



Notes on Numerical Linear Algebra

Dr. George W Benthien

December 9, 2006

E-mail: Benthien@cox.net

Contents

| | |
|--------------------------------------------------------------------|-----------|
| Preface | 5 |
| 1 Mathematical Preliminaries | 6 |
| 1.1 Matrices and Vectors | 6 |
| 1.2 Vector Spaces | 7 |
| 1.2.1 Linear Independence and Bases | 8 |
| 1.2.2 Inner Product and Orthogonality | 8 |
| 1.2.3 Matrices As Linear Transformations | 9 |
| 1.3 Derivatives of Vector Functions | 9 |
| 1.3.1 Newton's Method | 10 |
| 2 Solution of Systems of Linear Equations | 11 |
| 2.1 Gaussian Elimination | 11 |
| 2.1.1 The Basic Procedure | 12 |
| 2.1.2 Row Pivoting | 14 |
| 2.1.3 Iterative Refinement | 16 |
| 2.2 Cholesky Factorization | 16 |
| 2.3 Elementary Unitary Matrices and the QR Factorization | 19 |
| 2.3.1 Gram-Schmidt Orthogonalization | 19 |

| | | |
|----------|-----------------------------------------------|-----------|
| 2.3.2 | Householder Reflections | 20 |
| 2.3.3 | Complex Householder Matrices | 22 |
| 2.3.4 | Givens Rotations | 26 |
| 2.3.5 | Complex Givens Rotations | 27 |
| 2.3.6 | QR Factorization Using Householder Reflectors | 28 |
| 2.3.7 | Uniqueness of the Reduced QR Factorization | 29 |
| 2.3.8 | Solution of Least Squares Problems | 32 |
| 2.4 | The Singular Value Decomposition | 32 |
| 2.4.1 | Derivation and Properties of the SVD | 33 |
| 2.4.2 | The SVD and Least Squares Problems | 36 |
| 2.4.3 | Singular Values and the Norm of a Matrix | 39 |
| 2.4.4 | Low Rank Matrix Approximations | 39 |
| 2.4.5 | The Condition Number of a Matrix | 41 |
| 2.4.6 | Computation of the SVD | 42 |
| 3 | Eigenvalue Problems | 44 |
| 3.1 | Reduction to Tridiagonal Form | 46 |
| 3.2 | The Power Method | 46 |
| 3.3 | The Rayleigh Quotient | 47 |
| 3.4 | Inverse Iteration with Shifts | 47 |
| 3.5 | Rayleigh Quotient Iteration | 48 |
| 3.6 | The Basic QR Method | 48 |
| 3.6.1 | The QR Method with Shifts | 52 |
| 3.7 | The Divide-and-Conquer Method | 55 |
| 4 | Iterative Methods | 61 |

| | | |
|-----|-----------------------------------------|----|
| 4.1 | The Lanczos Method | 61 |
| 4.2 | The Conjugate Gradient Method | 64 |
| 4.3 | Preconditioning | 69 |
| | Bibliography | 71 |

List of Figures

| | | |
|-----|------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Householder reflection | 20 |
| 2.2 | Householder reduction of a matrix to bidiagonal form. | 42 |
| 3.1 | Graph of $f(\lambda) = 1 + \frac{.5}{1-\lambda} + \frac{.5}{2-\lambda} + \frac{.5}{3-\lambda} + \frac{.5}{4-\lambda}$ | 58 |
| 3.2 | Graph of $f(\lambda) = 1 + \frac{.5}{1-\lambda} + \frac{.01}{2-\lambda} + \frac{.5}{3-\lambda} + \frac{.5}{4-\lambda}$ | 59 |

Preface

The purpose of these notes is to present some of the standard procedures of numerical linear algebra from the perspective of a user and not a computer specialist. You will not find extensive error analysis or programming details. The purpose is to give the user a general idea of what the numerical procedures are doing. You can find more extensive discussions in the references

- Applied Numerical Linear Algebra by J. Demmel, SIAM 1997
- Numerical Linear Algebra by L. Trefethen and D. Bau, Siam 1997
- Matrix Computations by G. Golub and C. Van Loan, Johns Hopkins University Press 1996

The notes are divided into four chapters. The first chapter presents some of the notation used in this paper and reviews some of the basic results of Linear Algebra. The second chapter discusses methods for solving linear systems of equations, the third chapter discusses eigenvalue problems, and the fourth discusses iterative methods. Of course we cannot discuss every possible method, so I have tried to pick out those that I believe are the most used. I have assumed that the user has some basic knowledge of linear algebra.

Chapter 1

Mathematical Preliminaries

In this chapter we will describe some of the notation that will be used in these notes and review some of the basic results from Linear Algebra.

1.1 Matrices and Vectors

A matrix is a two-dimensional array of real or complex numbers arranged in rows and columns. If a matrix A has m rows and n columns, we say that it is an $m \times n$ matrix. We denote the element in the i -th row and j -th column of A by a_{ij} . The matrix A is often written in the form

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}.$$

We sometimes write $A = (a_1, \dots, a_n)$ where a_1, \dots, a_n are the columns of A . A vector (or n -vector) is an $n \times 1$ matrix. The collection of all n -vectors is denoted by \mathbb{R}^n if the elements (components) are all real and by \mathbb{C}^n if the elements are complex. We define the sum of two $m \times n$ matrices componentwise, i.e., the i, j entry of $A + B$ is $a_{ij} + b_{ij}$. Similarly, we define the multiplication of a scalar α times a matrix A to be the matrix whose i, j component is αa_{ij} .

If A is a real matrix with components a_{ij} , then the transpose of A (denoted by A^T) is the matrix whose i, j component is a_{ji} , i.e. rows and columns are interchanged. If A is a matrix with complex components, then A^H is the matrix whose i, j -th component is the complex conjugate of the j, i -th component of A . We denote the complex conjugate of a by \bar{a} . Thus, $(A^H)_{ij} = \overline{a_{ji}}$. A real matrix A is said to be symmetric if $A = A^T$. A complex matrix is said to be Hermitian if $A = A^H$. Notice that the diagonal elements of a Hermitian matrix must be real. The $n \times n$ matrix whose diagonal components are all one and whose off-diagonal components are all zero is called the identity matrix and is denoted by I .

If A is an $m \times k$ matrix and B is an $k \times n$ matrix, then the product AB is the $m \times n$ matrix with components given by

$$(AB)_{ij} = \sum_{r=1}^k a_{ir}b_{rj}.$$

The matrix product AB is only defined when the number of columns of A is the same as the number of rows of B . In particular, the product of an $m \times n$ matrix A and an n -vector x is given by

$$(Ax)_i = \sum_{k=1}^n a_{ik}x_k \quad i = 1, \dots, m.$$

It can be easily verified that $IA = A$ if the number of columns in I equals the number of rows in A . It can also be shown that $(AB)^T = B^T A^T$ and $(AB)^H = B^H A^H$. In addition, we have $(A^T)^T = A$ and $(A^H)^H = A$.

1.2 Vector Spaces

\mathbb{R}^n and \mathbb{C}^n together with the operations of addition and scalar multiplication are examples of a structure called a vector space. A vector space \mathcal{V} is a collection of vectors for which addition and scalar multiplication are defined in such a way that the following conditions hold:

1. If x and y belong to \mathcal{V} and α is a scalar, then $x + y$ and αx belong to \mathcal{V} .
2. $x + y = y + x$ for any two vectors x and y in \mathcal{V} .
3. $x + (y + z) = (x + y) + z$ for any three vectors x , y , and z in \mathcal{V} .
4. There is a vector 0 in \mathcal{V} such that $x + 0 = x$ for all x in \mathcal{V} .
5. For each x in \mathcal{V} there is a vector $-x$ in \mathcal{V} such that $x + (-x) = 0$.
6. $(\alpha\beta)x = \alpha(\beta x)$ for any scalars α , β and any vector x in \mathcal{V} .
7. $1x = x$ for any x in \mathcal{V} .
8. $\alpha(x + y) = \alpha x + \alpha y$ for any x and y in \mathcal{V} and any scalar α .
9. $(\alpha + \beta)x = \alpha x + \beta x$ for any x in \mathcal{V} and any scalars α , β .

A subspace of a vector space \mathcal{V} is a subset that is also a vector space in its own right.

1.2.1 Linear Independence and Bases

A set of vectors v_1, \dots, v_r is said to be linearly independent if the only way we can have $\alpha_1 v_1 + \dots + \alpha_r v_r = 0$ is for $\alpha_1 = \dots = \alpha_r = 0$. A set of vectors v_1, \dots, v_n is said to span a vector space \mathcal{V} if every vector x in \mathcal{V} can be written as a linear combination of the vectors v_1, \dots, v_n , i.e., $x = \alpha_1 v_1 + \dots + \alpha_n v_n$. The set of all linear combinations of the vectors v_1, \dots, v_r is a subspace denoted by $\langle v_1, \dots, v_r \rangle$ and called the span of these vectors. If a set of vectors v_1, \dots, v_n is linearly independent and spans \mathcal{V} it is called a basis for \mathcal{V} . If a vector space \mathcal{V} has a basis consisting of a finite number of vectors, then the space is said to be finite dimensional. In a finite-dimensional vector space every basis has the same number of vectors. This number is called the dimension of the vector space. Clearly \mathbb{R}^n and \mathbb{C}^n have dimension n . Let e_k denote the vector in \mathbb{R}^n or \mathbb{C}^n that consists of all zeroes except for a one in the k -th position. It is easily verified that e_1, \dots, e_n is a basis for either \mathbb{R}^n or \mathbb{C}^n .

1.2.2 Inner Product and Orthogonality

If x and y are two n -vectors, then the inner (dot) product $x \cdot y$ is the scalar value defined by $x^H y$. If the vector space is real we can replace x^H by x^T . The inner product $x \cdot y$ has the properties:

1. $y \cdot x = \overline{x \cdot y}$
2. $x \cdot (\alpha y) = \alpha(x \cdot y)$
3. $x \cdot (y + z) = x \cdot y + x \cdot z$
4. $x \cdot x \geq 0$ and $x \cdot x = 0$ if and only if $x = 0$.

Vectors x and y are said to be orthogonal if $x \cdot y = 0$. A basis v_1, \dots, v_n is said to be orthonormal if

$$v_i \cdot v_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}.$$

We define the norm $\|x\|$ of a vector x by $\|x\| = \sqrt{x \cdot x} = \sqrt{|x_1|^2 + \dots + |x_n|^2}$. The norm has the properties

1. $\|\alpha x\| = |\alpha| \|x\|$
2. $\|x\| = 0$ implies that $x = 0$
3. $\|x + y\| \leq \|x\| + \|y\|$.

