1)q^*(z)$. We can write $1-p^*(\lambda_k)$ at each of the extremal points λ_k as

$$1 - p^*(\lambda_k) = \rho e^{i\theta_k}$$

where θ_k is a real number and ρ is real and positive. Then it is easily seen that

$$1 - p^*(z) = \frac{\sum_{k=2}^{m+2} e^{i\theta_k} l_k(z)}{\sum_{k=2}^{m+2} e^{i\theta_k} l_k(\lambda_1)}, \tag{6.48}$$

where each $l_k(z)$ is the Lagrange polynomial:

$$l_k(z) = \prod_{\substack{j=2 \\ j\neq k}}^{m+2} \frac{z-\lambda_j}{\lambda_k-\lambda_j}\ . \tag{6.49}$$

Indeed, $1-p^*(z)$, which is of degree m, takes the values $\rho e^{i\theta_k}$ at the $m+1$ points $\lambda_2,\ldots,\lambda_{m+2}$. Therefore it is uniquely determined by the Lagrange formula

$$1 - p^*(z) = \sum_{k=2}^{m+2} \rho e^{i\theta_k} l_k(z)\ .$$

In addition $1-p^*(\lambda_1) = 1$ and this determines ρ as the inverse of $\sum_{k=2}^{m+2} e^{i\theta_k} l_k(\lambda_1)$, yielding the relation (6.48). This establishes the following theorem.

Theorem 6.6 *There are r eigenvalues in $\Omega = \Lambda(A)\backslash\{\lambda_1\}$, where $m+1 \le r \le 2m+1$, at which the optimal polynomial $1-p^*(z) = 1-(z-\lambda_1)q^*(z)$ reaches its maximum value. In addition, given any subset of $m+1$ among these r eigenvalues, which can be labeled $\lambda_2, \lambda_3, \ldots, \lambda_{m+2}$, the polynomial can be represented by (6.48). In particular,*

$$\epsilon_1^{(m+1)} = \frac{1}{\sum_{k=2}^{m+2} e^{i\theta_k} \prod_{\substack{j=2 \\ j\neq k}}^{m+2} \frac{\lambda_1-\lambda_j}{\lambda_k-\lambda_j}}\ . \tag{6.50}$$

Proof. The result was proved above. Note that $\epsilon_1^{(m+1)}$ is equal to ρ which is the inverse of the denominator in (6.48). □

In the first edition of this book, it was shown that when $r = m + 1$, then the sign $e^{i\theta_k}$ in the denominator of (6.50) becomes equal to the conjugate of the sign of $l_k(\lambda_1)$, which is the product term in the denominator of (6.50). In this case, (6.50) simplifies to

$$\epsilon_1^{(m+1)} = \frac{1}{\sum_{k=2}^{m+2} \prod_{\substack{j=2 \\ j\neq k}}^{m+2} \left| \frac{\lambda_1 - \lambda_j}{\lambda_k - \lambda_j} \right|} . \tag{6.51}$$

The result in the first edition of this book was stated incorrectly for the general complex case, because the lemma on which it was based is only valid for real functions. The result holds true only for the situation when the spectrum is real or when it is known that $r = m + 1$ (e.g., when $N = m + 1$).

We denote by $\mathbb{P}_m^*$ the set of polynomials p of degree $\leq m$ such that $p(\lambda_1) = 0$. We seek the best uniform approximation of the function $f(z) = 1$ by polynomials of degree $\leq m$ in $\mathbb{P}_m^*$. Note that $\mathbb{P}_m^*$ is of dimension m. Let the set of r extremal points be $\lambda_2, \ldots, \lambda_{r+1}$ (See theorem 6.5). According to Theorem 6.6, given any subset of $m + 1$ among these r extremal points, which we label $\lambda_2, \lambda_3, \ldots, \lambda_{m+2}$, the best polynomial can be represented by (6.48) in which $e^{i\theta_k} = sign(1 - p^*(\lambda_k))$.

Not much can be said from this result in the general situation. However, when $r = m+1$, then we can determine $\max |1-p^*(z)|$. In this situation the necessary and sufficient conditions of Theorem 6.5 express the extremal points as follows. Let us set $\xi_j \equiv \mu_j[\overline{f(z_j) - p^*(z_j)}]$ for $j = 1, \cdots m + 1$, and select any basis $\phi_1, \cdots, \phi_m$ of the polynomial subspace $\mathbb{P}_m^*$. Then, the condition (6.47) translates to

$$\sum_{k=1}^{m+1} \xi_k \phi_j(\lambda_k) = 0 \quad \text{for} \quad j = 1, \ldots, m. \tag{6.52}$$

The above equations constitute an underdetermined system of linear equations with the unknowns ξ_k. In fact, since the ξ_k's are all nonzero, we can fix any one component, and the rest will then be determined uniquely. This is best done in a more convenient *basis* of polynomials given by:

$$\omega_j(z) = (z - \lambda_1)\hat{l}_j(z), \quad j = 2, \ldots, m + 1, \tag{6.53}$$

where $\hat{l}_j$ is the Lagrange polynomial of degree $m - 1$,

$$\hat{l}_j(z) = \prod_{\substack{k=2 \\ k\neq j}}^{m+1} \frac{z - \lambda_k}{\lambda_j - \lambda_k}, \quad j = 2, \ldots, m + 1. \tag{6.54}$$

With this we can prove the following lemma.

Lemma 6.3 *The underdetermined linear system of m equations and $m+1$ unknowns $\xi_k, k = 2, \ldots, m+2$*

$$\sum_{k=2}^{m+2} \omega_j(\lambda_k)\xi_k = 0, \quad j = 2, 3, \ldots, m+1 \tag{6.55}$$

admits the nontrivial solution

$$\xi_k = \prod_{\substack{j=2 \\ j\neq k}}^{m+2} \frac{\lambda_1 - \lambda_j}{\lambda_k - \lambda_j}, \quad k = 2, \ldots, m+2. \tag{6.56}$$

Proof. Because of the Haar condition, the system of polynomials $\{\omega_j\}_{j=2,\ldots,m+1}$, forms a basis and therefore there exists a nontrivial solution to the above linear system. By the definition of the Lagrange polynomials, all the terms in the $i-$th equation vanish except those corresponding to $j = i$ and to $j = m+2$. Thus, the i^{th} equation can be rewritten as

$$(\lambda_i - \lambda_1)z_i + z_{m+2}(\lambda_{m+2} - \lambda_1) \prod_{\substack{k=2 \\ k\neq i}}^{m+1} \frac{\lambda_{m+2} - \lambda_k}{\lambda_i - \lambda_k} = 0.$$

The unknown z_{m+2} can be assigned an arbitrary nonzero value (since the system is underdetermined) and then the other unknowns are determined uniquely by:

$$\frac{z_i}{z_{m+2}} = -\frac{(\lambda_{m+2} - \lambda_1)}{(\lambda_i - \lambda_1)} \prod_{\substack{k=2, \\ k\neq i}}^{m+1} \frac{\lambda_{m+2} - \lambda_k}{\lambda_i - \lambda_k} = -\prod_{\substack{k=1 \\ k\neq i}}^{m+1} \frac{(\lambda_{m+2} - \lambda_k)}{(\lambda_i - \lambda_k)}.$$

Multiplying numerator and denominator by $(\lambda_i - \lambda_{m+2})$ we get

$$z_i = \frac{C}{\lambda_1 - \lambda_i} \prod_{\substack{k=2 \\ k\neq i}}^{m+2} \frac{1}{\lambda_i - \lambda_k}$$

where C is the following constant, which depends on the choice of z_{m+2},

$$C \equiv z_{m+2} \prod_{k=1}^{m+2} (\lambda_{m+2} - \lambda_k) .$$

The result follows by choosing z_{m+2} so that,

$$C = \prod_{k=2}^{m+2} (\lambda_1 - \lambda_k). \qquad \square$$

We can now prove the desired result.

Theorem 6.7 *Let p^* be the (unique) polynomial of degree m satisfying the constraint $p(\lambda_1) = 0$, and which is the best uniform approximation to the function $f(z) = 1$ on a compact set Ω consisting of at least $m+1$ points. Assume that there are $m+1$ extremal points labeled $\lambda_2, \ldots, \lambda_{m+2}$ and let $\xi_k, k = 2, \ldots, m+2$ be any solution of the linear system (6.55). Write each ξ_k in the form $\xi_k = \delta_k e^{-i\theta_k}$ where δ_k is real and positive and θ_k is real. Then, p^* can be expressed as*

$$1 - p^*(z) = \frac{\sum_{k=2}^{m+2} e^{i\theta_k} l_k(z)}{\sum_{k=2}^{m+2} |l_k(\lambda_1)|}, \tag{6.57}$$

where l_k is the Lagrange polynomial of degree m

$$l_k(z) = \prod_{\substack{j=2 \\ j\neq k}}^{m+2} \frac{z - \lambda_j}{\lambda_k - \lambda_j}.$$

As a consequence,

$$\epsilon_1^{(m+1)} = \left(\sum_{j=2}^{m+1} \prod_{k=2, k\neq j}^{m+1} \frac{|\lambda_k - \lambda_1|}{|\lambda_k - \lambda_j|} \right)^{-1}. \tag{6.58}$$

Proof. Equation (6.47) states that

$$1 - p^*(z) = \rho \sum_{k=2}^{m+2} e^{i\theta_k} l_k(z) \quad \text{with} \quad \rho = \frac{1}{\sum_{k=2}^{m+2} e^{i\theta_k} l_k(\lambda_1)}. \tag{6.59}$$

We now apply Theorem 6.5 which states that at the extremal points (now known to be unique) there are $m+1$ positive coefficients μ_j such that

$$\xi_k \equiv \mu_k \overline{[1 - p^*(\lambda_k)]} = \rho \mu_k e^{-i\theta_k} \tag{6.60}$$

satisfy the system (6.52). As was already mentioned, the solution to (6.52) is uniquely determined if we set any one of its components. Set $\xi_{m+2} = l_{m+2}(\lambda_1)$. Then, according to Lemma 6.3, we must have $\xi_k = l_k(\lambda_1)$, for $k = 2, \ldots, m+2$. Since ρ and μ_k are positive, (6.60) shows that

$$e^{-i\theta_k} = sign(l_k(\lambda_1)) \quad \rightarrow \quad e^{i\theta_k} = \frac{\overline{l_k(\lambda_1)}}{|l_k(\lambda_1)|} \rightarrow \quad \rho = \frac{1}{\sum_{k=2}^{m+2} |l_k(\lambda_1)|}.$$

The result (6.58) is an expanded version of the above expression for ρ. □

For the case where the eigenvalue is in the outermost part of the spectrum, the above expression can be interpreted as follows. In general, the distances $|\lambda_k - \lambda_1|$ are larger than the corresponding distances $|\lambda_k - \lambda_j|$ of the denominator. This

is illustrated in Figure (6.2). Therefore, many of the products will be large when m is large and the inverse of their sum will be small. This is independent of the actual locations of the critical points which are not known. The conclusion is that the eigenvalues that are in the outermost part of the spectrum are likely to be well approximated.

We can illustrate the above theorem with a few examples.

Example 6.4. Assume that

$$\lambda_k = \frac{k-1}{n-1}, \quad k = 1, 2, \ldots n,$$

and consider the special case when $m = n - 1$. Then,

$$\epsilon_1^{(m)} = \frac{1}{2^m - 1}.$$

Indeed, since $m = n - 1$ there is no choice for the λ_j's in the theorem but to be the remaining eigenvalues and (6.58) yields,

$$\begin{aligned}
(\epsilon_1^{(m)})^{-1} &= \sum_{j=2}^{m+1} \prod_{\substack{k=2 \\ k \neq j}}^{m+1} \frac{|k-1|}{|k-j|} \\
&= \sum_{j=2}^{m+1} \frac{m!}{(j-1)!(m+j-1)!} \\
&= \sum_{j=1}^{m} \binom{j}{m} = 2^m - 1
\end{aligned}$$

□

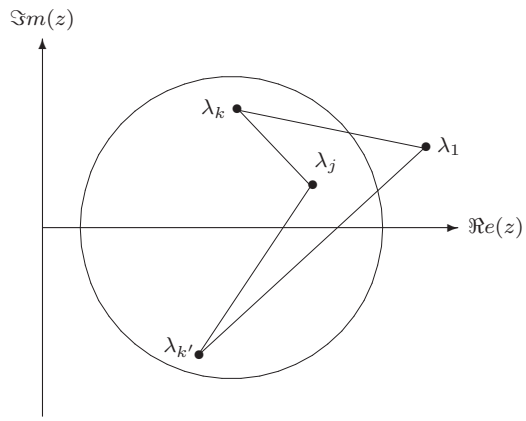


Figure 6.2: Illustration of Theorem 6.7 for λ_1 in the outermost part of the spectrum of A.

Example 6.5. Consider now a uniform distribution of eigenvalues over a circle instead of a real line,

$$\lambda_k = e^{i\frac{2(k-1)\pi}{n}},\ k = 1, 2, \ldots, n\ .$$

We assume once more that $m = n - 1$. Then we have

$$\epsilon_1^{(m)} = \frac{1}{m}\ .$$

To prove the above formula, we utilize again the fact that the eigenvalues involved in the theorem are known to be $\lambda_2, \lambda_3, ..., \lambda_n$. We define $\omega = e^{2i\pi/n}$ and write each product term in the formula (6.58) as

$$\begin{aligned}\prod_{\substack{k=2\\k\neq j}}^{m+1} \frac{|\omega^{k-1} - 1|}{|\omega^{k-1} - \omega^{j-1}|} &= \prod_{\substack{k=1\\k\neq j}}^{m} \frac{|\omega^{k} - 1|}{|\omega^{k} - \omega^{j}|} \\ &= \left[\prod_{k=1}^{m} |\omega^k - 1|\right] \left[|1 - \omega^j| \prod_{\substack{k=1\\k\neq j}}^{m} |\omega^k - \omega^j|\right]^{-1}\ .\end{aligned}$$

Recalling that the ω^k's are the powers of the n-th root of 1, a simple renumbering of the products in the denominator will show that the numerator and denominator have the same modulus. Hence the above product term is equal to one and by summing these products and inverting, we will get the desired result. □

The above two examples show a striking difference in behavior between two seemingly similar situations. The complex uniform distribution of eigenvalues over a circle is a much worse situation than that of the uniform distribution over a line segment. It indicates that there are cases where the eigenvalues will converge extremely slowly. Note that this poor convergence scenario may even occur if the matrix A is normal, since it is only the distribution of the eigenvalues that cause the difficulty.

Apart from the qualitative interpretation given above, it is also possible to give a simple explicit upper bounds for $\epsilon_i^{(m)}$.

Proposition 6.10 *Let $C(c, \rho)$ be a circle of center c and radius ρ that encloses all the eigenvalues of A except λ_1. Then,*

$$\epsilon_1^{(m)} \leq \left(\frac{\rho}{|\lambda_1 - c|}\right)^{m-1}\ .$$

Proof. An upper bound is obtained by using the particular polynomial $q(z) = (z - c)^{m-1}/(\lambda_1 - c)^{m-1}$ from which we get

$$\epsilon_1^{(m)} \leq \max_{j=2,3,\ldots,n} \left(\frac{|\lambda_j - c|}{|\lambda_1 - c|}\right)^{m-1} \leq \rho^{m-1}/|\lambda_1 - c|^{m-1}\ .$$ □

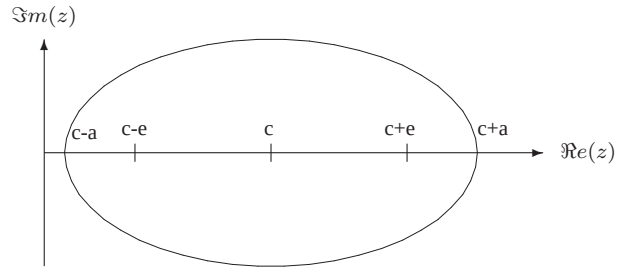


Figure 6.3: Ellipse containing the spectrum of A.

It was seen in Chapter 4 (Lemma 4.3) that the polynomial used in the proof is actually optimal.

Still from what was seen on Chebyshev polynomials in Chapter 4. we may be able to get a better estimate of $\epsilon_1^{(m)}$ if we can enclose the eigenvalues of the matrix A in an ellipse centered at c with focal distance e and major semi-axis a, as is illustrated in Figure 6.3. In this situation the results on the Chebyshev polynomials of the first kind allow us to state the following theorem.

Theorem 6.8 *Assume that all the eigenvalues of A expect λ_1 lie inside the ellipse centered at c, with foci $c+e, c-e$ and with major semi axis a. Then,*

$$\epsilon_1^{(m)} \leq \frac{C_{m-1}\left(\frac{a}{e}\right)}{|C_{m-1}\left(\frac{\lambda_1 - c}{e}\right)|} \tag{6.61}$$

where C_{m-1} is the Chebyshev polynomial of degree $m-1$ of the first kind. In addition, the relative difference between the left and the right hand sides tends to zero as m tends to infinity.

PROBLEMS

P-6.1 To measure the degree of invariance of a subspace X with respect to a matrix A, we define the measure $v(X,A) = \|(I-P)AP\|_2$ where P is the orthogonal projector onto the subspace. (1) Show that if X is invariant then $v(X,A) = 0$. (2) Show that when X is the m-th Krylov subspace generated from some initial vector v, then $v(X,A) = \beta_{m+1}$. (3) Let $r_i, i = 1, \ldots, m$ be the residual vectors associated with the approximate eigenvalues obtained from an orthogonal projection process onto X, and let $R = [r_1, ..., r_m]$. Show that $v(X,A) = \|R\|_2$.

P-6.2 Consider the matrix

$$A = \begin{pmatrix} 0 & & & & & 1 \\ 1 & & & & & 0 \\ & 1 & & & & 0 \\ & & 1 & & & \vdots \\ & & & \ddots & & \vdots \\ & & & & 1 & 0 \end{pmatrix}$$

(1) What are eigenvalues of A? (2) What is the m-th Krylov subspace associated with A when $v_1 = e_1$, the first column of the identity matrix? (3) What are the approximate eigenvalues obtained from Arnoldi's method in this case? How does this relate to Example 6.5?

P-6.3 Assume that k Schur vectors have already been computed and let P be an orthogonal projector associated with the corresponding invariant subspace. Assume that Arnoldi's method is applied to the matrix $(I - P)A$ starting with a vector that is orthogonal to the invariant subspace. Show that the Hessenberg matrix thus obtained is the same as the lower $(m - k) \times (m - k)$ principal submatrix obtained from an implicit deflation procedure. Show that an approximate Schur vector associated with the corresponding projection procedure is an approximate Schur vector for A. This suggests another implementation of the implicit deflation procedure seen in Section 6.2.3 in which only the $(m - k) \times (m - k)$ Hessenberg matrix is used. Give a corresponding new version of Algorithm 6.4. What are the advantages and disadvantages of this approach?

P-6.4 Show that for the Lanczos algorithm one has the inequality

$$\max_{i=1,2,\ldots,m} [\beta_{i+1}^2 + \alpha_i^2 + \beta_{i-1}^2]^{1/2} \leq \max_{j=1,\ldots,n} |\lambda_j|$$

Show a similar result in which $\max$ is replaced by $\min$.

P-6.5 Consider a matrix A that is *skew-Hermitian.* (1) Show that the eigenvalues of A are purely imaginary. What additional property do they satisfy in the particular case when A is $real$ skew-symmetric? [Hint: eigenvalues of real matrices come in complex conjugate pairs...] What can you say of a real skew-symmetric matrix of *odd* dimension n? (2) Assume that Arnoldi's procedure is applied to A starting with some arbitrary vector v_1. Show that the algorithm will produce scalars h_{ij} such that

$$\begin{aligned} &h_{ij} = 0, \text{ for } i < j - 1 \\ &\Re e[h_{jj}] = 0, j = 1, 2, ..., m \\ &h_{j,j+1} = -h_{j+1,j} j = 1, 2, ..., m \end{aligned}$$

(3) From the previous result show that in the particular where A is real skew-symmetric and v_1 is real, then the Arnoldi vectors v_j satisfy a two term recurrence of the form

$$\beta_{j+1} v_{j+1} = A v_j + \beta_j v_{j-1}$$

(4) Show that the approximate eigenvalues of A obtained from the Arnoldi process are also purely imaginary. How do the error bounds of the Lanczos algorithm (Hermitian case) extend to this case?

P-6.6 How do the results of the previous problem extend to the case where $A = \alpha I + S$ where α is a real scalar and S is skew-Hermitian or skew symmetric real?

P-6.7 We consider the following tridiagonal matrix A_n of size $n \times n$

$$A_n = \begin{pmatrix} 2 & 1 & & & & \\ 1 & 2 & . & & & \\ & 1 & . & 1 & & \\ & & . & 2 & 1 & \\ & & & 1 & 2 & \end{pmatrix}.$$

(1) Consider the vector z of length n whose $j-th$ component is $\sin j\theta$ where θ is a real parameter such that $0 < \theta \leq \pi/2$. Show that

$$(2(1+\cos\theta)I - A_n)z = \sin((n+1)\theta)e_n$$

where $e_n = (0, 0, ...0, 1)^H$. (2) Using the previous question find all the eigenvalues and corresponding eigenvectors of A_n. (3) Assume now that m steps of the Lanczos algorithm are performed on A_n with the starting vector $v_1 = e_1 = (1, 0, ..., 0)^H$. (3.a) Show that the Lanczos vectors v_j are such that $v_j = e_j$, $j = 1, 2, ..., m$. (3.b) What is the matrix T_m obtained from the Lanczos procedure? What are the approximate eigenvalues and eigenvectors? (Label all the eigenvalues in decreasing order). (3.c) What is the residual vector and the residual norm associated with the first approximate eigenvalue $\lambda_1^{(m)}$? [Hint: It will be admitted that

$$\sin^2\frac{\pi}{(m+1)} + \sin^2\frac{2\pi}{(m+1)} + ... + \sin^2\frac{m\pi}{(m+1)} = \frac{m+1}{2}]$$

How would you explain the fact that convergence is much slower than expected?

P-6.8 Show that the vector v_{m+1} obtained at the last step of Arnoldi's method is of the form $v_{m+1} = \gamma p_m(A)v_1$, in which γ is a certain normalizing scalar and p_m is the characteristic polynomial of the Hessenberg matrix H_m.

P-6.9 Develop a modified version of the non-Hermitian Lanczos algorithm that produces a sequence of vectors v_i, w_i that are such that each v_i is orthogonal to every w_j with $j \neq i$ and $\|v_i\|_2 = \|w_i\|_2 = 1$ for all i. What does the projected problem become?

P-6.10 Develop a version of the non-Hermitian Lanczos algorithm that produces a sequence of vectors v_i, w_i which satisfy $(v_i, v_j) = \pm\delta_{ij}$, but such that the matrix T_m is Hermitian tridiagonal. What does the projected problem become in this situation? How can this version be combined with the version defined in the previous exercise?

P-6.11 Using the notation of Section 6.3.2 prove that $q_{j+k}(x) = x^k p_j(x)$ is orthogonal to the polynomials $p_1, p_2, \ldots, p_{j-k}$, assuming that $k \leq j$. Show that if we orthogonalized q_{j+k} against $p_1, p_2, \ldots, p_{j-k}$, we would obtain a polynomial that is orthogonal to all polynomials of degree $<j+k$. Derive a general look-ahead non-Hermitian Lanczos procedure based on this observation.

P-6.12 It was stated after the proof of Lemma (6.3) that the solution of the linear system (6.55) is independent of the basis chosen to establish the result in the proof of the lemma. 1) Prove that this is the case. 2) Compute the solution directly using the power basis, and exploiting Vandermonde determinants.

NOTES AND REFERENCES. Several papers have been published on Arnoldi's method and its variants for solving eigenproblems. The original paper by Arnoldi [1] came out about one year after Lanczos' breakthrough paper [112] and is quite different in nature. The author hints that his method can be viewed as a projection method and that it might be used to approximate eigenvalues of large matrices. Note that the primary goal of the method is to reduce an arbitrary (dense) matrix to Hessenberg form. At the time, the QR algorithm was not yet invented, so the Hessenberg form was desired only because it leads to a simlpe recurrence for the characteristic polynomial. The 1980 paper by Saad [169] showed that the method could indeed be quite useful as a projection method to solve large eigenproblems, and gave a few variations of it. Later, sophisticated versions have been developed and used in realistic applications, see [27, 133, 134, 144, 152, 183], among others. During roughly the same period, much work was devoted to exploiting the basic non-Hermitian Lanczos algorithm by Parlett and co-workers [154] and by Cullum and Willoughby [34, 35] and Cullum, Kerner and Willoughby [33]. The first successful application of the code in a real life problem seems to be in the work by Carnoy and Geradin [19] who used a version of the algorithm in a finite element model.

The block Lanczos algorithm seems to have been developed first by Golub and Underwood [79]. The equivalent Block Arnoldi algorithm, has not been given much attention, except in control problems where it is closely associated with the notion of controllability for the multiple-input case [11]. In fact Arnoldi's method (single input case) and its block analogue (multiple input case) are useful in many areas in control, see for example [178, 179].

The error bounds on the Hermitian Lanczos algorithm are from [168]. Bounds of a different type have been proposed by Kaniel [103] (however there were a few errors for the case $i>1$ in Kaniel's original paper and some of these errors were later corrected by Paige [141]). We have omitted to discuss similar bounds for the Block Lanczos algorithm but these were also developed in Saad [168]. The convergence theory for the Arnoldi process is adapted from Saad [171].

The various implementations of the Lanczos algorithm in the Hermitian case are covered in detail in Parlett's book [148]. Implementations on massively parallel machines have recently been discussed by Petiton [156] on the CM-2 and by Scott [190] on the iPSC/2.

These notes stated the following in the first edition of this book: "*Concerning software, there is little that is publically available. Cullum and Willoughby offer a FORTRAN code for the Hermitian case in their book [36] based on the Lanczos algorithm without any form of reorthogonalization. A similar (research) code was also developed by Parlett and Reid [151].*" An implementation of the Look-Ahead Lanczos algorithm was also mentioned [64]. The situation has changed subtantially since then. ARPACK, a package developed by Lecoucq and Yang [118] and based on implicitly restarted Arnoldi method has become a de-facto standard now for Krylov subspace methods for eigenvalue problems. ARPACK is used in particular in Matlab to implement the `eigs` function. A number of other packages have appeared, see e.g., [197, 107, 14] to cite just a few, and it is likely that others will follow suite.

As was mentioned earlier, the first edition of this book contained an incorrect statement for theorem 6.7, which was corrected in [7] (see also the expanded version [6].) ■

Chapter 7

FILTERING AND RESTARTING TECHNIQUES

The early algorithms for eigenvalue extraction were often based on exploiting the powers of the matrix A. The prototype of these techniques is the power method, a technique that is attractive because of its simplicity but whose convergence rate may be unacceptably slow.

In this chapter we will present a number of techniques that are commonly termed polynomial 'acceleration' or 'filtering' techniques. These techniques exploit polynomial iterations of the form $z_q = p_q(A)z_0$ where p_q is a polynomial of degree q which is often determined from some knowledge on the distribution of the eigenvalues of A. A fundamental problem, which will utilize ideas from approximation theory, lies in computing a good polynomial p_q. Filtering methods can be valuable tools for speeding up the convergence of standard algorithms for computing eigenvalue and eigenvectors. They have had a great success as a means to accelerate the subspace iteration algorithm in the past. More recently, filtering techniques have been nicely blended with the Arnoldi procedure and gave rise to the important class of so-called 'implicit restarting schemes'.

7.1 Polynomial Filtering

The main goal of polynomial filtering is to enhance the basic projection scheme (whether Arnoldi or Lanczos, or Subspace iteration for example) by processing, e.g., the initial vectors, or initial subspace, so as to reduce their components in the unwanted parts of the spectrum relative to those in the wanted parts.

We begin with the simplest way of using filters starting with a case of Hermitian matrix A with eigenvalues

$$\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n,$$

and associated (orthonormal) eigenvectors $u_1, \cdots, u_n$. If we are interested in the largest eigenvalue, λ_1, we could use the power method (assuming $|\lambda_1| > |\lambda_i|$ for $i > 1$). This amounts to using the polynomial $p_q(t) = t^q$. However, we can ask ourselves if we can do better than the power series, or, more specifically, *what is the best polynomial that can be used if we want convergence to be the fastest?*

Ignoring implementation for now, we are interested in an iteration of the form

$$x_q = p_q(A)x_0.$$

If x_0 is expanded in the eigenbasis as

$$x_0 = \sum_{i=1}^{n} \gamma_i u_i,$$

then

$$\begin{aligned} x_q &= p_q(A)x_0 = \sum_{i=1}^{n} p_q(\lambda_i)\gamma_i u_i \\ &= p_q(\lambda_1)\gamma_1 u_1 + \sum_{i=2}^{n} p_q(\lambda_i)\gamma_i u_i. \end{aligned} \tag{7.1}$$

If the goal is to extract the first eigenpair, then we would like the component $p_q(\lambda_1)\gamma_1$ in (7.1) to be much larger than the other components, $p_q(\lambda_i)\gamma_i,\ i > 1$. Within this setting some scaling should be applied to the polynomial p_q and we can, for example, require that $p_q(\lambda_1) = 1$. In practice λ_1 is not known but the scaling will be based on an approximate eigenvalue $\tilde{\lambda}_1$.

The γ_i's are usually not known, so we might seek a polynomial p_q whose maximum absolute value over the λ_i's for $i > 1$ is the smallest possible. Since these λ_i's are again not known, a more realistic problem to solve is to seek a polynomial p_q whose value at λ_1 is one and whose maximum absolute value in some interval $[\alpha\ \beta]$ containing all other eigenvalues is the smallest possible. A mathematical formulation of this problem is

$$\begin{cases} \min\limits_{p_q \in \mathbb{P}_q} \max\limits_{t \in [\alpha,\ \beta]} |p_q(t)| \,. \\ p_q(\lambda_1) = 1 \end{cases}$$

We have encountered this problem and discussed its solution in Chapter 4. The optimal polynomial, as stated by Theorem 4.8, is the shifted and scaled Chebyshev polynomial of the first kind of degree q:

$$\hat{C}_q(t) \equiv \frac{C_q\left(1 + 2\frac{t-\beta}{\beta-\alpha}\right)}{C_q\left(1 + 2\frac{\lambda_1-\beta}{\beta-\alpha}\right)}\,.$$

Because of the attractive 3-term recurrence of Chebyshev polynomials, it is easy to write a simple vector recurrence for updating $x_q = \hat{C}_q(A)x_0$. This will be discussed in detail for the more general situation of complex Chebyshev polynomials, see Section 7.4.

Chebyshev polynomials can now be combined with projection-type methods in a number of ways. The next two sections discusses two options.

7.2 Explicitly Restarted Arnoldi's Method

An important property of Arnoldi's method seen in Chapter 4, is that if the initial vector v_1 is exactly in an invariant subspace of dimension r and not in any invariant subspace of smaller dimension, i.e., if the grade of v_1 is r, then the algorithm stops at step $m = r$, because we will obtain $\|\hat{v}_{r+1}\| = 0$. However, as Proposition 6.2 shows, in this case K_r will be invariant, which implies by Proposition 4.3 that the r computed eigenvalues are all exact.

This suggests that a good choice for the initial vector v_1 in Arnoldi's method would be to take a vector which is close to being in an invariant subspace of small dimension. Polynomial filtering can help construct such vectors. A filter polynomial can be selected so that after the filtering step is applied to some initial vector v the resulting vector will have small components in any eigenvalue that is inside a 'wanted' region and large components for eigenvalues outside this region. If there is a small number of such eigenvalues in addition to the wanted ones $\lambda_1, \lambda_2, ..., \lambda_r$, then the Arnoldi projection process will compute them with a good accuracy and they will be used to compute the next filter polynomial.

A possible approach is to select a vector z_0 which is a certain linear combination of approximate eigenvectors or Schur vectors obtained from a previous iteration and apply to it a certain polynomial filter. The result is then normalized to yield the initial vector for the next Arnoldi loop. This 'enhanced initial vector approach' does not work too well in practice and the reasons for this were explained by Morgan [131]. The process can be slow of even diverge in some cases when the eigenvalues are poorly separated. An alternative which works better is to target one eigenvalue-eigenvector pair at a time and proceed just as for the restarted Arnoldi method with deflation described in Chapter 4..

The algorithm is in fact very similar in structure to Algorithm 6.4. The only difference is that the initial vector in the outer loop is now preprocessed by a filtering acceleration. The implementation uses a single basis $v_1, v_2, ..., v_m$ whose first vectors are the Schur vectors of A that have already converged. If the $\nu - 1$ vectors $v_1, v_2, ..., v_{\nu-1}$ have converged then we start by choosing a vector v_ν which is orthogonal to $v_1, ..., v_{\nu-1}$ and of norm 1. We then perform $m - \nu$ steps of an Arnoldi process, orthogonalizing the vector v_j against all previous v_i's including $v_1, ..., v_{\nu-1}$. Finally, we restart as in the previous algorithm, taking v_1 to be $p_q(A)z_0$, where z_0 is the approximate Schur vector produced by the Arnoldi process. The algorithm is sketched below.

ALGORITHM **7.1 (Deflated Arnoldi with filtering)**

A. Start: *Choose an initial vector* v_1*, with* $\|v_1\|_2 = 1$.

B. Eigenvalue Loop: *For* $l = 1, 2, ..., p$ *do:*

1. *Arnoldi Iteration. For* $j = l, l+1, ..., m$ *do:*
   - *Compute* $w := Av_j$*;*
   - *Compute a set of* j *coefficients* h_{ij} *such that* $w := w - \sum_{i=1}^{j} h_{ij} v_i$ *is orthogonal to all previous* v_i*'s,* $i = 1, 2, ..., j$*;*

- *Compute* $h_{j+1,j} = \|w\|_2$ *and* $v_{j+1} = w/h_{j+1,j}$.

2. *Compute a desired Ritz pair* $\tilde{u}_l, \tilde{u}_l$, *and corresponding residual norm* ρ_l.
3. *Using the approximate Ritz values obtain a new filter polynomial* p_q.
4. *Compute* $z_q = p_q(A)z_0$, *with* $z_0 = \tilde{u}$.
5. *Orthonormalize* z_q *against all previous* v_j*'s to get the approximate Schur vector* $\tilde{u}_l$ *and define* $v_l := \tilde{u}_l$.
6. *If* ρ_j *is small enough then accept* $\tilde{v}_l$ *as the next Schur vector, compute* $h_{i,l} = (Av_l, v_i)\ i = 1, .., l$. *Else go to (B.1).*

Recall that in the B-loop, the Schur vectors associated with the eigenvalues $\lambda_1, ..., \lambda_{l-1}$ are frozen and so is the corresponding upper triangular matrix corresponding to these vectors.

The new polynomial p_q in Step B-3 can be chosen in a number of ways. One can compute a new Chebyshev polynomial for example, and this is described in detail in Section 7.4. Another option is simply to use a polynomial of the form

$$p_q(t) = (t - \theta_1)(t - \theta_2) \ldots (t - \theta_q). \tag{7.2}$$

where the θ_i's are a set (possibly all) of the unwanted values among the computed Ritz values. This has the effect of making p_q small on the unwanted set and large elsewhere. An elegant way to implement this idea is described in the next section.

7.3 Implicitly Restarted Arnoldi's Method

In the discussion of the Arnoldi process we saw that restarting was necessary in practice because as m increases, cost becomes prohibitive. The goal of implicitly restarted Arnoldi's method is to devise a procedure which is equivalent to applying a polynomial filter to the initial vector of the Arnoldi process. The idea blends three ingredients: polynomial filtering, the Arnoldi (or Lanczos) procedure, and the QR algorithm for computing eigenvalues. We consider a polynomial filter which is factored in the form (7.2). Assume that we have obtained from the Arnoldi procedure the Arnoldi decomposition

$$AV_m = V_m H_m + \beta_m v_{m+1} e_m^T \tag{7.3}$$

and consider applying the first factor: $(t - \theta_1)$ to all the basis vectors v_i:

$$(A - \theta_1 I)V_m = V_m(H_m - \theta_1 I) + \beta_m v_{m+1} e_m^T$$

Let $H_m - \theta_1 I = Q_1 R_1$. Then,

$$\begin{aligned}
(A - \theta_1 I)V_m &= V_m Q_1 R_1 + \beta_m v_{m+1} e_m^T & (7.4)\\
(A - \theta_1 I)(V_m Q_1) &= (V_m Q_1) R_1 Q_1 + \beta_m v_{m+1} e_m^T Q_1 & (7.5)\\
A(V_m Q_1) &= (V_m Q_1)(R_1 Q_1 + \theta_1 I) + \beta_m v_{m+1} e_m^T Q_1 & (7.6)
\end{aligned}$$

We now introduce the notation:

$$\begin{aligned} H_m^{(1)} &\equiv R_1 Q_1 + \theta_1 I; \\ (b_{m+1}^{(1)})^T &\equiv e_m^T Q_1; \\ V_m^{(1)} &\equiv V_m Q_1. \end{aligned}$$

With this notation the relation (7.6) translates to

$$AV_m^{(1)} = V_m^{(1)} H_m^{(1)} + v_{m+1}(b_{m+1}^{(1)})^T. \tag{7.7}$$

We now examine the result of Equation (7.7) in further detail. Our first observation is that $R_1 Q_1 + \theta_1 I$ is the matrix that would result from one step of the standard QR algorithm with shift θ_1 applied to H_m. In particular, from what is known about the QR algorithm, the matrix $H_m^{(1)}$ remains an upper Hessenberg matrix. As a result, (7.6) resembles an ordinary Arnoldi decomposition such as the one of Equation (7.3), except that the vector e_m^T is now replaced by the vector $(b_{m+1}^{(1)})^T$.

A second observation is that the first column of $V_m^{(1)}$ is a multiple of $(A - \theta_1 I)v_1$. This follows from multiplying (7.4) by the column e_1, and recalling that R_1 is upper triangular (with entries denoted by r_{ij})

$$(A - \theta_1 I)V_m e_1 = (V_m Q_1)R_1 e_1 + \beta_m v_{m+1} e_m^T e_1 \rightarrow (A - \theta_1 I)v_1 = r_{11} v_1^{(1)}.$$

Our third and final observation is that the columns of $V_m^{(1)}$ are orthonormal because they result from applying rotations to the columns of V_m, an operation which maintains orthonormality.

We can now apply the second shift in the same way:

$$(A - \theta_2 I)V_m^{(1)} = V_m^{(1)}(H_m^{(1)} - \theta_2 I) + v_{m+1}(b_{m+1}^{(1)})^T$$

By a similar process $(H_m^{(1)} - \theta_2 I) = Q_2 R_2$ and upon multiplying by Q_2 to the right we obtain:

$$(A - \theta_2 I)V_m^{(1)} Q_2 = (V_m^{(1)} Q_2)(R_2 Q_2) + v_{m+1}(b_{m+1}^{(1)})^T Q_2$$

leading to the following analogue of (7.7)

$$AV_m^{(2)} = V_m^{(2)} H_m^{(2)} + v_{m+1}(b_{m+1}^{(2)})^T \ , \tag{7.8}$$

where, $H_m^{(2)} \equiv R_2 Q_2 + \theta_2 I$, and $V_m^{(2)} \equiv V_m^{(1)} Q_2$.

The same argument as above will show that the first column of $V_m^{(2)}$ is a multiple of $(A - \theta_2 I)v_1^{(1)}$. Hence,

$$\begin{aligned} V_m^{(2)} e_1 &= \text{scalar} \times (A - \theta_2 I)v_1^{(1)} \\ &= \text{scalar} \times (A - \theta_2 I)(A - \theta_1 I)v_1. \end{aligned}$$

Note that Q_1 and Q_2 are both Hessenberg matrices and that

$$(b_{m+1}^{(2)})^T = (b_{m+1}^{(1)})^T Q_2 = e_m^T Q_1 Q_2 = [0, 0, \cdots, 0, \eta_1, \eta_2, \eta_3]$$

Consider the matrix $\hat{V}_{m-2} = [\hat{v}_1, \ldots, \hat{v}_{m-2}]$ consisting of the first $m-2$ columns of $V_m^{(2)}$ and the matrix $\hat{H}_{m-2}$ the leading $(m-2) \times (m-2)$ principal submatrix of H_m. Then,

$$\begin{aligned} A\hat{V}_{m-2} &= \hat{V}_{m-2}\hat{H}_{m-2} + \hat{\beta}_{m-1}\hat{v}_{m-1}e_m^T \qquad \text{with} \\ \hat{\beta}_{m-1}\hat{v}_{m-1} &\equiv \eta_1 v_{m+1} + h_{m-1,m-2}^{(2)} v_{m-1}^{(2)} \end{aligned}$$

Note that $\|\hat{v}_{m-1}\|_2 = 1$. The remarkable result is that the above decomposition is identical with one that would have been obtained from performing $(m-2)$ steps of the Arnoldi process starting with the filtered vector $\hat{v}_1 = w/\|w\|_2$, where $w = (A - \theta_2 I)(A - \theta_1 I)v_1$. This means we know how to *implicitly* apply polynomial filtering, in this case of degree 2, to the initial vector of an $(m-2)$-step Arnoldi procedure. The procedure exploits the QR algorithm within the Arnoldi procedure. The process can now be continued from step $m-2$, e.g., by performing two additional steps to perform the full m steps of an m-step Arnoldi procedure if desired. In terms of cost we have not gained or lost anything when matrix-vector products are counted: We started with m Arnoldi steps, performed two basis rotations and added two more Arnoldi steps to complete m full steps of Arnoldi. The total is $(m+2)$ matvecs the same as if we just started by computing $\hat{v}_1$ (2 matvecs) and the m additional matvecs for the m-step Arnoldi procedure. Of course implicit restarts presents a number of advantages over explicit restarting. First, it is a very stable procedure, consisting mostly of plane rotations. Second, it blends nicely with the Arnoldi process and allows one to get the desired shifts θ_i as the procedure progresses.

We have described the process for a degree 2 filter but clearly this can be extended to higher degrees. Notation is somewhat simplified by setting $k \equiv m - q$, where q is the degree. To describe the algorithm in a succinct way, we need to recall the implicit-shift QR procedure which is a major ingredient of the procedure.

ALGORITHM **7.2** *q-step Shifted QR*

For $j = 1, \cdots, q$ *Do*
  $(H - \theta_j I) = QR$
  $H := RQ + \theta_j I$
EndDo

Each instance of the above loop performs one step of the QR algorithm and the resulting matrix $\tilde{H}$ is similar to the original matrix since,

$$Q\tilde{H}Q^H = Q(RQ + \theta_j I)Q^H = QR + \theta_j I = H.$$

The implicit-Q theorem allows to perform these q steps in an effective way with plane rotations, a procedure known as bulge-chasing. Details are beyond the scope

of this book and can be found in standard text such as [77] or [204]. We will denote by

$$[\hat{H}, Q] = \text{QR}(H, \theta_1, \cdots, \theta_q)$$

the results of this operation, where Q is the unitary matrix which transforms H to $\hat{H}$. Using the usual notation we then end up with the following algorithm.

ALGORITHM **7.3** *Implicitly Restarted Arnoldi Algorithm*

Perform an m-step Arnoldi procedure to get the factorization:
$AV_m = V_m H_m + \hat{v}_{m+1} e_m^T$
Select the q shifts $\theta_1, \cdots, \theta_q$ from the eigenvalues of H_m
Perform a q-step QR with these shifts:
$[H_m, Q] := QR(H_m, \theta_1, \cdots, \theta_k)$
Set $k = m - q$, $H_k = H_m(1:k, 1:k)$, $V_k := V_k Q$
Set $\hat{v}_{k+1} := \hat{v}_{k+1} + \eta_k \hat{v}_{m+1}$ with $\eta_k = Q_{m,k}$;
Continue the resulting Arnoldi factorization
$AV_k = V_k H_k + \hat{v}_{k+1} e_k^T$
with q additional Arnoldi steps.

Recall our notation : $\hat{v}_{k+1} \equiv h_{k+1,k} v_{k+1}$ is the unscaled Arnoldi vector of the k-th step. As discussed above, a common implementation consists of keeping m, the dimension of the Krylov subspace, constant, and q, the degree of the filter, fixed.

7.3.1 Which Filter Polynomials?

Any polynomial can be used in the procedure just described as long as it is provided in the factored form (7.2). However, common implementations of the implicitly restarted Arnoldi algorithm, use for the 'shifts' θ_i approximate eigenvalues obtained from the Hessenberg matrix, i.e., Ritz values. The Ritz values are divided in two groups: 'wanted' and 'unwanted'. For example if our goal is to compute the ten eigenvalues of A with the smallest (algebraic) real parts we can set the ten leftmost Ritz values as wanted and the rest as unwanted. The θ_i's are selected among the unwanted Ritz values. The resulting filter polynomial will be zero on these roots, and so it likely to be small on the unwanted part of the spectrum, and it will have larger values on the wanted part of the spectrum.

The reader familiar with the QR algorithm may have noticed that exact eigenvalues of H_k are used as shifts in the QR algorithm and in this situation the output matrix $\hat{H}_k$ from the q-step QR procedure will have a partial upper triangular form.

7.4 Chebyshev Iteration

Chebyshev filters where among the first to be used for solving large eigenvalue problems, as well as linear systems of equations. Let A be a real nonsymmetric (or non Hermitian complex) matrix of dimension n and consider the eigenvalue problem, $Au = \lambda u$. Let $\lambda_1, \cdots, \lambda_n$ be the eigenvalues of A labeled in decreasing

order of their real parts, and suppose that we are interested in λ_1 which is initially assumed to be real.

We consider a polynomial iteration of the form: $z_k = p_k(A)z_0$, where z_0 is some initial vector and where p_k is a polynomial of degree k. We would like to choose p_k in such a way that the vector z_k converges rapidly towards an eigenvector of A associated with λ_1 as k tends to infinity. Assuming for simplicity that A is diagonalizable, we expand z_0 in the eigenbasis $\{u_i\}$ as,

$$z_0 = \sum_{i=1}^{n} \theta_i u_i,$$

which leads to the following expression for $z_k = p_k(A)z_0$:

$$z_k = \sum_{i=1}^{n} \theta_i p_k(\lambda_i) u_i = \theta_1 p_k(\lambda_1) u_1 + \sum_{i=2}^{n} \theta_i p_k(\lambda_i) u_i. \tag{7.9}$$

The above expansion shows that if z_k is to be a good approximation of the eigenvector u_1, then the second term must be much smaller that the first and this can be achieved by making every $p_k(\lambda_j)$, with $j \neq 1$, small in comparison with $p_k(\lambda_1)$. This leads us to seek a polynomial which takes 'small' values on the discrete set

$$R = \{\lambda_2, \lambda_3, \cdots, \lambda_n\},$$

and which satisfies the normalization condition

$$p_k(\lambda_1) = 1. \tag{7.10}$$

An ideal such polynomial would be one which minimizes the (discrete) uniform norm on the discrete set R over all polynomials of degree k satisfying (7.10). However, this polynomial is impossible to compute without the knowledge of all eigenvalues of A and as a result this approach has little practical value. A simple and common alternative, is to replace the discrete min-max polynomial by the continuous one on a domain containing R but excluding λ_1. Let E be such a domain in the complex plane, and let $\mathbb{P}_k$ denote the space of all polynomials of degree not exceeding k. We are thus seeking a polynomial p_k which achieves the minimum

$$\min_{p \in \mathbb{P}_k,\ p(\lambda_1)=1} \quad \max_{\lambda \in E} |p(\lambda)|. \tag{7.11}$$

For an arbitrary domain E, it is difficult to solve explicitly the above min-max problem. Iterative methods can be used, however, and the exploitation of the resulting min-max polynomials for solving eigenvalue problems constitutes a promising research area. A preferred alternative is to restrict E to be an ellipse having its center on the real line, and containing the unwanted eigenvalues $\lambda_i, i = 2, \cdots, n$.

Let $E(c, e, a)$ be an ellipse containing the set

$$R = \{\lambda_2, \lambda_3, \cdots, \lambda_n\},$$

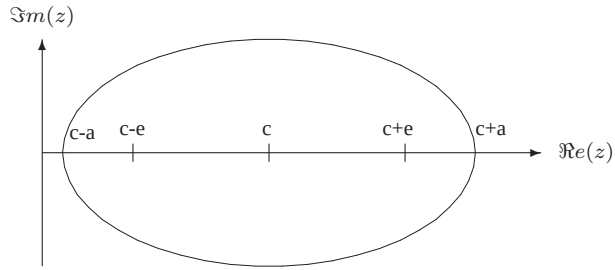


Figure 7.1: Ellipse containing the spectrum of A with e real.

and having (real) center c, foci $c+e$, $c-e$, and major semi-axis a. When A is real the spectrum of A is symmetric with respect to the real axis, so we can restrict $E(c,e,a)$ to being symmetric as well. In other words, the main axis of the ellipse is either the real axis or a line parallel to the imaginary axis. Therefore, a and e are either both real or both purely imaginary. These two cases are illustrated in Figure 7.1 and Figure 7.2 respectively.

A result that is known in approximation theory and shown in Section 4.4 of Chapter 4, is that when E is the ellipse $E(c,e,a)$ in (7.11), an asymptotically best min-max polynomial is the polynomial

$$p_k(\lambda) = \frac{C_k[(\lambda - c)/e]}{C_k[(\lambda_1 - c)/e]}\ , \tag{7.12}$$

where C_k is the Chebyshev polynomial of degree k of the first kind.

The computation of $z_k = p_k(A)z_0, k = 1, 2, \cdots$, is simplified by the three–term recurrence for the Chebyshev polynomials,

$$\begin{aligned} &C_1(\lambda) = \lambda, \qquad C_0(\lambda) = 1, \\ &C_{k+1}(\lambda) = 2\lambda C_k(\lambda) - C_{k-1}(\lambda), \qquad k = 1, 2, \cdots\ . \end{aligned}$$

Letting $\rho_k = C_k[(\lambda_1 - c)/e], k = 0, 1, \cdots$, we obtain

$$\rho_{k+1}p_{k+1}(\lambda) = C_{k+1}[\frac{\lambda - c}{e}] = 2\ \frac{\lambda - c}{e}\rho_k p_k(\lambda) - \rho_{k-1}p_{k-1}(\lambda).$$

We can simplify this further by defining $\sigma_{k+1} \equiv \rho_k/\rho_{k+1}$,

$$p_{k+1}(\lambda) = 2\sigma_{k+1}\frac{\lambda - c}{e}p_k(\lambda) - \sigma_k\sigma_{k+1}p_{k+1}(\lambda).$$

A straightforward manipulation using the definitions of σ_i, ρ_i and the three–term recurrence relation of the Chebyshev polynomials shows that $\sigma_i, i = 1, 2, \cdots$, can be obtained from the recurrence,

$$\begin{aligned} \sigma_1 &= \frac{e}{\lambda_1 - c}; \\ \sigma_{k+1} &= \frac{1}{\frac{2}{\sigma_1} - \sigma_k}\ , \quad k = 1, 2, \cdots\ . \end{aligned}$$

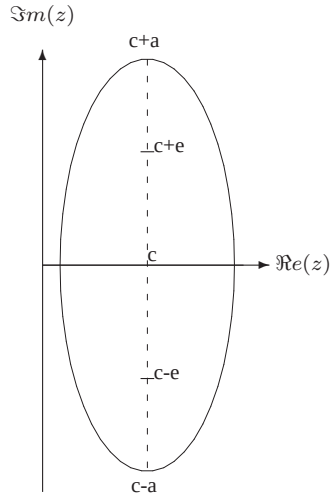


Figure 7.2: Ellipse containing the spectrum of A, with e purely imaginary.

The above two recursions defining z_k and σ_k can now be assembled together to yield a basic algorithm for computing $z_k = p_k(A)z_0,\ k \geq 1$. Although λ_1 is not known, recall that it is used in the denominator of (7.12) for scaling purposes only, so we may replace it by some approximation ν in practice.

ALGORITHM **7.4 Chebyshev Iteration**

1. *Start: Choose an arbitrary initial vector z_0 and compute*

$$\sigma_1 = \frac{e}{\lambda_1 - c}, \tag{7.13}$$

$$z_1 = \frac{\sigma_1}{e}(A - cI)z_0. \tag{7.14}$$

2. *Iterate: For $k = 1, 2, \cdots$, until convergence do:*

$$\sigma_{k+1} = \frac{1}{2/\sigma_1 - \sigma_k}, \tag{7.15}$$

$$z_{k+1} = 2\frac{\sigma_{k+1}}{e}(A - cI)z_k - \sigma_k\sigma_{k+1}z_{k-1}. \tag{7.16}$$

An important detail, which we have not considered for the sake of clarity, concerns the case when e is purely imaginary. It can be shown quite easily that even in this situation the above recursion can still be carried out in real arithmetic. The reason for this is that the scalars σ_{k+1}/e and $\sigma_{k+1}\sigma_k$ in the above algorithm

are real numbers. The primary reason for scaling by $C_k[(\lambda_1 - c)/e]$ in (7.12) is to avoid overflow but we have just given another reason, namely avoid complex arithmetic when e is purely imaginary.

7.4.1 Convergence Properties.

In order to understand the convergence properties of the sequence of approximations z_k we consider its expansion (7.9) and examine the behavior of each coefficient of u_i, for $i \neq 1$. By the definition of p_k we have:

$$p_k(\lambda_i) = \frac{C_k[(\lambda_i - c)/e]}{C_k[(\lambda_1 - c)/e]}.$$

From the standard definition of the Chebyshev polynomials in the complex plane seen in Chapter 4, the above expression can be rewritten as

$$p_k(\lambda_i) = \frac{w_i^k + w_i^{-k}}{w_1^k + w_1^{-k}}, \tag{7.17}$$

where w_i represents the root of largest modulus of the equation in w:

$$\frac{1}{2}(w + w^{-1}) = \frac{\lambda_i - c}{e}. \tag{7.18}$$

From (7.17), $p_k(\lambda_i)$ is asymptotic to $[w_i/w_1]^k$, hence the following definition.

Definition 7.1 *We will refer to $\kappa_i = |w_i/w_1|$ as the damping coefficient of λ_i relative to the parameters c, e. The convergence ratio $\tau(\lambda_1)$ of λ_1 is the largest damping coefficient κ_i for $i \neq 1$.*

The meaning of the definition is that each coefficient in the eigenvector u_i of the expansion (7.9) behaves like κ_i^k, as k tends to infinity. The damping coefficient $\kappa(\lambda)$ can obviously be also defined for any value λ in the complex plane, not necessarily an eigenvalue. Given a set of r wanted eigenvalues, $\lambda_1, \lambda_2, \ldots, \lambda_r$, the definition 7.1 can be extended for an eigenvalue λ_j $j \leq r$ as follows. The damping coefficient for any 'unwanted' eigenvalue $\lambda_i, i > r$ must simply be redefined as $|w_i/w_j|$ and the convergence ratio $\tau(\lambda_j)$ with respect to the given ellipse is the largest damping coefficient κ_l, for $l = r + 1, \ldots, n$.

One of the most important features in Chebyshev iteration lies in the expression (7.18). There are infinitely many points λ in the complex plane whose damping coefficient $\kappa(\lambda)$ has the same value κ. These points λ are defined by $(\lambda - c)/e = (w + w^{-1})/2$ and $|w/w_1| = \kappa$ where κ is some constant, and belong to the same confocal ellipse $E(c, e, a(\kappa))$. Thus a great deal of simplification can be achieved by locating those points *that are real* as it is preferable to deal with real quantities than imaginary ones in the above expression defining κ_i. As was seen in Section 4-4.4 the mapping $J(w) = \frac{1}{2}(w + w^{-1})$, transforms a circle into an ellipse in the complex plane. More precisely, for $w = \rho e^{i\theta}$, $J(w)$ belongs to an ellipse of center the origin, focal distance 1, and major semi-axis $\rho = \frac{1}{2}(\rho + \rho^{-1})$.

Moreover, given the major semi-axis α of the ellipse, the radius ρ is determined by $\rho = \frac{1}{2}[\alpha+(\alpha^2-1)^{1/2}]$. As a consequence the damping coefficient κ_i is simply ρ_i/ρ_1 where $\rho_i \equiv \frac{1}{2}[\alpha_i+(\alpha_i^2-1)^{1/2}]$ and α_i is the major semi-axis of the ellipse centered at the origin, with focal distance one and passing through $(\lambda_j - c)/e$. Since $\alpha_1 > \alpha_i, i = 2, 3, \cdots, n$, it is easy to see that $\rho_1 > \rho_i$, $i > 1$, and hence that the process will converge. Note that there is a further mapping between λ_j and $(\lambda_j - c)/e$ which transforms the ellipse $E(c, e, a_j)$ into the ellipse $E(0, 1, \alpha_j)$ where a_j and α_j are related by $\alpha_j = a_j/e$. Therefore, the above expression for the damping coefficient can be rewritten as:

$$\kappa_i = \frac{\rho_i}{\rho_1} = \frac{a_i + (a_i^2 - 1)^{1/2}}{a_1 + (a_1^2 - 1)^{1/2}}, \tag{7.19}$$

where a_i is the major semi-axis of the ellipse of center c, focal distance e, passing through λ_i. From the expansion (7.9), the vector z_k converges to $\theta_1 u_1$, and the error behaves like $\tau(\lambda_1)^k$.

The algorithm described above does not address a certain number of practical issues. For example, the parameters c and e will not generally be known beforehand, and their estimation is required. The estimation is typically done in a dynamic manner. In addition, the algorithm does not handle the computation of more than one eigenvalue. In particular what can we do in case λ_1 is complex, i.e., when λ_1 and $\lambda_2 = \bar{\lambda}_1$ form a complex pair?

7.4.2 Computing an Optimal Ellipse

We would like to find the 'best' ellipse enclosing the set R of unwanted eigenvalues, i.e., the eigenvalues other than the ones with the r algebraically largest real parts. We must begin by clarifying what is meant by 'best' in the present context. Consider Figure 7.3 representing a spectrum of some matrix A and suppose that we are interested in the r rightmost eigenvalues, i.e., $r = 4$ in the figure.

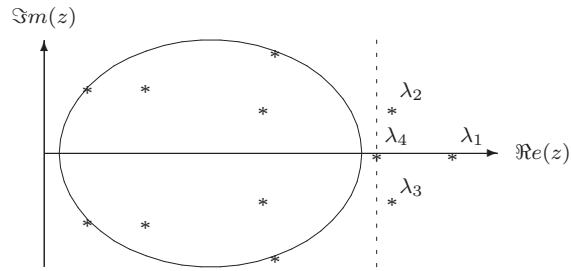


Figure 7.3: Example of a spectrum and the enclosing best ellipse for $r = 4$.

If $r = 1$ then one may simply seek the best ellipse in the sense of minimizing the convergence ratio $\tau(\lambda_1)$. This situation is identical to that of Chebyshev Iteration for linear systems for which much work has been done.

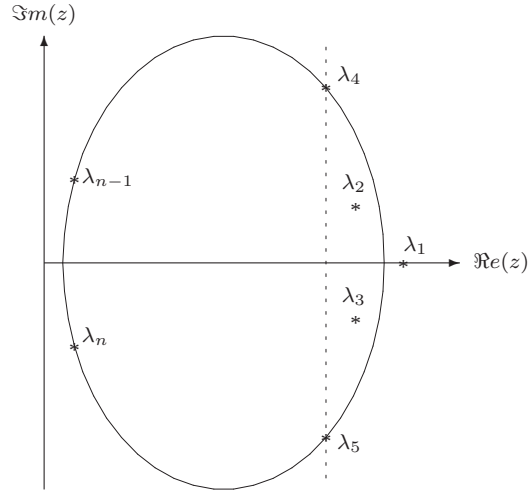


Figure 7.4: Case where $\mu = \lambda_r$ (complex): the eigenvalues λ_2 and λ_3 are inside the 'best' ellipse.

When $r > 1$, then we have several convergence ratios, each corresponding to one of the desired eigenvalues $\lambda_i, i = 1, \cdots, r$, and several possible strategies may be defined to try to optimize the process.

Initially, assume that λ_r is real (Figure 7.3) and consider any ellipse $E(c, e, a)$ including the set R of unwanted eigenvalues and not the eigenvalues

$$\{\lambda_1, \lambda_2, \cdots, \lambda_r\}.$$

It is easily seen from our comments of subsection 7.4.1 that if we draw a vertical line passing through the eigenvalue λ_r, all eigenvalues to the right of the line will converge faster than those to the left. Therefore, when λ_r is real, we may simply define the ellipse as the one that minimizes the convergence ratio of λ_r with respect to the two parameters c and e.

When λ_r is not real, the situation is more complicated. We could still attempt to maximize the convergence ratio for the eigenvalue λ_r, but the formulas giving the optimal ellipse do not readily extend to the case where λ_r is complex and the best ellipse becomes difficult to determine. But this is not the main reason why this choice is not suitable. A close look at Figure 7.3, in which we assume $r = 5$, reveals that the best ellipse for λ_r may not be a good ellipse for some of the desired eigenvalues. For example, in the figure the eigenvalues λ_2, λ_3 should be computed before the pair λ_4, λ_5 since their real parts are larger. However, because they are enclosed by the best ellipse for λ_5 they may not converge until many other eigenvalues will have converged including $\lambda_4, \lambda_5, \lambda_n, \lambda_{n-1}$ and possibly other unwanted eigenvalues not shown in the figure.

The difficulty comes from the fact that this strategy will not favor the eigenvalues with largest real parts but those belonging to the outermost confocal ellipse. It can be resolved by just maximizing the convergence ratio of λ_2 instead of λ_5 in this case. In a more complex situation it is unfortunately more difficult to determine at which particular eigenvalue λ_k or more generally at which value μ it is best to maximize $\tau(\mu)$. Clearly, one could solve the problem by taking $\mu = \Re e(\lambda_r)$, but this is likely to result in a suboptimal choice.

As an alternative, we can take advantage of the previous ellipse, i.e., an ellipse determined from previous purification steps. We determine a point μ on the real line *having the same convergence ratio as λ_r, with respect to the previous ellipse.* The next 'best' ellipse is then determined so as to maximize the convergence ratio for this point μ. This reduces to the previous choice $\mu = \Re e(\lambda_r)$ when λ_r is real. At the very first iteration one can set μ to be $\Re e(\lambda_r)$. This is illustrated in Figure7.5. In Figure 7.5 the ellipse in solid is the optimal ellipse obtained from some previous calculation from the dynamic process. In dashed line is an ellipse that is confocal to the previous ellipse which passes through λ_r. The point μ is defined as one of the two points where this ellipse crosses the real axis.

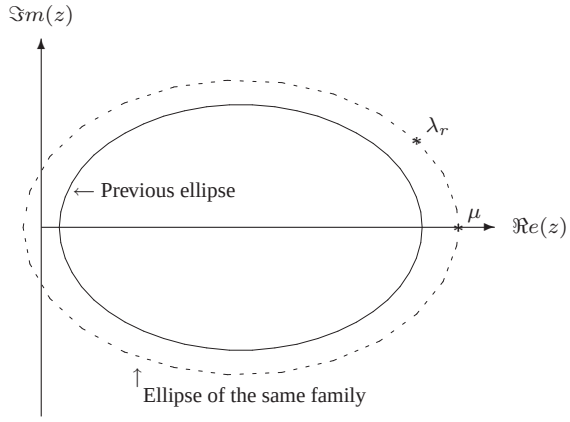


Figure 7.5: Point on the real axis whose convergence is equivalent with that of λ_r with respect to the previous ellipse.

The question which we have not yet fully answered concerns the practical determination of the best ellipse. At a typical step of the Arnoldi process we are given m approximations $\tilde{\lambda}_i, i = 1, \cdots, m$, of the eigenvalues of A. This approximate spectrum is divided in two parts: the r wanted eigenvalues $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_r$ and the set $\tilde{R}$ of the remaining eigenvalues $\tilde{R} = \{\tilde{\lambda}_{r+1}, \cdots, \tilde{\lambda}_m\}$. From the previous ellipse and the previous sets $\tilde{R}$, we would like to determine the next estimates for the optimal parameters c and e.

A similar problem was solved in the context of linear systems of equations and the technique can easily be adapted to our situation. We refer the reader to the two articles by Manteuffel [126, 127]. The change of variables $\xi = (\mu - \lambda)$

easily transforms μ into the origin in the ξ–plane and the problem of maximizing the ratio $\tau(\mu)$ is transformed into one of maximizing a similar ratio in the ξ–plane for the origin, with respect to the parameters c and e. An effective technique for solving this problem has been developed in [125], [127] but its description is rather tedious and will be omitted. We only indicate that there exist reliable software that will deliver the optimal values of $\mu - c$ and e at output if given the shifted eigenvalues $\mu - \tilde{\lambda}_j, \quad j = r+1, \cdots, m$ on input.

We now wish to deal with a minor difficulty encountered when λ_1 is complex. Indeed, it was mentioned in Section 7.4 that the eigenvalue λ_1 in (7.13) should, in practice, be replaced by some approximation ν of λ_1. Initially, ν can be set to some initial guess. Then, when the approximation $\tilde{\lambda}_1$ as computed from the outer loop of the procedure, becomes available it can be used. If it is real then we can take $\nu = \tilde{\lambda}_1$ and the iteration can be carried out in real arithmetic as was already shown, even when e is purely imaginary. However, the iteration will become complex if $\tilde{\lambda}_1$ is complex. To avoid this it suffices to take ν to be one of the two points where the ellipse $E(c, e, a_1)$ passing through $\tilde{\lambda}_1$, crosses the real axis. The effect of the corresponding scaling of the Chebyshev polynomial will be identical with that using $\tilde{\lambda}_1$ but will present the advantage of avoiding complex arithmetic.

7.5 Chebyshev Subspace Iteration

We will use the same notation as in the previous sections. Suppose that we are interested in the rightmost r eigenvalues and that the ellipse $E(c, e, a)$ contains the set R of all the remaining eigenvalues. Then the principle of the Chebyshev acceleration of subspace iteration is simply to replace the powers A^k in the first part of the basic algorithm 5.1 described in Chapter 5, by $p_k(A)$ where p_k is the polynomial defined by (7.12). It can be shown that the approximate eigenvector $\tilde{u}_i, i = 1, \cdots, r$ converges towards u_i, as $C_k(a/e)/C_k[(\lambda_i - c)/e]$, which, using arguments similar to those of subsection (7.4.1) is equivalent to η_i^k where

$$\eta_i = \frac{a + [a^2 - 1]^{1/2}}{a_i + [a_i^2 - 1]^{1/2}}. \tag{7.20}$$

The above convergence ratio can be far superior to the standard ratio $|\lambda_{r+1}/\lambda_i|$ which is achieved by the non-accelerated algorithm. However, we recall that subspace iteration computes the eigenvalues of largest moduli. Therefore, the unaccelerated and the accelerated subspace iteration methods are not always comparable since they achieve different objectives.

On the practical side, the best ellipse is obtained dynamically in the same way as was proposed for the Chebyshev–Arnoldi process. The accelerated algorithm will then have the following form.

ALGORITHM **7.5** *Chebyshev Subspace Iteration*

1. **Start:** $Q \leftarrow X$.
2. **Iterate:** *Compute* $Q \leftarrow p_k(A)Q$.

3. **Project:** *Orthonormalize Q and get eigenvalues and Schur vectors of $C = Q^T AQ$. Compute $Q \leftarrow QF$, where F is the matrix of Schur vectors of C.*

4. **Test for convergence:** *If Q is a satisfactory set of Schur vectors then stop, else get new best ellipse and go to 2.*

Most of the ideas described for the Arnoldi process extend naturally to this algorithm, and we now discuss briefly a few of them.

7.5.1 Getting the Best Ellipse.

The construction of the best ellipse is identical with that seen in subsection 7.4.2. The only potential difficulty is that the additional eigenvalues that are used to build the best ellipse may now be far less accurate than those provided by the more powerful Arnoldi technique.

7.5.2 Parameters k and m.

Here, one can take advantage of the abundant work on subspace iteration available in the literature. All we have to do is replace the convergences $|\lambda_{r+1}/\lambda_i|$ of the basic subspace iteration by the new ratios η_i of (7.20). For example, one way to determine the number of Chebyshev steps k, proposed in Rutishauser [167] and in Jennings and Stewart [96] is

$$n \approx \frac{1}{2}[1 + \ln(\epsilon^{-1})/\ln(\eta_1)],$$

where ϵ is some parameter depending on the unit round–off. The goal of this choice is to prevent the rounding errors from growing beyond the level of the error in the most slowly converging eigenvector. The parameter k is also limited from above by a user supplied bound $n_{\max}$, and by the fact that if we are close to convergence a smaller k can be determined to ensure convergence at the next projection step.

The same comments as in the Arnoldi–Chebyshev method can be made concerning the choice of m, namely that m should be at least $r + 2$, but preferably even larger although in a lesser extent than for Arnoldi. For the symmetric case it is often suggested to take to be a small multiple of r, e.g., $m = 2r$ or $m = 3r$.

7.5.3 Deflation

Another special feature of the subspace iteration is the deflation technique which consists of working only with the non-converged eigenvectors, thus 'locking' those that have already converged. Clearly, this can be used in the accelerated subspace iteration as well and will enhance its efficiency. For the more stable versions such as those based on Schur vectors, a similar device can be applied to the Schur vectors instead of the eigenvectors.

7.6 Least Squares - Arnoldi

The choice of ellipses as enclosing regions in Chebyshev acceleration may be overly restrictive and ineffective if the shape of the convex hull of the unwanted eigenvalues bears little resemblance to an ellipse. This has spurred much research in which the acceleration polynomial is chosen so as to minimize an L_2-norm of the polynomial p on the boundary of the convex hull of the unwanted eigenvalues with respect to some suitable weight function ω. The only restriction to this technique is that the degree of the polynomial is limited because of cost and storage requirements. This, however, is overcome by compounding low degree polynomials. The stability of the computation is enhanced by employing a Chebyshev basis and by a careful implementation in which the degree of the polynomial is taken to be the largest one for which the Gram matrix has a tolerable conditioning. The method for computing the least squares polynomial is fully described in [172] but we present a summary of its main features below.

7.6.1 The Least Squares Polynomial

Suppose that we are interested in computing the r eigenvalues of largest real parts $\lambda_1, \lambda_2, \ldots \lambda_r$ and consider the vector

$$z_k = p_k(A)z_0 \tag{7.21}$$

where p_k is a degree k polynomial. Referring to the expansion (7.9) we wish to choose among all polynomials p of degree $\leq k$ one for which $p(\lambda_i), i{>}r$ are small relative to $p(\lambda_i), i \leq r$. Assume that by some adaptive process, a polygonal region H which encloses the remaining eigenvalues becomes available to us. We then arrive at the problem of approximation theory which consists of finding a polynomial of degree k whose value inside some (polygonal) region is small while its values at r particular points (possibly complex) outside the region are large. For simplicity we start with the case where $r = 1$, i.e., only the eigenvalue λ_1 and its associated eigenvectors are sought. We seek a polynomial that is large at λ_1 and small elsewhere. For convenience we can always normalize the polynomial so that

$$p_k(\lambda_1) = 1. \tag{7.22}$$

The desired polynomial satisfying the above constraint can be sought in the form

$$p_k(\lambda) \equiv 1 - (\lambda - \lambda_1)s_k(\lambda) \tag{7.23}$$

where s_k is a polynomial of degree $k - 1$.

Since it is known that the maximum modulus of an analytic function over a region of the complex plane is reached on the boundary of the region, one solution to the above problem is to minimize an L_2-norm associated with some weight function ω, over all polynomials of degree k satisfying the constraint (7.22). We need to choose a weight function ω that will facilitate practical computations.

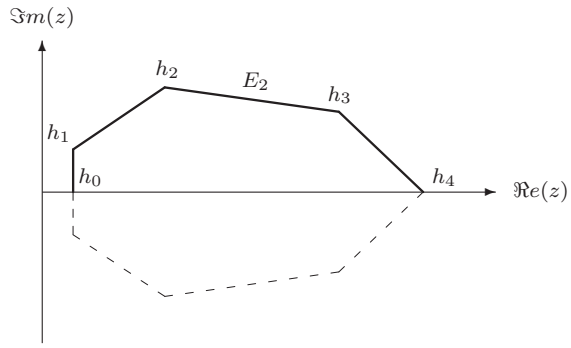


Figure 7.6: Polygon H containing the spectrum of A.

Let the region H of the complex plane, containing the eigenvalues $\lambda_{r+1}, \ldots \lambda_n$, be a polygon consisting of μ edges $E_1, E_2, \ldots E_\mu$, each edge E_j linking two successive vertices h_{j-1} and h_j of H, see Figure 7.6. Denoting by $c_j = \frac{1}{2}(h_j + h_{j-1})$ the center of the edge E_j and by $d_j = \frac{1}{2}(h_j - h_{j-1})$ its half-width, we define the following Chebyshev weight function on each edge:

$$\omega_j(\lambda) = \frac{2}{\pi}|d_j^2 - (\lambda - c_j)^2|^{-1/2}. \tag{7.24}$$

The weight ω on the boundary ∂H of the polygonal region is defined as the function whose restriction to each edge E_j is ω_j. Finally, the L_2-inner-product over ∂H is defined by

$$< p, q >_\omega = \int_{\partial H} p(\lambda)\overline{q(\lambda)}\omega(\lambda)|d\lambda| \tag{7.25}$$

$$= \sum_{j=1}^{\mu} \int_{E_j} p(\lambda)\overline{q(\lambda)}\omega_j(\lambda)|d\lambda|, \tag{7.26}$$

and the corresponding L_2-norm is denoted by $\|.\|_\omega$.