If v_1, \dots, v_n is an orthonormal basis and $x = \alpha_1 v_1 + \dots + \alpha_n v_n$, then it can be shown that $\|x\|^2 = |\alpha_1|^2 + \dots + |\alpha_n|^2$. The norm and inner product satisfy the inequality

$$|x \cdot y| \leq \|x\| \|y\|. \quad \text{Cauchy Inequality}$$

1.2.3 Matrices As Linear Transformations

An $m \times n$ matrix A can be considered as a mapping of the space \mathbb{R}^n (\mathbb{C}^n) into the space \mathbb{R}^m (\mathbb{C}^m) where the image of the n -vector x is the matrix-vector product Ax . This mapping is linear, i.e., $A(x + y) = Ax + Ay$ and $A(\alpha x) = \alpha Ax$. The range of A (denoted by $\text{Range}(A)$) is the space of all m -vectors y such that $y = Ax$ for some n -vector x . It can be shown that the range of A is the space spanned by the columns of A . The null space of A (denoted by $\text{Null}(A)$) is the vector space consisting of all n -vectors x such that $Ax = 0$. An $n \times n$ square matrix A is said to be invertible if it is a one-to-one mapping of the space \mathbb{R}^n (\mathbb{C}^n) onto itself. It can be shown that a square matrix A is invertible if and only if the null space $\text{Null}(A)$ consists of only the zero vector. If A is invertible, then the inverse A^{-1} of A is defined by $A^{-1}y = x$ where x is the unique n -vector satisfying $Ax = y$. The inverse has the properties $A^{-1}A = AA^{-1} = I$ and $(AB)^{-1} = B^{-1}A^{-1}$. We denote $(A^{-1})^T$ and $(A^T)^{-1}$ by A^{-T} .

If A is an $m \times n$ matrix, x is an n -vector, and y is an m -vector; then it can be shown that

$$(Ax) \cdot y = x \cdot (A^H y).$$

1.3 Derivatives of Vector Functions

The central idea behind differentiation is the local approximation of a function by a linear function. If f is a function of one variable, then the locus of points $(x, f(x))$ is a plane curve \mathcal{C} . The tangent line to \mathcal{C} at $(x, f(x))$ is the graphical representation of the best local linear approximation to f at x . We call this local linear approximation the differential. We represent this local linear approximation by the equation $dy = f'(x)dx$. If f is a function of two variables, then the locus of points $(x, y, f(x, y))$ represents a surface \mathcal{S} . Here the best local linear approximation to f at (x, y) is graphically represented by the tangent plane to the surface \mathcal{S} at the point $(x, y, f(x, y))$. We will generalize this idea of a local linear approximation to vector-valued functions of n variables. Let f be a function mapping n -vectors into m -vectors. We define the derivative $Df(x)$ of f at the n -vector x to be the unique linear transformation ($m \times n$ matrix) satisfying

$$f(x + h) = f(x) + Df(x)h + o(\|h\|) \quad (1.1)$$

whenever such a transformation exists. Here the o notation signifies a function with the property

$$\lim_{\|h\| \rightarrow 0} \frac{o(\|h\|)}{\|h\|} = 0.$$

Thus, $Df(x)$ is a linear transformation that locally approximates f .

We can also define a directional derivative $\delta_h f(x)$ in the direction h by

$$\delta_h f(x) = \lim_{\lambda \rightarrow 0} \frac{f(x + \lambda h) - f(x)}{\lambda} = \left. \frac{d}{d\lambda} f(x + \lambda h) \right|_{\lambda=0} \quad (1.2)$$

whenever the limit exists. This directional derivative is also referred to as the *variation* of f in the direction h . If $Df(x)$ exists, then

$$\delta_h f(x) = Df(x)h.$$

However, the existence of $\delta_h f(x)$ for every direction h does not imply the existence of $Df(x)$. If we take $h = e_i$, then $\delta_h f(x)$ is just the partial derivative $\frac{\partial f(x)}{\partial x_i}$.

1.3.1 Newton's Method

Newton's method is an iterative scheme for finding the zeroes of a smooth function f . If x is a guess, then we approximate f near x by

$$f(x + h) = f(x) + Df(x)h.$$

If $x + h$ is the zero of this linear approximation, then

$$h = -(Df(x))^{-1} f(x)$$

or

$$x + h = x - (Df(x))^{-1} f(x). \tag{1.3}$$

We can take $x + h$ as an improved approximation to the nearby zero of f . If we keep iterating with equation (1.3), then the $(k + 1)$ -iterate $x^{(k+1)}$ is related to the k -iterate $x^{(k)}$ by

$$x^{(k+1)} = x^{(k)} - (Df(x^{(k)}))^{-1} f(x^{(k)}). \tag{1.4}$$

Chapter 2

Solution of Systems of Linear Equations

2.1 Gaussian Elimination

Gaussian elimination is the standard way of solving a system of linear equations $Ax = b$ when A is a square matrix with no special properties. The first known use of this method was in the Chinese text *Nine Chapters on the Mathematical Art* written between 200 BC and 100 BC. Here it was used to solve a system of three equations in three unknowns. The coefficients (including the right-hand-side) were written in tabular form and operations were performed on this table to produce a triangular form that could be easily solved. It is remarkable that this was done long before the development of matrix notation or even a notation for variables. The method was used by Gauss in the early 1800s to solve a least squares problem for determining the orbit of the asteroid Pallas. Using observations of Pallas taken between 1803 and 1809, he obtained a system of six equations in six unknowns which he solved by the method now known as Gaussian elimination. The concept of treating a matrix as an object and the development of an algebra for matrices were first introduced by Cayley [2] in the paper *A Memoir on the Theory of Matrices*.

In this paper we will first describe the basic method and show that it is equivalent to factoring the matrix into the product of a lower triangular and an upper triangular matrix, i.e., $A = LU$. We will then introduce the method of row pivoting that is necessary in order to keep the method stable. We will show that row pivoting is equivalent to a factorization $PA = LU$ or $A = PLU$ where P is the identity matrix with its rows permuted. Having obtained this factorization, the solution for a given right-hand-side b is obtained by solving the two triangular systems $Ly = Pb$ and $Ux = y$ by simple processes called forward and backward substitution.

There are a number of good computer implementations of Gaussian elimination with row pivoting. Matlab has a good implementation obtained by the call `[L,U,P]=lu(A)`. Another good implementation is the LAPACK routine SGESV (DGESV,CGESV). It can be obtained in either Fortran or C from the site www.netlib.org.

We will end by showing how the accuracy of a solution can be improved by a process called

iterative refinement.

2.1.1 The Basic Procedure

Gaussian elimination begins by producing zeroes below the diagonal in the first column, i.e.,

$$\begin{pmatrix} \times & \times & \dots & \times \\ \times & \times & \dots & \times \\ \vdots & \vdots & & \vdots \\ \times & \times & \dots & \times \end{pmatrix} \rightarrow \begin{pmatrix} \times & \times & \dots & \times \\ 0 & \times & \dots & \times \\ \vdots & \vdots & & \vdots \\ 0 & \times & \dots & \times \end{pmatrix}. \quad (2.1)$$

If a_{ij} is the element of A in the i -th row and the j -th column, then the first step in the Gaussian elimination process consists of multiplying A on the left by the lower triangular matrix L_1 given by

$$L_1 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -a_{21}/a_{11} & 1 & 0 & \dots & 0 \\ -a_{31}/a_{11} & 0 & 1 & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ -a_{n1}/a_{11} & 0 & \dots & 0 & 1 \end{pmatrix}, \quad (2.2)$$

i.e., zeroes are produced in the first column by adding appropriate multiples of the first row to the other rows. The next step is to produce zeroes below the diagonal in the second column, i.e.,

$$\begin{pmatrix} \times & \times & \dots & \times \\ 0 & \times & \dots & \times \\ \vdots & \vdots & & \vdots \\ 0 & \times & \dots & \times \end{pmatrix} \rightarrow \begin{pmatrix} \times & \times & \times & \dots & \times \\ 0 & \times & \times & & \times \\ 0 & 0 & \times & & \times \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \times & \dots & \times \end{pmatrix}. \quad (2.3)$$

This can be obtained by multiplying $L_1 A$ on the left by the lower triangular matrix L_2 given by

$$L_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & & 0 \\ 0 & -a_{32}^{(1)}/a_{22}^{(1)} & 1 & 0 & & 0 \\ 0 & -a_{42}^{(1)}/a_{22}^{(1)} & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ 0 & -a_{n2}^{(1)}/a_{22}^{(1)} & 0 & \dots & 0 & 1 \end{pmatrix} \quad (2.4)$$

where $a_{ij}^{(1)}$ is the i, j -th element of $L_1 A$. Continuing in this manner, we can define lower triangular matrices L_3, \dots, L_{n-1} so that $L_{n-1} \cdots L_1 A$ is upper triangular, i.e.,

$$L_{n-1} \cdots L_1 A = U. \quad (2.5)$$

Taking the inverses of the matrices L_1, \dots, L_{n-1} , we can write A as

$$A = L_1^{-1} \cdots L_{n-1}^{-1} U. \quad (2.6)$$

Let

$$L = L_1^{-1} \cdots L_{n-1}^{-1}. \quad (2.7)$$

Then it follows from equation (2.6) that

$$A = LU. \quad (2.8)$$

We will now show that L is lower triangular. Each of the matrices L_k can be written in the form

$$L_k = I - u^{(k)} e_k^T \quad (2.9)$$

where e_k is the vector whose components are all zero except for a one in the k -th position and $u^{(k)}$ is a vector whose first k components are zero. The term $u^{(k)} e_k^T$ is an $n \times n$ matrix whose elements are all zero except for those below the diagonal in the k -th column. In fact, the components of $u^{(k)}$ are given by

$$u_i^{(k)} = \begin{cases} 0 & 1 \leq i \leq k \\ a_{ik}^{(k-1)} / a_{kk}^{(k-1)} & k < i \end{cases} \quad (2.10)$$

where $a_{ij}^{(k-1)}$ is the i, j -th element of $L_{k-1} \cdots L_1 A$. Since $e_k^T u^{(k)} = u_k^{(k)} = 0$, it follows that

$$\begin{aligned} (I + u^{(k)} e_k^T)(I - u^{(k)} e_k^T) &= I + u^{(k)} e_k^T - u^{(k)} e_k^T - u^{(k)} e_k^T u^{(k)} e_k^T \\ &= I - u^{(k)} (e_k^T u^{(k)}) e_k^T \\ &= I, \end{aligned} \quad (2.11)$$

i.e.,

$$L_k^{-1} = I + u^{(k)} e_k^T. \quad (2.12)$$

Thus, L_k^{-1} is the same as L_k except for a change of sign of the elements below the diagonal in column k . Combining equations (2.7) and (2.12), we obtain

$$L = (I + u^{(1)} e_1^T) \cdots (I + u^{(n-1)} e_{n-1}^T) = I + u^{(1)} e_1^T + \cdots + u^{(n-1)} e_{n-1}^T. \quad (2.13)$$

In this expression the cross terms dropped out since

$$u^{(i)} e_i^T u^{(j)} e_j^T = u_i^{(j)} u^{(i)} e_j^T = 0 \quad \text{for } i < j.$$

Equation (2.13) implies that L is lower triangular and that the k -th column of L looks like the k -th column of L_k with the signs reversed on the elements below the diagonal, i.e.,

$$L = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_{21}/a_{11} & 1 & 0 & & 0 \\ a_{31}/a_{11} & a_{32}^{(1)}/a_{22}^{(1)} & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1}/a_{11} & a_{n2}^{(1)}/a_{22}^{(1)} & & & 1 \end{pmatrix}. \quad (2.14)$$

Having the LU factorization given in equation (2.8), it is possible to solve the system of equations

$$Ax = LUx = b$$

for any right-hand-side b . If we let $y = Ux$, then y can be found by solving the triangular system $Ly = b$. Having y , x can be obtained by solving the triangular system $Ux = y$. Triangular systems are very easy to solve. For example, in the system $Ux = y$, the last equation can be solved for x_n (the only unknown in this equation). Having x_n , the next to the last equation can be solved for x_{n-1} (the only unknown left in this equation). Continuing in this manner we can solve for the remaining components of x . For the system $Ly = b$, we start by computing y_1 and then work our way down. Solving an upper triangular system is called *back substitution*. Solving a lower triangular system is called *forward substitution*.