Often, the matrix A is real and the convex hull may be taken to be symmetric with respect to the real axis. In this situation it is better to define the convex hull as the union of two polygons H^+ and H^- which are symmetric to each other. These two are represented in solid line and dashed line respectively in the figure 7.6. Then, when the coefficients of p and q are *real*, we only need to compute the integrals over the edges of the upper part H^+ of H because of the relation

$$< p, q >_\omega = 2\Re e \left[\int_{\partial H^+} p(\lambda)\overline{q(\lambda)}\omega(\lambda)|d\lambda| \right] . \tag{7.27}$$

Having defined an inner product we now define in the simplest case where $r = 1$, the 'least-squares' polynomial that minimizes

$$\|1 - (\lambda - \lambda_1)s(\lambda)\|_\omega. \tag{7.28}$$

Note that there are other ways of defining the least squares polynomial. Assume that we use a certain basis $t_0, \ldots, t_{k-1}$. and let us express the degree $k-1$ polynomial $s(\lambda)$ in this basis as

$$s(\lambda) = \sum_{j=0}^{k-1} \eta_j t_j(\lambda) \,. \tag{7.29}$$

Each polynomial $(\lambda - \lambda_1)t_j(\lambda)$ is of degree $j+1$ and can be expressed as

$$(\lambda - \lambda_1)t_j(\lambda) = \sum_{i=0}^{j+1} \tau_{ij} t_i(\lambda)$$

Denote by η the vector of the η_j's for $j = 0, \ldots, k-1$ and by γ the vector of coefficients $\gamma_j, j = 0, \ldots, k$ of $(\lambda - \lambda_1)s(\lambda)$ in the basis $t_0, \ldots, t_k$ and define $\tau_{ij} = 0$ for $i > j+1$. Then the above two relations state that

$$(\lambda - \lambda_1)s(\lambda) = \sum_{j=0}^{k-1} \eta_j \sum_{i=0}^{k} \tau_{ij} t_i(\lambda) = \sum_{i=0}^{k} \left(\sum_{j=0}^{k-1} \tau_{ij}\eta_j \right) t_i(\lambda)$$

In matrix form this means that

$$\gamma = T_k \eta$$

where T_k is the $(k+1) \times k$ matrix of coefficients t_{ij}'s, which is upper Hessenberg. In fact, it will seen that the matrix T_k is tridiagonal when Chebyshev bases are used.

The least-squares problem (7.28) will translate into a linear least-squares problem for the vector η. We will discuss some of the details of this approach next. There are two critical parts in this technique. The first concerns the choice of the basis and the second concerns the solution least-squares problem.

7.6.2 Use of Chebyshev Bases

To motivate our choice of the basis $\{t_j\}$, we assume at first that the best polynomial is expressed in the 'power' basis

$$1, \lambda, \lambda^2, \cdots \,.$$

Then, the solution of the least-squares problem (7.28) requires the factorization of the Gram matrix consisting of all the inner products $< \lambda^{i-1}, \lambda^{j-1} >_\omega$:

$$M_k = \{< t_j, t_i >_\omega\}_{i,j=0,\ldots,k}.$$

This matrix, often referred to as the moment matrix , can become extremely ill-conditioned and methods based on the use of the power basis will generally be limited to low degree calculations, typically not exceeding 10. A more reliable alternative is to replace the basis $\{\lambda^{i-1}\}$ by a more stable basis. One such basis,

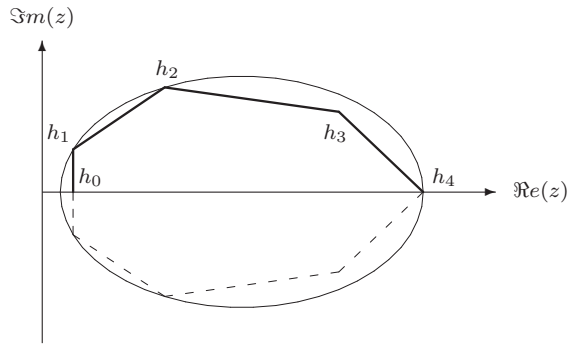


Figure 7.7: The ellipse of smallest area containing the convex hull of $\Lambda(A)$.

well understood in the real case, is that consisting of Chebyshev polynomials over an ellipse that contains the convex hull. The solution polynomial (7.42) will be expressed in terms of a Chebyshev basis associated with the ellipse of smallest area containing H. Such an ellipse is illustrated in Figure 7.7.

Computing the ellipse of smallest area that encloses H is a rather easy task, far easier than that of computing ellipses which minimize convergence rates for Chebyshev acceleration, see Exercise P-7.3 for details.

7.6.3 The Gram Matrix

The next step in the procedure for computing the best polynomial, is to evaluate the Gram matrix M_k. For the Chebyshev basis, the Gram matrix M_k can be constructed recursively *without any numerical integration*.

The entries of the Gram matrix are defined by,

$$m_{ij} =< t_{j-1}, t_{i-1} >_\omega , \quad i, j = 1, \ldots, k+1 .$$

Note that because of the symmetry of the domain, the matrix M_k has real coefficients. We start by expressing each polynomial $t_i(\lambda)$ in terms of the Chebyshev polynomials

$$C_l\left(\frac{\lambda - c_\nu}{d_\nu}\right) \equiv C_l(\xi)$$

for each of the μ edges E_ν, $\nu = 1, \ldots, \mu$. The variable ξ takes real values when λ lies on the edge E_ν. In other words we express each t_i as

$$t_i(\lambda) = \sum_{l=0}^{i} \gamma_{l,\nu}^{(i)} C_l(\xi) , \tag{7.30}$$

$$\xi = \frac{\lambda - c_\nu}{d_\nu} . \tag{7.31}$$

Each polynomial t_i will have μ different expressions of this type, one for each edge E_ν. Clearly, these expressions are redundant since one of them is normally enough to fully determine the polynomial t_i. However, this redundancy is useful from the practical point of view as it allows to perform an efficient computation in a stable manner. The following proposition enables us to compute the Gram matrix from the expressions (7.30).

Proposition 7.1 *Assuming the expressions* (7.30) *for each of the polynomials* t_i, *the coefficients of the Gram matrix* M_k *are given by*

$$m_{i+1,j+1} = 2\,\Re e\left\{\sum_{\nu=1}^{\mu}\left(2\,\gamma_{0,\nu}^{(i)}\overline{\gamma_{0,\nu}^{(j)}} + \sum_{l=1}^{j}\gamma_{l,\nu}^{(i)}\overline{\gamma_{l,\nu}^{(j)}}\right)\right\}\,, \tag{7.32}$$

for all i, j *such that* $0 \le i \le j \le k$.

Proof. The result follows from the orthogonality of the Chebyshev polynomials, the change of variables (7.31) and the expression (7.27). □

We now need to be able to compute the expansion coefficients. Because of the three term-recurrence of the Chebyshev polynomials it is possible to carry the computation of these coefficients in a recursive manner. We rewrite the recurrence relation for the shifted Chebyshev polynomials in the form

$$\beta_{i+1}t_{i+1}(\lambda) = (\lambda - \alpha_i)t_i(\lambda) - \delta_i t_{i-1}(\lambda), i = 0, 1, \ldots, k, \ldots, \tag{7.33}$$

with the convention that $t_{-1} \equiv= 0$ and $\delta_0 = 0$. Using the definitions (7.30) and (7.31), we get for each edge,

$$\beta_{i+1}t_{i+1}(\lambda) = (d_\nu\xi + c_\nu - \alpha_i)\sum_{l=0}^{i}\gamma_{l,\nu}^{(i)}C_l(\xi) - \delta_i\sum_{l=0}^{i-1}\gamma_{l,\nu}^{(i-1)}C_l(\xi)$$

which provides the expressions for t_{i+1} from those of t_i and t_{i-1} by exploiting the relations

$$\begin{aligned}\xi C_l(\xi) &= \frac{1}{2}[C_{l+1}(\xi) + C_{l-1}(\xi)] \quad l > 0\,,\\ \xi C_0(\xi) &= C_1(\xi)\,.\end{aligned}$$

The result is expressed in the following proposition.

Proposition 7.2 *For* $\nu = 1, 2, \ldots \mu$, *the expansion coefficients* $\gamma_{l,\nu}^{(i)}$ *satisfy the recurrence relation,*

$$\beta_{i+1}\gamma_{l,\nu}^{(i+1)} = \frac{d_\nu}{2}\left[\gamma_{l+1,\nu}^{(i)} + \gamma_{l-1,\nu}^{(i)}\right] + (c_\nu - \alpha_i)\gamma_{l,\nu}^{(i)} - \delta_i\gamma_{l,\nu}^{(i-1)} \tag{7.34}$$

for $l = 0, 1, \ldots, i+1$ *with the notational convention,*

$$\gamma_{-1,\nu}^{(i)} \equiv \gamma_{1,\nu}^{(i)}, \quad \gamma_{l,\nu}^{(i)} = 0 \quad \textit{for} \quad l{>}i\,.$$

The total number of operations required for computing a Gram matrix with the help of the above two propositions is $O(\mu k^3/3)$. This cost may be high for high degree polynomials. However, this cost will in general not be significant relatively to the total number of operations required with the matrix A which is typically very large. It is also not recommended to compute least squares polynomials of degree higher than 40 or 50.

7.6.4 Computing the Best Polynomial

In the simple case where we are computing λ_1 and the associated eigenvector, we need the polynomial $s(\lambda)$ which minimizes:

$$J(\eta) = \|1 - (\lambda - \lambda_1)s(\lambda)\|_w \tag{7.35}$$

where $s(\lambda)$ is the unknown polynomial of degree $k-1$ expressed in the form (7.29).

Let T_k be the $(k+1) \times k$ tridiagonal matrix

$$T_k = \begin{pmatrix} \alpha_0 & \delta_1 & & & \\ \beta_1 & \alpha_1 & \delta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-2} & \alpha_{k-2} & \delta_{k-1} \\ & & & \beta_{k-1} & \alpha_{k-1} \\ & & & & \beta_k \end{pmatrix} \tag{7.36}$$

whose coefficients $\alpha_i, \delta_i, \beta_i$ are those of the three-term recurrence (7.33). Given two polynomials of degree k

$$p(\lambda) = \sum_{i=0}^{k} \gamma_i t_i(\lambda) \quad \text{and} \quad q(\lambda) = \sum_{i=0}^{k} \theta_i t_i(\lambda)$$

it is easy to show that the inner product of these two polynomials can be computed from

$$< p, q >_\omega = (M_k \gamma, \theta) \tag{7.37}$$

where $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_k)^T$ and $\theta = (\theta_0, \theta_1, \ldots, \theta_k)^T$. Therefore, an alternative expression for $J(\eta)$ is

$$J(\eta)^2 = [e_1 - (T_k - \lambda_1 I)\eta]^H M_k [e_1 - (T_k - \lambda_1 I)\eta]$$

and as a consequence, we can prove the following theorem.

Theorem 7.1 *Let*

$$M_k = LL^T$$

be the Cholesky factorization of the $(k+1) \times (k+1)$ *Gram matrix* M_k *and denote by* H_k *the* $(k+1) \times k$ *upper Hessenberg matrix*

$$H_k = L^T(T_k - \lambda_1 I),$$

where T_k is the tridiagonal matrix (7.36) defined from the three-term recurrence of the basis t_i. Then the function $J(\eta)$ satisfies the relation,

$$J(\eta) = \|l_{11}e_1 - H_k\eta\|_2. \tag{7.38}$$

Therefore the computation of the best polynomial requires the solution of a $(k+1) \times k$ least squares problem. This is best done by reducing the Hessenberg matrix H_k into upper triangular form by using Givens rotations.

The above theorem does not deal with the case where we have several eigenvalues to compute, i.e., with the case $r > 1$. For this situation, we need to redefine the problem slightly. The following development is also of interest because it gives an alternative formulation to the least squares polynomial even for the case $r = 1$.

We start by introducing what is referred to as *kernel polynomials*,

$$K_k(\xi, \lambda) = \sum_{j=0}^{k} \overline{\pi_j(\xi)}\pi_j(\lambda) \tag{7.39}$$

in which the π_j's are the orthogonal polynomials with respect to the appropriate inner product, here $< ., . >_\omega$. Then the following well-known theorem holds [42].

Theorem 7.2 *Among all polynomials of degree k normalized so that $p(\lambda_1) = 1$, the one with the smallest ω-norm is given by*

$$q_k(\lambda) = \frac{K_k(\lambda_1, \lambda)}{K_k(\lambda_1, \lambda_1)}. \tag{7.40}$$

This gives an interesting alternative to the polynomial derived previously. We will now generalize this result and discuss its practical implementation.

We begin by generalizing the constraint (7.22) by normalizing the polynomial at the points $\lambda_1, \lambda_2, \ldots, \lambda_r$ as follows,

$$\sum_{j=1}^{r} \mu_j p(\lambda_j) = 1 \tag{7.41}$$

in which the μ_j's , $j = 1, \ldots r$ constitute r different weights.

Then we have the following generalization of the above theorem.

Theorem 7.3 *Let $\{\pi_i\}_{i=0,\ldots,k}$ be the first $k+1$ orthonormal polynomials with respect to the L_2-inner-product (7.26). Then among all polynomials p of degree k satisfying the constraint (7.41), the one with smallest ω-norm is given by*

$$p_k(\lambda) = \frac{\sum_{i=0}^{k} \phi_i \pi_i(\lambda)}{\sum_{i=0}^{k} |\phi_i|^2}\ , \tag{7.42}$$

where $\phi_i = \overline{\sum_{j=1}^{r} \mu_j \pi_i(\lambda_j)}$.

Proof. We recall the reproducing property of kernel polynomials [42],

$$< p, K_k(\xi, \lambda) >_\omega = p(\xi) \, , \tag{7.43}$$

in which the integration is with respect to the variable λ. It is easily verified that the polynomial (7.42) satisfies the constraint (7.41) and that p_k can be recast as

$$p_k(\lambda) = C \sum_{j=0}^{k} \overline{\mu_j} K_k(\lambda_j, \lambda) \tag{7.44}$$

where C is some constant. Next, we consider any polynomial p satisfying the constraint (7.41) and write p in the form

$$p(\lambda) \; = \; p_k(\lambda) + E(\lambda),$$

from which we get,

$$\|p\|_\omega^2 = \|p_k\|_\omega^2 + \|E\|_\omega^2 + 2\Re e\{< E, p_k >_\omega\}. \tag{7.45}$$

Since both p and p_k satisfy the constraint (7.41) we must have

$$\sum_{j=1}^{r} \mu_j E(\lambda_j) = 0. \tag{7.46}$$

From (7.44) and from the reproducing property (7.43) we see that

$$\begin{aligned} < E, p_k >_\omega \quad &= \quad C \sum_{j=1}^{r} \mu_j < E, K_k(\lambda_j, \lambda) >_\omega \\ &= \quad C \sum_{j=1}^{r} \mu_j E(\lambda_j) \, . \end{aligned}$$

Hence, from (7.46) $< E, p_k >_\omega = 0$ and (7.45) shows that $\|p\|_\omega \geq \|p_k\|_\omega$ for any p of degree $\leq k$. □

As is now explained, the practical computation of the best polynomial p_k can be carried out by solving a linear system with the Gram matrix M_k. We could also compute the orthogonal polynomials π_j and take their linear combination (7.42) but this would not be as economical.

We consider the unscaled version of the polynomial (7.42) used in (7.44),

$$\hat{p}_k(\lambda) = \sum_{j=1}^{r} \bar{\mu}_j K_k(\lambda_j, \lambda) \, , \tag{7.47}$$

which satisfies a property stated in the next proposition.

Proposition 7.3 *Let t be the $(k+1)$-vector with components*

$$\tau_i = \sum_{j=1}^{r} \mu_j t_{i-1}(\lambda_j) \;, \quad i = 0, \ldots, k \;.$$

Then the coefficients of the polynomial $\hat{p}_k$ in the basis $\{t_j\}$ are the conjugates of the components of the k-vector,

$$\eta = M_k^{-1} t \;.$$

Proof. Consider the Cholesky factorization $M_k = LL^T$ of the Gram matrix M_k. If we represent by $\underline{p}(\lambda)$ and $\underline{t}(\lambda)$ the vectors of size $k+1$ defined by

$$\underline{p}(\lambda) = (\pi_0(\lambda), \pi_1(\lambda), \ldots, \pi_k(\lambda))^T$$

and

$$\underline{t}(\lambda) = (t_0(\lambda), t_1(\lambda), \ldots, t_k(\lambda))^T$$

then we have the important relation,

$$\underline{p}(\lambda) = L^{-1}\underline{t}(\lambda) \tag{7.48}$$

which can be easily verified from (7.37). Notice that $K_k(\xi, \eta) = (\underline{p}(\lambda), \underline{p}(\xi))$ where $(.,.)$ is the complex inner product in $\mathbb{C}^{k+1}$, and therefore, from (7.47) and (7.48) we get

$$\begin{aligned}
\hat{p}_k(\lambda) &= \sum_{j=1}^{r} \bar{\mu}_j \left(\underline{p}(\lambda), \underline{p}(\lambda_j)\right) \\
&= \sum_{j=1}^{r} \bar{\mu}_j \left(L^{-1}\underline{t}(\lambda), L^{-1}\underline{t}(\lambda_j)\right) = \sum_{j=1}^{r} \bar{\mu}_j \left(\underline{t}(\lambda), M_k^{-1}\underline{t}(\lambda_j)\right) \\
&= \left(\underline{t}(\lambda), M_k^{-1} \sum_{j=1}^{r} \mu_j \underline{t}(\lambda_j)\right) = \left(\underline{t}(\lambda), M_k^{-1} t\right) \\
&= \sum_{l=1}^{k+1} \bar{\eta}_l t_{l-1}(\lambda) \;,
\end{aligned}$$

which completes the proof. □

The proposition allows to obtain the best polynomial directly in the desired basis. Note that since the matrix M_k is real, if the τ_i's are real then the coefficient vector η is real if the λ_j's are selected in pairs of conjugate complex numbers.

7.6.5 Least Squares Arnoldi Algorithms

A resulting hybrid method similar to the Chebyshev Arnoldi Algorithm can be easily derived. The algorithm for computing the r eigenvalues with largest real parts is outlined next.

ALGORITHM **7.6 Least Squares Arnoldi Algorithm**

1. **Start:** *Choose the degree k of the polynomial p_k, the dimension m of the Arnoldi subspaces and an initial vector v_1.*
2. **Projection step:**
   (a) *Using the initial vector v_1, perform m steps of the Arnoldi method and get the m approximate eigenvalues $\{\tilde{\lambda}_1, \ldots \tilde{\lambda}_m\}$ of the matrix H_m.*
   (b) *Estimate the residual norms $\rho_i, i = 1, \ldots, r$, associated with the r eigenvalues of largest real parts $\{\tilde{\lambda}_1, \ldots \tilde{\lambda}_r\}$ If satisfied then Stop.*
   (c) *Adapt: From the previous convex hull and the set $\{\tilde{\lambda}_{r+1}, \ldots \tilde{\lambda}_m\}$ construct a new convex hull of the unwanted eigenvalues.*
   (d) *Obtain the new least squares polynomial of degree k.*
   (e) *Compute a linear combination z_0 of the approximate eigenvectors $\tilde{u}_i, i = 1, \ldots, r$.*
3. **Polynomial iteration:**
   Compute $z_k = p_k(A)z_0$. Compute $v_1 = z_k/\|z_k\|$ and goto 2.

As can be seen the only difference with the Chebyshev algorithm is that the polynomial must now be computed. We must explain how the vector z_k can be computed. We will call w_i the auxiliary sequence of vectors $w_i = t_i(A)z_0$. One possibility would be to compute all the w_i's first and then accumulate their linear combination to obtain z_k. However, the w_i can also be accumulated at the same time as they are computed. More precisely, we can use a coupled recurrence as described in the next algorithm.

ALGORITHM **7.7 (For Computing $z_k = p_k(A)z_0$)**

1. **Start:** $\delta_0 := 0, w_0, y_0 = \eta_0 z_0$.
2. **Iterate:** *For $i = 1, 2, \ldots, k$ do:*

$$\begin{aligned} w_{i+1} &= \frac{1}{\beta_{i+1}}\left[(A - \alpha_i I)w_i - \delta_i w_{i-1}\right] , \\ y_i &= y_{i-1} + \eta_i w_{i+1} . \end{aligned}$$

3. **Finish:** $z_k = y_k$.

The intermediate vectors y_i are not related to the vectors z_i only the last vector y_k is.

We cannot, for reasons of space, describe all the details of the implementation. However, we mention that the linear combination at the end of step 3, is usually taken as follows:

$$z_0 = \sum_{i=1}^{r} \rho_i \tilde{u}_i$$

as for the Chebyshev iteration. Note that it is difficult, in general, to choose a linear combination that leads to balanced convergence because it is hard to represent a whole subspace by a single vector. This translates into divergence in many cases especially when the number of wanted eigenvalues r is not small. There is always the possibility of increasing the space dimension m, at a high cost, to ensure convergence but this solution is not always satisfactory from the practical point of view. Use of deflation constitutes a good remedy against this difficulty because it allows to compute one eigenvalue at a time which is much easier than computing a few of them at once. We omit the description of the corresponding algorithm whose general structure is identical with that of Algorithm 7.1.

One attractive feature of the deflation techniques is that the information gathered from the determination of the eigenvalue λ_i is still useful when iterating to compute the eigenvalue λ_{i+1}. The simplest way in which the information can be exploited is by using at least part of the convex hull determined during the computation of λ_i. Moreover, a rough approximate eigenvector associated with λ_{i+1} can be inexpensively determined during the computation of the eigenvalue λ_i and then used as initial vector in the next step for computing λ_{i+1}.

Another solution is to improve the separation of the desired eigenvalues by replacing A by a polynomial in A. This will be seen in the next chapter.

PROBLEMS

P-7.1 Prove that the relation (7.27) holds when the polynomials p and q are real and the polygon is symmetric with respect to the real line.

P-7.2 Show that the recurrence (7.15)-(7.16) can be performed in real arithmetic when A is real but e is complex. Rewrite the recurrence accordingly.

P-7.3 The purpose of this exercise is to develop formulas for the ellipse $E(c, e, a)$ of smallest area enclosing a polygon H. It is assumed that the polygon is symmetric about the real axis. Therefore the ellipse is also symmetric about the real axis. The following result will be assumed, see for example [126]: The best ellipse is either an ellipse that passes through 3 vertices of H and encloses H or an ellipse of smallest area passing through two vertices of H. Formulas for the first case have been established in the literature, see Manteuffel [126]. Therefore, we must only consider the second case. Let $\lambda_1 = (x_1, y_1)$

and $\lambda_2 = (x_2, y_2)$ two points in $\mathbb{R}^2$. We set

$$A = \frac{1}{2}(x2 - x1), \quad B = \frac{1}{2}(x1 + x2),$$
$$S = \frac{1}{2}(y2 - y1), \quad T = \frac{1}{2}(y1 + y2)$$

and define the variable $z = c - B$. At first, assume that $S \neq 0$ and define $Q = (S/T + T/S)/2$. Show that for a given z (which defines c) the only ellipse that passes through λ_1, λ_2 is defined by

$$\begin{aligned} e^2 &= \frac{1}{z}\left[(z + AT/S)(z + AS/T)(z - ST/A)\right] \\ a^2 &= (z + AT/S)(z + AS/T)\ . \end{aligned}$$

Then show that the optimal z is given by

$$z = \frac{A}{\sqrt{Q^2 + 3} \pm Q}$$

where $\pm$ is the sign of AS. In the particular case where $S = 0$ the above formulas break down. But then $c = B$ and one is lead to minimize the area as a function of a. Show that the minimum is reached for $a^2 = 2A^2$ and that the corresponding d is given by $d^2 = 2(A^2 - T^2)$.

P-7.4 Polynomials of degree 2 can be used to calculate intermediate eigenvalues of Hermitian matrices. Suppose we label the eigenvalues increasingly and that we have estimates for $\lambda_1, \lambda_{i-1}, \lambda_i, \lambda_{i+1}, \lambda_n$. Consider the family of quadratic polynomials that take the value 1 at λ_i and whose derivative at λ_i is zero. Find one such polynomial that will be suitable for computing λ_i and the associated eigenvector. Is this a good polynomial? Find a good polynomial for computing the eigenvalue λ_i.

P-7.5 Establish formula (7.37).

P-7.6 Prove Theorem 7.1.

NOTES AND REFERENCES. Part of the material in this Chapter is taken from Saad [173, 172, 174, 176, 171]. The idea of Chebyshev acceleration for eigenvalue problems is an old one and seems to have been first advocated by Flanders and Shortley [61]. However, in a work that has been vastly ignored, Lanczos also did some very interesting contemporary work in acceleration technique [113], see also the related paper [115]. Lanczos' approach is radically different from that of Flanders and Shortley, which is the approach most numerical analysts are familiar with. Concerned about the difficulty in getting eigenvalue estimates, Lanczos proposed as an alternative to compute a polynomial approximation to the Dirac function based on the wanted eigenvalue. The approximation is made over an interval containing the spectrum, which can easily be obtained from Gerschgorin estimates. This turns out to lead to the so-called Fejer kernel in the theory of approximation by trigonometric functions and then naturally to Chebyshev polynomials. His approach is a least squares technique akin to the one proposed by Stiefel [207] and later Saad [172]. Some ideas on implementing Chebyshev acceleration in the complex plane were introduced by Wrigley [224] but the technique did not mature until the 1975 PhD thesis by Manteuffel [125] in which a FORTRAN implementation for solving linear systems appeared. The work in [173] was based on adapting Manteuffel's implementation for the eigenvalue problem. The least squares polynomial approach presented in this chapter is based on the technical report [172] and its revised published version [174]. In my experience, the least squares approach does seem to perform slightly better in practice than the Chebyshev approach. Its drawbacks

(mainly, having to use relatively low degree polynomials) are rather minor in practice. Implicit restarts have been developed in the early 1990s by Sorensen [196], see also [117], and later resulted in the development of the package ARPACK [118]. By one measure the use of explicit restarts, or the use of polynomial filtering may be viewed as obsolete. However, there are situations where explicit filtering is mandatory. In electronic structure calculations, a form of nonlinear Subspace iteration based on Chebyshev polynomials gave superior results to any other methods we have tried, see [227, 225, 226] for details. ■

Chapter 8

PRECONDITIONING TECHNIQUES

The notion of preconditioning is better known for linear systems than it is for eigenvalue problems. A typical preconditioned iterative method for linear systems amounts to replacing the original linear system $Ax = b$ by (for example) the equivalent system $B^{-1}Ax = B^{-1}b$, where B is a matrix close to A in some sense and defined as the product of a lower by an upper sparse triangular matrices. This equivalent system is then handled by a Krylov subspace method. For eigenvalue problems, the best known preconditioning is the so-called shift-and-invert technique which we already mentioned in Chapter 4. If the shift σ is suitably chosen the shifted and inverted matrix $B = (A - \sigma I)^{-1}$, will have a spectrum with much better separation properties than that of the original matrix A and this will result in faster convergence. The term 'preconditioning' here is quite appropriate since the better separation of the eigenvalues around the desired eigenvalue implies that the corresponding eigenvector is likely to be better conditioned.

8.1 Shift-and-invert Preconditioning

One of the most effective techniques for solving large eigenvalue problems is to iterate with the shifted and inverted matrix,

$$(A - \sigma I)^{-1} \tag{8.1}$$

for standard problems and with (for example)

$$(A - \sigma B)^{-1} B \tag{8.2}$$

for a generalized problem of the form $Ax = \lambda Bx$. These methods fall under the general suggestive name shift-and-invert techniques. There are many possible ways of deriving efficient techniques based on shift-and-invert. In this section we will discuss some of the issues with one particular implementation in mind which involves a shift-and-invert preconditioning of Arnoldi's Algorithm.

8.1.1 General Concepts

Typically shift-and-invert techniques are combined with an efficient projection method such as Arnoldi's method or the Subspace iteration. The simplest possible scheme is to choose a shift σ and run Arnoldi's method on the matrix $(A-\sigma I)^{-1}$. Since the eigenvectors of A and $(A-\sigma I)^{-1}$ are identical one can recover the eigenvalues of A from the computed eigenvectors. Note that this can be viewed as an acceleration of the inverse iteration algorithm seen in Chapter 4, by Arnoldi's method, in the same way that the usual Arnoldi method was regarded as an acceleration of the power method. It requires only one factorization with the shifted matrix.

More elaborate algorithms involve selecting automatically new shifts and performing a few factorizations. Strategies for adaptively choosing new shifts and deciding when to refactor $(A-\sigma B)$ are usually referred to as shift-and-invert strategies. Thus, shift-and-invert simply consists of transforming the original problem $(A-\lambda I)x = 0$ into $(A-\sigma I)^{-1}x = \mu x$. The transformed eigenvalues μ_i are usually far better separated than the original ones which results in better convergence in the projection type algorithms. However, there is a trade-off when using shift-and-invert, because the original matrix by vector multiplication which is usually inexpensive, is now replaced by the more complex solution of a linear system at every step. When a new shift σ is selected, the LU factorization of the matrix $(A-\sigma I)$ is performed and subsequently, at every step of Arnoldi's algorithm (or any other projection algorithm), an upper and a lower triangular systems are solved. Moreover, the cost of the initial factorization can be quite high and in the course of an eigenvalue calculation, several shifts, and therefore several factorizations, may be required. Despite these additional costs shift-and-invert is often an extremely useful technique, especially for generalized problems.

If the shift σ is suitably selected the matrix $C = (A-\sigma I)^{-1}$ will have a spectrum with much better separation properties than the original matrix A and therefore should require far less iterations to converge. Thus, the rationale behind the Shift-and-Invert technique is that factoring the matrix $(A-\sigma I)$ once, or a few times during a whole run in which σ is changed a few times, is a price worth paying because the number of iterations required with C is so much smaller than that required with A that the expense of the factorizations is amortized. For the symmetric generalized eigenvalue problem $Bx = \lambda Ax$ there are further compelling reasons for employing shift-and-invert. These reasons are well-known and have been discussed at length in the recent literature, see for example, [146, 148, 56, 189]. The most important of these is that since we must factor one of the matrices A or B in any case, there is little incentive in not factoring $(A-\sigma B)$ instead, to gain faster convergence. Because of the predominance of generalized eigenvalue problems in structural analysis, shift-and-invert has become a fairly standard tool in this application area.

For nonsymmetric eigenvalue problems, shift-and-invert strategies are not as well-known, although the main arguments supporting such techniques are the same as in the Hermitian case. Let us consider the case where the matrices B

and A are real and banded but the shift σ is complex. One possibility is to work entirely in complex arithmetic. This is probably a fine alternative. If the matrix is real, it seems that the approach is a little wasteful and also unnatural. For example, it is known that the eigenvalues of the original matrix pencil come in complex conjugate pairs (at least in the case where B is positive definite). It would be desirable to have algorithms that deliver complex conjugate pairs as well. This is mainly because there may be a few close pairs of computed eigenvalues and it will become difficult to match the various pairs together if the conjugates are only approximately computed. A wrong match may in fact give incorrect eigenvectors. In the next section we consider the problem of performing the computations in real arithmetic.

8.1.2 Dealing with Complex Arithmetic

Let A be real and assume that we want to use a complex shift

$$\sigma = \rho + i\theta \; . \tag{8.3}$$

One can factor the matrix $(A - \sigma I)$ in (8.1) and proceed with an algorithm such as Arnoldi's method working with complex arithmetic. However, an alternative to using complex arithmetic is to replace the complex operator $(A - \sigma)^{-1}$ by the real one

$$B_+ = \Re e \left[(A - \sigma I)^{-1}\right] = \frac{1}{2}\left[(A - \sigma I)^{-1} + (A - \bar{\sigma} I)^{-1}\right] \tag{8.4}$$

whose eigenvectors are the same as those of the original problem and whose eigenvalues μ_i^+ are related to the eigenvalues λ_i of A by

$$\mu_i^+ = \frac{1}{2}\left(\frac{1}{\lambda_i - \sigma_i} + \frac{1}{\lambda_i - \bar{\sigma}_i}\right). \tag{8.5}$$

We can also use

$$B_- = \Im m \left[(A - \sigma I)^{-1}\right] = \frac{1}{2i}\left[(A - \sigma I)^{-1} - (A - \bar{\sigma} I)^{-1}\right] \; . \tag{8.6}$$

Again, the eigenvectors are the same as those of A and the eigenvalues μ_i^- are given by

$$\mu_i^- = \frac{1}{2i}\left(\frac{1}{\lambda_i - \sigma_i} + \frac{1}{\lambda_i - \bar{\sigma}_i}\right) \; . \tag{8.7}$$

A few additional possibilities are the following

$$B(\alpha, \beta) = \alpha B_+ + \beta B_- \; ,$$

for any nonzero pair α, β and

$$B_* = (A - \sigma I)^{-1}(A - \bar{\sigma})^{-1}. \tag{8.8}$$

This last option is known as the double shift approach and has been used by J.G.F. Francis in 1961/62 [63] in the context of the QR algorithm to solve a similar dilemma. The inverse of B_* is

$$(A - \sigma I)(A - \bar{\sigma} I) = [(A - \rho I)^2 + \theta^2 I].$$

This matrix, which is real, and is a quadratic polynomial in A and again shares A's eigenvectors. An interesting observation is that (8.8) is redundant with (8.6).

Proposition 8.1 *The matrices B_* and B_- are related by*

$$B_- = \theta B_* \,. \tag{8.9}$$

The proof is left as an exercise, see Exercise P-8.4.

An obvious advantage in using either (8.4) or (8.6) in place of (8.1) is that the first operator is real and therefore all the work done in the projection method can be performed in real arithmetic. A nonnegligible additional benefit is that the complex conjugate pairs of eigenvalues of original problem are also approximated by complex conjugate pairs thus removing some potential difficulties in distinguishing these pairs when they are very close. In a practical implementation, the matrix $(A - \sigma I)$ must be factored into the product LU of a lower triangular matrix L and an upper triangular matrix U. Then every time the vector $w = \Re e[(A - \sigma I)^{-1}]v$ must be computed, the forward and backward solves are processed in the usual way, possibly using complex arithmetic, and then the real part of the resulting vector is taken to yield w.

An interesting question that might be asked is which of (8.4) or (8.6) is best? The experiments in [152] reveal that the choice is not an easy one. It is readily verified that as $\lambda \rightarrow \sigma$,

$$\mu^+ \approx \frac{1}{2(\lambda - \sigma)} \,, \quad \mu^- \approx \frac{1}{2i(\lambda - \sigma)}.$$

showing that B_+ and B_- give equal enhancement to eigenvalues near σ. In contrast, as $\lambda \rightarrow \infty$, B_- dampens the eigenvalues more strongly than does B_+ since,

$$\mu^+ = \frac{\lambda - \rho}{(\lambda - \sigma)(\lambda - \bar{\sigma})} \,, \quad \mu^- = \frac{\theta}{(\lambda - \sigma)(\lambda - \bar{\sigma})} \,. \tag{8.10}$$

The only conclusion from all this is that whichever of the two options is used the performance is not likely to be substantially different from the other or from that of the standard (8.1).

In the following discussion we choose to single out B_+, but all that is said about B_+ is also true of B_-. In practice it is clear that the matrix B_+ should not be computed explicitly. In fact either of these matrices is full in general and would be prohibitive to compute. Instead, we first factor the matrix $(A - \sigma I)$ at the outset. This is done in complex arithmetic or by implementing complex arithmetic with real arithmetic. For example, if A is banded, to preserve bandedness and still use real arithmetic, one can represent the j-th component $x_j = \xi_j + i\zeta_j$ of a

vector z of $\mathbb{C}^n$ by the components $\eta_{2j-1} = \xi_j$ and $\eta_{2j} = \zeta_j$ of the real 2n-vector y of the components η_j, $j = 1, \ldots, 2n$. Translating the matrix $(A - \sigma I)$ into this transformation gives a $(2n) \times (2n)$ real banded matrix. Once the matrix is factored, a projection type method, e.g., subspace iteration, is applied using as operator $B_+ = \Re e(A - \sigma I)$. Matrix-vector products with the matrix B_+ are required in the subspace iteration. Each of these can be performed by first solving $(A - \sigma I)w = v$, possibly in complex arithmetic, and then setting $B_+ v = \Re e(w)$ (respectively $B_- v = \Im m(w)$).

8.1.3 Shift-and-Invert Arnoldi

We now consider the implementation of shift-and-invert with an algorithm such as Arnoldi's method. Suppose we must compute the p eigenvalues closest to a shift σ_0. In the symmetric case an important tool is available to determine which of the approximate eigenvalues should be considered in order to compute all the desired eigenvalues in a given interval only once. This tool is Sylvester's inertia theorem which gives the number of eigenvalues to the right and left of σ by counting the number of negative entries in the diagonal elements of the U part of the LU factorization of the shifted matrix. In the non Hermitian case a similar tool does not exist. In order to avoid the difficulty we exploit deflation in the following manner. As soon as an approximate eigenvalue has been declared satisfactory we proceed to a deflation process with the corresponding Schur vector. The next run of Arnoldi's method will attempt to compute some other eigenvalue close to σ_0. With proper implementation, the next eigenvalue will usually be the next closest eigenvalue to σ_0. However, there is no guarantee for this and so we cannot be sure that eigenvalues will not be missed. This is a weakness of projection methods in the non Hermitian case, in general.

Our experimental code ARNINV based on this approach implements a simple strategy which requires two parameters m, k_{rest} from the user and proceeds as follows. The code starts by using σ_0 as an initial shift and calls Arnoldi's algorithm with $(A - \sigma_0 I)^{-1}$ Arnoldi to compute the eigenvalue of A closest to σ_0. Arnoldi's method is used with restarting, i.e., if an eigenvalue fails to converge after the Arnoldi loop we reran Arnoldi's algorithm with the initial vector replaced by the eigenvalue associated with the eigenvalue closest to σ_0. The strategy for changing the shift is dictated by the second parameter k_{rest}. If after k_{rest} calls to Arnoldi with the shift σ_0 the eigenpair has not yet converged then the shift σ_0 is changed to the best possible eigenvalue close to σ_0 and we repeat the process. As soon as the eigenvalue has converged we deflate it using Schur deflation as described in the previous section. The algorithm can be summarized as follows.

ALGORITHM **8.1 Shift-and-Invert Arnoldi**

1. **Initialize:**

   Choose an initial vector v_1 of norm unity, an initial shift σ, and the dimension and restart parameters m and k_{rest}.

2. **Eigenvalue loop:**

   (a) *Compute the LU factorization of* $(A - \sigma I)$.

   (b) *If* $k > 1$ *then (re)-compute*

$$h_{ij} = ((A - \sigma I)^{-1} v_j, v_i)\ i, j = 1, k-1\ .$$

   (c) *Arnoldi Iteration. For* $j = k, k+1, ..., m$ *do:*

   - *Compute* $w := (A - \sigma I)^{-1} v_j$.
   - *Compute a set of* j *coefficients* h_{ij} *so that* $w := w - \sum_{i=1}^{j} h_{ij} v_i$ *is orthogonal to all previous* v_i*'s,* $i = 1, 2, ..., j$.
   - *Compute* $h_{j+1,j} := \|w\|_2$ *and* $v_{j+1} := w/h_{j+1,j}$.

   (d) *Compute eigenvalue of* H_m *of largest modulus, corresponding approximate eigenvector of* $(A-\sigma I)^{-1}$*, and associated (estimated) residual norm* ρ_k.

   (e) *Orthonormalize this eigenvector against all previous* v_j*'s to get the approximate Schur vector* $\tilde{u}_k$ *and define* $v_k := \tilde{u}_k$.

   (f) *If* ρ_k *is small enough then accept* v_k *as the next Schur vector. Set* $k : k+1$*; if* $k<p$ *goto 2.*

   (g) *If the number of restarts with the same shift exceeds* k_{rest} *select a new shift and goto 1. Else restart Arnoldi's algorithm, i.e., goto 2-(c).*

A point of detail in the algorithm is that the $(k-1) \times (k-1)$ principal submatrix of the Hessenberg matrix H_m is recomputed whenever the shift changes. The reason is that this submatrix represents the matrix $(A - \sigma I)^{-1}$ in the first $k-1$ Schur vectors and therefore it must be updated as σ changes. This is in contrast with the simple Arnoldi procedure with deflation described earlier in Chapter 6. However, there exists a simpler implementation that avoids this, see Exercise P-8.2. The above algorithm is described for general complex matrix and there is no attempt in it to avoid complex arithmetic in case the original matrix is real. In this situation, we must replace $(A - \sigma I)^{-1} v_j$ in B.2 by $\Re e[(A - \sigma I)^{-1} v_j]$ and ensure that we select the eigenvalues corresponding to the eigenvalues of A closest to σ. We also need to replace the occurrences of eigenvectors by the pair of real parts and imaginary parts of the eigenvectors.

Example 8.1. We consider the test problem on Chemical reactions described in Chapter 3. This coupled system is discretized in the interval $[0, 1]$ using $n_x + 1$ points with $n_x = 100$ which yields a matrix of size $n = 200$. We tested ARNINV to compute the six rightmost eigenvalues of A. We took as initial shift the value $\sigma = 0$, and $m = 15$, $k_{rest} = 10$. In this case ARNINV delivered all the desired eigenvalues by making four calls to the Arnoldi subroutine and there was no need to change shifts. The tolerance imposed was $\epsilon = 10^{-7}$. The result of the execution is shown in Table 8.1. What is shown in the figure is the progress of the algorithm

Eig.	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
1	0.1807540453D-04	0.2139497548D+01	0.212D-09
	0.1807540453D-04	-0.2139497548D+01	0.212D-09
	-0.6747097569D+00	0.2528559918D+01	0.224D-06
	-0.6747097569D+00	-0.2528559918D+01	0.224D-06
3	-0.6747097569D+00	0.2528559918D+01	0.479D-13
	-0.6747097569D+00	-0.2528559918D+01	0.479D-13
	-0.2780085122D+01	0.2960250300D+01	0.336D-01
	-0.2780085122D+01	-0.2960250300D+01	0.336D-01
5	-0.1798530837D+01	0.3032164644D+01	0.190D-06
	-0.1798530837D+01	-0.3032164644D+01	0.190D-06
5	-0.1798530837D+01	0.3032164644D+01	0.102D-11
	-0.1798530837D+01	-0.3032164644D+01	0.102D-11
	-0.2119505960D+02	0.1025421954D+00	0.749D-03

Table 8.1: Convergence history of ARNINV for chemical reaction test problem. Each separate outer iteration corresponds to a call to Arnoldi's module

after each projection (Arnoldi) step. The eigenvalue loop number indicates the eigenvalue that is being computed at the particular Arnoldi call. Thus, when trying to compute the eigenvalue number 3, the algorithm has already computed the first two (in this case a complex conjugate pair), and has deflated them. We print the eigenvalue of interest, i.e., the one we are trying to compute, plus the one (or the pair of complex conjugate eigenvalues) that is likely to converge after it. The last column shows the actual residual norm achieved for the eigenvalues shown. After execution, we computed the average error for the 6 computed eigenvalues and found that it was equal to 0.68×10^{-14}. The total execution time on an Alliant FX-8 computer was about 2.13 seconds.

We reran the above test with a larger number of eigenvalues to compute, namely $nev = 10$. The initial shift σ, was changed to $\sigma_0 = -0.5 + 0.2i$ and we also changed k_{rest} to $k_{rest} = 3$. Initially, the run looked similar to the previous one. A pair of complex conjugate eigenvalues were found in the first Arnoldi iteration, then another pair in the second iteration, then none in the third iteration and one pair in the fourth iteration. It took two more iterations to get the eigenvalues number 7 and 8. For the last eigenvalue a new shift was taken because it took three Arnoldi iterations without success. However the next shift that was taken was already an excellent approximation and the next eigenvalue was computed in the next iteration. The cost was higher than the previous run with the CPU time on the Alliant FX-8 climbing to approximately 5.65 seconds. □

8.2 Polynomial Preconditioning

We have seen in the previous chapter a few different ways of exploiting polynomials in A to accelerate simple algorithms such as Arnoldi's method or subspace iteration. In this section we will show another way of combining a projection type technique such as Arnoldi's method with these polynomials.

For a classical eigenvalue problem, one alternative is to use polynomial preconditioning as is described next. The idea of polynomial preconditioning is to replace the operator B by a simpler matrix provided by a polynomial in A. Specifically, we consider the polynomial in A

$$B_k = p_k(A) \tag{8.11}$$

where p_k is a degree k polynomial. Ruhe [165] considers a more general method in which p_k is not restricted to be a polynomial but can be a rational function. When an Arnoldi type method is applied to B_k, we do not need to form B_k explicitly, since all we will ever need in order to multiply a vector x by the matrix B_k is k matrix-vector products with the original matrix A and some linear combinations.

For fast convergence, we would ideally like that the r wanted eigenvalues of largest real parts of A be transformed by p_k into r eigenvalues of B_k that are very large as compared with the remaining eigenvalues. Thus, we can proceed as in the previous chapter by attempting to minimize some norm of p_k in some region subject to constraints of the form,

$$p(\lambda_1) = 1 \quad \text{or} \quad \sum_{j=1}^{r} \mu_j p(\lambda_j) = 1 \ . \tag{8.12}$$

Once again we have freedom in choosing the norm of the polynomials, to be either the infinity norm or the L_2-norm. Because the L_2-norm offers more flexibility and performs usually slightly better than the infinity norm, we will only consider a technique based on the least squares approach. We should emphasize, however, that a similar technique using Chebyshev polynomials can easily be developed. Therefore, we are faced again with the function approximation problem described in Section 3.3.

Once p_k is calculated, the preconditioned Arnoldi process consists of using Arnoldi's method with the matrix A replaced by $B_k = p_k(A)$. This will provide us with approximations to the eigenvalues of B_k which are related to those of A by $\lambda_i(B_k) = p_k(\lambda_i(A))$ It is clear that the approximate eigenvalues of A can be obtained from the computed eigenvalues of B_k by solving a polynomial equation. However, the process is complicated by the fact that there are k roots of this equation for each value $\lambda_i(B_k)$ that are candidates for representing one eigenvalue $\lambda_i(A)$. The difficulty is by no means unsurmountable but we have preferred a more expensive but simpler alternative based on the fact that the eigenvectors of A and B_k are identical. At the end of the Arnoldi process we obtain an orthonormal basis V_m which contains all the approximations to these eigenvectors.

A simple idea is to perform a Galerkin process for A onto $\mathrm{span}[V_m]$ by explicitly computing the matrix $A_m = V_m^H A V_m$ and its eigenvalues and eigenvectors. Then the approximate eigenvalues of A are the eigenvalues of A_m and the approximate eigenvectors are given by $V_m y_i^{(m)}$ where $y_i^{(m)}$ is an eigenvector of A_m associated with the eigenvalue $\tilde{\lambda}_i$. A sketch of the algorithm for computing nev eigenvalues is as follows.

ALGORITHM **8.2 Least-Squares Preconditioned Arnoldi**

1. **Start:** *Choose the degree k of the polynomial p_k, the dimension parameter m and an initial vector v_1. Set $iev = 1$.*
2. **Initial Arnoldi Step:** *Using the initial vector v_1, perform m steps of the Arnoldi method with the matrix A and get initial set of Ritz values for A.*
3. **Eigenvalue Loop:**
   (a) *Adapt: From the previous convex hull and the new set of Ritz values construct a new convex hull of the unwanted eigenvalues. Obtain the new least squares polynomial p_k of degree k.*
   (b) *Update H_m: If $iev > 1$ then (re)-compute*
   $$h_{ij} = (p_k(A)v_j, v_i)\ i, j = 1, iev - 1\ .$$
   (c) *Arnoldi Iteration: For $j = iev, iev + 1, ..., m$ do:*
   - *Compute $w := p_k(A)v_j$*
   - *Compute a set of j coefficients h_{ij} so that $w := w - \sum_{i=1}^{j} h_{ij} v_i$ is orthogonal to all previous v_i's, $i = 1, 2, ..., j$.*
   - *Compute $h_{j+1,j} := \|w\|_2$ and $v_{j+1} := w/h_{j+1,j}$.*

   (d) *Projection Step: Compute the matrix $A_m = V_m^T A V_m$ and its m eigenvalues $\{\tilde{\lambda}_1, \ldots \tilde{\lambda}_m\}$.*
   (e) *Select the next wanted approximate eigenvalue $\tilde{\lambda}_{iev}$ and compute corresponding eigenvector $\tilde{z}$. Orthonormalize this eigenvector against $v_1, \ldots, v_{iev-1}$ to get the approximate Schur vector $\tilde{u}_{iev}$ and define $v_{iev} := \tilde{u}_{iev}$.*
   (f) *Test. If ρ_{iev} is small enough then accept v_{iev} as the next Schur vector and set $iev := iev + 1$.*
   (g) *Restart: if $iev < nev$ goto 2.*

The general structure of the algorithm is quite close to that of shift-and-invert with deflation. What differentiates the two algorithms is essentially the fact that here we need to adaptively compute a polynomial, while the shift-and-invert algorithm computes an LU factorization of a shifted matrix. Practically, we must be careful about the number of factorizations needed in shift-and-invert whereas the computational cost of calculating a new polynomial is rather low. The difference

between this method and those of the previous chapter is that here the polynomial iteration is an inner iteration and the Arnoldi iteration is the outer loop, while in the hybrid method, the two processes are serially following each other. Both approaches can be viewed as means of accelerating the Arnoldi method.

It is clear that a version without the Schur-Wielandt deflation technique can also be applied to the polynomial preconditioned Arnoldi method but this is not recommended.

Example 8.2. We take the same example as in the previous section and illustrate the use of an experimental least squares Arnoldi program called ARNLS on the above example. We fixed the dimension of the Krylov subspace to be always equal to $m = 15$. The degree of the polynomial was taken to be 20. However, note that the program has the capability to lower the degree by as much as is required to ensure a well conditioned Gram matrix in the least squares polynomial problem. This did not happen in this run however, i.e. the degree was always 20. Again, ARNLS was asked to compute the six rightmost eigenvalues. The run was much longer so its history cannot be reproduced here. Here are however a few statistics.

- Total number of matrix by vector multiplications for the run = 2053;
- Number of calls to the projection subroutines = 9;
- Total CPU time used on an Alliant FX-8 = 3.88 sec.

Note that the number of projection steps is more than twice that required for shift-and-invert. The execution time is also more than 80 % higher. We reran the same program by changing only two parameters: m was increased to $m = 20$ and the degree of the polynomial was set to $k = 15$. The statistics are now as follows:

- Total number of matrix by vector multiplications for the run = 1144;
- Number of calls to the projection subroutines = 5;
- Total CPU time used = 2.47 sec.

Both the number of projection steps and the execution times have been drastically reduced and have come closer to those obtained with shift-and-invert. □

One of the disadvantages of polynomial preconditionings is precisely this wide variation in performance depending on the choice of the parameters. To some extent there is a similar dependence of the performance of ARNINV on the initial shift, although in practice a good initial shift is often known. A superior feature of shift-and-invert is that it allows to compute eigenvalues inside the spectrum. Polynomial preconditioning can be generalized to this case but does not perform too well. We should also comment on the usefulness of using polynomial preconditioning in general. A commonly heard argument against polynomial

preconditioning is that is it suboptimal: In the Hermitian case the conjugate gradient and the Lanczos methods are optimal polynomial processes in that they provide the best possible approximation, in some sense, to the original problem from Krylov subspaces. Hence the argument that polynomial preconditioning would not perform as well since it si likely to require a larger number of matrix by vector multiplications. However, in the non Hermitian case the optimality result is no longer valid. In fact even in the symmetric case the optimality result is only true in exact arithmetic, which is far from real situations in which loss of orthogonality can be rather severe. A notable difference with the situation of linear system solutions is that the overhead in computing the best ellipse and best polynomial may now be amortized over several eigenvalues. In fact one single outer loop may enable one to compute a few eigenvalues/eigenvectors and not just one.

The next question is whether or not a simple restarted Arnoldi algorithm would perform better than a polynomial preconditioned method. The answer is a definite no. A run with ARNIT [177] an iterative Arnoldi method with deflation failed even to deliver the first eigenvalue of the test matrix used in the above example. The initial vector was the same and we tried two cases $m = 15$, which did not show any sign of convergence and $m = 20$ which might have eventually converged but was extremely slow. The nonrestarted Arnoldi method would, however be of interest, if not for its excessive memory requirement.

8.3 Davidson's Method

Davidson's method is a generalization of the Lanczos algorithm in that like the Lanczos algorithm it uses projections of the matrix over a sequence of subspaces of increasing dimension. It is indeed a preconditioned version of the Lanczos method. The difference with the Lanczos algorithm is that the amount of work required at each step increases at each iteration because, just like in Arnoldi's method, we must orthogonalize against all previous vectors. From the implementation point of view the method is akin to Arnoldi's method. For example, the process must be restarted periodically with the current best estimate of the wanted eigenvector.

The basic idea of the algorithm is rather simple. It consists of generating an orthogonal set of vectors onto which a projection is performed. At each step j, (this is the equivalent to the j-th step in the Lanczos algorithm) the residual vector of the current approximation $\tilde{\lambda}, \tilde{u}$ to the desired eigenpair is computed. The resulting vector is then multiplied by $(M - \tilde{\lambda}I)^{-1}$, where M is some preconditioning matrix. In the original algorithms M was simply the diagonal of the matrix A.

Thus, the algorithm consists of two nested loops. The process for computing the largest (resp. smallest) eigenvalue of A, can be described as follows.

ALGORITHM **8.3 Davidson's method.**

1. **Start:** *Choose an initial unit vector* v_1.
2. **Iterate:** *Until convergence do:*

3. **Inner Loop:** *for* $j = 1, \ldots, m$ *do:*

- *Compute* $w := Av_j$.
- *Compute* $V_j^T w$, *the last column of* $H_j := V_j^T A V_j$.
- *Compute the largest eigenpair* λ, y *of* H_j.
- *Compute the Ritz vector* $u := V_j y$ *and its associated residual vector* $r := Au - \lambda u$.
- *Test for convergence. If satisfied Return.*
- *Compute* $t := M_j r$ *(skip when* $j = m$*).*
- *Orthogonalize* t *against* V_j *via Modified Gram-Schmidt:* $V_{j+1} := MGS([V_j, t])$ *(skip when* $j = m$*).*

4. **Restart:** *Set* $v_1 := u$ *and go to 3.*

The preconditioning matrix M_j is normally some approximation of $(A - \lambda I)^{-1}$. As was already mentioned the simplest and most common preconditioner M_j is $(D - \lambda I)^{-1}$ where D is the main diagonal of A (Jacobi Preconditioner). It can only be effective when A is nearly diagonal, i.e., when matrix of eigenvectors is close to the identity matrix. The fact that this is often the situation in Quantum Chemistry explains the popularity of the method in this field. However, the preconditioner need not be as simple. It should be noticed that, without preconditioning, i.e., when if $M_j = I$ for all j, then the sequence of vectors v_j coincide with those produced by the Lanczos algorithm, so that the Lanczos and Davidson algorithms are equivalent in this case.

When several eigenvalues are sought or when it is known that there is a cluster of eigenvalues around the desired eigenvalue then a block version of the algorithm may be preferred. Then several eigenpairs of H_j will be computed at the same time and several vectors are added to the basis V_j instead of one.

We state a general convergence result due to Sadkane [182]. In the following, we assume that we are seeking to compute the largest eigenvalue λ_1. We denote by $\mathcal{P}_j$ the projection onto a subspace K_j spanned by an orthonormal basis V_j. Thus, the *nonrestarted* Davidson algorithm is just a particular case of this situation.

Theorem 8.1 *Assuming that the Ritz vector* $u_1^{(j)}$ *belongs to* K_{j+1}*, then the sequence of Ritz values* $\lambda_1^{(j)}$ *is an increasing sequence that is convergent. If, in addition, the preconditioning matrices are uniformly bounded and uniformly positive definite in the orthogonal complement of* K_j *and if the vector* $(I - \mathcal{P}_j)M_j r_j$ *belongs to* K_{j+1} *for all* j *then the limit of* $\lambda_1^{(j)}$ *as* $j \to \infty$ *is an eigenvalue of* A *and* $u_1^{(j)}$ *admits a subsequence that converges to an associated eigenvector.*

Proof. For convenience the subscript 1 is dropped from this proof. In addition we assume that all matrices are *real* symmetric. That $\lambda^{(j)}$ is an increasing sequence is a consequence of the assumptions and the min-max theorem. In addition, the $\lambda^{(j)}$ is bounded from above by λ and as result it converges.

To prove the second part of the theorem, let us define $z_j = (I - \mathcal{P}_j)M_j r_j$ and $w_j = z_j/\|z_j\|_2$. Note that since $u^{(j)} \perp z_j$ and $r_j \perp K_j$ we have,

$$\begin{aligned} z_j^H A u^{(j)} &= z_j^H(\lambda^{(j)}u^{(j)} + r_j) \\ &= r_j^H M_j(I - \mathcal{P}_j)r_j \\ &= r_j^H M_j r_j. \end{aligned} \tag{8.13}$$

Consider the 2-column matrix $W_j = [u^{(j)}, w_j]$ and let

$$B_j = W_j^H A W_j = \begin{pmatrix} \lambda^{(j)} & \alpha_j \\ \alpha_j & \beta_j \end{pmatrix} \tag{8.14}$$

in which we have set $\alpha_j = w_j^H A u^{(j)}$ and $\beta_j = w_j^H A w_j$. Note that by the assumptions $\mathrm{span}\{W_j\}$ is a subspace of K_{j+1}. Therefore, by Cauchy's interlace theorem and the optimality properties of the Rayleigh Ritz procedure the smallest of two eigenvalues $\mu_1^{(j)}, \mu_2^{(j)}$ of B_j satisfies the relation

$$\lambda^{(j)} \leq \mu_1^{(j)} \leq \lambda^{(j+1)}.$$

The eigenvalues of B_j are defined by $(\mu - \lambda^{(j)})(\mu - \beta_j) - \alpha_j^2 = 0$ and as a result of $|\mu_1^{(j)}| \leq \|A\|_2$ and $|\beta_j| \leq \|A\|_2$ we

$$\alpha_j^2 \leq 2(\mu_1^{(j)} - \lambda^{(j)})\|A\|_2 \leq (\lambda^{(j+1)} - \lambda^{(j)})\|A\|_2 \,.$$

The right hand side of the above inequality converges to zero as $j \to \infty$ and so $\lim_{j\to\infty} = 0$. From (8.13),

$$r_j^H M_j r_j = \|z_j\|_2 \alpha_j \leq \|(I - \mathcal{P}_j)M_j r_j\|\alpha_j \leq \|M_j r_j\|\alpha_j \,.$$

Since we assume that M_j is uniformly bounded and using the fact that $\|r_j\|_2 \leq 2\|A\|_2$ the above inequality shows that

$$\lim_{j\to\infty} r_j^H M_j r_j = 0.$$

In addition, since r_j belongs to the orthogonal complement of K_j and by the uniform positive definiteness of the sequence M_j, $r_j^H M_j r_j \geq \gamma \|r_j\|_2{}^2$ where γ is some positive constant. Therefore, $\lim_{j\to\infty} r_j = 0$. To complete the proof, let $\bar{\lambda}$ the limit of the sequence $\lambda^{(j)}$. The $u^{(j)}$'s are bounded since they are all of norm unity so they admit a limit point. Taking the relation $r_j = (A - \lambda^{(j)}I)u^{(j)}$, to the limit, we see that any such limit point $\bar{u}$, must satisfy $(A - \bar{\lambda}I)\bar{u} = 0$. □

The result given by Sadkane includes the more general case where more than one eigenvalue is computed by the algorithm and is therefore more general, see Exercise P-8.1 for details. The restriction on the positive definiteness of the M_j 's is a rather severe condition in the case where the eigenvalue to be computed is not the largest one. The fact that M_j must remain bounded is somewhat less restrictive. However, in shift-and-invert preconditioning, for example, an unbounded

M_j is sought rather than avoided. If we want to achieve rapid convergence, it is desirable to have M_j close to some $(A - \sigma I)^{-1}$ in some sense and σ close to the desired eigenvalue. The assumptions of the theorem do not allow us to take σ too close from the desired eigenvalue. Nevertheless, this result does establish convergence in some instances and we should add that little else is known concerning the convergence of Davidson's algorithm.

8.4 The Jacobi-Davidson approach

The Jacobi-Davidson approach can be viewed as an improved version of the Davidson approach and it is best introduced via a perturbation argument which in effect describes the Newton approach for solving nonlinear systems of equations.

8.4.1 Olsen's Method

We assume that we have a preconditioner, i.e., an approximation M to the original matrix A and write

$$M = A + E. \tag{8.15}$$

Along with this, an approximate eigenpair (μ, z) of A is available which satisfies

$$Az = \mu z + r \tag{8.16}$$

where r is the residual $r \equiv (A - \mu I)z$.

Our goal is to find an improved eigenpair $(\mu+\eta, z+v)$ to the current eigenpair (μ, z). For this we can set as a goal to solve the following equation for the desired eigenpair:

$$A(z + v) = (\mu + \eta)(z + v).$$

Neglecting the second order term ηv, replacing A by its preconditioner M, and rearranging the equation we arrive at the so-called *correction equation*

$$(M - \mu I)v - \eta z = -r. \tag{8.17}$$

The unknowns are η (a scalar) and v (a vector). This is clearly an underdertermined system and a constraint must be added. For example, we could require that the new vector $z + v$ be of 2-norm unity. This will yield the quadratic constraint, $(z + v)^H(z + v) = 1$ from which second-order terms can be ignored to yield the linear condition

$$z^H v = 0. \tag{8.18}$$

Note that a more general constraint of the form $w^H v = 0$ can also be imposed where w is some other vector.

The equations (8.17) and (8.18) can be put in the following matrix form:

$$\begin{pmatrix} M - \mu I & -z \\ z^H & 0 \end{pmatrix} \begin{pmatrix} v \\ \eta \end{pmatrix} = \begin{pmatrix} -r \\ 0 \end{pmatrix}. \tag{8.19}$$

The unknown v can be eliminated from the second equation. This is done by assuming that $M - \mu I$ is nonsingular and extracting $v = (M - \mu I)^{-1}[\eta z - r]$ from the first equation which is substituted in the second equation to yield $z^H v = z^h(M-\mu I)^{-1}[\eta z - r] = 0$ or $z^H(M-\mu I)^{-1}\eta z = (M-\mu I)^{-1}r$. This determines η which is substituted in the first part of (8.19). In the end, the solution to the system (8.19) is:

$$\eta = \frac{z^H(M-\mu I)^{-1}r}{z^H(M-\mu I)^{-1}z} \qquad v = -(M-\mu I)^{-1}(r - \eta z). \tag{8.20}$$

This solution was proposed by Olsen et al [139] and the corresponding correction is sometimes known as Olsen's method.

It is worthwhile to generalize the above equations slightly. As was mentioned earlier we can replace the orthogonality condition (8.18) by some other orthogonality condition of the form

$$w^H v = 0. \tag{8.21}$$

In this situation, (8.19) becomes

$$\begin{pmatrix} M-\mu I & -z \\ w^H & 0 \end{pmatrix} \begin{pmatrix} v \\ \eta \end{pmatrix} = \begin{pmatrix} -r \\ 0 \end{pmatrix}. \tag{8.22}$$

and its solution (8.20) is replaced by

$$\eta = \frac{w^H(M-\mu I)^{-1}r}{w^H(M-\mu I)^{-1}z} \qquad v = -(M-\mu I)^{-1}(r - \eta z). \tag{8.23}$$

8.4.2 Connection with Newton's Method

We already mentioned the relationship of this approach with Newton's method. Indeed, consider one step of Newton's method for solving the (nonlinear) system of equations

$$\begin{cases} (A-\lambda I)u &= 0 \\ \frac{1}{2}u^T u - 1 &= 0 \end{cases}.$$

The unknown is the pair $\begin{pmatrix} u \\ \lambda \end{pmatrix}$ and the current approximation is $\begin{pmatrix} z \\ \mu \end{pmatrix}$. One step of Newton's method corresponds to the following operation:

$$\begin{pmatrix} z_{new} \\ \mu_{new} \end{pmatrix} = \begin{pmatrix} z \\ \mu \end{pmatrix} - \begin{pmatrix} (A-\mu I) & -z \\ z^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} r \\ 0 \end{pmatrix}.$$

Compare the Jacobian matrix on the right-hand side with that of the correction equation (8.17). If we replace the matrix A by its approximation M in the Jacobian matrix, and write $u_{new} = z + v$, $\lambda_{new} = \lambda_{new} + \eta$, we would obtain the exact same solution (8.20) as that obtained from Olsen's method. The result (8.20) is therefore equivalent to using an approximate Newton step, whereby the matrix A in the Jacobian is replaced by the preconditioner M.

8.4.3 The Jacobi-Davidson Approach

Yet another way to characterize the solution to the system (8.19) is to express v as the unique solution of the form $v = (M - \mu I)^{-1}(\eta z - r)$ which is perpendicular to z, i.e., we need to

$$\text{Find} \quad v = (M - \mu I)^{-1}(-r + \eta z), \quad \text{such that} \quad z^H v = 0. \tag{8.24}$$

Indeed, the first equation of (8.17) yields the form of v, which is $v = (M - \mu I)^{-1}(\eta z - r)$ and the second stipulates that this solution must be orthogonal to z. Note that in the above equation, η is a parameter which is selected so that the orthogonality $z^H v = 0$ is satisfied, where v depends on η. The Jacobi-Davidson formulation developed by Fokkema et al [62] rewrites the solution using projectors.

Let P_z be a projector in the direction of z which leaves r invariant, i.e., such that

$$P_z z = 0; \quad P_z r = r.$$

It can be easily seen that any such projector is of the form

$$P_z = I - \frac{z s^H}{s^H z}, \tag{8.25}$$

where s is a vector that is orthogonal to r. The η-parameterized equation in (8.24) yields the relation $(M - \mu I)v = -r + \eta z$. Multiplying by P_z simplifies this relation to

$$P_z(M - \mu I)v = -r. \tag{8.26}$$

This can now be viewed as a singular system of equations which has infinitely many solutions. Indeed, it is a consistent system since it has the particular solution $-(M - \mu I)^{-1} r$ due to the relation $P_z r = r$. In fact one may ask what are all the solutions to the above system?

Any vector v satisfying the relation

$$(M - \mu I)v = -r + \eta z \quad \text{for} \quad \eta \in \mathbb{C} \tag{8.27}$$

is solution to the system (8.26) as can be readily verified by multiplying (8.27) by P_z and recalling that $P_z r = r$ and $P_z z = 0$. Conversely, for any solution v to (8.26) the vector $t = (M - \mu I)v$ is such that $P_z t = -r$. Therefore, the expression (8.25) of P_z implies $(I - \frac{z s^H}{s^H z})t = -r$ showing that $t = -r + \eta z$, with $\eta = s^H t / s^H z$. Hence, this vector v is a member of the solution set described by (8.27).

In other words *systems (8.26) and (8.27) are mathematically equivalent* in that they have the sets of solutions. The solutions of (8.27) are clearly the vectors v_η given by

$$v_\eta = -(M - \mu I)^{-1} r + \eta (M - \mu I)^{-1} z, \quad \eta \in \mathbb{C}. \tag{8.28}$$

So far we have ignored the constraint (8.21). Let P_w any projector in the direction of w, so that $P_w w = 0$. The constraint that v is orthogonal to w can be

expressed by the relation $P_w v = v$. So to obtain the desired solution from (8.26) we only need to replace this system by

$$[P_z(M - \mu I)P_w]v = -r \tag{8.29}$$

with the understanding that the solution is $P_w v$ instead of v.

We have therefore rewritten the system completely using projectors. The result is summarized in the following proposition which states that the solutions of the system (8.29) and that given by (8.23) are identical.

Proposition 8.2 *All the solutions of the (singular) system (8.26) are given by (8.28). The unique solution among these that is orthogonal to w is given by (8.23) and it is the unique solution $v = P_w v$ of (8.29).*

In orthogonal projection methods (e.g. Arnoldi) we have $r \perp z$, so we can take $P_z = I - zz^H$ assuming $\|z\|_2 = 1$. As for P_w, if we use to the common constraint (8.18) instead of the more general constraint (8.21) it is natural to take again $P_w = I - zz^H$. With these assumptions, the Jacobi-Davidson correction using a single vector z, consists in finding v such that :

$$(I - zz^H)(M - \mu I)(I - zz^H)v = -r \qquad v \perp z.$$

The main attraction of this viewpoint is that we can use iterative methods for the solution of the correction equation, i.e., exact solutions of systems with the matrix M are not explicitly required.

Block generalizations of the above scheme are straightforward. Instead of a vector z, we will use an orthogonal matrix Z, and the above system becomes

$$(I - ZZ^H)(M - \mu I)(I - ZZ^H)v = -r.$$

An intrepretation of the above equation is that we need to solve the correction in a reduced subspace, namely one that is orthogonal to the span of Z. This will tend to maximize 'new' information injected to the approximation.

8.5 The CMS – AMLS connection

A method for computing eigenvalues of partitioned matrices was introduced in structural dynamics by [32, 92] and was later extended [8] to a method known as the Algebraic Multi-Level Substructuring (AMLS). The method takes its root from domain decomposition ideas and it can be recast in the framework of the correction equations seen earlier.

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric real matrix, partitioned as

$$A = \begin{pmatrix} B & E \\ E^T & C \end{pmatrix}, \tag{8.30}$$

where $B \in \mathbb{R}^{(n-p)\times(n-p)}$, $C \in \mathbb{R}^{p \times p}$ and $E \in \mathbb{R}^{(n-p)\times p}$. The underlying context here is that the above matrix arises from the discretization of a certain

operator (e.g., a Laplacean) on a domain Ω which is then partitioned into several subdomains. An illustration is shown in Figure 8.1 for the simplest case of two subdomains. The subdomains, which may overlap, are separated by an interface Γ. The unknowns in the interior of each subdomain Ω_i are completely decoupled from the unknowns in all other subdomains. Coupling among subdomains is through the unknowns of the interface Γ and the unknowns in each Ω_i that are adjacent to Γ. With the situation just described, the matrix B is a block diagonal matrix. Each diagonal block will represent the unknowns which are interior for each domain. The C block correspond to all the interface variables.

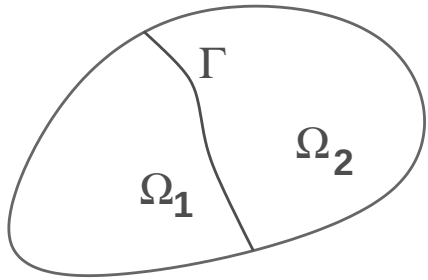


Figure 8.1: The simple case of two subdomains Ω_1, Ω_2 and an interface Γ (source of figure: [5]).

The eigenvalue problem $Au = \lambda u$, can be written

$$\begin{pmatrix} B & E \\ E^T & C \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \lambda \begin{pmatrix} u \\ y \end{pmatrix}, \tag{8.31}$$

where $u \in \mathbb{C}^{n-p}$ and $y \in \mathbb{C}^p$. The method of Component Mode Synthesis (`CMS`), was introduced in the 1960s in structural dynamics for computing eigenvalues of matrices partitioned in this form, see [32, 92]. The first step of the method is to solve the problem $Bv = \mu v$. This amounts to solving each of the decoupled smaller eigenvalue problems corresponding to each subdomain Ω_i separately. The method then injects additional vectors to account for the coupling among subdomains. With the local eigenvectors and the newly injected eigenvectors, a Rayleigh-Ritz projection procedure is then performed. We now consider these steps in detail.

Consider the matrix

$$U = \begin{pmatrix} I & -B^{-1}E \\ 0 & I \end{pmatrix}. \tag{8.32}$$

This *block Gaussian eliminator* for matrix (8.30) is selected so that

$$U^T A U = \begin{pmatrix} B & 0 \\ 0 & S \end{pmatrix},$$

where S is the Schur complement

$$S = C - E^T B^{-1} E. \tag{8.33}$$

The original problem (8.31) is equivalent to the generalized eigenvalue problem $U^T A U u = \lambda U^T U u$, which becomes

$$\begin{pmatrix} B & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix} = \lambda \begin{pmatrix} I & -B^{-1}E \\ -E^T B^{-1} & M_S \end{pmatrix} \begin{pmatrix} u \\ y \end{pmatrix}, \tag{8.34}$$

where $M_S = I + E^T B^{-2} E$.

The next step of CMS is to neglect the coupling matrices (blocks in positions (1,2) and (2,1)) in the right-hand side matrix of (8.34). This yields the uncoupled problem

$$Bz = \mu\, z \tag{8.35}$$

$$Sw = \eta\, M_S w. \tag{8.36}$$

Once the wanted eigenpairs have been obtained from (8.35–8.36), they are utilized in a projection method (Rayleigh-Ritz) applied to the original problem (8.34). The basis used for this projection is of the form

$$\left\{ \hat{z}_i = \begin{pmatrix} z_i \\ 0 \end{pmatrix} \quad i = 1, \ldots, m_B; \qquad \hat{w}_j = \begin{pmatrix} 0 \\ w_j \end{pmatrix} \quad j = 1, \ldots, m_S \right\}, \tag{8.37}$$

where $m_B < (n-p)$ and $m_S < p$. It is important to note that the projection is applied to (8.34) rather than to the original problem (8.31). There is an inherent change of basis between the two and, for reasons that will become clear shortly, the basis $\{\hat{z}_i\}_i, \{\hat{w}_j\}_j$, is well suited for the transformed problem rather than the original one.

We now consider this point in detail. We could also think of using the transformed basis

$$\left\{ \hat{z}_i = \begin{pmatrix} z_i \\ 0 \end{pmatrix} \; i = 1, \ldots, m_B; \quad \hat{u}_j = \begin{pmatrix} -EB^{-1} w_j \\ w_j \end{pmatrix} \; j = 1, \ldots, m_S \right\}, \tag{8.38}$$

for solving the original problem (8.31) instead of basis (8.37). As can be easily seen, these two options are mathematically equivalent.