To compute L requires approximately $n^3/3$ operations where an operation consists of an addition and a multiplication. For each right-hand-side, solving the two triangular systems requires approximately n^2 operations. Thus, as far as solving systems of equations is concerned, having the LU factorization of A is just as good as having the inverse of A and is less costly to compute.

2.1.2 Row Pivoting

There is one problem with Gaussian elimination that has yet to be addressed. It is possible for one of the diagonal elements $a_{kk}^{(k-1)}$ that occur during Gaussian elimination to be zero or to be very small. This causes a problem since we must divide by this diagonal element. If one of the diagonals is exactly zero, the process obviously blows up. However, there can still be a problem if one of the diagonals is small. In this case large elements are produced in both the L and U matrices. These large entries lead to a loss of accuracy when there are subtractions involving these big numbers. This problem can occur even for well behaved matrices. To eliminate this problem we introduce row pivoting. In performing Gaussian elimination, it is not necessary to take the equations in the order they are given. Suppose we are at the stage where we are zeroing out the elements below the diagonal in the k -th column. We can interchange any of the rows from the k -th row on without changing the structure of the matrix. In row pivoting we find the largest in magnitude of the elements $a_{kk}^{(k-1)}, a_{k+1,k}^{(k-1)}, \dots, a_{nk}^{(k-1)}$ and interchange rows to bring that element to the k, k -position. Mathematically we can perform this row interchange by multiplying on the left by the matrix P_k that is like the identity matrix with the appropriate rows interchanged. The matrix P_k has the property $P_k P_k = I$, i.e., P_k is its own inverse. With row pivoting equation (2.5) is replaced by

$$L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1 A = U. \quad (2.15)$$

We can write this equation in the form

$$L_{n-1} (P_{n-1} L_{n-2} P_{n-1}^{-1}) (P_{n-1} P_{n-2} L_{n-3} P_{n-2}^{-1} P_{n-1}^{-1}) \cdots \\ (P_{n-1} \cdots P_2 L_1 P_2^{-1} \cdots P_{n-1}^{-1}) (P_{n-1} \cdots P_1) A = U. \quad (2.16)$$

Define $L'_{n-1} = L_{n-1}$ and

$$L'_k = P_{n-1} \cdots P_{k+1} L_k P_{k+1}^{-1} \cdots P_{n-1}^{-1} \quad k = 1, \dots, n-2. \quad (2.17)$$

Then equation (2.16) can be written

$$(L'_{n-1} \cdots L'_1)(P_{n-1} \cdots P_1)A = U. \quad (2.18)$$

Note that multiplying by P_j on the left only modifies rows j up to n . Similarly, multiplying by $P_j^{-1} = P_j$ on the right only modifies columns j up to n . Therefore,

$$\begin{aligned} L'_k &= (P_{n-1} \cdots P_{k+1})(I + u^{(k)}e_k^T)(P_{k+1} \cdots P_{n-1}) \\ &= I + (P_{n-1} \cdots P_{k+1})u^{(k)}e_k^T(P_{k+1} \cdots P_{n-1}) \\ &= I + v^{(k)}e_k^T \end{aligned} \quad (2.19)$$

where $v^{(k)}$ is like $u^{(k)}$ except the components $k + 1$ to n are permuted by $P_{n-1} \cdots P_{k+1}$. Since L'_k has the same form as L_k , it follows that the matrix $L = (L'_1)^{-1} \cdots (L'_{n-1})^{-1}$ is lower triangular. Thus, if we define $P = P_{n-1} \cdots P_1$, equation (2.18) becomes

$$PA = LU. \quad (2.20)$$

Of course, in practice we don't need to explicitly construct the matrix P since the interchanges can be kept track of using a vector. To solve a system of equations $Ax = b$ we replace the system by $P Ax = P b$ and proceed as before.

It is also possible to do column interchanges as well as row interchanges, but this is seldom used in practice. By the construction of L all its elements are less than or equal to one in magnitude. The elements of U are usually not very large, but there are some peculiar cases where large entries can appear in U even with row pivoting. For example, consider the matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 1 & 0 & \cdots & 0 & 1 \\ -1 & -1 & 1 & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & 0 & 1 \\ -1 & -1 & & & 1 & 1 \\ -1 & -1 & -1 & \cdots & -1 & 1 \end{pmatrix}.$$

In the first step no pivoting is necessary, but the elements 2 through n in the last column are doubled. In the second step again no pivoting is necessary, but the elements 3 through n are doubled. Continuing in this manner we arrive at

$$U = \begin{pmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & \ddots & \vdots \\ & & & & 2^{n-1} \end{pmatrix}.$$

Although growth like this in the size of the elements of U is theoretically possible, there are no reports of this ever having happened in the solution of a real-world problem. In practice Gaussian elimination with row pivoting has proven to be very stable.

2.1.3 Iterative Refinement

If the solution of $Ax = b$ is not sufficiently accurate, the accuracy can be improved by applying Newton's method to the function $f(x) = Ax - b$. If $x^{(k)}$ is an approximate solution to $f(x) = 0$, then a Newton iteration produces an approximation $x^{(k+1)}$ given by

$$x^{(k+1)} = x^{(k)} - (Df(x^{(k)}))^{-1} f(x^{(k)}) = x^{(k)} - A^{-1}[Ax^{(k)} - b]. \quad (2.21)$$

An iteration step can be summarized as follows:

1. compute the residual $r^{(k)} = Ax^{(k)} - b$;
2. solve the system $Ad^{(k)} = r^{(k)}$ using the LU factorization of A ;
3. Compute $x^{(k+1)} = x^{(k)} - d^{(k)}$.

The residual is usually computed in double precision. If the above calculations were carried out exactly, the answer would be obtained in one iteration as is always true when applying Newton's method to a linear function. However, because of roundoff errors, it may require more than one iteration to obtain the desired accuracy.

2.2 Cholesky Factorization

Matrices that are Hermitian ($A^H = A$) and positive definite ($x^H Ax > 0$ for all $x \neq 0$) occur sufficiently often in practice that it is worth describing a variant of Gaussian elimination that is often used for this class of matrices. Recall that Gaussian elimination amounted to a factorization of a square matrix A into the product of a lower triangular matrix and an upper triangular matrix, i.e., $A = LU$. The Cholesky factorization represents a Hermitian positive definite matrix A by the product of a lower triangular matrix and its conjugate transpose, i.e., $A = LL^H$. Because of the symmetries involved, this factorization can be formed in roughly half the number of operations as are needed for Gaussian elimination.

Let us begin by looking at some of the properties of positive definite matrices. If e_i is the i -th column of the identity matrix and $A = (a_{ij})$ is positive definite, then $a_{ii} = e_i^T A e_i > 0$, i.e., the diagonal components of A are real and positive. Suppose X is a nonsingular matrix of the same size as the Hermitian, positive definite matrix A . Then

$$x^H (X^H A X)x = (Xx)^H A (Xx) > 0 \quad \text{for all } x \neq 0.$$

Thus, A Hermitian positive definite implies that $X^H A X$ is Hermitian positive definite. Conversely, suppose $X^H A X$ is Hermitian positive definite. Then

$$A = (X X^{-1})^H A (X X^{-1}) = (X^{-1})^H (X^H A X) (X^{-1}) \quad \text{is Hermitian positive definite.}$$

Next we will show that the component of largest magnitude of a Hermitian positive definite matrix A always lies on the diagonal. Suppose that $|a_{kl}| = \max_{i,j} |a_{ij}|$ and $k \neq l$. If $a_{kl} = |a_{kl}|e^{i\theta_{kl}}$, let $\alpha = -e^{-i\theta_{kl}}$ and $x = e_k + \alpha e_l$. Then

$$x^H Ax = e_k^T A e_k + \bar{\alpha} e_l^T A e_k + \alpha e_k^T A e_l + |\alpha|^2 e_l^T A e_l = a_{kk} + a_{ll} - 2|a_{kl}| \leq 0.$$

This contradicts the fact that A is positive definite. Therefore, $\max_{i,j} |a_{ij}| = \max_i a_{ii}$. Suppose we partition the Hermitian positive definite matrix A as follows

$$A = \begin{pmatrix} B & C^H \\ C & D \end{pmatrix}$$

If y is a nonzero vector compatible with D , let $x^H = (0, y^H)$. Then

$$x^H Ax = (0, y^H) \begin{pmatrix} B & C^H \\ C & D \end{pmatrix} \begin{pmatrix} 0 \\ y \end{pmatrix} = y^H D y > 0,$$

i.e., D is Hermitian positive definite. Similarly, letting $x^H = (y^H, 0)$, we can show that B is Hermitian positive definite.

We will now show that if A is a Hermitian, positive-definite matrix, then there is a unique lower triangular matrix L with positive diagonals such that $A = LL^H$. This factorization is called the Cholesky factorization. We will establish this result by induction on the dimension n . Clearly, the result is true for $n = 1$. For in this case we can take $L = (\sqrt{a_{11}})$. Suppose the result is true for matrices of dimension $n - 1$. Let A be a Hermitian, positive-definite matrix of dimension n . We can partition A as follows

$$A = \begin{pmatrix} a_{11} & w^H \\ w & K \end{pmatrix} \quad (2.22)$$

where w is a vector of dimension $n - 1$ and K is a $(n - 1) \times (n - 1)$ matrix. It is easily verified that

$$A = \begin{pmatrix} a_{11} & w^H \\ w & K \end{pmatrix} = B^H \begin{pmatrix} 1 & 0 \\ 0 & K - \frac{ww^H}{a_{11}} \end{pmatrix} B \quad (2.23)$$

where

$$B = \begin{pmatrix} \sqrt{a_{11}} & \frac{w^H}{\sqrt{a_{11}}} \\ 0 & I \end{pmatrix}. \quad (2.24)$$

We will first show that the matrix B is invertible. If

$$Bx = \begin{pmatrix} \sqrt{a_{11}} & \frac{w^H}{\sqrt{a_{11}}} \\ 0 & I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{a_{11}}x_1 + \frac{w^H x_2}{\sqrt{a_{11}}} \\ x_2 \end{pmatrix} = 0,$$

then $x_2 = 0$ and $\sqrt{a_{11}}x_1 = x_1 = 0$. Therefore, B is invertible. From our discussion at the beginning of this section it follows from equation (2.23) that the matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & k - \frac{ww^H}{a_{11}} \end{pmatrix}$$

is Hermitian positive definite. By the results on the partitioning of a positive definite matrix, it follows that the matrix

$$K - \frac{ww^H}{a_{11}}$$

is Hermitian positive definite. By the induction hypothesis, there exists a unique lower triangular matrix \hat{L} with positive diagonals such that

$$K - \frac{ww^H}{a_{11}} = \hat{L}\hat{L}^H. \quad (2.25)$$

Substituting equation (2.25) into equation (2.23), we get

$$A = B^H \begin{pmatrix} 1 & 0 \\ 0 & \hat{L}\hat{L}^H \end{pmatrix} B = B^H \begin{pmatrix} 1 & 0 \\ 0 & \hat{L} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \hat{L}^H \end{pmatrix} B = \begin{pmatrix} \sqrt{a_{11}} & 0 \\ \frac{w}{\sqrt{a_{11}}} & \hat{L} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & \frac{w^H}{\sqrt{a_{11}}} \\ 0 & \hat{L}^H \end{pmatrix} \quad (2.26)$$

which is the desired factorization of A . To show uniqueness, suppose that

$$A = \begin{pmatrix} a_{11} & w^H \\ w & K \end{pmatrix} = \begin{pmatrix} l_{11} & 0 \\ v & \hat{L} \end{pmatrix} \begin{pmatrix} l_{11} & v^H \\ 0 & \hat{L}^H \end{pmatrix} \quad (2.27)$$

is a Cholesky factorization of A . Equating components in equation (2.27), we see that $l_{11}^2 = a_{11}$ and hence that $l_{11} = \sqrt{a_{11}}$. Also $l_{11}v = w$ or $v = w/l_{11} = w/\sqrt{a_{11}}$. Finally, $vv^H + \hat{L}\hat{L}^H = K$ or $K - vv^H = K - ww^H/a_{11} = \hat{L}\hat{L}^H$. Since $\hat{L}\hat{L}^H$ is the unique factorization of the $(n-1) \times (n-1)$ Hermitian, positive-definite matrix $K - ww^H/a_{11}$, we see that the Cholesky factorization of A is unique. It now follows by induction that there is a unique Cholesky factorization of any Hermitian, positive-definite matrix.

The factorization in equation (2.23) is the basis for the computation of the Cholesky factorization. The matrix B^H is lower triangular. Since the matrix $K - ww^H/a_{11}$ is positive definite, it can be factored in the same manner. Continuing in this manner until the center matrix becomes the identity matrix, we obtain lower triangular matrices L_1, \dots, L_n such that

$$A = L_1 \cdots L_n L_n^H \cdots L_1^H.$$

Letting $L = L_1 \cdots L_n$, we have the desired Cholesky factorization.

As was mentioned previously, the number of operations in the Cholesky factorization is about half the number in Gaussian elimination. Unlike Gaussian elimination the Cholesky method does not need pivoting in order to maintain stability. The Cholesky factorization can also be written in the form

$$A = LDL^H$$

where D is diagonal and L now has all ones on the diagonal.

2.3 Elementary Unitary Matrices and the QR Factorization

In Gaussian elimination we saw that a square matrix A could be reduced to triangular form by multiplying on the left by a series of elementary lower triangular matrices. This process can also be expressed as a factorization $A = LU$ where L is lower triangular and U is upper triangular. In least squares problems the number of rows m in A is usually greater than the number of columns n . The standard technique for solving least-squares problems of this type is to make use of a factorization $A = QR$ where Q is an $m \times m$ unitary matrix and R has the form

$$R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$$

with \hat{R} an $n \times n$ upper triangular matrix. The usual way of obtaining this factorization is to reduce the matrix A to triangular form by multiplying on the left by a series of elementary unitary matrices that are sometimes called Householder matrices (reflectors). We will show how to use this QR factorization to solve least square problems. If \hat{Q} is the $m \times n$ matrix consisting of the first n columns of Q , then

$$A = \hat{Q}\hat{R}.$$

This factorization is called the reduced QR factorization. Elementary unitary matrices are also used to reduce square matrices to a simplified form (Hessenberg or tridiagonal) prior to eigenvalue calculation.

There are several good computer implementations that use the Householder QR factorization to solve the least squares problem. The LAPACK routine is called SGELS (DGELS,CGELS). In Matlab the solution of the least squares problem is given by $A \setminus b$. The QR factorization can be obtained with the call $[Q,R]=qr(A)$.

2.3.1 Gram-Schmidt Orthogonalization

A reduced QR factorization can be obtained by an orthogonalization procedure known as the Gram-Schmidt process. Suppose we would like to construct an orthonormal set of vectors q_1, \dots, q_n from a given linearly independent set of vectors a_1, \dots, a_n . The process is recursive. At the j -th step we construct a unit vector q_j that is orthogonal to q_1, \dots, q_{j-1} using

$$v_j = a_j - \sum_{i=1}^{j-1} (q_i^H a_j) q_i$$
$$q_j = v_j / \|v_j\|.$$

The orthonormal basis constructed has the additional property

$$\langle q_1, \dots, q_j \rangle = \langle a_1, \dots, a_j \rangle \quad j = 1, 2, \dots, n.$$

If we consider a_1, \dots, a_n as columns of a matrix A , then this process is equivalent to the matrix factorization $A = \hat{Q}\hat{R}$ where $\hat{Q} = (q_1, \dots, q_n)$ and \hat{R} is upper triangular. Although the Gram-Schmidt process is very useful in theoretical considerations, it does not lead to a stable numerical procedure. In the next section we will discuss Householder reflectors, which lead to a more stable procedure for obtaining a QR factorization.

2.3.2 Householder Reflections

Let us begin by describing the Householder reflectors. In this section we will restrict ourselves to real matrices. Afterwards we will see that there are a number of generalizations to the complex case. If v is a fixed vector of dimension m with $\|v\| = 1$, then the set of all vectors orthogonal to v is an $(m - 1)$ -dimensional subspace called a hyperplane. If we denote this hyperplane by H , then

$$H = \{u : v^T u = 0\}. \quad (2.28)$$

Here v^T denotes the transpose of v . If x is a point not on H , let \bar{x} denote the orthogonal projection of x onto H (see Figure 2.1). The difference $\bar{x} - x$ must be orthogonal to H and hence a multiple of v , i.e.,

$$\bar{x} - x = \alpha v \quad \text{or} \quad \bar{x} = x + \alpha v. \quad (2.29)$$

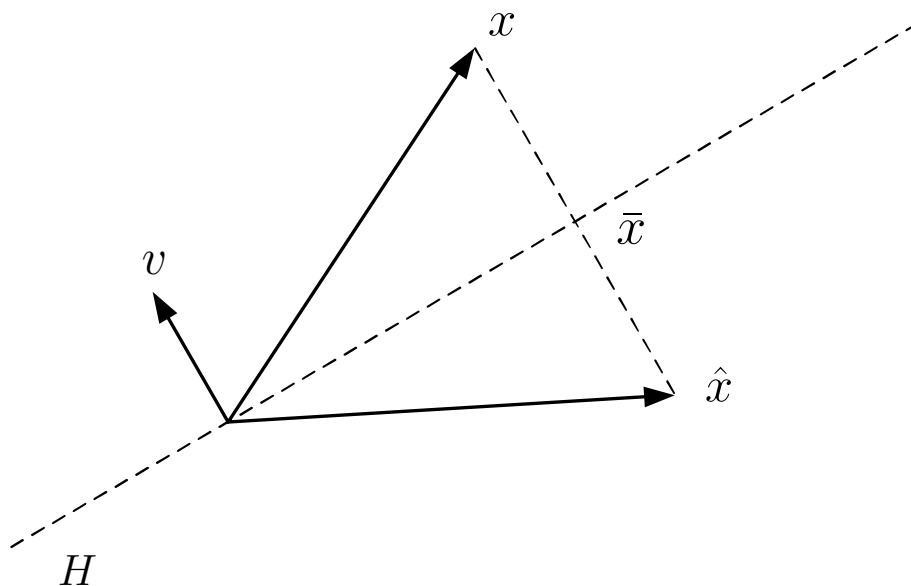


Figure 2.1: Householder reflection

Since \bar{x} lies on H and $v^T v = \|v\|^2 = 1$, we must have

$$v^T \bar{x} = v^T x + \alpha v^T v = v^T x + \alpha = 0. \quad (2.30)$$

Thus, $\alpha = -v^T x$ and consequently

$$\bar{x} = x - (v^T x)v = x - vv^T x = (I - vv^T)x. \quad (2.31)$$

Define $P = I - vv^T$. Then P is a projection matrix that projects vectors orthogonally onto H . The projection \bar{x} is obtained by going a certain distance from x in the direction $-v$. Figure 2.1 suggests that the reflection \hat{x} of x across H can be obtained by going twice that distance in the same direction, i.e.,

$$\hat{x} = x - 2(v^T x)v = x - 2vv^T x = (I - 2vv^T)x. \quad (2.32)$$

With this motivation we define the Householder reflector Q by

$$Q = I - 2vv^T \quad \|v\| = 1. \quad (2.33)$$

An alternate form for the Householder reflector is

$$Q = I - \frac{2uu^T}{\|u\|^2} \quad (2.34)$$

where here u is not restricted to be a unit vector. Notice that, in this form, replacing u by a multiple of u does not change Q . The matrix Q is clearly symmetric, i.e., $Q^T = Q$. Moreover,

$$Q^T Q = Q^2 = (I - 2vv^T)(I - 2vv^T) = I - 2vv^T - 2vv^T + 4vv^T vv^T = I, \quad (2.35)$$

i.e., Q is an orthogonal matrix. As with all orthogonal matrices Q preserves the norm of a vector, i.e.,