Lemma 8.1 *The Rayleigh-Ritz process using the basis (8.37) for problem (8.34) is mathematically equivalent to the Rayleigh-Ritz process using the basis (8.38) for problem (8.31).*

Proof. For a given matrix A, and a given basis (not necessarily orthogonal) consisting of the columns of a certain matrix Z, the Rayleigh Ritz process can be written as

$$Z^T A Z \underline{v} = \underline{\lambda} Z^T Z \underline{v}$$

If Z is the basis (8.37) then the basis (8.38) is nothing but UZ. Comparing the two projection processes gives the result. □

8.5.1 AMLS and the Correction Equation

It is possible to view AMLS from the angle of the correction equation. First, AMLS *corrects* the eigenvectors of B in an attempt to obtain better approximations to the eigenvectors of the whole matrix A. This is done by exploiting the Schur complement matrix and constructing a good (improved) basis to perform a Rayleigh Ritz projection.

Consider eigenvectors of B associated with a few of its smallest eigenvalues $\mu_i, i = 1, \cdots, m_B$:

$$Bz_i = \mu_i z_i \ .$$

One can consider the expanded version of z_i into the whole space,

$$\hat{z}_i = \begin{pmatrix} z_i \\ 0 \end{pmatrix}$$

as approximate eigenvectors of A. The eigenvectors obtained in this manner amount to neglecting all the couplings and are likely to yield very crude approximations. It is possible to improve this approximation via a *correction equation* as is was done for the Davidson or Jacobi-Davidson approach, or in an inverse iteration approach.

An interesting observation is that the residuals $r_i = (A - \mu I)\hat{z}_i$ have components only on the interface variables, i.e., they have the shape:

$$r_i = \begin{pmatrix} 0 \\ E^T z_i \end{pmatrix}. \tag{8.39}$$

where the partitioning corresponds to the one above.

Consider now a Davidson-type correction in which the preconditioner is the matrix $(A - \mu I)$. We are to solve the equation $(A - \mu I)u_i = r_i$ where r_i is given by (8.39) so that the system to solve is

$$(A - \mu I)u_i = \begin{pmatrix} 0 \\ E^T z_i \end{pmatrix} \tag{8.40}$$

The matrix $A - \mu I$ can be block-factored as

$$(A - \mu I) = \begin{pmatrix} I & 0 \\ E^T(B - \mu I)^{-1} & I \end{pmatrix} \begin{pmatrix} B - \mu I & E \\ 0 & S(\mu) \end{pmatrix} \tag{8.41}$$

where $S(\mu)$ is the Schur complement

$$S(\mu) = C - \mu I - E^T(B - \mu I)^{-1}E \ .$$

The particular form of the right-hand side (8.40) leads to a solution that simplifies considerably. We find that

$$u_i = \begin{pmatrix} -(B - \mu I)^{-1}Ew_i \\ w_i \end{pmatrix} \quad \text{with} \quad w_i = S(\mu)^{-1}E^T z_i. \tag{8.42}$$

There is a striking similarity between the result of this correction when the shift $\mu = 0$ is used, and the basis vectors used in the projection process in AMLS. The basis (8.38) used in the AMLS correction consists of the vectors

$$\begin{pmatrix} z_i \\ 0 \end{pmatrix} \quad \begin{pmatrix} -B^{-1}Ew_j \\ w_j \end{pmatrix} \tag{8.43}$$

where the w_i's are (generalized) eigenvectors of the Schur complement problem (8.36), and the z_i's are eigenvectors of the B block. In contrast, Jacobi-Davidson computes w_i's from a form of inverse iteration applied to $S(\mu)$. Notice that vectors to which the inverse of $S(\mu)$ is applied to vectors in the range of E^T.

Next consider a correction obtained from a single vector inverse iteration. In this case, for each approximate eigenvector z_i we would seek a new approximation x_i by solving an equation of the type $(A - \mu I)x_i = \hat{z}_i$, where μ is a certain shift, often a constant one during the inverse iteration process to reduce the cost of the factorization.

Let us define

$$t_i = (B - \mu I)^{-1} z_i \tag{8.44}$$

$$w_i = -S(\mu)^{-1} E^T t_i. \tag{8.45}$$

Then, taking the factorization (8.41) and the particular structure of $\hat{z}_i$ into account we find that the solution x_i the system $(A - \mu I)x_i = \hat{z}_i$ is

$$x_i = \begin{pmatrix} t_i - (B - \mu I)^{-1} E w_i \\ w_i \end{pmatrix}. \tag{8.46}$$

Note that the w_i's are again different from those of AMLS or the Davidson approach. If the z_i are eigenvectors of the B block then this basis would be equivalent to the one where each z_i is replaced by $B^{-1}z_i = \mu_i^{-1} z_i$.

8.5.2 Spectral Schur Complements

The following equations result from (8.34)

$$Bu = \lambda(u - B^{-1}Ey), \tag{8.47}$$

$$Sy = \lambda(-E^T B^{-1} u + M_S y). \tag{8.48}$$

It is easyly shown that when E is of full rank and B is nonsingular, then (λ, u) cannot be an eigenpair of B. This is because if (λ, u) is an eigenpair for B, then we would have $B^{-1}Ey = 0$ and since E if of full rank, then y would be zero. However, since y and u cannot be both equal to zero, $E^T B^{-1} u \neq 0$ and (8.48) would imply that $\lambda = 0$ which contradict the assumption that B is nonsingular. The result is that any pair (λ, u), where (λ, u, y) is a solution of (8.34), cannot be an eigenpair for B.

A consequence of this is that when λ is an eigenvalue of B, then equation (8.47) always has a solution in the orthogonal of the eigenspace of B associated

with this eigenvalue. We can therefore always solve the system $(B - \lambda I)u = -B^{-1}y$ derived from (8.47) provided the inverse of B is interpreted as a pseudo-inverse. In what follows, $(B - \lambda I)^{-1}$ will mean the pseudo-inverse $(B - \lambda I)^{\dagger}$ when λ is an eigenvalue of B. The notation is not changed because this is a situation of little practical interest in general as it occurs rarely.

We can substitute (8.47) into (8.48) to obtain

$$Sy = \lambda \left(\lambda E^T B^{-1} (B - \lambda I)^{-1} B^{-1} E y + M_s y \right),$$

which results in the equivalent nonlinear eigenvalue problem

$$\left[S - \lambda \left(E^T B^{-2} E \right) - \lambda^2 E^T B^{-1} (B - \lambda I)^{-1} B^{-1} E \right] \; y \; = \; \lambda y. \tag{8.49}$$

Rewriting the above problem in an expanded notation we obtain the following nonlinear eigenvalue problem

$$\left[C - E^T B^{-1} (B + \lambda I + \lambda^2 (B - \lambda I)^{-1}) B^{-1} E \right] y = \lambda y. \tag{8.50}$$

We can show that the above problem is equivalent to a nonlinear eigenvalue problem involving the spectral Schur complement

$$S(\lambda) = C - E^T (B - \lambda I)^{-1} E. \tag{8.51}$$

The first resolvent equality (3.15) seen in Chapter 3, yields

$$(B - \lambda I)^{-1} - B^{-1} = \lambda (B - \lambda I)^{-1} B^{-1}.$$

Substitute the above relation to transform the term $\lambda^2 (B - \lambda I)^{-1}) B^{-1}$ in the expression of the left hand matrix in (8.50) which we denote by $\hat{S}(\lambda)$:

$$\begin{aligned}
\hat{S}(\lambda) &= C - E^T B^{-1} \left(I + \lambda B^{-1} + \lambda (B - \lambda I)^{-1} - \lambda B^{-1} \right) E \\
&= C - E^T \left(B^{-1} + \lambda B^{-1} (B - \lambda I)^{-1} \right) E \\
&= C - E^T \left(B^{-1} - B^{-1} + (B - \lambda I)^{-1} \right) E \\
&= C - E^T (B - \lambda I)^{-1} E \\
&= S(\lambda).
\end{aligned}$$

In fact, the Schur complement S can be viewed as the first term of the Taylor series expansion of $S(\lambda)$ with respect to λ around $\lambda = 0$. The standard expansion of the resolvent, see, equation (3.14) in Chapter 3, which in our situation can be written as

$$(B - \lambda I)^{-1} = B^{-1} \sum_{k=0}^{\infty} (\lambda B^{-1})^k = \sum_{k=0}^{\infty} \lambda^k B^{-k-1}, \tag{8.52}$$

leads to the following series expansion for $S(\lambda)$

$$\begin{aligned}
S(\lambda) &= C - E^T \sum_{k=0}^{\infty} (\lambda^k B^{-k-1}) E \\
&= C - E^T \left(B^{-1} + \lambda B^{-2} + \lambda^2 B^{-3} + \cdots \right) E. \qquad (8.53)
\end{aligned}$$

In AMLS the second part of the projection basis Z (see (8.37)) consists of eigenvectors associated with the smallest eigenvalues of the generalized eigenproblem $Sw = \lambda M_S w$ which translates to

$$[C - E^T B^{-1} E]w = \lambda(I + E^T B^{-2} E)w$$

or equivalently,

$$\left(C - E^T(B^{-1} + \lambda B^{-2})E\right) w = \lambda w. \tag{8.54}$$

In light of relation (8.53), the above eigenproblem can clearly be considered as a truncated version of the original nonlinear problem (8.50), where the terms $\lambda^k B^{-k-1}$ for $k \geq 2$ in the expansion of the resolvent $(B - \lambda I)^{-1}$ are dropped. Thus, the eigenvector w can be seen as a direct approximation to the bottom part y of the exact eigenvector of (8.34).

The above observations immediately lead to some possible suggestions on how to improve the approximation by including additional terms of the infinite expansion. We can for example devise a second order approximation to (8.50) obtained by adding the term $\lambda^2 B^{-3}$, see [5] for details.

Next we analyze how AMLS expands the approximation of the lower part y to an approximation $[u; y]^T$ of an eigenvector of the complete problem (8.34).

8.5.3 The Projection Viewpoint

Consider again the nonlinear Schur complement (8.51). The eigenvalues of the original problem, which do not belong to the spectrum of B, can be obtained from those of the nonlinear eigenvalue problem

$$S(\lambda)x = \lambda x.$$

Proposition 8.3 *Let λ, y be an eigenpair of the nonlinear eigenvalue problem*

$$S(\lambda)y = \lambda y$$

where $S(\lambda)$ is defined by (8.51). Then, λ is an eigenvalue of (8.31) with associated eigenvector:

$$\begin{pmatrix} -(B - \lambda I)^{-1} E y \\ y \end{pmatrix} \tag{8.55}$$

Proof. The proof consists of a simple verification. □

Now assume that we have a good approximation to the nonlinear Schur complement problem, i.e., to a solution λ and y of the nonlinear problem (8.49). It is clear that the best we can do to retrieve the corresponding eigenvector of (8.31) is to use substitution, i.e., to compute the top part of (8.55):

$$u = -(B - \lambda I)^{-1} E y, \tag{8.56}$$

which will give us the top part of the exact eigenvector. This entails factoring the matrix $(B - \lambda I)$ for each different eigenvalue λ, which is not practical. As was seen in the previous section, AMLS extracts an approximation to the nonlinear problem (8.49) by solving the generalized eigenvalue problem (8.54) and then it replaces the substitution step (8.56) by a projection step. Specifically, once an approximate pair λ, y is obtained, AMLS computes approximate eigenvectors to the original problem by a projection process using the space spanned by the family of vectors:

$$\left\{ \begin{pmatrix} v_i^B \\ 0 \end{pmatrix} \right\}, \qquad \left\{ \begin{pmatrix} -B^{-1}Ey_j \\ y_j \end{pmatrix} \right\}, \tag{8.57}$$

in which v_i^B are eigenvectors of B associated with its smallest eigenvalues. Note that these two sets of vectors are of the form $U \begin{pmatrix} v_i^B \\ 0 \end{pmatrix}$, for the first, and $U \begin{pmatrix} 0 \\ y_j \end{pmatrix}$ for the second, where U was defined earlier by equation (8.32). The question is: why is this a good way to replace the substitution step (8.56)? Another issue is the quality we might expect from this process.

Ideally, the prolongation matrix U should be replaced by one which depends on the eigenvalue λ, namely

$$U(\lambda) = \begin{pmatrix} I & -(B - \lambda I)^{-1}E \\ 0 & I \end{pmatrix}.$$

Indeed, if we were to use the prolongator $U(\lambda)$ instead of U, then $U(\lambda) \begin{pmatrix} 0 \\ y_j \end{pmatrix}$ would be an exact eigenvector (if the approximation to y that we use were exact).

It is not appealing in practice to use a different prolongator $U(\lambda)$ for each different eigenvalue λ. What is interesting and important to note is that $U(\lambda)$ and U are likely to be close to each other for small (in modulus) eigenvalues λ. Furthermore, *the difference between the two consists mostly of components related to eigenvectors of B which are associated with eigenvalues close to λ*. It is helpful to examine this difference:

$$\begin{aligned} [U(\lambda) - U] \begin{pmatrix} 0 \\ y \end{pmatrix} &= \begin{pmatrix} 0 & -((B - \lambda I)^{-1} - B^{-1})E \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ y \end{pmatrix} \\ &= \begin{pmatrix} -\lambda\,(B - \lambda I)^{-1}B^{-1}Ey \\ 0 \end{pmatrix}. \end{aligned}$$

In order to compensate for this difference, it is natural to add to the subspace eigenvectors of B in which this difference will have large components, i.e., eigenvectors corresponding to the smallest eigenvalues of B. This is at the heart of the approximation exploited by AMLS which incorporates the first set in (8.57).

PROBLEMS

P-8.1 Consider a more general framework for Theorem 8.1, in which one is seeking l eigenvalues at a time. The new vectors are defined as

$$t_{i,j} = M_{i,j}^{-1} r_{i,j} \quad i = 1, 2, \ldots l.$$

where i refers to the eigenvalue number and j is the step number. As a result the dimension of the Davidson subspace increases by l at every step instead of one. The assumptions on each of the individual preconditioners for each eigenvalue are the same as in Theorem 8.1.
(1) Establish that the first part of Theorem 8.1 is still valid.
(2) To show the second part, define $z_{ij} = (I - \mathcal{P}_j)M_{i,j}r_{ij}$ and similarly $w_{ij} = z_{ij}/\|z_{ij}\|_2$ and

$$W_j = [u_1^{(j)}, u_2^{(j)}, \ldots, u_i^{(j)}, w_{ij}].$$

Show that W_j is orthonormal and that the matrix $B_{i,j} = W_j^H A W_j$ has the form,

$$B_{i,j} = \begin{pmatrix} \lambda_1^{(j)} & & & \alpha_{1j} \\ & \ddots & & \vdots \\ & & \lambda_i^{(j)} & \alpha_{ij} \\ \alpha_{1j} & \cdots & \alpha_{ij} & \beta_j \end{pmatrix} \tag{8.58}$$

in which we set $\alpha_{kj} = w_{ij}^H A u_k^{(j)})$ and $\beta_j = w_{ij}^H A w_{ij}$.
(3) Show that

$$\lambda_k^{(j)} \leq \mu_k^{(j)} \leq \lambda_k^{(j+1)} \quad k = 1, 2, \ldots, i.$$

(4) Taking the Frobenius norm of $B_{i,j}$ and using the fact that

$$\mathrm{tr}(B_{i,j}) = \sum_{k=1}^{i+1} \mu_k^{(j)} = \beta_j + \sum_{k=1}^{i} \lambda_k^{(j)}$$

show that

$$\begin{aligned} 2\sum_{k=1}^{i} \alpha_{kj}^2 &= \sum_{k=1}^{i} (\mu_k^{(j)} - \lambda_k^{(j)})(\mu_k^{(j)} + \lambda_k^{(j)} - \mu_{i+1}^{(j)} - \beta_j) \\ &\leq 4\|A\|_2 \sum_{k=1}^{i} (\mu_k^{(j)} - \lambda_k^{(j)}) \, . \end{aligned}$$

(5) Complete the proof of the result similarly to Theorem 8.1.

P-8.2 Using the result of Exercise P-6.3 write a simpler version of the shift-and-invert Arnoldi Algorithm with deflation, Algorithm 8.1, which does not require the $(k-1) \times (k-1)$ principal submatrix of H_m, i.e., the (quasi) upper triangular matrix representing of $(A - \sigma I)^{-1}$ in the computed invariant subspace.

P-8.3 How can one get the eigenvalues of A from those of B_+ or B_-. What happens if the approximate eigenvalues are close and complex? What alternative can you suggest for recovering approximate eigenvalues of A from a given projection process applied to either of these two real operators.

P-8.4 Establish the relation (8.9).

NOTES AND REFERENCES. The notion of preconditioning is well-known for linear systems but it is not clear who defined this notion first. In the survey paper by Golub and O'Leary [78] it is stated that "The term *preconditioning* is used by Turing (1948) and by then seems standard terminology for problem transforming in order to make solutions easier. The first application of the work to the

idea of improving the convergence of an iterative method may be by Evans (1968), and Evans (1973) and Axelsson (1974) apply it to the conjugate gradient algorithm". However, the idea of polynomial preconditioning is clearly described in a 1952 paper by Lanczos [113], although Lanczos does not use the term "preconditioning" explicitly. The idea was suggested later for eigenvalue calculations by Stiefel who employed least-squares polynomials [207] and Rutishauser [166] who combined the QD algorithm with Chebyshev acceleration. The section on Shift-and-Invert preconditioning is adapted from [152]. Davidson's method as described in [41] can be viewed as a cheap version of Shift-and-Invert , in which the solution of the linear systems are solved (very) inaccurately. The method is well-known to the physicists or quantum chemists but not as well known to numerical analysts. The lack of theory of the method might have been one reason for the neglect. Generalizations and extensions of the method are proposed by Morgan and Scott [132] in the Hermitian case but little has been done in the non-Hermitian case.

The Jacobi-Davidson enhancement was developed in the mid 1990s [62] though simplified forms of the method existed already, since the method represents in effect a Newton type approach. The viewpoint of projections and the insight provided by the ensuing articles gave an important impetus to this approach. A few packages have been written based on the Jacobi-Davidson approach. For example, JADAMILU [1], developed by Matthias Bollhöfer and Yvan Notay [14] is a package written in fortran-77 and exploits incomplete LU factorizations and iterative solvers.

The AMLS approach was developed mainly as a powerful replacement to the shift-and-invert Block-Lanczos algorithm which was used in certain structural engineering problems [8]. It is in essence a form of shift-and-invert approach based on the shift $\sigma = 0$ and the use of domain-decomposition concepts for factoring the matrix. Because it uses a single shift, and it is a single-shot method, its accuracy tends to be limited for the eigenvalues that are far from zero. Some of the material on the new sections 8.4 and 8.5 is adapted from [158], and [5]. ■

[1] `http://homepages.ulb.ac.be/~jadamilu/`

Chapter 9

NON-STANDARD EIGENVALUE PROBLEMS

Many problems arising in applications are not of the standard form $Ax = \lambda x$ but of the 'generalized' form $Ax = \lambda Bx$. In structural engineering, the A matrix is called the stiffness matrix and B is the mass matrix. In this situation, both are symmetric real and often B is positive definite. Other problems are quadratic in nature, i.e., they take the form

$$\lambda^2 Ax + \lambda Bx + Cx = 0.$$

This chapter gives a brief overview of these problems and of some specific techniques that are used to solve them. In many cases, we will seek to convert a nonstandard problems into a standard one in order to be able to exploit the methods and tools of the previous chapters.

9.1 Introduction

Many eigenvalue problems arising in applications are either generalized, i.e., of the form

$$Ax = \lambda Bx \tag{9.1}$$

or quadratic,

$$\lambda^2 Ax + \lambda Bx + Cx = 0.$$

Such problems can often be reduced to the standard form $Ax = \lambda x$ under a few mild assumptions. For example when B is nonsingular, then (9.1) can be rewritten as

$$B^{-1}Ax = \lambda x\ . \tag{9.2}$$

As will be explained later, the matrix $C = B^{-1}A$ need not be computed explicitly in order to solve the problem. Similarly, the quadratic eigen-problem can be transformed into a generalized eigen-problem of size $2n$, in a number of different ways.

Thus, it might appear that these nonstandard problems may be regarded as particular cases of the standard problems and that no further discussion is warranted. This is not the case. First, a number of special strategies and techniques must be applied to improve efficiency. For example, when A is symmetric and B

is symmetric positive definite then an alternative transformation of (9.1) will lead to a Hermitian problem. Second, there are some specific issues that arise, such as the situation where both A and B are singular matrices, which have no equivalent in the standard eigenvalue context.

9.2 Generalized Eigenvalue Problems

In this section we will summarize some of the results known for the generalized eigenvalue problem and describe ways of transforming it into standard form. We will then see how to adapt some of the techniques seen in previous chapters.

9.2.1 General Results

The pair of matrices A, B in the problem (9.1) is often referred to as a *matrix pencil*. We will more often use the term *matrix pair* than vmatrix pencil. If there is no particular reason why one of the two matrices A and B should play a special role, then the most natural way of defining eigenvalues of a matrix pair is to think of them as pairs (α, β) of complex numbers. Thus, (α, β) is an eigenvalue of the pair (A, B) if by definition there is a vector u, called an associated eigenvector, such that

$$\beta A u = \alpha B u. \tag{9.3}$$

Equivalently, (α, β) is an eigenvalue if and only if

$$\det(\beta A - \alpha B) = 0 \, .$$

When (α, β) is an eigenvalue pair for (A, B), then $(\bar{\alpha}, \bar{\beta})$ is an eigenvalue pair for (A^H, B^H) since $\det(\beta A - \alpha B)^H = 0$. The left eigenvector for A, B is defined as a vector for which

$$(\beta A - \alpha B)^H w = 0. \tag{9.4}$$

This extension of the notion of eigenvalue is not without a few drawbacks. First, we note that the trivial pair $(0, 0)$ always satisfies the definition. Another difficulty is that there are infinitely many pairs (α, β) which can be termed 'generalized eigenvalues' to represent the same 'standard eigenvalue'. This is because we can multiply a given (α, β) by any complex scalar and still get an eigenvalue for the pair. Thus, the standard definition of an eigenvalue corresponds to the case where $B = I$ and $\beta = 1$. There are three known ways out of the difficulty. A popular way is to take the ratio α/β as an eigenvalue, which corresponds to selecting the particular pair $(\alpha, 1)$ in the set. When β is zero, the eigenvalue takes the value infinity and this may not be satisfactory from the numerical point of view. A second way would be to use pairs (α, β) but scale them by some norm in $\mathbb{C}^2$, e.g., so that $|\alpha|^2 + |\beta|^2 = 1$. Finally, a third way, adopted by Stewart and Sun [205] is to denote by $\langle \alpha, \beta \rangle$ the set of all pairs that satisfy (9.3). The eigenvalue is then a set instead of an element in $\mathbb{C}^2$. We will refer to this set as a *generalized eigenvalue*. However, we will sacrifice a little of rigor for convenience, and also

call any representative element (α, β), in the set, at the exclusion of $(0, 0)$, an *eigenvalue pair*. Note the distinction between the notation of an eigenvalue pair $(.,.)$ and the set to which it belongs to, i.e., the generalized eigenvalue, denoted by $\langle .,. \rangle$. This definition is certainly radically different from, and admittedly more complicated than, the usual definition, which corresponds to arbitrarily selecting the pair corresponding to $\beta = 1$. On the other hand it is more general. In particular, the pair $\langle 1, 0 \rangle$ is well defined whereas with the usual definition it becomes an infinite eigenvalue.

To illustrate the various situations that can arise we consider two by two matrices in the following examples.

Example 9.1. Let

$$A = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

By the definition (α, β) is an eigenvalue if $\det(\beta A - \alpha B) = 0$ which gives the set of pairs (α, β) satisfying the relation $\beta = \pm i\alpha$. In other words, the two generalized eigenvalues are $\langle 1, i \rangle$ and $\langle 1, -i \rangle$. This example underscores the fact that the eigenvalues of a symmetric real (or Hermitian complex) pair are not necessarily real. □

Example 9.2. Let

$$A = \begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

Here $\det(\beta A - \alpha B) = \alpha\beta$, so the definition shows that $\langle 0, 1 \rangle$ and $\langle 1, 0 \rangle$ are generalized eigenvalues. Note that both matrices are singular. □

Example 9.3. Let

$$A = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

In this case any pair $\langle \alpha, \beta \rangle$ is an eigenvalue since $\det(\beta A - \alpha B) = 0$ independently of the two scalars α and β. Note that this will occur whenever the two matrices are singular and have a common null space. Any vector of the null-space can then be viewed as a degenerate eigenvector associated with an arbitrary scalar. Such pairs are said to be singular. □

Example 9.4. Let

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 2 \\ 0 & 2 \end{pmatrix}.$$

This is another example where any pair (α, β) is an eigenvalue since $\det(\beta A - \alpha B) = 0$ independently of α and β. The two matrices are again singular but here

their two null spaces do not intersect. Any 'eigenvalue' (α, β) has the associated 'eigenvector' $(2\alpha, -\beta)^H$. □

The above examples suggests an important case that may cause difficulties numerically. This is the case of 'singular pairs'.

Definition 9.1 *A matrix pair (A, B) is called singular if $\beta A - \alpha B$ is singular for all α, β. A matrix pair that is not singular is said to be regular .*

The added complexity due for example to one (or both) of the matrices being singular means that special care must be exercised when dealing with generalized eigen-problems. However, the fact that one or both of the matrices A or B is singular does not mean that trouble is lurking. In fact generalized eigenvalue problem can be quite well behaved in those situations, if handled properly.

We now state a number of definitions and properties. If we multiply both components A and B of the pair (A, B) to the left by the same nonsingular matrix Y then the eigenvalues and right eigenvectors are preserved. Similarly, if we multiply them to the right by the same non-singular matrix X then the eigenvalues and the left eigenvectors are preserved. The left eigenvectors are multiplied by Y^{-H} in the first case and the right eigenvectors are multiplied by X^{-1} in the second case. These transformations generalize the similarity transformations of the standard problems.

Definition 9.2 *If X and Y are two nonsingular matrices, the pair*

$$(YAX, YBX)$$

is said to be equivalent to the pair (A, B).

We will now mention a few properties. Recall that if (α, β) is an eigenvalue pair for (A, B), then $(\bar{\alpha}, \bar{\beta})$ is an eigenvalue pair for (A^H, B^H). The corresponding eigenvector is called the left eigenvector of the pair (A, B).

A rather trivial property, which may have some nontrivial consequences, is that the *eigenvectors* of (A, B) are the same as those of (B, A). A corresponding eigenvalue pair (α, β) is simply permuted to (β, α).

In the standard case we know that a left and a right eigenvector associated with two distinct eigenvalues are orthogonal. We will now show a similar property for the generalized problem.

Proposition 9.1 *Let $\lambda_i = \langle \alpha_i, \beta_i \rangle$ and $\lambda_j = \langle \alpha_j, \beta_j \rangle$ two distinct generalized eigenvalues of the pair (A, B) and let u_i be a right eigenvector associated with λ_i and w_j a left eigenvector associated with λ_j. Then,*

$$(Au_i, w_j) = (Bu_i, w_j) = 0. \tag{9.5}$$

Proof. Writing the definition for λ_i yields,

$$\beta_i A u_i - \alpha_i B u_i = 0.$$

Therefore,

$$0 = (\beta_i A u_i - \alpha_i B u_i, w_j) = (u_i, (\bar{\beta}_i A^H - \bar{\alpha}_i B^H) w_j) \ . \tag{9.6}$$

We can multiply both sides of the above equation by β_j and use the fact that $(\bar{\alpha}_j, \bar{\beta}_j)$ is an eigenvalue for A^H, B^H with associated eigenvector w_j to get,

$$\begin{aligned} 0 &= (u_i, \bar{\beta}_i \bar{\beta}_j A^H w_j - \bar{\alpha}_i \bar{\beta}_j B^H w_j) \\ 0 &= (u_i, (\bar{\beta}_i \bar{\alpha}_j - \bar{\alpha}_i \bar{\beta}_j) B^H w_j) \\ 0 &= (\beta_i \alpha_j - \alpha_i \beta_j)(B u_i, w_j). \end{aligned}$$

This implies that $(Bu_i, w_j) = 0$ because

$$\beta_i \alpha_j - \alpha_i \beta_j = \begin{vmatrix} \beta_i & \beta_j \\ \alpha_i & \alpha_j \end{vmatrix}$$

must be nonzero by the assumption that the two eigenvalues are distinct. Finally, to show that $(Au_i, w_j) = 0$ we can redo the proof, this time multiplying both sides of (9.6) by α_j instead of β_j, or we can simply observe that we can interchange the roles of A and B, and use the fact that (A, B) and (B, A) have the same set of eigenvectors. □

The proposition suggests that when all eigenvalues are distinct, we may be able to simultaneously diagonalize A and B. In fact if all eigenvalues are distinct then the proposition translates into

$$W^H A U = D_A, \quad W^H B U = D_B$$

in which D_A and D_B are two diagonals, U is the matrix of the right eigenvectors and W the matrix of left eigenvectors (corresponding to eigenvalues listed in the same order as for U). There are two points that are still unclear. The first is that we do not know how many distinct eigenvalues there can be. We would like to show that when the pair is regular then there are n of them so that the matrices U and W in the above equality are $n \times n$ matrices. The second point is that we do not know yet whether or not the eigenvectors associated with these distinct eigenvalues are linearly independent. When either A or B are nonsingular then the eigenvectors associated with distinct eigenvectors are linearly independent. This can be seen by observing that the eigenvectors of the pair (A, B) are the same as those of $(B^{-1}A, I)$ in case B is nonsingular or $(I, A^{-1}B)$ when A is nonsingular. As it turns out this extends to the case when the pair is regular. When the pair (A, B) is a regular pair, then there are two scalars σ_*, τ_* such that the matrix $\tau_* A - \sigma_* B$ is nonsingular. We would like to construct *linearly transformed pairs* that have the same eigenvectors as (A, B) and such that one of the two matrices in the pair is nonsingular. The following theorem will help establish the desired result.

Theorem 9.1 *Let (A, B) any matrix pair and consider the transformed pair (A_1, B_1) defined by*

$$A_1 = \tau_1 A - \sigma_1 B \ , \quad B_1 = \tau_2 B - \sigma_2 A \ , \tag{9.7}$$

for any four scalars $\tau_1, \tau_2, \sigma_1, \sigma_2$ such that the 2×2 matrix

$$\Omega = \begin{pmatrix} \tau_2 & \sigma_1 \\ \sigma_2 & \tau_1 \end{pmatrix}$$

is nonsingular. Then the pair (A_1, B_1) has the same eigenvectors as the pair (A, B). An associated eigenvalue $(\alpha^{(1)}, \beta^{(1)})$ of the transformed pair (A_1, B_1) is related to an eigenvalue pair (α, β) of the original pair (A, B) by

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \Omega \begin{pmatrix} \alpha^{(1)} \\ \beta^{(1)} \end{pmatrix} . \tag{9.8}$$

Proof. Writing that $(\alpha^{(1)}, \beta^{(1)})$ is an eigenvalue pair of (A_1, B_1) with associated eigenvector u we get

$$\beta^{(1)}(\tau_1 A - \sigma_1 B)u = \alpha^{(1)}(\tau_2 B - \sigma_2 A)u$$

which after grouping the Au and Bu terms together yields,

$$(\tau_1\beta^{(1)} + \sigma_2\alpha^{(1)})Au = (\tau_2\alpha^{(1)} + \sigma_1\beta^{(1)})Bu . \tag{9.9}$$

The above equation shows that u is an eigenvector for the original pair (A, B) associated with the eigenvalue (α, β) with

$$\beta = \tau_1\beta^{(1)} + \sigma_2\alpha^{(1)}, \quad \alpha = \tau_2\alpha^{(1)} + \sigma_1\beta^{(1)}. \tag{9.10}$$

Note that (α, β) is related by (9.8) to $(\alpha^{(1)}, \beta^{(1)})$ and as a result α and β cannot both vanish because of the nonsingularity of Ω. Conversely, to show that any eigenvector of (A, B) is an eigenvector of (A_1, B_1) we can show that A and B can be expressed by relations similar to those in (9.7) in terms of A_1 and B_1. This comes from the fact that Ω is nonsingular. n □

A result of the above theorem is that we can basically identify a regular problem with one for which one of the matrices in the pair is nonsingular. Thus, the choice $\sigma_1 = \sigma_*, \tau_1 = \tau_*$ and $\sigma_2 = \sigma_1, \tau_2 = -\tau_1$ makes the matrix A_1 nonsingular with a non-singular Ω transformation. In fact once τ_1, σ_1 are selected any choice of τ_2 and σ_2 that makes Ω nonsingular will be acceptable.

Another immediate consequence of the theorem is that when (A, B) is regular then there are n eigenvalues (counted with their multiplicities).

Corollary 9.1 *Assume that the pair (A, B) has n distinct eigenvalues. Then the matrices U and W of the n associated right and left eigenvectors respectively, are nonsingular and diagonalize the matrices A and B simultaneously, i.e., there are two diagonal matrices D_A, D_B such that,*

$$W^H AU = D_A , \quad W^H BU = D_B .$$

The equivalent of the Jordan canonical form is the Weierstrass-Kronecker form. In the following we denote by $\mathrm{diag}\ (X, Y)$ a block diagonal matrix with X in the (1,1) block and Y in the (2,2) block.

Theorem 9.2 *A regular matrix pair (A, B) is equivalent to a matrix pair of the form*

$$(\mathrm{diag}\ (J, I)\ ,\ \mathrm{diag}\ (I, N))\ , \tag{9.11}$$

in which the matrices are partitioned in the same manner, and where J and N are in Jordan canonical form and N is nilpotent.

The equivalent of the Schur canonical form would be to simultaneously reduce the two matrices A and B to upper triangular form. This is indeed possible and can be shown by a simple generalization of the proof of Schur's theorem seen in Chapter 1.

Theorem 9.3 *For any regular matrix pair (A, B) there are two unitary matrices Q_1 and Q_2 such that*

$$Q_1^H A Q_2 = R_A \quad \textit{and} \quad Q_1^H B Q_2 = R_B$$

are two upper triangular matrices.

9.2.2 Reduction to Standard Form

When one of the components of the pair (A, B) is nonsingular, there are simple ways to get a standard problem from a generalized one. For example, when B is nonsingular, we can transform the original system

$$\beta A u = \alpha B u$$

into

$$B^{-1} A u = \alpha u$$

taking $\beta = 1$. This simply amounts to multiplying both matrices in the pair by B^{-1}, thus transforming (A, B) into the equivalent pair $(B^{-1}A, I)$. Other transformations are also possible. For example, we can multiply on the right by B^{-1} transforming (A, B) into the equivalent pair (AB^{-1}, I). This leads to the problem

$$AB^{-1} y = \alpha y \quad \text{with} \quad u = B^{-1} y.$$

Similarly, when A is nonsingular, we can solve the problem

$$A^{-1} B u = \alpha u$$

setting $\beta = 1$ or, again using the variable $y = A^{-1}u$,

$$BA^{-1} y = \alpha y.$$

Note that all the above problems are non Hermitian in general. When A and B are both Hermitian and, in addition, B is positive definite, a better alternative may be to exploit the Cholesky factorization of B. If $B = LL^T$, we get after multiplying from the left by L^{-1} and from the right by L^{-T}, the standard problem

$$L^{-1}AL^{-T}y = \alpha y. \tag{9.12}$$

None of the above transformations can be used when both A and B are singular. In this particular situation, one can shift the matrices, i.e., use a transformation of the form described in theorem (9.1). If the pair is regular then there will be a matrix Ω that will achieve the appropriate transformation. In practice these transformations are not easy to perform since we need to verify whether or not a transformed matrix is singular. If a pair is regular but both A and B are singular, then chances are that a slight linear transformation will yield a pair with one or both of the matrices nonsingular. However, this is not easy to check in practice. First, there is the difficulty of determining whether or not a matrix is deemed nonsingular. Second, in case the two matrices have a nontrivial common null space, then this trial-and-error approach cannot work since any pair α, β will yield a singular $\beta A - \alpha B$, and this information will not be enough to assert that the pair (A, B) is singular.

The particular case where both components A and B are singular and their null spaces have a nontrivial intersection, i.e.,

$$\mathrm{Null}(A) \cap \mathrm{Null}(B) \;\neq\; \{0\}$$

deserves a few more words. This is a special singular problem. In practice, it may sometimes be desirable to 'remove' the singularity, and compute the eigenvalues associated with the restriction of the pair to the complement of the null space. This can be achieved provided we can compute a basis of the common null space, a task that is not an easy one for large sparse matrices, especially if the dimension of the null space is not small.