$$\|Qx\|^2 = (Qx)^T Qx = x^T Q^T Qx = x^T x = \|x\|^2. \quad (2.36)$$

To reduce a matrix to one that is upper triangular it is necessary to zero out columns below a certain position. We will show how to construct a Householder reflector so that its action on a given vector x is a multiple of e_1 , the first column of the identity matrix. To zero out a vector below row k we can use a matrix of the form

$$Q = \begin{pmatrix} I & 0 \\ 0 & Q \end{pmatrix}$$

where I is the $(k-1) \times (k-1)$ identity matrix and Q is a $(m-k+1) \times (m-k+1)$ Householder matrix. Thus, for a given vector x we would like to choose a vector u so that Qx is a multiple of the unit vector e_1 , i.e.,

$$Qx = x - \frac{2(u^T x)}{\|u\|^2} u = \alpha e_1. \quad (2.37)$$

Since Q preserves norms, we must have $|\alpha| = \|x\|$. Therefore, equation (2.37) becomes

$$Qx = x - \frac{2(u^T x)}{\|u\|^2} u = \pm \|x\| e_1. \quad (2.38)$$

It follows from equation (2.38) that u must be a multiple of the vector $x \mp \|x\| e_1$. Since u can be replaced by a multiple of u without changing Q , we let

$$u = x \mp \|x\| e_1. \quad (2.39)$$

It follows from the definition of u in equation (2.39) that

$$u^T x = \|x\|^2 \mp \|x\| x_1 \quad (2.40)$$

and

$$\|u\|^2 = u^T u = \|x\|^2 \mp \|x\|x_1 \mp \|x\|x_1 + \|x\|^2 = 2(\|x\|^2 \mp \|x\|x_1). \quad (2.41)$$

Therefore,

$$\frac{2(u^T x)}{\|u\|^2} = 1, \quad (2.42)$$

and hence Qx becomes

$$Qx = x - \frac{2(u^T x)}{\|u\|^2}u = x - u = \pm\|x\|e_1 \quad (2.43)$$

as desired. From what has been discussed so far, either of the signs in equation (2.39) would produce the desired result. However, if x_1 is very large compared to the other components, then it is possible to lose accuracy through subtraction in the computation of $u = x \mp \|x\|e_1$. To prevent this we choose u to be

$$u = x + \text{sign}(x_1)\|x\|e_1 \quad (2.44)$$

where $\text{sign}(x_1)$ is defined by

$$\text{sign}(x_1) = \begin{cases} +1 & x_1 \geq 0 \\ -1 & x_1 < 0. \end{cases} \quad (2.45)$$

With this choice of u , equation (2.43) becomes

$$Qx = -\text{sign}(x_1)\|x\|e_1. \quad (2.46)$$

In practice, u is often scaled so that $u_1 = 1$, i.e.,

$$u = \frac{x + \text{sign}(x_1)\|x\|e_1}{x_1 + \text{sign}(x_1)\|x\|}. \quad (2.47)$$

With this choice of u ,

$$\|u\|^2 = \frac{2\|x\|}{\|x\| + |x_1|}. \quad (2.48)$$

The matrix Q applied to a general vector y is given by

$$Qy = y - 2\frac{u^T y}{\|u\|^2}u. \quad (2.49)$$

2.3.3 Complex Householder Matrices

There are several ways to generalize Householder matrices to the complex case. The most obvious is to let

$$U = I - 2\frac{uu^H}{\|u\|^2}$$

where the superscript H denotes conjugate transpose. It can be shown that a matrix of this form is both Hermitian ($U = U^H$) and unitary ($U^H U = I$). However, it is sometimes convenient

to be able to construct a U such that $U^H x$ is a real multiple of e_1 . This is especially true when converting a Hermitian matrix to tridiagonal form prior to an eigenvalue computation. For in this case the tridiagonal matrix becomes a real symmetric matrix even when starting with a complex Hermitian matrix. Thus, it is not necessary to have a separate eigenvalue routine for the complex case. It turns out that there is no Hermitian unitary matrix U , as defined above, that is guaranteed to produce a real multiple of e_1 . Therefore, linear algebra libraries such as LAPACK use elementary unitary matrices of the form

$$U = I - \sigma w w^H \quad (2.50)$$

where σ can be complex. These matrices are not in general Hermitian. If U is to be unitary, we must have

$$I = U^H U = (I - \bar{\sigma} w w^H)(I - \sigma w w^H) = I - (\bar{\sigma} + \sigma - |\sigma|^2 \|w\|^2) w w^H$$

and hence

$$|\sigma|^2 \|w\|^2 = 2 \operatorname{Re}(\sigma). \quad (2.51)$$

Notice that replacing w by w/η and σ by $|\eta|^2 \sigma$ in equation (2.50) leaves U unchanged. Thus, a scaling of w can be absorbed in σ . We would like to choose w and σ so that

$$U^H x = x - \bar{\sigma}(w^H x)w = \gamma \|x\| e_1 \quad (2.52)$$

where $\gamma = \pm 1$. It can be seen from equation (2.52) that w must be proportional to the vector $x - \gamma \|x\| e_1$. Since the factor of proportionality can be absorbed in σ , we choose

$$w = x - \gamma \|x\| e_1. \quad (2.53)$$

Substituting this expression for w into equation (2.52), we get

$$U^H x = x - \bar{\sigma}(w^H x)(x - \gamma \|x\| e_1) = (1 - \bar{\sigma} w^H x)x + \bar{\sigma} \gamma (w^H x) \|x\| e_1 = \gamma \|x\| e_1. \quad (2.54)$$

Thus, we must have

$$\bar{\sigma}(w^H x) = 1 \quad \text{or} \quad \sigma = \frac{1}{x^H w}. \quad (2.55)$$

This choice of σ gives

$$U^H x = \gamma \|x\| e_1.$$

It follows from equation (2.53) that

$$x^H w = \|x\|^2 - \gamma \|x\| \bar{x}_1 \quad (2.56)$$

and

$$\begin{aligned} \|w\|^2 &= (x^H - \gamma \|x\| e_1^T)(x - \gamma \|x\| e_1) = \|x\|^2 - \gamma \|x\| x_1 - \gamma \|x\| \bar{x}_1 + \|x\|^2 \\ &= 2(\|x\|^2 - \gamma \|x\| \operatorname{Re}(x_1)) \end{aligned} \quad (2.57)$$

Thus, it follows from equations (2.55)–(2.57) that

$$\begin{aligned}
\frac{2 \operatorname{Re}(\sigma)}{|\sigma|^2} &= \frac{\sigma + \bar{\sigma}}{\sigma \bar{\sigma}} = \frac{1}{\sigma} + \frac{1}{\bar{\sigma}} = x^H w + w^H x \\
&= (\|x\|^2 - \gamma \|x\| \bar{x}_1) + (\|x\|^2 - \gamma \|x\| x_1) \\
&= 2(\|x\|^2 - 2\gamma \|x\| \operatorname{Re}(x_1)) \\
&= \|w\|^2,
\end{aligned}$$

i.e., the condition in equation (2.51) is satisfied. It follows that the matrix U defined by equation (2.50) is unitary when w is defined by equation (2.53) and σ is defined by equation (2.55). As before we choose γ to prevent the loss of accuracy due to subtraction in equation (2.53). In this case we choose $\gamma = -\operatorname{sign}(\operatorname{Re}(x_1))$. Thus, w becomes

$$w = x + \operatorname{sign}(\operatorname{Re}(x_1)) \|x\| e_1. \quad (2.58)$$

Let us define a real constant ν by

$$\nu = \operatorname{sign}(\operatorname{Re}(x_1)) \|x\|. \quad (2.59)$$

With this definition w becomes

$$w = x + \nu e_1. \quad (2.60)$$

It follows that

$$x^H w = \|x\|^2 + \nu \bar{x}_1 = \nu^2 + \nu \bar{x}_1 = \nu(\nu + \bar{x}_1), \quad (2.61)$$

and hence

$$\sigma = \frac{1}{\nu(\nu + \bar{x}_1)}. \quad (2.62)$$

In LAPACK w is scaled so that $w_1 = 1$, i.e.,

$$w = \frac{x + \nu e_1}{x_1 + \nu}. \quad (2.63)$$

With this w , σ becomes

$$\sigma = \frac{|x_1 + \nu|^2}{\nu(\nu + \bar{x}_1)} = \frac{(x_1 + \nu)(\bar{x}_1 + \nu)}{\nu(\nu + \bar{x}_1)} = \frac{x_1 + \nu}{\nu}. \quad (2.64)$$

Clearly this σ satisfies the inequality

$$|\sigma - 1| = \frac{|x_1|}{|\nu|} = \frac{|x_1|}{\|x\|} \leq 1. \quad (2.65)$$

It follows from equation (2.64) that σ is real when x_1 is real. Thus, U is Hermitian when x_1 is real.

An alternate approach to defining a complex Householder matrix is to let

$$U = I - \frac{2ww^H}{\|w\|^2}. \quad (2.66)$$

This U is Hermitian and

$$U^H U = \left(I - \frac{2ww^H}{\|w\|^2} \right) \left(I - \frac{2ww^H}{\|w\|^2} \right) = I - \frac{2ww^H}{\|w\|^2} - \frac{2ww^H}{\|w\|^2} + \frac{4\|w\|^2 ww^H}{\|w\|^4} = I, \quad (2.67)$$

i.e., U is unitary. We want to choose w so that

$$U^H x = Ux = x - \frac{2w^H x}{\|w\|^2} w = \gamma \|x\| e_1 \quad (2.68)$$

where $|\gamma| = 1$. Multiplying equation (2.68) by x^H , we get

$$x^H Ux = x^H U^H x = \overline{x^H Ux} = \gamma \|x\| \overline{x_1}. \quad (2.69)$$

Since $x^H Ux$ is real, it follows that $\gamma \overline{x_1}$ is real. If $x_1 = |x_1| e^{i\theta_1}$, then γ must have the form

$$\gamma = \pm e^{i\theta_1}. \quad (2.70)$$

It follows from equation (2.68) that w must be proportional to the vector $x \mp e^{i\theta_1} \|x\| e_1$. Since multiplying w by a constant factor doesn't change U , we take

$$w = x \mp e^{i\theta_1} \|x\| e_1. \quad (2.71)$$

Again, to avoid accuracy problems, we choose the plus sign in the above formula, i.e.,

$$w = x + e^{i\theta_1} \|x\| e_1. \quad (2.72)$$

It follows from this definition that

$$\begin{aligned} \|w\|^2 &= (x^H + e^{-i\theta_1} \|x\| e_1^T)(x + e^{-i\theta_1} \|x\| e_1) \\ &= \|x\|^2 + |x_1| \|x\| + |x_1| \|x\| + \|x\|^2 = 2\|x\|(\|x\| + |x_1|) \end{aligned} \quad (2.73)$$

and

$$w^H x = (x^H + e^{-i\theta_1} \|x\| e_1^T)x = \|x\|^2 + e^{-i\theta_1} x_1 \|x\| = \|x\|(\|x\| + |x_1|). \quad (2.74)$$

Therefore,

$$\frac{2w^H x}{\|w\|^2} = 1, \quad (2.75)$$

and hence

$$Ux = x - w = x - (x + e^{i\theta_1} \|x\| e_1) = -e^{i\theta_1} \|x\| e_1. \quad (2.76)$$

This alternate form for the Householder matrix has the advantage that it is Hermitian and that the multiplier of ww^H is real. However, it can't in general map a given vector x into a real multiple of e_1 . Both EISPACK and LINPACK use elementary unitary matrices similar to this. The LAPACK form is not Hermitian, involves a complex multiplier of ww^H , but can produce a real multiple of e_1 when acting on x . As stated before, this can be a big advantage when reducing matrices to triangular form prior to an eigenvalue computation.

where $c^2 + s^2 = 1$. The matrix $G(i, j)$ is clearly orthogonal. In terms of components

$$G(i, j)_{kl} = \begin{cases} 1 & k = l, k \neq i \text{ and } k \neq j \\ c & k = l, k = i \text{ or } k = j \\ -s & k = i, l = j \\ s & k = j, l = i \\ 0 & \text{otherwise} \end{cases}. \quad (2.78)$$

Multiplying a vector by $G(i, j)$ only affects the i and j components. If $y = G(i, j)x$, then

$$y_k = \begin{cases} x_k & k \neq i \text{ and } k \neq j \\ cx_i - sx_j & k = i \\ sx_i + cx_j & k = j \end{cases}. \quad (2.79)$$

Suppose we want to make $y_j = 0$. We can do this by setting

$$c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}} \quad \text{and} \quad s = \frac{-x_j}{\sqrt{x_i^2 + x_j^2}}. \quad (2.80)$$

With this choice for c and s , y becomes

$$y_k = \begin{cases} x_k & k \neq i \text{ and } k \neq j \\ \sqrt{x_i^2 + x_j^2} & k = i \\ 0 & k = j \end{cases}. \quad (2.81)$$

Multiplying a matrix A on the left by $G(i, j)$ only alters rows i and j . Similarly, Multiplying A on the right by $G(i, j)$ only alters columns i and j

2.3.5 Complex Givens Rotations

For the complex case we replace R in the previous section by

$$R = \begin{pmatrix} c & -\bar{s} \\ s & c \end{pmatrix} \quad \text{where } c \text{ is real.} \quad (2.82)$$

It can be easily verified that R is unitary if and only if c and s satisfy

$$c^2 + |s|^2 = 1.$$

Given a 2-vector x , we want to choose R so that Rx is a multiple of e_1 . For R unitary, we must have

$$Rx = \gamma \|x\| e_1 \quad \text{where } |\gamma| = 1. \quad (2.83)$$

Multiplying equation (2.83) by R^H , we get

$$x = RR^H x = \gamma \|x\| R^H e_1 = \gamma \|x\| \begin{pmatrix} c \\ -s \end{pmatrix} \quad (2.84)$$

or

$$c = \frac{x_1}{\gamma \|x\|} \quad \text{and} \quad s = \frac{-x_2}{\gamma \|x\|}. \quad (2.85)$$

We define $\text{sign}(u)$ for u complex by

$$\text{sign}(u) = \begin{cases} u/|u| & u \neq 0 \\ 1 & u = 0. \end{cases} \quad (2.86)$$

If c is to be real, γ must have the form

$$\gamma = \epsilon \text{sign}(x_1) \quad \epsilon = \pm 1.$$

With this choice of γ , c and s become

$$c = \frac{|x_1|}{\epsilon \|x\|} \quad \text{and} \quad s = \frac{-x_2}{\epsilon \text{sign}(x_1) \|x\|}. \quad (2.87)$$

If we want the complex case to reduce to the real case when x_1 and x_2 are real, then we can choose $\epsilon = \text{sign}(\text{Re}(x_1))$. As before, we can construct $G(i, j)$ by replacing the (i, i) and (j, j) components of the identity matrix by c , the (i, j) component by $-\bar{s}$, and the (j, i) component by s . In the expressions for c and s in equation (2.87), we replace x_1 by x_i , x_2 by x_j , and $\|x\|$ by $\sqrt{|x_i|^2 + |x_j|^2}$.

2.3.6 QR Factorization Using Householder Reflectors

Let A be an $m \times n$ matrix with $m > n$. Let Q_1 be a Householder matrix that maps the first column of A into a multiple of e_1 . Then $Q_1 A$ will have zeroes below the diagonal in the first column. Now let

$$Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & \hat{Q}_2 \end{pmatrix}$$

where \hat{Q}_2 is an $(m-1) \times (m-1)$ Householder matrix that will zero out the entries below the diagonal in the second column of $Q_1 A$. Continuing in this manner, we can construct Q_2, \dots, Q_{n-1} so that

$$Q_{n-1} \cdots Q_1 A = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} \quad (2.88)$$

where \hat{R} is an $n \times n$ triangular matrix. The matrices Q_k have the form

$$Q_k = \begin{pmatrix} I & 0 \\ 0 & \hat{Q}_k \end{pmatrix} \quad (2.89)$$

where \hat{Q}_k is an $(m - k + 1) \times (m - k + 1)$ Householder matrix. If we define

$$Q^H = Q_{n-1} \cdots Q_1 \quad \text{and} \quad R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}, \quad (2.90)$$

then equation (2.88) can be written

$$Q^H A = R. \quad (2.91)$$

Moreover, since each Q_k is unitary, we have

$$Q^H Q = (Q_{n-1} \cdots Q_1)(Q_1^H \cdots Q_{n-1}^H) = I, \quad (2.92)$$

i.e., Q is unitary. Therefore, equation (2.91) can be written

$$A = QR. \quad (2.93)$$

Equation (2.93) is the desired factorization. The operations count for this factorization is approximately mn^2 where an operation is an addition and a multiplication. In practice it is not necessary to construct the matrix Q explicitly. Usually only the vectors v defining each Q_k are saved.

If \hat{Q} is the matrix consisting of the first n columns of Q , then

$$A = \hat{Q} \hat{R} \quad (2.94)$$

where \hat{Q} is a $m \times n$ matrix with orthonormal columns and \hat{R} is a $n \times n$ upper triangular matrix. The factorization in equation (2.94) is the reduced QR factorization.

2.3.7 Uniqueness of the Reduced QR Factorization

In this section we will show that a matrix A of full rank has a unique reduced QR factorization if we require that the triangular matrix R has positive diagonals. All other reduced QR factorizations of A are simply related to this one with positive diagonals.

The reduced QR factorization can be written

$$A = (a_1, a_2, \dots, a_n) = (q_1, q_2, \dots, q_n) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}. \quad (2.95)$$

If A has full rank, then all of the diagonal elements r_{jj} must be nonzero. Equating columns in equation (2.95), we get

$$a_j = \sum_{k=1}^j r_{kj} q_k = r_{jj} q_j + \sum_{k=1}^{j-1} r_{kj} q_k$$

or

$$q_j = \frac{1}{r_{jj}} \left(a_j - \sum_{k=1}^{j-1} r_{kj} q_k \right). \quad (2.96)$$

When $j = 1$ equation (2.96) reduces to

$$q_1 = \frac{a_1}{r_{11}}. \quad (2.97)$$

Since q_1 must have unit norm, it follows that

$$|r_{11}| = \|a_1\|. \quad (2.98)$$

Equations (2.97) and (2.98) determine q_1 and r_{11} up to a factor having absolute value one, i.e., there is a d_1 with $|d_1| = 1$ such that

$$r_{11} = d_1 \hat{r}_{11} \quad q_1 = \frac{\hat{q}_1}{d_1}$$

where $\hat{r}_{11} = \|a_1\|$ and $\hat{q}_1 = a_1/\hat{r}_{11}$.

For $j = 2$, equation (2.96) becomes

$$q_2 = \frac{1}{r_{22}} (a_2 - r_{12} q_1).$$

Since the columns q_1 and q_2 must be orthonormal, it follows that

$$0 = q_1^H q_2 = \frac{1}{r_{22}} (q_1^H a_2 - r_{12})$$

and hence that

$$r_{12} = q_1^H a_2 = d_1 \hat{q}_1^H a_2. \quad (2.99)$$

Here we have used the fact that $\overline{d_1} = 1/d_1$. Since q_2 has unit norm, it follows that

$$1 = \|q_2\| = \frac{1}{|r_{22}|} \|a_2 - r_{12} q_1\| = \frac{1}{|r_{22}|} \|a_2 - (d_1 \hat{q}_1^H a_2) \hat{q}_1 / d_1\| = \frac{1}{|r_{22}|} \|a_2 - (\hat{q}_1^H a_2) \hat{q}_1\|$$

and hence that

$$|r_{22}| = \|a_2 - (\hat{q}_1^H a_2) \hat{q}_1\| \equiv \hat{r}_{22}.$$

Therefore, there exists a scalar d_2 with $|d_2| = 1$ such that

$$r_{22} = d_2 \hat{r}_{22} \quad \text{and} \quad q_2 = \hat{q}_2 / d_2$$

where $\hat{q}_2 = (a_2 - (\hat{q}_1^H a_2) \hat{q}_1) / \hat{r}_{22}$.