9.2.3 Deflation

For practical purposes, it is important to define deflation processes for the generalized eigenvalue problem. In particular we would like to see how we can extend, in the most general setting, the Wielandt deflation procedure seen in Chapter 4. Assuming we have computed an eigenvector u_1 associated with some eigenvalue $\lambda_1 = \langle \alpha, \beta \rangle$, of (A, B) the most general way of defining analogues of the deflated matrix A_1 of Chapter 4 is to deflate the matrices A and B as follows:

$$A_1 = A - \sigma_1 B u_1 v^H , \tag{9.13}$$

$$B_1 = B - \sigma_2 A u_1 v^H . \tag{9.14}$$

We assume, as in the standard case, that $v^H u_1 = 1$. We can easily verify that the eigenvector u_1 is still an eigenvector of the pair (A_1, B_1). The corresponding

eigenvalue pair (α', β') must satisfy

$$\beta' A_1 u_1 = \alpha' B_1 u_1$$

from which we get the relation

$$(\beta' + \sigma_2 \alpha') A u_1 = (\alpha' + \sigma_1 \beta') B u_1 \ .$$

Thus we can identify $\alpha' + \sigma_1 \beta'$ with α and $\beta' + \sigma_2 \alpha'$ with β, to get

$$\alpha = \alpha' + \sigma_1 \beta', \quad \beta = \beta' + \sigma_2 \alpha' \ . \tag{9.15}$$

Inverting the relations, we get

$$\alpha' = \frac{\alpha - \sigma_1 \beta}{1 - \sigma_1 \sigma_2}, \quad \beta' = \frac{\beta - \sigma_2 \alpha}{1 - \sigma_1 \sigma_2} \tag{9.16}$$

assuming that $1 - \sigma_1 \sigma_2 \neq 0$. The scaling by $1 - \sigma_1 \sigma_2$ can be ignored to obtain the simpler relations,

$$\alpha' = \alpha - \sigma_2 \beta \ , \quad \beta' = \beta - \sigma_1 \alpha \tag{9.17}$$

which can be rewritten as

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \begin{pmatrix} 1 & -\sigma_1 \\ -\sigma_2 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \ . \tag{9.18}$$

In the standard case we have $B = I$, $\beta = \beta' = 1$ and $\sigma_2 = 0$, so the standard eigenvalue is changed to $\alpha' = \alpha - \sigma_1$ as was seen in Chapter 4.

Using Proposition 9.1, we can show that the left eigenvectors not associated with λ_1 are preserved. The particular choice $v = Bw_1$, in which w_1 is the left eigenvector associated with the eigenvalue λ_1 preserves both left and right eigenvectors and is a generalization of Hotelling's deflation, see Exercise P-9.3.

9.2.4 Shift-and-Invert

Before defining the analogue of the standard shift-and-invert technique we need to know how to incorporate linear shifts. From Theorem 9.1 seen in Section 9.2.1, for any pair of scalars σ_1, σ_2, the pair $(A - \sigma_1 B, B - \sigma_2 A)$ has the same eigenvectors as the original pair (A, B). An eigenvalue (α', β') of the transformed matrix pair is related to an eigenvalue pair (α, β) of the original matrix pair by

$$\begin{aligned} \alpha &= \alpha' + \sigma_1 \beta' \ , \\ \beta &= \beta' + \sigma_2 \alpha' \ . \end{aligned}$$

Computing (α', β') from (α, β) we get, assuming $1 - \sigma_1 \sigma_2 \neq 0$,

$$\alpha' = \frac{\alpha - \sigma_1 \beta}{1 - \sigma_1 \sigma_2}, \quad \beta' = \frac{\beta - \sigma_2 \alpha}{1 - \sigma_1 \sigma_2}.$$

In fact, since the eigenvalues are defined up to a scaling factor, we can write

$$\alpha' = \alpha - \sigma_1\beta\ , \quad \beta' = \beta - \sigma_2\alpha\ . \tag{9.19}$$

It is common to take one of the two shifts, typically σ_2, to be zero. In this special situation:

$$\alpha' = \alpha - \sigma_1\beta\ , \quad \beta' = \beta$$

which gives the usual situation corresponding to $\beta = 1$.

Shift-and-invert for the generalized problems corresponds to multiplying through the two matrices of the shifted pair by the inverse of one of them, typically the first. Thus the shifted-and-inverted pair would be

$$\left(I\ ,\ (A - \sigma_1 B)^{-1}(B - \sigma_2 A)\right).$$

This is now a problem which has the same eigenvalues as the pair $(A - \sigma_1 B, B - \sigma_2 A)$, i.e., its generic eigenvalue pair (α', β') is related to the original pair (α, β) of (A, B) via (9.19). It seems as if we have not gained anything as compared with the pair $(A - \sigma_1 B, B - \sigma_2 A)$. However, the A -matrix for the new pair is the identity matrix.

The most common choice is $\sigma_2 = 0$ and $\sigma_1\beta$ close to an eigenvalue of the original matrix.

9.2.5 Projection Methods

The projection methods seen in Chapter 4 are easy to extend to generalized eigenproblems. In the general framework of oblique projection methods, we are given two subspaces K and L and seek an approximate eigenvector $\tilde{u}$ in the subspace K and an approximate eigenvalue $(\tilde{\alpha}, \tilde{\beta})$ such that

$$(\tilde{\beta}A - \tilde{\alpha}B)\tilde{u} \perp L. \tag{9.20}$$

Given two bases $V = \{v_1, \ldots, v_m\}$, and $W = \{w_1, \ldots, w_m\}$ of K and L, respectively, and writing $\tilde{u} = Vy$, the above conditions translate into the generalized eigenvalue problem

$$\tilde{\beta}W^H AVy = \tilde{\alpha}W^H BVy\ .$$

Note that we can get a standard projected problem if we can find a pair W, V that is such that $W^H BV = I$. For orthogonal projection methods ($K = L$), this will be the case in particular when B is Hermitian positive definite, and the system of vectors $\{v_i\}_{i=1,\ldots m}$ is B-orthonormal.

When the original pair is Hermitian definite, i.e., when A and B are Hermitian positive definite and when B is positive definite, the projected problem will also be Hermitian definite. The approximate eigenvalues will also be real and all of the properties seen for the Hermitian case in Chapter 1 will extend in a straightforward way.

9.2.6 The Hermitian Definite Case

We devote this section to the important particular case where both A and B are Hermitian and one of them, say B, is positive definite. This situation corresponds to the usual Hermitian eigen-problem in the standard case. For example the eigenvalues are real and the eigenvectors from an orthogonal set with respect to the B–inner product defined by

$$(x, y)_B = (Bx, y) \; . \tag{9.21}$$

That this represents a proper inner product is well-known. The corresponding norm termed the B-norm is given by

$$\|x\|_B = (Bx, x)^{1/2} \; .$$

An important observation that is key to understanding this case is that even though the matrix $C = B^{-1}A$ of one of the equivalent standard eigenproblems is non-Hermitian with respect to the Euclidean inner product, it is *self-adjoint with respect to the B-inner product* in that

$$(Cx, y)_B = (x, Cy)_B \quad \forall \; x, y \; . \tag{9.22}$$

Therefore, one can expect that all the results seen for the standard problem for Hermitian case will be valid provided we replace Euclidean product by the B-inner product. For example, the min-max theorems will be valid provided we replace the Rayleigh quotient $(Ax, x)/(x, x)$ by

$$\mu(x) = \frac{(Cx, x)_B}{(x, x)_B} \; = \; \frac{(Ax, x)}{(Bx, x)} \; .$$

If we were to use the Lanczos algorithm we would have two options. The first is to factor the B matrix and use the equivalent standard formulation (9.12). This requires factoring the B-matrix and then solving a lower and an upper triangular system at each step of the Lanczos algorithm. An interesting alternative would be to simply employ the standard Lanczos algorithm for the matrix $C = B^{-1}A$ replacing the usual Euclidean inner product by the B inner product at each time that an inner product is invoked. Because of the self-adjointness of C with respect to the B inner product, we will obtain an algorithm similar to the one in the standard case, which is based on a simple three term recurrence. A naive implementation of the main loop in exact arithmetic would consist of the following steps,

$$\begin{aligned}
w &:= B^{-1}Av_j \; , && (9.23)\\
\alpha_j &:= (w, v_j)_B \; , && (9.24)\\
w &:= w - \alpha_j v_j - \beta_j v_{j-1} \; , && (9.25)\\
\beta_{j+1} &:= \|w\|_B \; , && (9.26)\\
v_{j+1} &:= w/\beta_{j+1} \; .
\end{aligned}$$

We observe that α_j in (9.24) is also equal to (Av_j, v_j) and this gives an easy way of computing the $\alpha's$, using standard Euclidean inner products. Before multiplying Av_j by B^{-1} in (9.23) α_j is computed and saved. The computation on β_{j+1} is a little more troublesome. The use of the definition of the B-inner product would require a multiplication by the matrix B. This may be perfectly acceptable if B is diagonal but could be wasteful in other cases. One way to avoid this matrix product is to observe that, by construction, the vector w in (9.26) is B-orthogonal to the vectors v_j and v_{j-1}. Therefore,

$$(Bw, w) = (Av_j, w) - \alpha_j(Bv_j, w) - \beta_j(Bv_{j-1}, w) = (Av_j, w).$$

As a result, if we save the vector Av_j computed in (9.23) until the end of the loop we can evaluate the B-norm of w with just an Euclidean inner product. Another alternative is to keep a three-term recurrence for the vectors $z_j = Bv_j$. Then Bw is available as

$$Bw = Av_j - \alpha_j z_j - \beta_j z_{j-1}$$

and the inner product (Bw, w) can be evaluated. Normalizing Bw by β_{j+1} yields z_{j+1}. This route requires two additional vectors of storage and a little additional computation but is likely to be more viable from the numerical point of view. Whichever approach is taken, a first algorithm will look as follows.

ALGORITHM **9.1 First Lanczos algorithm for matrix pairs**

1. **Start:** *Choose an initial vector* v_1 *of* B*-Norm unity. Set* $\beta_1 = 0, v_0 = 0$.
2. **Iterate:** *For* $j = 1, 2, \ldots, m$, *do:*
   - *(a)* $v := Av_j$,
   - *(b)* $\alpha_j := (v, v_j)$,
   - *(c)* $w := B^{-1}v - \alpha_j v_j - \beta_j v_{j-1}$,
   - *(d) Compute* $\beta_{j+1} = \|w\|_B$, *using* $\beta_{j+1} := \sqrt{(v, w)}$,
   - *(e)* $v_{j+1} = w/\beta_{j+1}$.

One difficulty in the above algorithm is the possible occurrence of a negative B norm of w in the presence of rounding errors.

A second algorithm which is based on keeping a three-term recurrence for the z_j's, implements a modified Gram-Schmidt version of the Lanczos algorithm, i.e., it is analogous to Algorithm 6.5 seen in Chapter 6.

ALGORITHM **9.2 Second Lanczos algorithm for matrix pairs**

1. **Start:** *Choose an initial vector* v_1 *of* B*-Norm unity. Set* $\beta_1 = 0, z_0 = v_0 = 0, z_1 = Bv_1$.
2. **Iterate:** *For* $j = 1, 2, \ldots, m$, *do*
   - *(a)* $v := Av_j - \beta_j z_{j-1}$,

(b) $\alpha_j = (v, v_j)$,
(c) $v := v - \alpha_j z_j$,
(d) $w := B^{-1}v$,
(e) $\beta_{j+1} = \sqrt{(w, v)}$,
(f) $v_{j+1} = w/\beta_{j+1}$ *and* $z_{j+1} = v/\beta_{j+1}$.

Note that the B-norm in (d) is now of the form $(B^{-1}v, v)$ and since B is Hermitian positive definite, this should not cause any numerical problems if computed properly.

In practice the above two algorithms will be unusable in the common situation when B is singular. This situation has been studied carefully in [137]. Without going into the geometric details, we would like to stress that the main idea here is to shift the problem so as to make $(A - \sigma B)$ nonsingular and then work in the subspace $\mathrm{Ran}(A - \sigma B)^{-1}B$. A simplification of the algorithm in [137] is given next. Here, σ is the shift.

ALGORITHM **9.3 Spectral Transformation Lanczos**

1. **Start:** *Choose an initial vector* w *in* $\mathrm{Ran}[\,(A - \sigma B)^{-1}B\,]$. *Compute* $z_1 = Bw$ *and* $\beta_1 := \sqrt{(w, z_1)}$. *Set* $v_0 := 0$.
2. **Iterate:** *For* $j = 1, 2, \ldots, m$, *do*
   (a) $v_j = w/\beta_j$ *and* $z_j := z_j/\beta_j$,
   (b) $z_j = (A - \sigma B)^{-1}w$,
   (c) $w := w - \beta_j v_{j-1}$,
   (d) $\alpha_j = (w, z_j)$,
   (e) $w := w - \alpha_j z_j$,
   (f) $z_{j+1} = Bw$,
   (g) $\beta_{j+1} = \sqrt{(z_{j+1}, w)}$.

Note that the algorithm requires only multiplications with the matrix B. As in the previous algorithm, the two most recent z_j's must be saved, possibly all of them if some form of B - reorthogonalization is to be included. We should point out a simple connection between this algorithm and the previous one. With the exception of the precaution taken to choose the initial vector, algorithm 9.3 is a slight reformulation of Algorithm 9.2, applied to the pair (A', B') where $A' = B$ and $B' = (A - \sigma B)$.

9.3 Quadratic Problems

The equation of motion for a structural system with viscous damping and without external forces is governed by the equation

$$M\ddot{q} + C\dot{q} + Kq = 0\ .$$

In vibration analysis, the generic solution of this equation is assumed to take the form $q = ue^{\lambda t}$ and this leads to the quadratic eigenvalue problem

$$(\lambda^2 M + \lambda C + K)u = 0\,. \tag{9.27}$$

These eigenvalue problems arise in dynamical systems where damping and other effects, e.g., gyroscopic, are taken into account. Such effects will define the C matrix. In the next subsections we will see how to adapt some of the basic tools to solve quadratic problems.

9.3.1 From Quadratic to Generalized Problems

The most common way of dealing with the above problem is to transform it into a (linear) generalized eigenvalue problem. For example, defining

$$v = \begin{pmatrix} \lambda u \\ u \end{pmatrix}$$

we can rewrite (9.27) as

$$\begin{pmatrix} -C & -K \\ I & 0 \end{pmatrix} v = \lambda \begin{pmatrix} M & 0 \\ 0 & I \end{pmatrix} v\,. \tag{9.28}$$

It is clear that there is a large number of different ways of rewriting (9.27), the one above being one of the simplest. One advantage of (9.27) is that when M is Hermitian positive definite, as is often the case, then so also is the second matrix of the resulting generalized problem (9.28). If all matrices involved, namely K, C, and M, are Hermitian it might be desirable to obtain a generalized problem with Hermitian matrices, even though this does not in any way guarantee that the eigenvalues will be real. We can write instead of (9.28)

$$\begin{pmatrix} C & K \\ K & 0 \end{pmatrix} v = \lambda \begin{pmatrix} -M & O \\ O & K \end{pmatrix} v\ . \tag{9.29}$$

An alternative to the above equation is

$$\begin{pmatrix} C & M \\ M & 0 \end{pmatrix} v = \mu \begin{pmatrix} -K & O \\ O & M \end{pmatrix} v \tag{9.30}$$

where we have set $\mu = 1/\lambda$. By comparing (9.29) and (9.30) we note the interesting fact that M and K have simply been interchanged. This could also have been observed directly from the original equation (9.27) by making the change of variable $\mu = 1/\lambda$. For practical purposes, we may therefore select between (9.30) and (9.29) the formulation that leads to the more economical computations. We will select (9.29) in the rest of this chapter.

While the difference between (9.30) and (9.29) may be insignificant, there are important practical implications in chosing between (9.28) and (9.29). Basically, the decision comes down to choosing an intrinsically non-Hermitian generalized

eigen-problem with *a Hermitian positive definite* B matrix, versus a generalized eigen-problem where *both matrices in the pair are Hermitian indefinite*. In the case where M is a (positive) diagonal matrix, then the first approach is not only perfectly acceptable, but may even be the method of choice in case Arnoldi's method using a polynomial preconditioning is to be attempted. In case all matrices involved are Hermitian positive definite, there are strong reasons why the second approach is to be preferred. These are explained by Parlett and Chen [149]. Essentially, one can use a Lanczos type algorithm, similar to one of versions described in subsection 9.2.6, in spite of the fact that the B matrix that defines the inner products is indefinite.

PROBLEMS

P-9.1 Examine how the eigenvalues and eigenvectors of a pair of matrices (A, B) change when both A and B are multiplied by the same nonsingular matrix to the left or to the right.

P-9.2 In section 9.2.4 and 9.2.3 the shifts σ_1, σ_2 were assumed to be such that $1-\sigma_1\sigma_2 \neq 0$. What happens if this were not to be the case? Consider both the linear shifts, Section 9.2.4 and Wielandt deflation 9.2.3.

P-9.3 Given the right and left eigenvectors u_1, and w_1 associated with an eigenvalue λ_1 of the pair A, B and such that $(Bu_1, Bw_1) = 1$, show that the matrix pair

$$A_1 = A - \sigma_1 B u_1 w_1^H B^H \ , \quad B_1 = B - \sigma_2 A u_1 w_1^H B^H$$

has the same left and right eigenvectors as A, B. The shifts σ_1, σ_2 are assumed to satisfy the condition $1 - \sigma_1\sigma_2 \neq 0$.

P-9.4 Show that when (A, B) are Hermitian and B is positive definite then $C = B^{-1}A$ is self-adjoint with respect to the B-inner product, i.e., that (9.22) holds.

P-9.5 Redo the proof of Proposition 9.1 with the usual definitions of eigenvalues ($Au = \lambda Bu$). What is gained? What is lost?

P-9.6 Show that algorithm 9.3 is a reformulation of Algorithm 9.2, applied to the pair (A', B') where $A' = B$ and $B' = (A - \sigma B)$.

NOTES AND REFERENCES. The reader is referred to Stewart and Sun [205] for more details and references on the theory of generalized eigenproblems. There does not seem to be any exhaustive coverage of the generalized eigenvalue problems, theory and algorithms, in one book. In addition, there seems to be a dichotomy between the need of users, mostly in finite elements modeling, and the numerical methods that numerical analysts develop. One of the first papers on the numerical solution of quadratic eigenvalue problems is Borri and Mantegazza [16]. Quadratic eigenvalue problems are rarely solved in structural engineering. The models are simplified first by neglecting damping and the leading eigenvalues of the resulting generalized eigenproblem are computed. Then the eigenvalues of the whole problem are approximated by performing a projection process onto the computed invariant subspace of the approximate problem [95]. This may very well change in the future, as models are improving and computer power is making rapid gains. ■

Chapter 10

ORIGINS OF MATRIX EIGENVALUE PROBLEMS

This chapter gives a brief overview of some applications that give rise to matrix eigenvalue problems. These applications can be classified in three different categories. The first category, by far the largest from the applications point of view, consists of problems related to the analysis of vibrations. These typically generate symmetric generalized eigenvalue problems. The second is the class of problems related to stability analysis, such as for example the stability analysis of an electrical network. In general, this second class of problems generates nonsymmetric matrices. The third category comprises physical applications related to quantum mechnical systems, specifically problems generated from the Schrödinger equation. The list of applications discussed in this chapter is by no means exhaustive. In fact the number of these applications is constantly growing. For example, an emerging application is one that is related to the broad area of data analysis, machine learning, and information sciences.

10.1 Introduction

The numerical computation of eigenvalues of large matrices is a problem of major importance in many scientific and engineering applications. We list below just a few of the applications areas where eigenvalue calculations arise:

- Structural dynamics
- Electrical Networks
- Combustion processes
- Macro-economics
- Normal mode techniques
- Quantum chemistry
- Markov chain techniques
- Chemical reactions
- Magnetohydrodynamics
- Control theory

One class of applications which has recently gained considerable ground is that related to linear algebra methods in data-mining, see for example, [109] for a survey. However, the most commonly solved eigenvalue problems today are those issued from the first item in the list, namely those problems associated with the vibration analysis of large structures. Complex structures such as those of an aircraft or a turbine are represented by finite element models involving a large number of degrees of freedom. To compute the natural frequencies of the structure one usually

solves a generalized eigenvalue problem of the form $Ku = \lambda Mu$ where typically, but not always, the stiffness and mass matrices K and M respectively, are both symmetric positive definite.

In the past decade tremendous advances have been achieved in the solution methods for symmetric eigenvalue problems especially those related to problems of structures. The well-known structural analysis package, NASTRAN, which was developed by engineers in the sixties and seventies now incorporates the state of the art in numerical methods for eigenproblems such as block Lanczos techniques.

Similar software for the nonsymmetric eigenvalue problem on the other hand remains lacking. There seems to be two main causes for this. First, in structural engineering where such problems occur in models that include damping, and gyroscopic effects, it is a common practice to replace the resulting quadratic problem by a small dense problem much less difficult to solve using heuristic arguments. A second and more general reason is due to a prevailing view among applied scientists that the large nonsymmetric eigenvalue problems arising from their more accurate models are just intractable or difficult to solve numerically. This often results in simplified models to yield smaller matrices that can be handled by standard methods. For example, one-dimensional models may be used instead of two-dimensional or three-dimensional models. This line of reasoning is not totally unjustified since nonsymmetric eigenvalue problems can be hopelessly difficult to solve in some situations due for example, to poor conditioning. Good numerical algorithms for non-Hermitian eigenvalue problems tend also to be far more complex that their Hermitian counterparts. Finally, as was reflected in earlier chapters, the theoretical results that justify their use are scarcer.

The goal of this chapter is mainly to provide motivation and it is independent of the rest of the book. We will illustrate the main ideas that lead to the various eigenvalue problems in some of the applications mentioned above. The presentation is simplified in order to convey the overall principles.

10.2 Mechanical Vibrations

Consider a small object of mass m attached to an elastic spring suspended from the lid of a rigid box, see Figure 10.1. When stretched by a distance Δl the spring will exert a force of magnitude $k\Delta l$ whose direction is opposite to the direction of the displacement. Moreover, if there is a fluid in the box, such as oil, a displacement will cause a damping, or drag force to the movement, which is usually proportional to the velocity of the movement. Let us call l the distance of the center of the object from the top of the box when the mass is at equilibrium and denote by y the position of the mass at time t, with the initial position $y = 0$ being that of equilibrium. Then at any given time there are four forces acting on m:

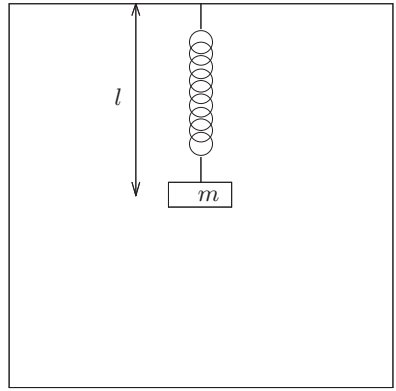


Figure 10.1: Model problem in mechanical vibrations

1. The gravity force mg pulling downward;
2. The spring force $-k(l+y)$;
3. The damping force $-c\dfrac{dy}{dt}$;
4. The external force $F(t)$.

By Newton's law of motion,

$$m\frac{d^2y}{dt^2} = mg - k(l+y) - c\frac{dy}{dt} + F(t)\ .$$

If we write the equation at steady state, i.e., setting $y \equiv 0$ and $F(t) \equiv 0$, we get $mg = kl$. As a result the equation simplifies into

$$m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky = F(t)\ . \tag{10.1}$$

Free vibrations occur when there are no external forces and when the damping effects are negligible. Then (10.1) becomes

$$m\frac{d^2y}{dt^2} + ky = 0 \tag{10.2}$$

the general solution of which is of the form

$$y(t) = R\cos\left(\frac{k}{m}t - \phi\right)$$

which means that the mass will oscillate about its equilibrium position with a period of $2\pi/\omega_0$, with $\omega_0 \equiv k/m$.

Damped free vibrations include the effect of damping but exclude any effects from external forces. They lead to the homogeneous equation:

$$m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky = 0$$

whose characteristic equation is $mr^2 + cr + k = 0$.

When $c^2 - 4km > 0$ then both solutions r_1, r_2 of the characteristic equation are negative and the general solution is of the form

$$y(t) = ae^{r_1 t} + be^{r_2 t}$$

which means that the object will return very rapidly to its equilibrium position. A system with this characteristic is said to be *overdamped*.

When $c^2 - 4km = 0$ then the general solution is of the form

$$y(t) = (a + bt)e^{-ct/2m}$$

which corresponds to critical damping. Again the solution will return to its equilibrium but in a different type of movement from the previous case. The system is said to be *critically damped*.

Finally, the case of *underdamping* corresponds to the situation when $c^2 - 4km < 0$ and the solution is of the form

$$y(t) = e^{-ct/2m}\,[a\cos\mu t + b\sin\mu t]$$

with

$$\mu = \frac{\sqrt{4km - c^2}}{2m}\ .$$

This time the object will oscillate around its equilibrium but the movement will die out quickly.

In practice the most interesting case is that of *forced vibrations*, in which the exterior force F has the form $F(t) = F_0\cos\omega t$. The corresponding equation is no longer a homogeneous equation, so we need to seek a particular solution to the equation (10.1) in the form of a multiple of $\cos(\omega t - \delta)$. Doing so, we arrive after some calculation at the solution

$$\eta(t) = \frac{F_0\cos(\omega t - \delta)}{\sqrt{(k - m\omega^2)^2 + c^2\omega^2}} \tag{10.3}$$

where

$$\tan\delta = \frac{c\omega}{k - m\omega^2}\ .$$

See Exercise P-10.3 for a derivation. The general solution to the equations with forcing is obtained by adding this particular solution to the general solution of the homogeneous equation seen earlier.

The above solution is only valid when $c \neq 0$. When $c = 0$, i.e., when there are no damping effects, we have what is referred to as *free forced vibrations* . In

this case, letting $\omega_0^2 = \frac{k}{m}$, a particular solution of the non-homogeneous equation is

$$\frac{F_0}{m(\omega_0^2 - \omega^2)} \cos \omega t$$

when $\omega \neq \omega_0$ and

$$\frac{F_0 t}{2m\omega_0} \sin \omega_0 t \tag{10.4}$$

otherwise. Now every solution is of the form

$$y(t) = a \cos \omega t + b \ \sin \omega t + \frac{F_0}{2m\omega_0} \ t \ \sin \omega_0 t.$$

The first two terms in the above solution constitute a periodic function but the last term represents an oscillation with a dangerously increasing amplitude.

This is referred to as a resonance phenomenon and has been the cause of several famous disasters in the past, one of the most recent ones being the Tacoma bridge disaster (Nov. 7, 1940). Another famous such catastrophe, is that of the Broughton suspension bridge near Manchester England. In 1831 a column of soldiers marched on it in step causing the bridge to enter into resonance and collapse. It has since become customary for soldiers to break step when entering a bridge. For an interesting accoint of the Tacoma bridge disaster mentioned above and other similar phenomena see Braun [17].

Note that in reality the case $c = 0$ is fallacious since some damping effects always exist. However, in practice when c is very small the particular solution (10.3) can become very large when $\omega^2 = k/m$. Thus, whether c is zero or simply very small, *dangerous oscillations can occur whenever the forcing function F has a period equal to that of the free vibration case*.

We can complicate matters a little in order to introduce matrix eigenvalue problems by taking the same example as before and add another mass suspended to the first one, as is shown in Figure 10.2.

Assume that at equilibrium, the center of gravity of the first mass is at distance l_1 from the top and that of the second is at distance l_2 from the first one. There are now two unknowns, the displacement y_1 from the equilibrium of the first mass and the displacement y_2 from its equilibrium position of the second mass. In addition to the same forces as those for the single mass case, we must now include the effect of the spring force pulling from the other spring. For the first mass this is equal to

$$k_2[l_2 - y_1 + y_2],$$

which clearly corresponds to a displacement of the second mass relative to the first one. A force equal to this one in magnitude but opposite in sign acts on the second mass in addition to the other forces. Newton's law now yields

$$m_1 \frac{d^2 y_1}{dt^2} = m_1 g - k_1(l_1 + y_1) - c_1 \frac{dy_1}{dt} + k_2(l_2 + y_2 - y_1) + F_1(t) \ ,$$

$$m_2 \frac{d^2 y_2}{dt^2} = m_2 g - k_2(l_2 + y_1) - c \frac{dy_2}{dt} - k_2(l_2 + y_2 - y_1) + F_2(t) \ .$$

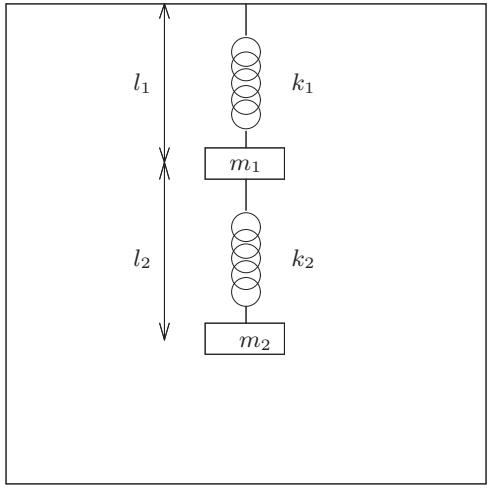


Figure 10.2: A spring system with two masses.

At equilibrium the displacements as well as their derivatives, and the external forces are zero. As a result we must have $0 = m_1 g - k_1 l_1 + k_2 l_2$, and $0 = m_2 g - 2k_2 l_2$. Hence the simplification

$$m_1 \frac{d^2 y_1}{dt^2} + c_1 \frac{dy_1}{dt} + (k_1 + k_2) y_1 - k_2 y_2 = F_1(t) \,, \tag{10.5}$$

$$m_2 \frac{d^2 y_2}{dt^2} + c_2 \frac{dy_2}{dt} - k_2 y_1 + 2k_2 y_2 = F_2(t) \,. \tag{10.6}$$

Using the usual notation of mechanics for derivatives, equations (10.5) and (10.6) can be written in condensed form as

$$\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix} \begin{pmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{pmatrix} + \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} + \begin{pmatrix} k_1 + k_2 & -k_2 \\ -k_2 & 2k_2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \tag{10.7}$$

or,

$$M\ddot{y} + C\dot{y} + Ky = F \tag{10.8}$$

in which M, C and K are 2×2 matrices. More generally, one can think of a very large structure, for example a high rise building, as a big collection of masses and springs that are interacting with each other just as in the previous example. In fact equation (10.8) is the typical equation considered in structural dynamics but the matrices $M, K,$ and C can be very large. One of the major problems in structural engineering it to attempt to avoid vibrations, i.e., the resonance regime explained

earlier for the simple one mass case. According to our previous discussion this involves avoiding the eigenfrequencies, ω_0 in the previous example, of the system. More exactly, an analysis is made before the structure is build and the proper frequencies are computed. There is usually a band of frequencies that must be avoided. For example, an earthquake history of the area may suggest avoiding specific frequencies. Here, the proper modes of the system are determined by simply computing oscillatory solutions of the form $y(t) = y_0 e^{i\omega t}$ that satisfies the free undamped vibration equation

$$M\ddot{y} + Ky = 0 \quad \text{or} \quad -\omega^2 M y_0 + K y_0 = 0 \, .$$

10.3 Electrical Networks.

Consider a simple electrical circuit consisting of a resistance or R Ohms, an inductance of L Henrys and a capacitor of C Farads connected in series with a generator of E volts.

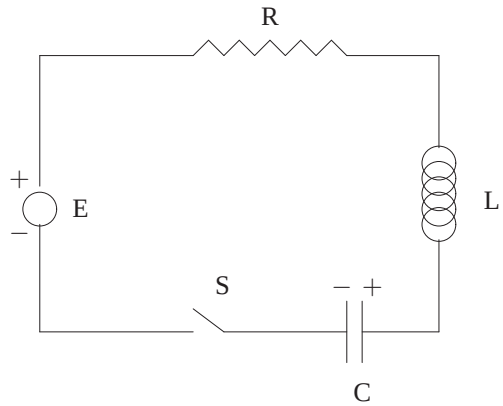


Figure 10.3: A simple series electric circuit.

In a closed circuit, the sum of the voltage drops is equal to the input voltage $E(t)$. The voltage drop across the resistance is RI where I is the intensity while it is $L\dot{I}$ across the inductance and Q/C across the capacitor where Q is the electric charge whose derivative is I. Therefore the governing equations can be written in terms of Q as follows,

$$L\ddot{Q} + R\dot{Q} + Q/C = E(t) \, ,$$

which resembles that of mechanical vibrations. Realistic electric networks can be modeled by a large number of circuits interconnected to each other. Resonance here might be sought rather than avoided, as occurs when tuning a radio to a given electromagnetic wave which is achieved by varying the capacity C.

The problem of power system networks is different in that there are instabilities of exponential type that occur in these systems under small disturbances. The problem there is to control these instabilities. Although very complex in nature, the problem of power systems instability can be pictured from the above simple circuit in which the resistance R is made negative, i.e., we assume that the resistance is an active device rather than a passive one. Then it can be seen that the circuit may become unstable because the solution takes the form $ae^{s_1 t} + be^{s_2 t}$ in which s_1, s_2 may have positive real parts, which leads to unstable solutions.

10.4 Electronic Structure Calculations

One of the greatest scientific achievements of humankind is the discovery, in the early part of the twentieth century, of quantum mechanical laws describing the behavior of matter. These laws make it possible, at least in principle, to predict the electronic properties of matter from the nanoscale to the macroscale. The progress that lead to these discoveries is vividly narrated in the book *"Thirty years that shook physics"* by George Gamov [68]. A series of discoveries, starting with the notion of quantas originated by Max Planck at the end of 1900, and ending roughly in the mid-1920's, with the emergence of the Schrödinger wave equation, set the stage for the new physics. Solutions of the Schrödinger wave equation resulted in essentially a complete understanding of the dynamics of matter at the atomic scale.

One could, formally at least, understand atomic and molecular phenomena from these equations, but solving these equations in their original form is nearly impossible, even today, except for systems with a very small number of electrons. The decades following the discovery of quantum mechanics have elaborated several methods for findind good approximations to the solutions. In terms of methodology and algorithms, the biggest steps forward were made in the sixties with the advent of two key new ideas. The first, *density functional theory*, enabled one to transform the initial problem into one which involves functions of only one space variables instead of N space variables, for N-particle systems in the original Schrödinger equation. Instead of dealing with functions in $\mathbb{R}^{3N}$, we only need to handle functions in $\mathbb{R}^3$. The second substantial improvement came with *pseudopotentials*. In short pseudopotentials allowed one to reduce the number of electrons to be considered by constructing special potentials, which would implicitly reproduce the effect of chemically inert core electrons and explicitly reproduce the properties of the chemically active valence electrons .

10.4.1 Quantum descriptions of matter

Consider N nucleons of charge Z_n at positions $\{\mathbf{R}_n\}$ for $n = 1, \cdots, N$ and M electrons at positions $\{\mathbf{r}_i\}$ for $i = 1, \cdots, M$. An illustration is shown in Figure 10.4.

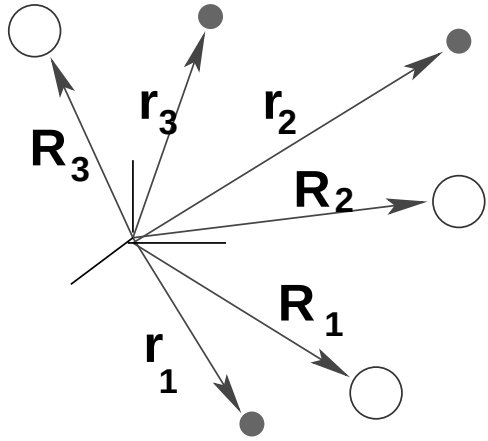


Figure 10.4: Atomic and electronic coordinates: Filled circles represent electrons, open circles represent nuclei (Source of figure: [181]).

The non-relativistic, time-independent Schrödinger equation for the electronic structure of the system can be written as:

$$\mathcal{H}\,\Psi = E\,\Psi \tag{10.9}$$

where the many-body wave function Ψ is of the form

$$\Psi \equiv \Psi(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \cdots; \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \cdots) \tag{10.10}$$

and E is the total electronic energy. The Hamiltonian $\mathcal{H}$ in its simplest form is:

$$\begin{aligned}\mathcal{H}(\mathbf{R}_1, \mathbf{R}_2, \cdots; \mathbf{r}_1, \mathbf{r}_2, \cdots) = & -\sum_{n=1}^{N} \frac{\hbar^2 \nabla_n^2}{2M_n} + \frac{1}{2} \sum_{\substack{n,n'=1,\\ n\neq n'}}^{N} \frac{Z_n Z_{n'} e^2}{|\mathbf{R}_n - \mathbf{R}_{n'}|} \\ & - \sum_{i=1}^{M} \frac{\hbar^2 \nabla_i^2}{2m} - \sum_{n=1}^{N}\sum_{i=1}^{M} \frac{Z_n e^2}{|\mathbf{R}_n - \mathbf{r}_i|} + \frac{1}{2} \sum_{\substack{i,j=1\\ i\neq j}}^{M} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}\,. \end{aligned} \tag{10.11}$$

Here, M_n is the mass of the nucleus, $\hbar$ is Planck's constant, h, divided by 2π, m is the mass of the electron, and e is the charge of the electron. The above Hamiltonian includes the kinetic energies for the nucleus (first sum in $\mathcal{H}$), and each electron (3rd sum), the inter-nuclei repulsion energies (2nd sum), the nuclei-electronic (Coulomb) attraction energies (4th sum), and the electron-electron repulsion energies (5th sum). Each Laplacean ∇_n^2 involves differentiation with respect to the coordinates of the n^{th} nucleus. Similarly the term ∇_i^2 involves differentiation with respect to the coordinates of the i^{th} electron.