For $j = 3$, equation (2.96) becomes

$$q_3 = \frac{1}{r_{33}} (a_3 - r_{13} q_1 - r_{23} q_2).$$

Since the columns q_1, q_2 and q_3 must be orthonormal, it follows that

$$\begin{aligned} 0 &= q_1^H q_3 = \frac{1}{r_{33}}(q_1^H a_3 - r_{13}) \\ 0 &= q_2^H q_3 = \frac{1}{r_{33}}(q_2^H a_3 - r_{23}) \end{aligned}$$

and hence that

$$\begin{aligned} r_{13} &= q_1^H a_3 = d_1 \hat{q}_1^H a_3 \\ r_{23} &= q_2^H a_3 = d_2 \hat{q}_2^H a_3. \end{aligned}$$

Since q_3 has unit norm, it follows that

$$1 = \|q_3\| = \frac{1}{|r_{33}|} \|a_3 - r_{13}q_1 - r_{23}q_2\| = \frac{1}{|r_{33}|} \|a_3 - (\hat{q}_1^H a_3)\hat{q}_1 - (\hat{q}_2^H a_3)\hat{q}_2\|$$

and hence that

$$|r_{33}| = \|a_3 - (\hat{q}_1^H a_3)\hat{q}_1 - (\hat{q}_2^H a_3)\hat{q}_2\| \equiv \hat{r}_{33}.$$

Therefore, there exists a scalar d_3 with $|d_3| = 1$ such that

$$r_{33} = d_3 \hat{r}_{33} \quad \text{and} \quad q_3 = \hat{q}_3/d_3 \tag{2.100}$$

where $\hat{q}_3 = (a_3 - (\hat{q}_1^H a_3)\hat{q}_1 - (\hat{q}_2^H a_3)\hat{q}_2)/\hat{r}_{33}$. Continuing in this way we obtain the unitary matrix $\hat{Q} = (\hat{q}_1, \dots, \hat{q}_n)$ and the triangular matrix

$$\hat{R} = \begin{pmatrix} \hat{r}_{11} & \hat{r}_{12} & \cdots & \hat{r}_{1n} \\ & \hat{r}_{22} & \cdots & \hat{r}_{2n} \\ & & \ddots & \vdots \\ & & & \hat{r}_{nn} \end{pmatrix}$$

such that $A = \hat{Q}\hat{R}$ is the unique reduced QR factorization of A with R having positive diagonal elements. If $A = QR$ is any other reduced QR factorization of A , then

$$R = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \hat{R}$$

and

$$Q = \hat{Q} \begin{pmatrix} 1/d_1 & & \\ & \ddots & \\ & & 1/d_n \end{pmatrix} = \hat{Q} \begin{pmatrix} \overline{d_1} & & \\ & \ddots & \\ & & \overline{d_n} \end{pmatrix}$$

where $|d_1| = \dots = |d_n| = 1$.

2.3.8 Solution of Least Squares Problems

In this section we will show how to use the QR factorization to solve the least squares problem. Consider the system of linear equations

$$Ax = b \quad (2.101)$$

where A is an $m \times n$ matrix with $m > n$. In general there is no solution to this system of equations. Instead we seek to find an x so that $\|Ax - b\|$ is as small as possible. In view of the QR factorization, we have

$$\|Ax - b\|^2 = \|QRx - b\|^2 = \|Q(Rx - Q^H b)\|^2 = \|Rx - Q^H b\|^2. \quad (2.102)$$

We can write Q in the partitioned form $Q = (Q_1, Q_2)$ where Q_1 is an $m \times n$ matrix. Then

$$Rx - Q^H b = \begin{pmatrix} \hat{R}x \\ 0 \end{pmatrix} - \begin{pmatrix} Q_1^H b \\ Q_2^H b \end{pmatrix} = \begin{pmatrix} \hat{R}x - Q_1^H b \\ -Q_2^H b \end{pmatrix}. \quad (2.103)$$

It follows from equation (2.103) that

$$\|Rx - Q^H b\|^2 = \|\hat{R}x - Q_1^H b\|^2 + \|Q_2^H b\|^2. \quad (2.104)$$

Combining equations (2.102) and (2.104), we get

$$\|Ax - b\|^2 = \|\hat{R}x - Q_1^H b\|^2 + \|Q_2^H b\|^2. \quad (2.105)$$

It can be easily seen from this equation that $\|Ax - b\|$ is minimized when x is the solution of the triangular system

$$\hat{R}x = Q_1^H b \quad (2.106)$$

when such a solution exists. This is the standard way of solving least square systems. Later we will discuss the singular value decomposition (SVD) that will provide even more information relative to the least squares problem. However, the SVD is much more expensive to compute than the QR decomposition.

2.4 The Singular Value Decomposition

The Singular Value Decomposition (SVD) is one of the most important and probably one of the least well known of the matrix factorizations. It has many applications in statistics, signal processing, image compression, pattern recognition, weather prediction, and modal analysis to name a few. It is also a powerful diagnostic tool. For example, it provides approximations to the rank and the condition number of a matrix as well as providing orthonormal bases for both the range and the null space of a matrix. It also provides optimal low rank approximations to a matrix. The SVD is applicable to both square and rectangular matrices. In this regard it provides a general solution to the least squares problem.

The SVD was first discovered by differential geometers in connection with the analysis of bilinear forms. Eugenio Beltrami [1] (1873) and Camille Jordan [10] (1874) independently discovered that the singular values of the matrix associated with a bilinear form comprise a complete set of invariants for the form under orthogonal substitutions. The first proof of the singular value decomposition for rectangular and complex matrices seems to be by Eckart and Young [5] in 1939. They saw it as a generalization of the principal axis transformation for Hermitian matrices.

We will begin by deriving the SVD and presenting some of its most important properties. We will then discuss its application to least squares problems and matrix approximation problems. Following this we will show how singular values can be used to determine the condition of a matrix (how close the rows or columns are to being linearly dependent). We will conclude with a brief outline of the methods used to compute the SVD. Most of the methods are modifications of methods used to compute eigenvalues and vectors of a square matrix. The details of the computational methods are beyond the scope of this presentation, but we will provide references for those interested.

2.4.1 Derivation and Properties of the SVD

Theorem 1. (Singular Value Decomposition) *Let A be a nonzero $m \times n$ matrix. Then there exists an orthonormal basis u_1, \dots, u_m of m -vectors, an orthonormal basis v_1, \dots, v_n of n -vectors, and positive numbers $\sigma_1, \dots, \sigma_r$ such that*

1. u_1, \dots, u_r is a basis of the range of A
2. v_{r+1}, \dots, v_n is a basis of the null space of A
3. $A = \sum_{k=1}^r \sigma_k u_k v_k^H$.

Proof: $A^H A$ is a Hermitian $n \times n$ matrix that is positive semidefinite. Therefore, there is an orthonormal basis v_1, \dots, v_n and nonnegative numbers $\sigma_1^2, \dots, \sigma_n^2$ such

$$A^H A v_k = \sigma_k^2 v_k \quad k = 1, \dots, n. \quad (2.107)$$

Since A is nonzero, at least one of the eigenvalues σ_k^2 must be positive. Let the eigenvalues be arranged so that $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_r^2 > 0$ and $\sigma_{r+1}^2 = \dots = \sigma_n^2 = 0$. Consider now the vectors Av_1, \dots, Av_n . We have

$$(Av_i)^H Av_j = v_i^H A^H Av_j = \sigma_j^2 v_i^H v_j = 0 \quad i \neq j, \quad (2.108)$$

i.e., Av_1, \dots, Av_n are orthogonal. When $i = j$

$$\begin{aligned} \|Av_i\|^2 &= v_i^H A^H Av_i = \sigma_i^2 v_i^H v_i = \sigma_i^2 > 0 & i = 1, \dots, r \\ &= 0 & i > r. \end{aligned} \quad (2.109)$$

Thus, $Av_{r+1} = \dots = Av_n = 0$ and hence v_{r+1}, \dots, v_n belong to the null space of A . Define u_1, \dots, u_r by

$$u_i = (1/\sigma_i) Av_i \quad i = 1, \dots, r. \quad (2.110)$$

Then u_1, \dots, u_r is an orthonormal set of vectors in the range of A that span the range of A . Thus, u_1, \dots, u_r is a basis for the range of A . The dimension r of the range of A is called the *rank of A* . If $r < m$, we can extend the set u_1, \dots, u_r of orthonormal vectors to an orthonormal basis u_1, \dots, u_m of m -space using the Gram-Schmidt process. If x is an n -vector, we can write x in terms of the basis v_1, \dots, v_n as

$$x = \sum_{k=1}^n (v_k^H x) v_k. \quad (2.111)$$

It follows from equations (2.110) and (2.111) that

$$Ax = \sum_{k=1}^n (v_k^H x) A v_k = \sum_{k=1}^r (v_k^H x) \sigma_k u_k = \sum_{k=1}^r \sigma_k u_k v_k^H x. \quad (2.112)$$

Since x in equation (2.112) was arbitrary, we must have

$$A = \sum_{k=1}^r \sigma_k u_k v_k^H. \quad (2.113)$$

The representation of A in equation (2.113) is called the singular value decomposition (SVD). If x belongs to the null space of A ($Ax = 0$), then it follows from equation (2.112) and the linear independence of the vectors u_1, \dots, u_r that $v_k^H x = 0$ for $k = 1, \dots, r$. It then follows from equation (2.111) that

$$x = \sum_{k=r+1}^n (v_k^H x) v_k,$$

i.e., v_{r+1}, \dots, v_n span the null space of A . Since v_{r+1}, \dots, v_n are orthonormal vectors belonging to the null space of A , they form a basis for the null space of A .

We will now express the SVD in matrix form. Define $U = (u_1, \dots, u_m)$, $V = (v_1, \dots, v_n)$, and $S = \text{diag}(\sigma_1, \dots, \sigma_r)$. If $r < \min(m, n)$, then the SVD can be written in the matrix form

$$A = U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H. \quad (2.114)$$

If $r = m < n$, then the SVD can be written in the matrix form

$$A = U (S \ 0) V^H. \quad (2.115)$$

If $r = n < m$, then the SVD can be written in the matrix form

$$A = U \begin{pmatrix} S \\ 0 \end{pmatrix} V^H. \quad (2.116)$$

If $r = m = n$, then the SVD can be written in the matrix form

$$A = USV^H. \quad (2.117)$$

Generally we write the SVD in the form (2.114) with the understanding that some of the zero portions might collapse and disappear.

We next give a geometric interpretation of the SVD. For this purpose we will restrict ourselves to the real case. Let x be a point on the unit sphere, i.e., $\|x\| = 1$. Since u_1, \dots, u_r is a basis for the range of A , there exist numbers y_1, \dots, y_r such that

$$\begin{aligned} Ax &= \sum_{k=1}^r y_k u_k \\ &= \sum_{k=1}^r \sigma_k (v_k^T x) u_k. \end{aligned}$$

Therefore, $y_k = \sigma_k (v_k^T x)$, $k = 1, \dots, r$. Since the columns of V form an orthonormal basis, we have

$$x = \sum_{k=1}^n (v_k^T x) v_k.$$

Therefore,

$$\|x\|^2 = \sum_{k=1}^n (v_k^T x)^2 = 1.$$

It follows that

$$\frac{y_1^2}{\sigma_1^2} + \dots + \frac{y_r^2}{\sigma_r^2} = (v_1^T x)^2 + \dots + (v_r^T x)^2 \leq 1.$$

Here equality holds when $r = n$. Thus, the image of x lies on or interior to the hyper ellipsoid with semi axes $\sigma_1 u_1, \dots, \sigma_r u_r$. Conversely, if y_1, \dots, y_r satisfy

$$\frac{y_1^2}{\sigma_1^2} + \dots + \frac{y_r^2}{\sigma_r^2} \leq 1,$$

we define $\alpha^2 = 1 - \sum_{k=1}^r (y_k/\sigma_k)^2$ and

$$x = \sum_{k=1}^r \frac{y_k}{\sigma_k} v_k + \alpha v_{r+1}.$$

Since v_{r+1} is in the null space of A and $Av_k = \sigma_k u_k$ ($k \leq r$), it follows that

$$Ax = \sum_{k=1}^r \frac{y_k}{\sigma_k} Av_k + \alpha Av_{r+1} = \sum_{k=1}^r y_k u_k.$$