In principle, the electronic structure of any system is completely determined by (10.9), or, to be exact, by minimizing the energy $< \Psi|\mathcal{H}|\Psi >$ under the constraint of normalized wave functions Ψ. This is nothing but the Rayleigh quotient of the Hamiltonian associated with the wave function Ψ and its minimum

is reached when Ψ is the eigenfunction associated with the smallest eigenvalue. Recall that Ψ has a probabilistic interpretation: for the minimizing wave function Ψ,

$$|\Psi(\mathbf{R}_1, \cdots, \mathbf{R}_N; \mathbf{r}_1, \cdots, \mathbf{r}_M)|^2 d^3\mathbf{R}_1 \cdots d^3\mathbf{R}_N d^3\mathbf{r}_1 \cdots d^3\mathbf{r}_M$$

represents the probability of finding particle 1 in volume $|\mathbf{R}_1 + d^3\mathbf{R}_1|$, particle 2 in volume $|\mathbf{R}_2 + d^3\mathbf{R}_2|$, etc.

From a computational point of view however, the problem is not tractable for systems which include more than just a few atoms and dozen electrons, or so. The main computational difficulty stems from the nature of the wave function which involves all coordinates of all particles (nuclei and electrons) simultaneously. For example, if we had just 10 particles, and discretized each coordinate using just 100 points for each of the x, y, z directions, we would have 10^6 points for each coordinate for a total of $\left(10^6\right)^{10} = 10^{60}$ variables altogether.

Several simplifications were made to develop techniques which are practical as well as sufficiently accurate. The goal for these approximations is to be able to compute both the ground state, *i.e.*, the state corresponding to minimum energy E, and excited state energies, or energies corresponding to higher eigenvalues E in (10.9), and this by using a reasonable number of degrees of freedom.

The first, and most basic, of these is the *Born-Oppenheimer* or *adiabatic* approximation. Since the nuclei are considerably more massive than the electrons, it can be assumed that the electrons will respond "instantaneously" to the nuclear coordinates. We can then separate the nuclear coordinates from the electronic coordinates. Under this approximation, the first term in (10.11) vanishes and the second becomes a constant. We are left with a new Hamiltonian:

$$\begin{aligned} \mathcal{H}(\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_M) &= \sum_{i=1}^{M} \frac{-\hbar^2 \nabla_i^2}{2m} - \sum_{n=1}^{N} \sum_{i=1}^{M} \frac{Z_n e^2}{|\mathbf{R}_n \mathbf{r}_i|} \\ &\quad + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{M} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} \,. \end{aligned} \tag{10.12}$$

This simplification in itself will not be sufficient to reduce the complexity of the Schrödinger equation to an acceptable level.

10.4.2 The Hartree approximation

If we were able to write the Hamiltonian $\mathcal{H}$ as a sum of individual (non-interacting) Hamiltonians, one for each electron, then it is easy to see that the problem would become *separable*. In this case the wave function Ψ can be written as a product of individual *orbitals*, $\phi_k(\mathbf{r}_k)$ each of which is an eigenfunction of the non-interacting Hamiltonian. This is an important concept and it is often characterized as the *one-electron* picture of a many-electron system.

The eigenfunctions of such a Hamiltonian determine orbitals (eigenfunctions) and energy levels (eigenvalues). For many systems, there are an infinite number of states, enumerated by quantum numbers. Each eigenvalue represents an "energy"

level corresponding to the orbital of interest. For example, in an atom such as hydrogen, an infinite number of bound states exist, each labeled by a set of three discrete integers. In general, the number of integers equal the spatial dimensionality of the system plus spin. In hydrogen, each state can be labeled by three indices (n, l, and m) and s for spin. In the case of a solid, there are essentially an infinite number of atoms and the energy levels can be labeled by quantum numbers, which are no longer discrete, but quasi-continuous. In this case, the energy levels form an *energy band*.

The energy states are filled by minimizing the total energy of the system, in agreement with the Pauli principle, *i.e.*, each electron has a unique set of quantum numbers, which label an orbital. The N lowest orbitals account for $2N$ electrons, i.e., a pair of a spin up and a spin down electrons for each orbital. Orbitals that are not occupied are called "virtual states." The lowest energy orbital configuration is called the *ground state*. The ground state can be used to determine a number of properties, *e.g.*, stable structures, mechanical deformations, phase transitions, and vibrational modes. The states above the ground state are known as *excited states*. These are helpful in calculating response functions of the solid, e.g., the dielectric and the optical properties of materials.

In mathematical terms, $\mathcal{H} \equiv \oplus \mathcal{H}^i$, the circled sum being a direct sum meaning that $\mathcal{H}^i$ acts *only* on particle number i, leaving the others unchanged. Hartree suggested to use this as an approximation technique whereby the basis resulting from this calculation will be substituted in $< \Psi|\mathcal{H}|\Psi > / < \Psi|\Psi >$, to yield an upper bound for the energy.

In order to make the Hamiltonian (10.12) non-interactive, we must remove the last term in (10.12), *i.e.*, we assume that the electrons do not interact with each other. Then the *electronic* part of the Hamiltonian becomes:

$$\mathcal{H}_{el} = \mathcal{H}_{el}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \cdots) = \sum_{i=1}^{M} \frac{-\hbar^2 \nabla_i^2}{2m} \; - \; \sum_{n=1}^{N} \sum_{i=1}^{M} \frac{Z_n e^2}{|\mathbf{R}_n - \mathbf{r}_i|} \tag{10.13}$$

which can be cast in the form

$$\mathcal{H}_{el} = \sum_{i=1}^{M} \left[\frac{-\hbar^2 \nabla_i^2}{2m} + \mathcal{V}_N(\mathbf{r}_i) \right] \equiv \bigoplus_{i=1}^{M} \mathcal{H}^i \tag{10.14}$$

where

$$\mathcal{V}_N(\mathbf{r}_i) = \; - \; \sum_{n=1}^{N} \frac{Z_n e^2}{|\mathbf{R}_n - \mathbf{r}_i|} \; . \tag{10.15}$$

This simplified Hamiltonian is separable and admits eigenfunctions of the form

$$\psi(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \cdots) = \phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)\phi_3(\mathbf{r}_3) \cdots \; , \tag{10.16}$$

where the $\phi_i(\mathbf{r})$ orbitals are determined from the "one-electron" equation:

$$\mathcal{H}^i \phi_i(\mathbf{r}) = E_i \phi_i(\mathbf{r}) \; . \tag{10.17}$$

The total energy of the system is the sum of the occupied eigenvalues, E_i.

This model is extremely simple, but not realistic. Physically, using the statistical interpretation mentioned above, writing Ψ as the product of ϕ_i's, only means that the electrons have independent probabilities of being located in a certain position in space. This lack of *correlation* between the particles causes the resulting energy to be overstated. In particular, the Pauli Principle states that no two electrons can be at the same point in space and have the same quantum numbers. The solutions Ψ computed in (10.16)–(10.17) is known as the *Hartree wave function*.

It can be shown that the individual orbitals, $\phi_i(\mathbf{r})$, are solutions of the eigenvalue problem

$$\left(\frac{-\hbar^2\nabla^2}{2m} + \mathcal{V}_N(\mathbf{r}) + \sum_{\substack{j=1 \\ j\neq i}}^{M} \int \frac{e^2|\phi_j(\mathbf{r}')|^2}{|\mathbf{r}'-\mathbf{r}|}\, d^3\mathbf{r}' \right) \phi_i(\mathbf{r}) = E_i\phi_i(\mathbf{r})\ . \quad (10.18)$$

The subscripts i, j of the coordinates have been removed as there is no ambiguity. The Hamiltonian related to each particle can be written in the form $\mathcal{H} = \frac{-\hbar^2\nabla^2}{2m} + \mathcal{V}_N + \mathcal{W}_H$, where $\mathcal{V}_N$ was defined earlier and

$$\mathcal{W}_H \equiv \sum_{\substack{j=1 \\ j\neq i}}^{M} \int \frac{e^2\phi_j(\mathbf{r})\phi_j(\mathbf{r})^* d^3\mathbf{r}'}{|\mathbf{r}'-\mathbf{r}|}\ . \quad (10.19)$$

This *Hartree potential*, or *Couloumb potential*, can be interpreted as the potential seen from each electron by averaging the distribution of the other electrons $|\phi_j(\mathbf{r})|^2$'s. It can be obtained from solving the Poisson equation with the charge density $e|\phi_j(\mathbf{r})|^2$ for each electron j. Note that both $\mathcal{V}_N$ and $\mathcal{W}_H$ depend on the electron i. Another important observation is that solving the eigenvalue problem (10.18), requires the knowledge of the other orbitals ϕ_j, *i.e.*, those for $j \neq i$. Also, the electron density of the orbital in question should not be included in the construction of the Hartree potential.

The solution of the problem requires a *self-consistent field* (SCF) iteration. One begins with some set of orbitals, and computes iteratively new sets by solving (10.18), using the most current set of $\phi_j's$ for $j \neq i$. This iteration is continued until the set of ϕ_i's is self-consistent. One difficulty is that the Hamiltonian depends on the orbital since the summation in (10.18) excludes the term $j = i$. This means that if there are M electrons, then M Hamiltonians must be considered and (10.18) solved for each of them at each SCF loop. This procedure can therefore be expensive.

A major weakness of the Hartree approximation is that it does not obey the Pauli exclusion principle [124]. The Hartree-Fock method, discussed next, is an attempt to remedy this weakness.

10.4.3 The Hartree-Fock approximation

Pauli's exclusion principle states that there can be only two electrons in the same orbit and they must be of opposite spin. The coordinates must include spin, so

we define $\mathbf{x}_i = \begin{pmatrix} \mathbf{r}_i \\ s_i \end{pmatrix}$ where s_i is the spin of the i^{th} electron. A canonical way to enforce the exclusion principle is to require that a wave function Ψ be an antisymmetric function of the coordinates $\mathbf{x}_i$ of the electrons in that by inter-changing any two of these its coordinates, the function must change its sign. In the Hartree-Fock approximation, many body wave functions with antisymmetric properties are constructed, typically cast as *Slater determinants*, and used to approximately solve the eigenvalue problem associated with the Hamiltonian (10.12).

Starting with one-electron orbitals, $\phi_i(\mathbf{x}) \equiv \phi(\mathbf{r})\sigma(s)$, the following functions meet the antisymmetry requirements:

$$\hat{\Psi}(\mathbf{x}_1, \mathbf{x}_2, \cdots) = \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \cdots & \phi_1(\mathbf{x}_M) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_M(\mathbf{x}_1) & \cdots & \cdots & \cdots & \phi_M(\mathbf{x}_M) \end{vmatrix} . \tag{10.20}$$

The actual Slater determinants are obtained from normalizing the auxiliary functions $\hat{P}si$ of (10.20): $\Psi = (M!)^{-1/2}\hat{\Psi}$. If two electrons occupy the same orbit, two rows of the determinant will be identical and Ψ will be zero. The determinant will also vanish if two electrons occupy the same point in generalized space (*i.e.*, $\mathbf{x}_i = \mathbf{x}_j$) as two columns of the determinant will be identical. Exchanging positions of two particles will lead to a sign change in the determinant.

If one uses a Slater determinant to evaluate the total electronic energy and maintains wave function normalization, the orbitals can be obtained from the following *Hartree-Fock* equations:

$$\mathcal{H}^i\phi_i(\mathbf{r}) = \left(\frac{-\hbar^2\nabla^2}{2m} + \mathcal{V}_N(\mathbf{r}) + \sum_{j=1}^{M} \int \frac{e^2\,|\phi_j(\mathbf{r}\,')|^2}{|\mathbf{r} - \mathbf{r}\,'|}\, d^3\mathbf{r}\,' \right) \phi_i(\mathbf{r})$$
$$- \sum_{j=1}^{M} \int \frac{e^2}{|\mathbf{r} - \mathbf{r}\,'|}\, \phi_j^*(\mathbf{r}\,')\phi_i(\mathbf{r}\,')\, d^3\mathbf{r}\,'\ \ \delta_{s_i,s_j}\ \phi_j(\mathbf{r}) = E_i\phi_i(\mathbf{r})\,. \tag{10.21}$$

It is customary to simplify this expression by defining an electronic charge density, ρ:

$$\rho(\mathbf{r}) = \sum_{j=1}^{M} |\phi_j(\mathbf{r}\,)|^2, \tag{10.22}$$

and an orbital dependent "*exchange-charge density*", ρ_i^{HF} for the i^{th} orbital:

$$\rho_i^{HF}(\mathbf{r}, \mathbf{r}\,') = \sum_{j=1}^{M} \frac{\phi_j^*(\mathbf{r}\,')\,\phi_i(\mathbf{r}\,')\,\phi_i^*(\mathbf{r}\,)\,\phi_j(\mathbf{r}\,)}{\phi_i^*(\mathbf{r}\,)\,\phi_i(\mathbf{r}\,)}\,\delta_{s_i,s_j}\,. \tag{10.23}$$

This "density" involves a spin dependent factor which couples only states (i, j) with the same spin coordinates (s_i, s_j).

With these charge densities defined, it is possible to define corresponding potentials. The *Coulomb* or *Hartree* potential, $\mathcal{V}_H$, is defined by

$$\mathcal{V}_H(\mathbf{r}) = \int \rho(\mathbf{r})\, \frac{e^2}{|\mathbf{r} - \mathbf{r}\,'|}\, d^3\mathbf{r}'\,. \tag{10.24}$$

and an *exchange* potential can be defined by

$$\mathcal{V}_x^i(\mathbf{r}) = \; -\int \rho_i^{HF}(\mathbf{r},\mathbf{r}\,')\,\frac{e^2}{|\mathbf{r}-\mathbf{r}\,'|}\,d^3\mathbf{r}'\,. \tag{10.25}$$

This combination results in the following Hartree-Fock equation:

$$\left(\frac{-\hbar^2\nabla^2}{2m} + \mathcal{V}_N(\mathbf{r}) + \mathcal{V}_H(\mathbf{r}) + \mathcal{V}_x^i(\mathbf{r})\right)\,\phi_i(\mathbf{r}) = E_i\phi_i(\mathbf{r})\,. \tag{10.26}$$

The number of electronic degrees of freedom in Hartree-Fock - based calculations grows rapidly with the number atoms often prohibiting an accurate solution, or even one's ability to store the resulting wave function. The method scales nominally as N^4 (N being the number of basis functions), though its practical scaling is closer to N^3. An alternate approach is based on Density Functional Theory (DFT) which scales nominally as N^3, or close to N^2 in practice.

10.4.4 Density Functional Theory

In a number of classic papers, Hohenberg, Kohn, and Sham established a theoretical basis for justifying the replacement of the many-body wave function by one-electron orbitals [123, 90, 108]. Their results put the charge density at center stage. The charge density is a distribution of probability, *i.e.*, $\rho(\mathbf{r}_1)d^3\mathbf{r}_1$ represents,in a probabilistic sense, the number of electrons (all electrons) in the infinitesimal volume $d^3\mathbf{r}_1$.

Specifically, the Hohenberg-Kohn results were as follows. The first Hohenberg and Kohn theorem states that *for any system of electrons in an external potential $\mathcal{V}_{ext}$, the Hamiltonian (specifically $\mathcal{V}_{ext}$ up to a constant) is determined uniquely by the ground-state density alone.* Solving the Schrödinger equation would result in a certain ground-state wave function Ψ, to which is associated a certain charge density,

$$\rho(\mathbf{r}_1) = \sum_{s_1=\uparrow,\downarrow} M \int |\Psi(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_M)|d\mathbf{x}_2\cdots d\mathbf{x}_\mathbf{M}\,. \tag{10.27}$$

From each possible state function Ψ one can obtain a (unique) probability distribution ρ. This mapping from the solution of the full Schrödinger equation to ρ is trivial. What is less obvious is that the reverse is true: Given a charge density, ρ, it is possible in theory to obtain a unique Hamiltonian and associated ground-state wave function, Ψ. Hohenberg and Kohn's first theorem states that this mapping is one-to-one, *i.e.*, we could get the Hamiltonian (and the wave function) solely from ρ.

The second Hohenberg-Kohn theorem provides the means for obtaining this reverse mapping: *The ground-state density of a system in a particular external potential can be found by minimizing an associated energy functional.* In principle, there is a certain energy functional, which is minimized by the unknown ground state charge density, ρ. This statement still remains at a formal level in the

sense that no practical means was given for computing Ψ or a potential, $\mathcal{V}$. The magnitude of the simplification suggests that the energy functional will be hard to construct. Indeed, this transformation changes the original problem with a total of $3N$ coordinates plus spin, to one with only 3 coordinates, albeit with N orbitals to be determined.

Later Kohn and Sham provided a workable computational method based on the following result: *For each interacting electron system, with external potential $\mathcal{V}_0$, there is a local potential $\mathcal{V}_{ks}$, which results in a density ρ equal to that of the interacting system.* Thus, the Kohn-Sham energy functional is formally written in the form

$$\mathcal{H}_{KS} = \frac{\hbar^2}{2m}\nabla^2 + \mathcal{V}_{eff}, \tag{10.28}$$

where the effective potential is defined as for a one-electron potential, *i.e.*, as in (10.14),

$$\mathcal{V}_{eff} = \mathcal{V}_N(\rho) + \mathcal{V}_H(\rho) + \mathcal{V}_{xc}(\rho). \tag{10.29}$$

Note that in contrast with (10.14), $\mathcal{V}_{xc}$ is now without an index, as it is only for one electron. Also note the dependence of each potential term on the charge density ρ, which is implicitly defined from the set of occupied eigenstates $\psi_i, i = 1, \cdots, N$ of (10.28) by Eq. (10.22).

The energy term associated with the nuclei-electron interactions is $\langle \mathcal{V}_N|\rho\rangle$, while that associated with the electron-electron interactions is $\langle \mathcal{V}_H|\rho\rangle$, where $\mathcal{V}_H$ is the Hartree potential,

$$\mathcal{V}_H = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}' \ .$$

The Kohn-Sham energy functional is of the following form:

$$\begin{aligned} E(\rho) \quad &= \quad -\frac{\hbar^2}{2m}\sum_{i=1}^{N}\int \phi_i^*(\mathbf{r})\nabla^2\phi_i(\mathbf{r})d\mathbf{r} \ + \ \int \rho(\mathbf{r})\mathcal{V}_{ion}(\mathbf{r})d\mathbf{r} \\ &\quad + \frac{1}{2}\int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' + E_{xc}(\rho(\mathbf{r})) \end{aligned} \tag{10.30}$$

The effective energy, or Kohn-Sham energy, may not represent the true, or "experimental energy," because the Hamiltonian has been approximated.

One of the issues left with the DFT approximation is to determine the Exchange and Correlation energy from which the potential $\mathcal{V}_{xc}$ in (10.29) can be obtained.

In contemporary theories, correlation energies are explicitly included in the energy functionals [123]. These energies have been determined by numerical studies performed on uniform electron gases resulting in local density expressions of the form: $\mathcal{V}_{xc}[\rho(\mathbf{r})] = \mathcal{V}_x[\rho(\mathbf{r})] + \mathcal{V}_c[\rho(\mathbf{r})]$, where $\mathcal{V}_c$ represents contributions to the total energy beyond the Hartree-Fock limit [20]. For the exchange energy, one of the simplest model in DFT consists of using the *Local Density Approximation* (LDA), originally suggested by Kohn and Sham [108]. Within LDA, one obtains

the following expression:

$$E_x[\rho] = -\frac{3e^2}{4\pi}(3\pi^2)^{1/3}\int[\rho(\mathbf{r})]^{4/3}\,d^3\mathbf{r}, \tag{10.31}$$

from which one obtains $\mathcal{V}_x[\rho]$ by taking the functional derivative:

$$\mathcal{V}_x[\rho] = \frac{\delta E_x[\rho]}{\delta\rho} = -\frac{e^2}{\pi}(3\pi^2\rho(\mathbf{r}))^{1/3}\,. \tag{10.32}$$

10.4.5 The Kohn-Sham equation

The *Kohn-Sham equation* [108] for the electronic structure of matter is

$$\left(\frac{-\hbar^2\nabla^2}{2m} + \mathcal{V}_N(\mathbf{r}) + \mathcal{V}_H(\mathbf{r}) + \mathcal{V}_{xc}[\rho(\mathbf{r})]\right)\phi_i(\mathbf{r}) = E_i\phi_i(\mathbf{r})\,. \tag{10.33}$$

This equation is nonlinear and it can be solved either as an optimization problem (minimize energy with respect to wavefunctions) or a nonlinear eigenvalue problem. In the optimization approach an initial wavefunction basis is selected and a gradient-type approach is used to iteratively refine the basis until a minimum energy is reached. In the second approach the Kohn-Sham equation is solved solved "self-consistently". An approximate charge is assumed to estimate the exchange-correlation potential, and this charge is used to determine the Hartree potential from (10.24). The output potential is then carefully mixed with the previous input potential (s) and the result is inserted in the Kohn-Sham equation and the total charge density determined as in (10.22). The "output" charge density is used to construct new exchange-correlation and Hartree potentials. The process is repeated until the input and output charge densities or potentials are identical to within some prescribed tolerance.

Due to its ease of implementation and overall accuracy, the Local Density Approximation (LDA) mentioned earlier is a popular choice for describing the electronic structure of matter. Recent developments have included so-called gradient corrections to the local density approximation. In this approach, the exchange-correlation energy depends on the local density and the gradient of the density. This approach is called the generalized gradient approximation (GGA) [155].

10.4.6 Pseudopotentials

A major difficulty in solving the eigenvalue problem arising from the Kohn-Sham equation is the length and energy scales involved. The inner (core) electrons are highly localized and tightly bound compared to the outer (valence electrons). A simple basis function approach is frequently ineffectual. For example, a plane wave basis (see next section) might require 10^6 waves to represent converged wave functions for a core electron, whereas only 10^2 waves are required for a valence electron[24]. In addition, the potential is singular near the core and this

cause difficulties in discretizing the Hamiltonian and in representing the wavefunctions. The use of pseudopotentials overcomes these problems by removing the core states from the problem and replacing the all-electron potential by one that replicates only the chemically active, valence electron states[24]. It is well-known that the physical properties of solids depend essentially on the valence electrons rather than on the core electrons, *e.g.*, the Periodic Table is based on this premise. By construction, the pseudopotential reproduces exactly the valence state properties such as the eigenvalue spectrum and the charge density outside the ion core.

The cores are composed of nuclei and inert inner electrons. Within this model many of the complexities of an all-electron calculation are avoided. A group IV solid such as C with 6 electrons is treated in a similar fashion to Pb with 82 electrons since both elements have 4 valence electrons.

The pseudopotential approximation takes advantage of this observation by removing the core electrons and introducing a potential that is weaker at the core, which will make the (pseudo)wave functions behave like the all-electron wave function near the locations of the valence electrons, *i.e.*, beyond a certain radius r_c away from the core region. The valence wave functions often oscillate rapidly in the core region because of the orthogonality requirement of the valence states to the core states. This oscillatory or nodal structure of the wave functions corresponds to the high kinetic energy in this region.

Pseudopotential calculations center on the accuracy of the valence electron wave function in the spatial region away from the core, *i.e.*, within the "chemically active" bonding region. The smoothly-varying pseudo wave function should be identical with the appropriate all-electron wave function in the bonding regions. The idea of pseudopotentials goes back to Fermi [58] who in 1934 introduced a similar construction to account for the shift in the wave functions of high-lying states of alkali atoms subject to perturbations from foreign atoms.

10.5 Stability of Dynamical Systems

Consider a dynamical system governed by the differential equation

$$\frac{dy}{dt} = F(y) \tag{10.34}$$

where $y \in \mathbb{R}^n$ is some vector-valued function of t and F is a function from $\mathbb{R}^n$ to itself. We will assume that the system is time autonomous in that the variable t does not appear in the right hand side of (10.34). Note that F can be a complicated partial differential operator and is usually nonlinear.

The stability of a nonlinear system that satisfies the equation $\dot{y} = F(y)$ is usually studied in terms of its steady state solution. The steady state solution $\bar{y}$ is, by definition, the limit of $y(t)$ as t tends to infinity. This limit, when it exists, will clearly depend on the initial conditions of the differential equation. The solution $\bar{y}$ can be found by solving the steady-state equation $F(y) = 0$ because the variation of y with respect to time will tend to zero at infinity. A system governed by

equation (10.34) is said to be locally stable if there exists an ϵ such that

$$\|y(t) - \bar{y}\| \to 0 \ , \ \text{as} \ t \to \infty$$

whenever $\|y(0) - \bar{y}\| \leq \epsilon$. For obvious reasons, it is said that the steady state solution is attracting. The important result on the stability of dynamical systems, is that in most cases the stability of the dynamical system can be determined by its linear stability, i.e., by the stability of the linear approximation of F at $\bar{y}$. In other words the system is stable if all the eigenvalues of the Jacobian matrix

$$J = \left\{ \frac{\partial f_i(\bar{y})}{\partial x_j} \right\}_{i,j=1,\ldots,n}$$

have negative real parts and unstable if at least one eigenvalue has a positive real part. If some eigenvalues of J lie on the imaginary axis, then the stability of the system cannot be determined by its linear stability, see [83]. In this case the system may or may not be stable depending on the initial condition among other things.

It is often the case that Jacobian matrices are very large nonsymmetric and sparse such as for example when F originates from the discretization of a partial differential operator. This is also the case when simulating electrical power systems, since the dimension of the Jacobian matrices will be equal to the number of nodes in the network multiplied by the number of unknowns at each node, which is usually four.

10.6 Bifurcation Analysis

The behavior of phenomena arising in many applications can be modeled by a parameter dependent differential equation of the form

$$\frac{dy}{dt} = F(y, \alpha) \tag{10.35}$$

where y is a vector valued function and α is typically a real parameter. There are several problems of interest when dealing with an equation of the form (10.35). A primary concern in some applications is to determine how stability properties of the system will change as the parameter α varies. For example α might represent a mass that is put on top of a structure to study its resistance to stress. When this mass increases to reach a critical value the structure will collapse. Another important application is when controlling the so-called panel flutter that causes wings of airplanes to disrupt after strong vibrations. Here the bifurcation parameter is the magnitude of the velocity of air. Christodoulou and Scriven solved a rather challenging problem involving bifurcation and stability analysis in fluid flow in [27]. Bifurcation theory comprises a set of analytical and numerical tools used to analyze the change of solution behavior as α varies and part of the spectrum of the Jacobian moves from the left half plane (stable plane) to the right half (unstable) plane.

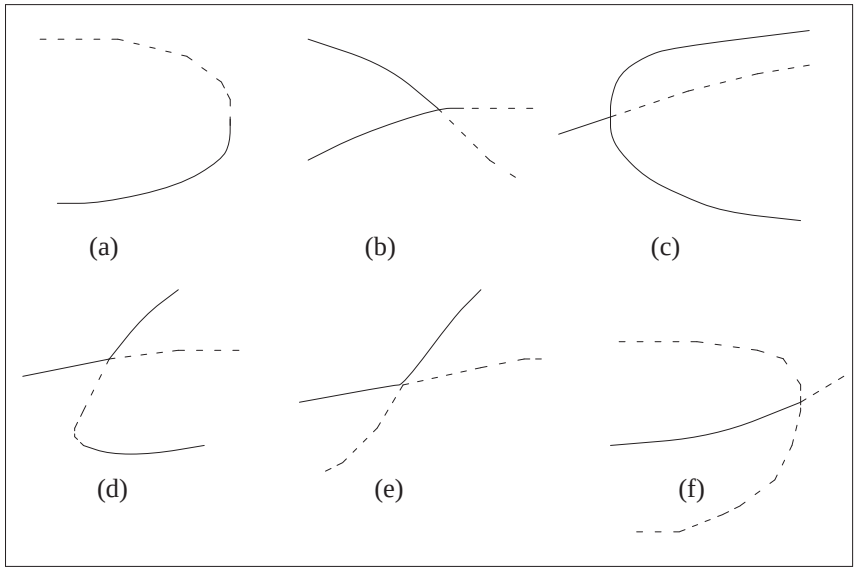


Figure 10.5: Bifurcation patterns. Stable branches solid lines, unstable branches dashed lines.

A typical situation is when one *real* eigenvalue passes from the left plane to the right half plane. Thus, the Jacobian becomes singular in between. This could correspond to either a 'turning 'point or a 'real bifurcation 'point. The change of behavior of the solution can happen in several different ways as is illustrated in Figure 4. Often bifurcation analysis amounts to the detection of all such points. This is done by a marching procedure along one branch until crossing the primary bifurcation point and taking all possible paths from there to detect the secondary bifurcation points etc..

An interesting case is when a pair of complex imaginary eigenvalues cross the imaginary axis. This is referred to as Hopf bifurcation. Then at the critical value of α where the crossing occurs, the system admits a periodic solution. Also, the trajectory of y, sometimes referred to as the phase curve in mechanics, forms a closed curve in the y plane referred to as the phase plane (this can be easily seen for the case $n = 2$ by using the parameter t to represent the curve).

10.7 Chemical Reactions

An increasing number of matrix eigenvalue problems arise from the numerical simulation of chemical reactions. An interesting class of such reactions are those where periodic reactions occur 'spontaneously 'and trigger a wave like regime. A well-known such example is the Belousov-Zhabotinski reaction which is modeled

by what is referred to as the Brusselator model. The model assumes that the reaction takes place in a tube of length one. The space variable is denoted by r, and the time variable by t. There are two chemical components reacting with one another. Their concentrations which are denoted by $x(t,r)$ and $y(t,r)$ satisfy the coupled partial differential equations

$$\begin{aligned} \frac{\partial x}{\partial t} &= \frac{D_1}{L}\frac{\partial^2 x}{\partial r^2} + A - B - (B+1)x + x^2 y \\ \frac{\partial y}{\partial t} &= \frac{D_2}{L}\frac{\partial^2 y}{\partial r^2} + Bx - x^2 y \end{aligned}$$

with the initial conditions,

$$x(0,r) = x_0(r), \quad y(0,r) = y_0(r), \quad 0 \leq r \leq 1$$

and the boundary conditions

$$x(t,0) = x(t,1) = A, \quad y(t,0) = y(t,1) = \frac{B}{A}\,.$$

A trivial stationary solution to the above system is $\bar{x} = A, \bar{y} = B/A$. The linear stability of the above system at the stationary solution can be studied by examining the eigenvalues of the Jacobian of the transformation on the right-hand-side of the above equations. This Jacobian can be represented in the form

$$J = \begin{pmatrix} \frac{D_1}{L}\frac{\partial^2}{\partial_r^2} - (B+1) + 2xy & x^2 \\ B - 2xy & \frac{D_2}{L}\frac{\partial^2}{\partial_r^2} - x^2 \end{pmatrix}\,.$$

This leads to a sparse eigenvalue problem after discretization. In fact the problem addressed by chemists is a bifurcation problem, in that they are interested in the critical value of L at which the onset of periodic behavior is triggered. This corresponds to a pair of purely imaginary eigenvalues of the Jacobian crossing the imaginary axis.

10.8 Macro-economics

We consider an economy which consists of n different sectors each producing one good and each good produced by one sector. We denote by a_{ij} the quantity of good number i that is necessary to produce one unit of good number j. This defines the coefficient matrix A known as the matrix of technical coefficients. For a given production $(x)_{i=1,\ldots,n}$, the vector Ax will represent the quantities needed for this production , and therefore $x - Ax$ will be the net production. This is roughly Leontiev's linear model of production.

Next, we would like to take into account labor and salary in the model. In order to produce a unit quantity of good j, the sector j employs w_j workers and we define the vector of workers $w = [w_1, w_2, \ldots, w_n]^T$. Let us assume that the salaries are the same in all sectors and that they are entirely used for consumption,

each worker consuming the quantity d_i of good number i. We define again the vector $d = [d_1, d_2, \ldots, d_n]^T$. The total consumption of item i needed to produce one unit of item j becomes

$$a_{ij} + w_j d_i \ .$$

This defines the so-called *socio-technical* matrix $B = A + w^T d$.

The additional assumptions on the model are that the needs of the workers are independent of their sector, and that there exists a pricing system that makes every sector profitable. By pricing system or strategy, we mean a vector $p = (p_i)_{i=1,\ldots,n}$ of the prices p_i of all the goods. The questions are

1) Does there exist a pricing strategy that will ensure a profit rate equal for all sectors? (balanced profitability)

2) Does there exist a production structure x that ensures the same growth rate τ to each sector? (balanced growth).

The answer is provided by the following theorem.

Theorem 10.1 *If the matrix B is irreducible there esists a pricing strategy p, a production structure x and a growth rate $r = \tau$ that ensure balanced profitability and balanced growth and such that*

$$B^T p = \frac{1}{1+r} p \ , \ Bx = \frac{1}{1+\tau} x.$$

In other words the desired pricing system and production structure are left and right eigenvectors of the matrix B respectively. The proof is a simple exercise that uses the Perron-Frobenius theorem. Notice that the profit rate r is equal to the growth rate τ; this is referred to as the golden rule of growth.

10.9 Markov Chain Models

A discrete state, discrete time Markov chain is a random process with a finite (or countable) number of possible states taking place at countable times $t_1, t_2, \ldots, t_k \ldots$, and such that the probability of an event depends only on the state of the system at the previous time. In what follows, both times and states will be numbered by natural integers. Thus, the conditional probability that the system be in state j at time k, knowing that it was under state j_1 at time 1, state j_2, at state 2 etc.., state $j_k - 1$ at time $k-1$ only depends on its state $j_k - 1$ at the time $k-1$, or

$$P(X_k = j \mid X_1 = j_1, X_2 = j_2, \ldots, X_{k-1} = j_{k-1}) \\ = P(X_k = j \mid X_{k-1} = j_{k-1})$$

where $P(E)$ is the probability of the event E and X is a random variable.

A system can evolve from a state to another by passing through different transitions. For example, if we record at every minute the number of people waiting

for the $7am$ bus at a given bus-stop, this number will pass from 0 at, say, instant 0 corresponding to $6:45$ am to say 10 at instant 15 corresponding to 7 am. Moreover, at any given time between instant 0 and 15, the probability of another passenger coming, i.e., of the number of passengers increasing by one at that instant, only depends on the number of persons already waiting at the bus-stop.

If we assume that there are N possible states, we can define at each instant k, an $N \times N$ matrix $P^{(k)}$, called transition probability matrix, whose entries $p_{ij}^{(k)}$ are the probabilities that a system passes from state i to state j at time k, i.e.,

$$p_{ij}^{(k)} = P(X_k = j | X_{k-1} = i)$$

The matrix $P^{(k)}$ is such that its entries are nonnegative, and the row sums are equal to one. Such matrices are called *stochastic*. One of the main problems associated with Markov chains is to determine the probabilities of every possible state of the system after a very long period of time.

The most elementary question that one faces when studying such models is: how is the system likely to evolve given that it has an initial probability distribution $q^{(0)} = (q_1^{(0)}, q_2^{(0)}, \ldots, q_N^{(0)})$? It is easy to see that at the first time $q^{(1)} = q^{(0)}P^{(0)}$, and more generally

$$q^{(k)} = q^{(k-1)}P^{(k-1)}.$$

Therefore,

$$q^{(k)} = q^{(0)}P^{(0)}P^{(1)} \ldots P^{(k-1)}P^{(k)}.$$

A homogeneous systems is one whose transition probability matrix $P^{(k)}$ is independent of time. If we assume that the system is homogeneous then we have

$$q^{(k)} = q^{(k-1)}P \tag{10.36}$$

and as a result if there is a stationary distribution $\pi = \lim q^{(k)}$ it must satisfy the equality $\pi = \pi P$. In other words π is a left eigenvector of P associated with the eigenvalue unity. Conversely, one might ask what are the conditions under which there is a stationary distribution.

All the eigenvalues of P do not exceed its 1-norm which is one because P is nonnegative. Therefore if we assume that P is irreducible then by the Perron-Frobenius theorem, one is the eigenvalue of largest modulus, and there is a corresponding left eigenvector π with positive entries. If we scale this eigenvector so that $\|\pi\|_1 = 1$ then this eigenvector will be a stationary probability distribution. Unless there is only one eigenvalue with modulus one, it is not true that a limit of q_k defined by (10.36) always exists. In case there is only eigenvalue of P of modulus one, then q_k will converge to π under mild conditions on the initial probability distributions q_0.

Markov chain techniques are very often used to analyze queuing networks and to study the performance of computer systems.

PROBLEMS

P-10.1 Generalize the model problems of Section 10.2 involving masses and springs to an arbitrary number of masses.

P-10.2 Compute the exact eigenvalues (analytically) of the matrix obtained from discretizing the Chemical reaction model problem in Section 10.7. Use the parameters listed in Chapter II for the example.

P-10.3 Show that when $F(t) = F_0 \cos \omega t$ then a particular solution to (10.1) is given by

$$\frac{F_0}{(k - m\omega^2)^2 + c^2\omega^2} \left[(k - m\omega^2) \cos \omega t + c\omega \sin \omega t\right].$$

Show that (10.3) is an alternative expression of this solution.

NOTES AND REFERENCES. At the time of the first edition of this book I stated that "Many of the emerging applications of eigenvalue techniques are related to fluid dynamics and bifurcation theory [19, 87, 98, 128, 130, 99, 186, 215] aero-elasticity [47, 48, 66, 129, 84, 85, 185], chemical engineering [28, 27, 160, 88, 161] and economics [38]." Recent interest of numerical analysts has turned to applications from two challenging and distinct areas: nanosciences (electronic structure calculations, see Sec. 10.4) and information sciences (machine learning, data analysis). Problems originating from quantum mechanical calculations are challenging not only because of the large sizes of the matrices encountered but also because the number of eigenvalues to be computed can be very large. Section 10.4 is an abbreviated version of the survey article [181]. A prototypical application in the second category is that of the google page-rank problem, see for example, [102, 74, 15]. For a description of linear algebra method for information retrieval (IR), see [9]. These application typically lead to singular value problems instead of eigenvalue problems. The eigenvalue problems which are encountered in the specific problem of dimension reduction are surveyed in [109]. The Lanczos algorithm can play a significant role in reducing the cost of these techniques as the required accuracy is typically not high, see for example [25, 10] for an illustration. ■

Bibliography

[1] W. E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

[2] O. Axelsson and V. A. Barker. *Finite Element Solution of Boundary Value Problems*. Academic Press, Orlando, FL, 1984.

[3] K. J. Bathé and E. L. Wilson. *Numerical Methods in Finite Elements Analysis*. Prentice Hall, Englewood Cliffs, New Jersey, 1976.

[4] F. L. Bauer. Das verfahren der treppeniteration und verwandte verfahren zur losung algebraischer eigenwertprobleme. *ZAMP*, 8:214–235, 1957.

[5] K. Bekas and Y. Saad. Computation of smallest eigenvalues using spectral Schur complements. *SIAM Journal on Scientific Computing*, 27(2):458–481, 2005.

[6] M. Bellalij, Y. Saad, and H. Sadok. On the convergence of the Arnoldi process for eigenvalue problems. Technical Report umsi-2007-12, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2007. In press.

[7] M. Bellalij, Y. Saad, and H. Sadok. On the convergence of the Arnoldi process for eigenvalue problems. *SIAM Journal on Numerical Analysis*, 48(2):393–407, 2010.

[8] J. K. Bennighof and R. B. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM. J. Sci. Comput.*, 25(6):2084–2106, 2004.

[9] M. Berry and M. Browne. *Understanding search engines: Mathematical Modeling and Text Retrieval. 2nd edition*. SIAM, 2005.

[10] Katarina Blom and Axel Ruhe. A krylov subspace method for information retrieval. *SIAM Journal on Matrix Analysis and Applications*, 26(2):566–582, 2005.

[11] D. Boley and G. H. Golub. The Lanczos-Arnoldi algorithm and controllability. *Systems and Control Letters*, 4:317–324, 1987.

[12] D. Boley, R. Maier, and J. Kim. A parallel QR algorithm for the nonsymmetric eigenvalue problem. *Computer Physics Communications*, 53:61–70, 1989.

[13] D. L. Boley, D. G. Truhlar, R. E. Wyatt, and L. E. Collins. *Practical Iterative Methods for Large Scale Computations*. North Holland, Amsterdam, 1989. Proceedings of Minnesota Supercomputer Institute Workshop.

[14] M. Bollhoefer and Y. Notay. JADAMILU: a software code for computing selected eigenvalues of large sparse symmetric matrices. *Computer Physics Communications*, 177:951–964, 2007.

[15] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *The Eleventh International World Wide Web Conference*, pages 415–429, 2001.

[16] M. Borri and P. Mantegazza. Efficient solution of quadratic eigenproblems arising in dynamic analysis of structures. *Comp. Meth. Appl. Mech. and Engng*, 12:19–31, 1977.

[17] M. Braun. *Differential equations and their applications*. Springer-Verlag, New York, 1983. Applied mathematical sciences series, Number 15.

[18] C. Brezinski. *Padé Type Approximation and General Orthogonal Polynomials*. Birkhäuser-Verlag, Basel-Boston-Stuttgart, 1980.

[19] E. Carnoy and M. Geradin. On the practical use of the Lanczos algorithm in finite element applications to vibration and bifurcation problems. In Axel Ruhe, editor, *Proceedings of the Conference on Matrix Pencils, Lulea, Sweden, March 1982*, pages 156–176, New York, 1982. University of Umea, Springer Verlag.

[20] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45:566–569, 1980.

[21] T. F. Chan and H. B. Keller. Arclength continuation and multi-grid techniques for nonlinear eigenvalue problems. *SIAM Journal on Scientific and Statistical Computing*, 3:173–194, 1982.

[22] F. Chatelin. *Spectral Approximation of Linear Operators*. Academic Press, New York, 1984.

[23] F. Chatelin. *Valeurs propres de matrices*. Masson, Paris, 1988.

[24] J. R. Chelikowsky and M. L. Cohen. Pseudopotentials for semiconductors. In T. S. Moss and P. T. Landsberg, editors, *Handbook of Semiconductors*. Elsevier, Amsterdam, 2nd edition, 1992.

[25] J. Chen and Y. Saad. Lanczos vectors versus singular vectors for effective dimension reduction. *IEEE Trans. on Knowledge and Data Engineering*, 21(9):1091–1103, 2009.

[26] C. C. Cheney. *Introduction to Approximation Theory*. McGraw Hill, NY, 1966.

[27] K. N. Christodoulou and L. E. Scriven. Finding leading modes of a viscous free surface flow: An asymmetric generalized eigenproblem. *J. Scient. Comput.*, 3:355–406, 1988.

[28] K. N. Christodoulou and L. E. Scriven. Operability limits of free surface flow systems by solving nonlinear eigenvalue problems. Technical report, University of Minnesota Supercomputer Institute, Minneapolis, MN, 1988.

[29] A. Clayton. Further results on polynomials having least maximum modulus over an ellipse in the complex plane. Technical Report AEEW-7348, UKAEA, Harewell-UK, 1963.

[30] A. K. Cline, G. H. Golub, and G. W. Platzman. Calculation of normal modes of oceans using a Lanczos method. In J. R. Bunch and D. C. Rose, editors, *Sparse Matrix Computations*, pages 409–426. Academic Press, 1976.

[31] M. Clint and A. Jennings. The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration method. *Journal of the Institute of Mathematics and its Applications*, 8:111–121, 1971.

[32] R. R. Craig, Jr. and M. C. C. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6:1313–1319, 1968.

[33] J. Cullum, W. Kerner, and R. Willoughby. A generalized nonsymmetric Lanczos procedure. *Computer Physics Communications*, 53, 1989.

[34] J. Cullum and R. Willoughby. A Lanczos procedure for the modal analysis of very large nonsymmetric matrices. In *Proceedings of the 23rd Conference on Decision and Control, Las Vegas*, 1984.

[35] J. Cullum and R. Willoughby. A practical procedure for computing eigenvalues of large sparse nonsymmetric matrices. Technical Report RC 10988 (49366), IBM, T. J. Watson Research center, Yorktown heights, NY, 1985.

[36] J. Cullum and R. A. Willoughby. *Lanczos algorithms for large symmetric eigenvalue computations*. Volumes 1 and 2. Birkhäuser, Boston, 1985.

[37] J. Cullum and R. A. Willoughby. *Large Scale Eigenvalue Problems*. North-Holland, 1986. Mathematics Studies series, Number 127.

[38] F. d' Almeida. Numerical study of dynamic stability of macroeconomical models- software for MODULECO. Technical report, INPG- University of Grenoble, Grenoble-France, 1980. Dissertation (French).

[39] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Math. Comput*, 30:772–795, 1976.

[40] B. N. Datta. *Numerical Linear Algebra and Applications, second edition*. SIAM, Philadelphia, PA, 2010.

[41] E. R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices. *Journal of Computational Physics*, 17:87–94, 1975.

[42] P. J. Davis. *Interpolation and Approximation*. Blaisdell, Waltham, MA, 1963.

[43] T. A. Davis. *Direct methods for sparse linear systems*. SIAM, Philadelphia, PA, 2006.

[44] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[45] J. J. Dongarra, I. S. Duff, D. Sorensen, and H. A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, Philadelphia, PA, 1991.

[46] J.J. Dongarra, J.R. Bunch, C.B. Moler, and G. W. Stewart. *LINPACK User's Guide*. SIAM, Philadelphia, PA, 1979.

[47] E. H. Dowell. Nonlinear oscillations of a fluttering plate, II. *AIAA*, 5:1856–1862, 1967.

[48] E. H. Dowell. *Aeroelasticity of Plates of Shells*. Nordhoff Internat., Leyden, 1975.

[49] I. S. Duff. A survey of sparse matrix research. In *Proceedings of the IEEE, 65*, pages 500–535, New York, 1977. Prentice Hall.

[50] I. S. Duff. Ma28 – a set of FORTRAN subroutines for sparse unsymmetric matrices. Technical Report R8730, A. E. R. E., Harewell, England, 1978.

[51] I. S. Duff. A survey of sparse matrix software. In W. R. Cowell, editor, *Sources and development of Mathematical software*. Prentice Hall, New York, 1982.

[52] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Clarendon Press, Oxford, 1986.

[53] I. S. Duff, R. G. Grimes, and J. G. Lewis. Sparse matrix test problems. *ACM Transactions on Mathematical Software*, 15:1–14, 1989.

[54] H. C. Elman, Y. Saad, and P. Saylor. A hybrid Chebyshev Krylov subspace algorithm for solving nonsymmetric systems of linear equations. *SIAM Journal on Scientific and Statistical Computing*, 7:840–855, 1986.

[55] I. Erdelyi. An iterative least squares algorithm suitable for computing partial eigensystems. *SIAM J. on Numer. Anal*, B 3. 2, 1965.

[56] T. Ericsson and A. Ruhe. The spectral transformation Lanczos method in the numerical solution of large sparse generalized symmetric eigenvalue problems. *Mathematics of Computation*, 35:1251–1268, 1980.

[57] L. E. Eriksson and A. Rizzi. Analysis by computer of the convergence of discrete approximations to the euler equations. In *Proceedings of the 1983 AIAA conference, Denver 1983*, pages 407–442, Denver, 1983. AIAA.

[58] E. Fermi. Tentativo di una teoria dei raggi beta. *Nuovo Cimento*, II:1–19, 1934.

[59] B. Fischer and R. W. Freund. On the constrained Chebyshev approximation problem on ellipses. *Journal of Approximation Theory*, 62:297–315, 1990.

[60] B. Fischer and R. W. Freund. Chebyshev polynomials are not always optimal. *Journal of Approximation Theory*, 65:261–272, 1991.

[61] D. A. Flanders and G. Shortley. Numerical determination of fundamental modes. *J. Appl. Phy.*, 21:1328–1322, 1950.

[62] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM J. Sci. Comput.*, 20(1):94–125, 1998.

[63] J. G. F. Francis. The QR transformations, parts i and ii. *Computer J.*, 4:362–363, and 332–345, 1961-1962.

[64] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the Look-Ahead Lanczos algorithm for non-Hermitian matrices, Part I. Technical Report 90-11, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1990.

[65] R. W. Freund and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, Part II. Technical Report 90-11, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1990.

[66] Y. C. Fung. *Introdiction to the Theory of Aeroelasticity*. John Wiley, New York, 1955.

[67] E. Gallopoulos and Y. Saad. On the parallel solution of parabolic equations. In R. De Groot, editor, *Proceedings of the International Conference on Supercomputing 1989, Heraklion, Crete, June 5-9, 1989*. ACM press, 1989.

[68] G. Gamov. *Thirty Years That Shook Physics: The Story of Quantum Theory*. Dover, 1966.

[69] F. R. Gantmacher. *The Theory of Matrices*. Chelsea, New York, 1959.

[70] W. Gautschi. On generating orthogonal polynomials. *SIAM Journal on Scientific and Statistical Computing*, 3:289–317, 1982.

[71] J. A. George and J. W-H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[72] M. Geradin. On the Lanczos method for solving large structural eigenvalue problems. *Z. Angew. Math. Mech.*, 59:T127–T129, 1979.

[73] S. Gerschgorin. On bounding the eigenvalues of a matrix (in german). *Izv. Akad. Nauk. SSSR Otd Mat. Estest.*, 1:749–754, 1931.

[74] David Gleich, Leonid Zhukov, and Pavel Berkhin. Fast parallel pagerank: A linear system approach. In *The Fourteenth International World Wide Web Conference, New York, NY*. ACM Press, 2005.

[75] S. K. Godunov and G. P. Propkopov. A method of minimal iteration for evaluating the eigenvalues of an elliptic operator. *Zh. Vichsl. Mat. Mat. Fiz.*, 10:1180–1190, 1970.

[76] Sergey K. Godunov. *Modern aspects of linear algebra*. Translations of mathematical monographs, volume 175. Americal Mathematical Society, 1998.

[77] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[78] G. H. Golub and D. P. O'Leary. Some history of the conjugate gradient and Lanczos algorithms: 1948-1976. *SIAM review*, 31:50–102, 1989.

[79] G. H. Golub and R. Underwood. The block Lanczos method for computing eigenvalues. In J. R. Rice, editor, *Mathematical Software III*, pages 361–377. Academic press, New York, 1977.

[80] G. H. Golub, R. Underwood, and J. H. Wilkinson. The Lanczos algorithm for the symmetric $Ax = \lambda Bx$ problem. Technical Report STAN-CS-72-720, Stanford University, Stanford, California, 1972.

[81] G. H. Golub and J. H. Wilkinson. Ill-conditioned eigensystems and the computation of the Jordan canonical form. *SIAM review*, 18:578–619, 1976.

[82] W. B. Gragg. Matrix interpretation and applications of continued fraction algorithm. *Rocky Mountain J. of Math.*, 4:213–225, 1974.

[83] J. Guckenheimer and P. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields*. Springer Verlag, New York, 1983.

[84] K. K. Gupta. Eigensolution of damped structural systems. *Internat. J. Num. Meth. Engng.*, 8:877–911, 1974.

[85] K. K. Gupta. On a numerical solution of the supersonic panel flutter eigenproblem. *Internat. J. Num. Meth. Engng.*, 10:637–645, 1976.

[86] P. R. Halmos. *Finite-Dimensional Vector Spaces*. Springer Verlag, New York, 1958.

[87] J. Heyvaerts, J. M. Lasry, M. Schatzman, and P. Witomski. Solar flares: A nonlinear problem in an unbounded domain. In C. Bardos, J. M. Lasry, and M. Schatzman, editors, *Bifurcation and nonlinear eigenvalue problems, Proceedings*, pages 160–192, New York, 1978. Springer Verlag. Lecture notes in Mathematics Series.

[88] H. Hlavacek and H. Hofmann. Modeling of chemical reactors XVI. *Chemical Eng. Sci.*, 25:1517–1526, 1970.

[89] D. H. Hodges. Aeromechanical stability of analysis for bearingless rotor helicopters. *J. Amer. Helicopter Soc.*, 24:2–9, 1979.

[90] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, 1964.

[91] A. S. Householder. *Theory of Matrices in Numerical Analysis*. Blaisdell Pub. Co., Johnson, CO, 1964.

[92] W. C. Hurty. Vibrations of structural systems by component-mode synthesis. *Journal of the Engineering Mechanics Division, ASCE*, 86:51–69, 1960.

[93] I. Ipsen and Y. Saad. The impact of parallel architectures on the solution of eigenvalue problems. In J. Cullum and R. A. Willoughby, editors, *Large Scale Eigenvalue Problems*, Amsterdam, The Netherlands, 1986. North-Holland, Vol. 127 Mathematics Studies Series.

[94] A. Jennings. *Matrix Computations for Engineers and Scientists*. Wiley, New York, 1977.

[95] A. Jennings. Eigenvalue methods and the analysis of structural vibrations. In I. S. Duff, editor, *Sparse Matrices and their Uses*, pages 109–138. Academic Press, New York, 1981.

[96] A. Jennings and W. J. Stewart. Simultaneous iteration for partial eigensolution of real matrices. *J. Math. Inst. Appl.*, 15:351–361, 1980.

[97] A. Jennings and W. J. Stewart. A simultaneous iteration algorithm for real matrices. *ACM, Trans. of Math. Software*, 7:184–198, 1981.

[98] A. Jepson. *Numerical Hopf Bifurcation*. PhD thesis, Cal. Inst. Tech., Pasadena, CA., 1982.

[99] D. D. Joseph and D. H. Sattinger. Bifurcating time periodic solutions and their stability. *Arch. Rat. Mech. Anal,*, 45:79–109, 1972.

[100] W. Kahan and B. N. Parlett. How far should you go with the Lanczos process? In J. R. Bunch and D. C. Rose, editors, *Sparse Matrix Computations*, pages 131–144. Academic Press, 1976.

[101] W. Kahan, B. N. Parlett, and E. Jiang. Residual bounds on approximate eigensystems of non-normal matrices. *SIAM Journal on Numerical Analysis*, 19:470–484, 1982.

[102] Sepandar D. Kamvar, Taher H. Haveliwala, and Gene H. Golub. Adaptive methods for the computation of pagerank. *Linear Algebra and its Applications*, 386:51–65, 2004.

[103] S. Kaniel. Estimates for some computational techniques in linear algebra. *Mathematics of Computation*, 20:369–378, 1966.

[104] T. Kato. On the upper and lower bounds of eigenvalues. *J. Phys. Soc. Japan*, 4:334–339, 1949.

[105] T. Kato. *Perturbation Theory for Linear Operators*. Springer–Verlag, New–York, 1976.

[106] L. Kleinrock. *Queueing Systems, vol. 2: Computer Applications*. John Wiley and Sons, New York, London, 1976.

[107] A. V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.*, 23(2):517–541, 2001.

[108] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

[109] E. Kokiopoulou, J. Chen, and Y. Saad. Trace optimization and eigenproblems in dimension reduction methods. Technical Report umsi-2009-31, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2009. To appear NLAA.

[110] M. A. Krasnoselskii et al. *Approximate Solutions of Operator Equations*. Wolters-Nordhoff, Groningen, 1972.

[111] A. N. Krylov. On the numerical solution of equations whose solution determine the frequency of small vibrations of material systems (in russian). *Izv. Akad. Nauk. SSSR Otd Mat. Estest.*, 1:491–539, 1931.

[112] C. Lanczos. 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*, 45:255–282, 1950.

[113] C. Lanczos. Chebyshev polynomials in the solution of large-scale linear systems. In *Proceedings of the ACM*, pages 124–133, 1952.

[114] C. Lanczos. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, 49:33–53, 1952.

[115] C. Lanczos. Iterative solution of large-scale linear systems. *J. Soc. Indust. Appl. Math*, 6:91–109, 1958.

[116] C. Lanczos. *Applied Analysis*. Dover, New York, 1988.

[117] R. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, 17:789–821, 1996.

[118] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK USERS GUIDE: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, 1998. Available at `http//www.caam.rice.edu/software/ARPACK/`.

[119] J. G. Lewis and H. D. Simon. Numerical experience with the spectral transformation Lanczos. Technical Report MM-TR-16, Boeing Computer Services, Seattle, WA, 1984.

[120] S. S. Lo, B. Philippe, and A. Sameh. A multiprocessor algorithm for symmetric tridiagonal eigenvaluie problem. *SIAM J. Stat. Sci. Comput.*, 8:s155–s165, 1987.

[121] D. E. Longsine and S. F. Mc Cormick. Simultaneous Rayleigh quotient minimization methods for $Ax = \lambda Bx$. *Linear Algebra and its Applications*, 34:195–234, 1980.

[122] G. G. Lorentz. *Approximation of functions*. Holt, Rinehart - Winston, New York, 1966.

[123] S. Lundqvist and N. H. March, editors. *Theory of the Inhomogeneous Electron Gas*. Plenum, 1983.

[124] O. Madelung. *Introduction to Solid State Theory*. Springer-Verlag, 1996.

[125] T. A. Manteuffel. An iterative method for solving nonsymmetric linear systems with dynamic estimation of parameters. Technical Report UIUCDCS-75-758, University of Illinois at Urbana-Champaign, Urbana, Ill., 1975. Ph. D. dissertation.

[126] T. A. Manteuffel. The Tchebychev iteration for nonsymmetric linear systems. *Numerische Mathematik*, 28:307–327, 1977.

[127] T. A. Manteuffel. Adaptive procedure for estimation of parameter for the nonsymmetric Tchebychev iteration. *Numerische Mathematik*, 28:187–208, 1978.

[128] J. E. Marsden and M. Mc Cracken. *The Hopf Bifurcation and its Applications*. Springer Verlag, New York, 1976.

[129] Y. Matsuzaki and Y. C. Fung. Unsteady fluid dynamic forces on a simply supported circular cylinder of finite length conveying a flow, with applications to stability. *Journal of Sound and Vibrations*, 54:317–330, 1977.

[130] R. K. Mehra and J. V. Caroll. Bifurcation analysis of aircraft high angle-of-attack flight dynamics. In P. J. Holmes, editor, *New Approaches to Nonlinear Problems in Dynamics - Proceedings of the Asilomar Conference Ground, Pacific Grove, California 1979*, pages 127–146. The Engineering Foundation, SIAM, 1980.

[131] R. B. Morgan. On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Mathematics of Computation*, 65:1212–1230, 1996.

[132] R. B. Morgan and D. S. Scott. Generalizations of davidson's method for computing eigenvalues of sparse symmetric matrices. *SIAM Journal on Scientific and Statistical Computing*, 7:817–825, 1986.

[133] R. Natarajan. An Arnoldi-based iterative scheme for nonsymmetric matrix pencils arising in finite element stability problems. *Journal of Computational Physics*, 100:128–142, 1992.

[134] R. Natarajan and A. Acrivos. The instability of the steady flow past spheres and disks. Technical Report RC 18235, IBM Res. div., T. J. Watson Res. ctr, Yorktown Heights, 1992.

[135] R. K. Nesbet. Algorithm for diagonalization of large matrices. *J. Chem. Phys.*, 42:311–312, 1965.

[136] B. Nour-Omid. Applications of the Lanczos algorithm. *Comput. Phys. Comm.*, 53(1-3):153–168, 1989.

[137] B. Nour-Omid, B. N. Parlett, T. Ericsson, and P. S. Jensen. How to implement the spectral transformation. *Math. Comput.*, 48:663–673, 1987.

[138] B. Nour-Omid, B. N. Parlett, and R. Taylor. Lanczos versus subspace iteration for the solution of eigenvalue problems. Technical Report UCB/SESM-81/04, University of California at Berkeley, Dept. of Civil Engineering, Berkeley, California, 1980.

[139] J. Olsen, P. Jørgensen, and J. Simons. Passing the one-billion limit in full configuration-interaction (fci) calculations. *Chemical Physics Letters*, 169:463–472, 1990.

[140] O. Osterby and Z. Zlatev. *Direct Methods for Sparse Matrices*. Springer Verlag, New York, 1983.

[141] C. C. Paige. *The computation of eigenvalues and eigenvectors of very large sparse matrices*. PhD thesis, London University, Institute of Computer Science, London, England, 1971.

[142] C. C. Paige. Practical use of the symmetric Lanczos process with reorthogonalization. *BIT*, 10:183–195, 1971.

[143] C. C. Paige. Computational variants of the Lanczos method for the eigenproblem. *Journal of the Institute of Mathematics and its Applications*, 10:373–381, 1972.

[144] P. C. Papanastasiou. *Numerical analysis of localization phenomena with application to deep boreholes*. PhD thesis, University of Minnesota, Dept. Civil and Mineral Engineering, Minneapolis, MN, 1990.

[145] B. N. Parlett. The Rayleigh quotient iteration and some generalizations for nonnormal matrices. *Math. Comput.*, 28:679–693, 1974.

[146] B. N. Parlett. How to solve $(K - \lambda M)z = 0$ for large K and M. In E. Asbi et al., editor, *Proceedings of the 2nd International Congress on Numerical Methods for Engineering (GAMNI 2)*, pages 97–106, Paris, 1980. Dunod.

[147] B. N. Parlett. The software scene in the extraction of eigenvalues from sparse matrices. *SIAM J. of Sci. Stat. Comput.*, 5(3):590–604, 1984.

[148] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Number 20 in Classics in Applied Mathematics. SIAM, Philadelphia, 1998.

[149] B. N. Parlett and H. C. Chen. Use of an indefinite inner product for computing damped natural modes. Technical Report PAM-435, Center for Pure and Applied Mathematics, University of California at Berkeley, Berkeley, CA, 1988.

[150] B. N. Parlett and B. Nour-Omid. The use of refined error bounds when updating eigenvalues of tridiagonals. *Linear Algebra and its Applications*, 68:179–219, 1985.

[151] B. N. Parlett and J. K. Reid. Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems. *IMA J. Num. Anal.*, 1:135–155, 1981.

[152] B. N. Parlett and Y. Saad. Complex shift and invert strategies for real matrices. *Linear Algebra and its Applications*, 88/89:575–595, 1987.

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

[154] B. N. Parlett, D. R. Taylor, and Z. S. Liu. A look-ahead Lanczos algorithm for nonsymmetric matrices. *Mathematics of Computation*, 44:105–124, 1985.

[155] J. P. Perdew, K. Burke, and Y. Wang. Generalized gradient approximation for the exchange-correlation hole of a many-electron system. *Phys. Rev. B*, 54:16533–16539, 1996.

[156] S. Petiton. Parallel subspace method for non-Hermitian eigenproblems on the connection machine (CM-2). Technical Report YALEU/DCS/RR-859, Yale University, Computer Science dept., New Haven, CT, 1991.

[157] B. Philippe and Y. Saad. Solving large sparse eigenvalue problems on supercomputers. In *Proceedings of International Workshop on Parallel Algorithms and Architectures, Bonas, France Oct. 3-6 1988*, Amsterdam, 1989. North-Holland.

[158] B. Philippe and Y. Saad. On correction equations and domain decomposition for computing invariant subspaces. *Computer Methods in Applied Mechanics and Engineering (special issue devoted to Domain Decomposition)*, 196:1471–1483, 2007.

[159] S. Pissanetzky. *Sparse Matrix Technology*. Academic Press, New York, 1984.

[160] A. B. Poore. A model equation arising in chemical reactor theory. *Arch. Rat. Mech. Anal.*, 52:358–388, 1973.

[161] P. Raschman, M. Kubicek, and M. Maros. Waves in distributed chemical systems: experiments and computations. In P. J. Holmes, editor, *New Approaches to Nonlinear Problems in Dynamics - Proceedings of the Asilomar Conference Ground, Pacific Grove, California 1979*, pages 271–288. The Engineering Foundation, SIAM, 1980.

[162] T. J. Rivlin. *The Chebyshev Polynomials: from Approximation Theory to Algebra and Number Theory*. J. Wiley and Sons, New York, 1990.

[163] A. Ruhe. Numerical methods for the solution of large sparse eigenvalue problems. In V. A. Barker, editor, *Sparse Matrix Techniques, Lect. Notes Math. 572*, pages 130–184, Berlin-Heidelberg-New York, 1976. Springer Verlag.

[164] A. Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Mathematics of Computation*, 33:680–687, 1979.

[165] A. Ruhe. Rational Krylov sequence methods for eigenvalue computations. *Linear Algebra and its Applications*, 58:391–405, 1984.

[166] H. Rutishauser. Theory of gradient methods. In *Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems*, pages 24–49, Basel-Stuttgart, 1959. Institute of Applied Mathematics, Zurich, Birkhäuser Verlag.

[167] H. Rutishauser. Computational aspects of F. L. Bauer's simultaneous iteration method. *Numerische Mathematik*, 13:4–13, 1969.

[168] Y. Saad. On the rates of convergence of the Lanczos and the block Lanczos methods. *SIAM J. Numer. Anal.*, 17:687–706, 1980.

[169] Y. Saad. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and its Applications*, 34:269–295, 1980.

[170] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of Computation*, 37:105–126, 1981.

[171] Y. Saad. Projection methods for solving large sparse eigenvalue problems. In B. Kagstrom and A. Ruhe, editors, *Matrix Pencils, proceedings, Pitea Havsbad*, pages 121–144, Berlin, 1982. University of Umea, Sweden, Springer Verlag. Lecture notes in Math. Series, Number 973.

[172] Y. Saad. Least-squares polynomials in the complex plane with applications to solving sparse nonsymmetric matrix problems. Technical Report 276, Yale University, Computer Science Dept., New Haven, Connecticut, 1983.

[173] Y. Saad. Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems. *Mathematics of Computation*, 42:567–588, 1984.

[174] Y. Saad. Least squares polynomials in the complex plane and their use for solving sparse nonsymmetric linear systems. *SIAM Journal on Numerical Analysis*, 24:155–169, 1987.

[175] Y. Saad. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Aut. Cont.*, 33:290–297, 1988.

[176] Y. Saad. Numerical solution of large nonsymmetric eigenvalue problems. *Computer Physics Communications*, 53:71–90, 1989.

[177] Y. Saad. Numerical solution of large nonsymmetric eigenvalue problems. *Computer Physics Communications*, 53:71–90, 1989.

[178] Y. Saad. Numerical solution of large Lyapunov equations. In M. A. Kaashoek, J. H. van Schuppen, and A. C. Ran, editors, *Signal Processing, Scattering, Operator Theory, and Numerical Methods. Proceedings of the international symposium MTNS-89, vol III*, pages 503–511, Boston, 1990. Birkhauser.

[179] Y. Saad. An overview of Krylov subspace methods with applications to control problems. In M. A. Kaashoek, J. H. van Schuppen, and A. C. Ran, editors, *Signal Processing, Scattering, Operator Theory, and Numerical Methods. Proceedings of the international symposium MTNS-89, vol III*, pages 401–410, Boston, 1990. Birkhauser.

[180] Y. Saad. SPARSKIT: A basic tool kit for sparse matrix computations. Technical Report RIACS-90-20, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffett Field, CA, 1990.

[181] Y. Saad, J. Chelikowsky, and S. Shontz. Numerical methods for electronic structure calculations of materials. *SIAM review*, 52:3–54, 2009.

[182] M. Sadkane. *Analyse Numérique de la Méthode de Davidson*. PhD thesis, Université de Rennes, UER mathematiques et Informatique, Rennes, France, 1989.

[183] M. Said, M. A. Kanesha, M. Balkanski, and Y. Saad. Higher excited states of acceptors in cubic semiconductors. *Physical Review B*, 35(2):687–695, 1988.

[184] A. H. Sameh and J. A. Wisniewski. A trace minimization algorithm for the generalized eigenvalue problem. *SIAM Journal on Numerical Analysis*, 19:1243–1259, 1982.

[185] G. Sander, C. Bon, and M. Geradin. Finite element analysis of supersonic panel flutter. *Internat. J. Num. Meth. Engng.*, 7:379–394, 1973.

[186] D. H. Sattinger. Bifurcation of periodic solutions of the navier stokes equations. *Arch. Rat. Mech. Anal,*, 41:68–80, 1971.

[187] D. S. Scott. *Analysis of the symmetric Lanczos process*. PhD thesis, University of California at Berkeley, Berkeley, CA., 1978.

[188] D. S. Scott. Solving sparse symmetric generalized eigenvalue problems without factorization. *SIAM J. Num. Anal.*, 18:102–110, 1981.

[189] D. S. Scott. The advantages of inverted operators in Rayleigh-Ritz approximations. *SIAM J. on Sci. and Statist. Comput.*, 3:68–75, 1982.

[190] D. S. Scott. Implementing Lanczos-like algorithms on Hypercube architectures. *Computer Physics Communications*, 53:271–282, 1989.

[191] E. Seneta. Computing the stationary distribution for infinite Markov chains. In H. Schneider A. Bjorck, R. J. Plemmons, editor, *Large Scale Matrix Problems*, pages 259–267. Elsevier North Holland, New York, 1981.

[192] A. H. Sherman. Yale Sparse Matrix Package – user's guide. Technical Report UCID-30114, Lawrence Livermore National Lab., Livermore, CA, 1975.

[193] H. D. Simon. *The Lanczos Algorithm for Solving Symmetric Linear Systems*. PhD thesis, University of California at Berkeley, Berkeley, CA., 1982.

[194] H. D. Simon. The Lanczos algorithm with partial reorthogonalization. *Mathematics of Computation*, 42:115–142, 1984.

[195] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler. *Matrix Eigensystem Routines – EISPACK Guide*, volume 6 of *Lecture Notes in Computer Science*. Springer–Verlag, New York, second edition, 1976.

[196] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13:357–385, 1992.

[197] A. Stathopoulos. Preconditioned iterative multimethod eigensolver: Methods and software description. *ACM Transaction on Mathematical Software*, 37(2):21:1–21:30, 2010.

[198] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[199] G. W. Stewart. A bibliographical tour of the large, sparse, generalized eigenvalue problem. In J. R. Bunch and D. C. Rose, editors, *Sparse Matrix Computations*, pages 113–130, New York, 1976. Academic Press.

[200] G. W. Stewart. Simultaneous iteration for computing invariant subspaces of non-Hermitian matrices. *Numerische Mathematik*, 25:123–136, 1976.

[201] G. W. Stewart. SRRIT - a FORTRAN subroutine to calculate the dominant invariant subspaces of a real matrix. Technical Report TR-514, University of Maryland, College Park, MD, 1978.

[202] G. W. Stewart. Perturbation bounds for the definite generalized eigenvalue problem. *Linear Algebra and its Applications*, 23:69–85, 1979.

[203] G. W. Stewart. A generalization of Saad's theorem on rayleigh-ritz approximations. *Linear Algebra and its Applications*, 327:115–119, 1999.

[204] G. W. Stewart. *Matrix Algorithms II: Eigensystems*. SIAM, Philadelphia, 2001.

[205] G. W. Stewart and J.G. Sun. *Matrix perturbation theory*. Academic Press Inc., Boston, MA, 1990.

[206] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, NJ, 1994.

[207] E. L. Stiefel. Kernel polynomials in linear algebra and their applications. *U. S. National Bureau of Standards, Applied Mathematics Series*, 49:1–24, 1958.

[208] G. Strang. *Introduction to Linear Algebra (4th edition)*. Wellesley-Cambridge Press, Wellesley, MA, 2009.

[209] D. Taylor. *Analysis of the look-ahead Lanczos algorithm*. PhD thesis, Department of Computer Science, Berkeley, CA, 1983.

[210] G. Temple. The accuracy of Rayleigh's method of calculating the natural frequencies of vibrating systems. *Proc. Roy. Soc. London Ser. A*, 211:204–224, 1958.

[211] L. N. Trefethen. Pseudospectra of matrices. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis, 1991*, pages 234–246. Longman, 1992.

[212] L. N. Trefethen. Pseudospectra of linear operators. *SIAM review*, pages 383–406, 1997.

[213] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, Philadelpha, PA, 1997.

[214] L. N. Trefethen and M. Embree. *Spectra and pseudospectra: The behavior of nonnormal matrices and operators*. Princetin University Press, Princeton, New Jersey, 2005.

[215] H. Troger. Application of bifurcation theory to the solution of nonlinear stability problems in mechanical engineering. In *Numerical methods for bifurcation problems*, number ISNM 70 in International series on numerical mathematics, pages 525–546. Birkhäuser Verlag, Basel, 1984.

[216] J. S. Vandergraft. Generalized Rayleigh methods with applications to finding eigenvalues of large matrices. *Linear Algebra and its Applications*, 4:353–368, 1971.

[217] R. S. Varga. *Matrix Iterative Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1962.

[218] Y. V. Vorobyev. *Method of Moments in Applied Mathematics*. Gordon and Breach, New York, 1965.

[219] E. L. Wachspress. *Iterative Solution of Elliptic Systems and Applications to the Neutron Equations of Reactor Physics*. Prentice Hall, Englewood Cliffs, NJ, 1966.

[220] H. F. Walker. Implementation of the GMRES method using Householder transformations. *SIAM Journal on Scientific Computing*, 9:152–163, 1988.

[221] O. Widlund. A Lanczos method for a class of non-symmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 15:801–812, 1978.

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

[223] J. H. Wilkinson and C. Reinsch. *Handbook for automatic computation, Vol. II, Linear Algebra*. Springer Verlag, New York, 1971.

[224] H. E. Wrigley. Accelerating the jacobi method for solving simultaneous equations by Chebyshev extrapolation when the eigenvalues of the iteration matrix are complex. *Computer Journal*, 6:169–176, 1963.

[225] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky. Self-consistent-field calculation using Chebyshev-filtered subspace iteration. *J. Comp. Phys.*, 219(1):172–184, 2006.

[226] Yunkai Zhou and Yousef Saad. A Chebyshev-Davidson algorithm for large symmetric eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 29(3):954–971, 2007.

[227] Yunkai Zhou, Yousef Saad, Murilo L. Tiago, and James R. Chelikowsky. Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration. *Phy. rev. E*, 74:066704, 2006.

[228] Z. Zlatev, K. Schaumburg, and J. Wasniewski. A testing scheme for subroutines solving large linear problems. *Computers and Chemistry*, 5:91–100, 1981.

Index