In addition,

$$\|x\|^2 = \sum_{k=1}^r \frac{y_k^2}{\sigma_k^2} + \alpha^2 = 1.$$

Thus, we have shown that the image of the unit sphere $\|x\| = 1$ under the mapping A is the hyper ellipsoid

$$\frac{y_1^2}{\sigma_1^2} + \cdots + \frac{y_r^2}{\sigma_r^2} \leq 1$$

relative to the basis u_1, \dots, u_r . When $r = n$, equality holds and the image is the surface of the hyper ellipsoid

$$\frac{y_1^2}{\sigma_1^2} + \cdots + \frac{y_r^2}{\sigma_n^2} = 1.$$

2.4.2 The SVD and Least Squares Problems

In least squares problems we seek an x that minimizes $\|Ax - b\|$. In view of the singular value decomposition, we have

$$\begin{aligned} \|Ax - b\|^2 &= \left\| U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H x - b \right\|^2 \\ &= \left\| U \left[\begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H x - U^H b \right] \right\|^2 \\ &= \left\| \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H x - U^H b \right\|^2. \end{aligned} \quad (2.118)$$

If we define

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = V^H x \quad (2.119)$$

$$\hat{b} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = U^H b. \quad (2.120)$$

then equation (2.118) can be written

$$\|Ax - b\|^2 = \left\| \begin{pmatrix} S y_1 - \hat{b}_1 \\ -\hat{b}_2 \end{pmatrix} \right\|^2 = \|S y_1 - \hat{b}_1\|^2 + \|\hat{b}_2\|^2. \quad (2.121)$$

It is clear from equation (2.121) that $\|Ax - b\|$ is minimized when $y_1 = S^{-1}\hat{b}_1$. Therefore, the y that minimizes $\|Ax - b\|$ is given by

$$y = \begin{pmatrix} S^{-1}\hat{b}_1 \\ y_2 \end{pmatrix} \quad y_2 \text{ arbitrary.} \quad (2.122)$$

In view of equation (2.119), the x that minimizes $\|Ax - b\|$ is given by

$$x = Vy = V \begin{pmatrix} S^{-1}\hat{b}_1 \\ y_2 \end{pmatrix} \quad y_2 \text{ arbitrary.} \quad (2.123)$$

Since V is unitary, it follows from equation (2.123) that

$$\|x\|^2 = \|S^{-1}\hat{b}_1\|^2 + \|y_2\|^2.$$

Thus, there is a unique x of minimum norm that minimizes $\|Ax - b\|$, namely the x corresponding to $y_2 = 0$. This x is given by

$$\begin{aligned} x &= V \begin{pmatrix} S^{-1}\hat{b}_1 \\ 0 \end{pmatrix} \\ &= V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} \\ &= V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H b. \end{aligned}$$

The matrix multiplying b on the right-hand-side of this equation is called the generalized inverse of A and is denoted by A^+ , i.e.,

$$A^+ = V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H. \quad (2.124)$$

Thus, the minimum norm solution of the least squares problem is given by $x = A^+b$. The $n \times m$ matrix A^+ plays the same role in least squares problems that A^{-1} plays in the solution of linear equations. We will now show that this definition of the generalized inverse gives the same result as the classical Moore-Penrose conditions.

Theorem 2. *If A has a singular value decomposition given by*

$$A = U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H,$$

then the matrix X defined by

$$X = A^+ = V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H$$

is the unique solution of the Moore-Penrose conditions:

1. $AXA = A$
2. $XAX = X$
3. $(AX)^H = AX$
4. $(XA)^H = XA$.

Proof:

$$\begin{aligned}
AXA &= U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H \\
&= U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} V^H \\
&= U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H \\
&= A,
\end{aligned}$$

i.e., X satisfies condition (1).

$$\begin{aligned}
XAX &= V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H \\
&= V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H \\
&= X,
\end{aligned}$$

i.e., X satisfies condition (2). Since

$$AX = U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H = U \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} U^H$$

and

$$XA = V \begin{pmatrix} S^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^H U \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} V^H = V \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} V^H,$$

it follows that both AX and XA are Hermitian, i.e., X satisfies conditions (3) and (4). To show uniqueness let us suppose that both X and Y satisfy the Moore-Penrose conditions. Then

$$\begin{aligned}
X &= XAX \quad \text{by (2)} \\
&= X(AX)^H = XX^H A^H \quad \text{by (3)} \\
&= XX^H (AYA)^H = XX^H A^H Y^H A^H \quad \text{by (1)} \\
&= XX^H A^H (AY)^H = XX^H A^H AY \quad \text{by (3)} \\
&= X(AX)^H AY = XAXAY \quad \text{by (3)} \\
&= XAY \quad \text{by (2)} \\
&= X(AYA)Y \quad \text{by (1)} \\
&= XA(YA)Y = XA(YA)^H Y = XAA^H Y^H Y \quad \text{by (4)} \\
&= (XA)^H A^H Y^H Y = A^H X^H A^H Y^H Y \quad \text{by (4)} \\
&= (AXA)^H Y^H Y = A^H Y^H Y \quad \text{by (1)} \\
&= (YA)^H Y = YAY \quad \text{by (4)} \\
&= Y \quad \text{by (2)}.
\end{aligned}$$

Thus, there is only one matrix X satisfying the Moore-Penrose conditions.

2.4.3 Singular Values and the Norm of a Matrix

Let A be an $m \times n$ matrix. By virtue of the SVD, we have

$$Ax = \sum_{k=1}^r \sigma_k (v_k^H x) u_k \quad \text{for any } n\text{-vector } x. \quad (2.125)$$

Since the vectors u_1, \dots, u_r are orthonormal, we have

$$\|Ax\|^2 = \sum_{k=1}^r \sigma_k^2 |v_k^H x|^2 \leq \sigma_1^2 \sum_{k=1}^r |v_k^H x|^2 \leq \sigma_1^2 \|x\|^2. \quad (2.126)$$

The last inequality comes from the fact that x has the expansion $x = \sum_{k=1}^n (v_k^H x) v_k$ in terms of the orthonormal basis v_1, \dots, v_n and hence

$$\|x\|^2 = \sum_{k=1}^n |v_k^H x|^2.$$

Thus, we have

$$\|Ax\| \leq \sigma_1 \|x\| \quad \text{for all } x. \quad (2.127)$$

Since $Av_1 = \sigma_1 u_1$, we have $\|Av_1\| = \sigma_1 = \sigma_1 \|v_1\|$. Hence,

$$\max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sigma_1, \quad (2.128)$$

i.e., A can't stretch the length of a vector by a factor greater than σ_1 . One of the definitions of the norm of a matrix is

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}. \quad (2.129)$$

It follows from equations (2.128) and (2.129) that $\|A\| = \sigma_1$ (the maximum singular value of A). If A is of full rank ($r=n$), then it follows by a similar argument that

$$\min_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \sigma_n.$$

If A is an $m \times n$ matrix and B is an $n \times p$ matrix, then for every p -vector x we have

$$\|ABx\| \leq \|A\| \|Bx\| \leq \|A\| \|B\| \|x\|$$

and hence $\|AB\| \leq \|A\| \|B\|$.

2.4.4 Low Rank Matrix Approximations

You can think of the rank of a matrix as a measure of redundancy. Matrices of low rank should have lots of redundancy and hence should be capable of specification by less parameters than the

total number of entries. For example, if the matrix consists of the pixel values of a digital image, then a lower rank approximation of this image should represent a form of image compression. We will make this concept more precise in this section.

One choice for a low rank approximation to A is the matrix $A_k = \sum_{i=1}^k \sigma_i u_i v_i^H$ for $k < r$. A_k is a truncated SVD expansion of A . Clearly

$$A - A_k = \sum_{i=k+1}^r \sigma_i u_i v_i^H. \quad (2.130)$$

Since the largest singular value of $A - A_k$ is σ_{k+1} , we have

$$\|A - A_k\| = \sigma_{k+1}. \quad (2.131)$$

Suppose B is another $m \times n$ matrix of rank k . Then the null space \mathcal{N} of B has dimension $n - k$. Let w_1, \dots, w_{n-k} be a basis for \mathcal{N} . The $n + 1$ n -vectors $w_1, \dots, w_{n-k}, v_1, \dots, v_{k+1}$ must be linearly dependent, i.e., there are constants $\alpha_1, \dots, \alpha_{n-k}$ and $\beta_1, \dots, \beta_{k+1}$, not all zero, such that

$$\sum_{i=1}^{n-k} \alpha_i w_i + \sum_{i=1}^{k+1} \beta_i v_i = 0.$$

Not all of the α_i can be zero since v_1, \dots, v_{k+1} are linearly independent. Similarly, not all of the β_i can be zero. Therefore, the vector h defined by

$$h = \sum_{i=1}^{n-k} \alpha_i w_i = - \sum_{i=1}^{k+1} \beta_i v_i$$

is a nonzero vector that belongs to both \mathcal{N} and $\langle v_1, \dots, v_{k+1} \rangle$. By proper scaling, we can assume that h is a vector with unit norm. Since h belongs to $\langle v_1, \dots, v_{k+1} \rangle$, we have

$$h = \sum_{i=1}^{k+1} (v_i^H h) v_i. \quad (2.132)$$

Therefore,

$$\|h\|^2 = \sum_{i=1}^{k+1} |v_i^H h|^2. \quad (2.133)$$

Since $Av_i = \sigma_i u_i$ for $i = 1, \dots, r$, it follows from equation (2.132) that

$$Ah = \sum_{i=1}^{k+1} (v_i^H h) Av_i = \sum_{i=1}^{k+1} (v_i^H h) \sigma_i u_i. \quad (2.134)$$

Therefore,

$$\|Ah\|^2 = \sum_{i=1}^{k+1} |v_i^H h|^2 \sigma_i^2 \geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} |v_i^H h|^2 = \sigma_{k+1}^2 \|h\|^2. \quad (2.135)$$

Since h belongs to the null space \mathcal{N} , we have

$$\|A - B\|^2 \geq \|(A - B)h\|^2 = \|Ah\|^2 \geq \sigma_{k+1}^2 \|h\|^2 = \sigma_{k+1}^2. \quad (2.136)$$

Combining equations (2.131) and (2.136), we obtain

$$\|A - B\| \geq \sigma_{k+1} = \|A - A_k\|. \quad (2.137)$$

Thus, A_k is the rank k matrix that is closest to A .

2.4.5 The Condition Number of a Matrix

Suppose A is an $n \times n$ invertible matrix and x is the solution of the system of equations $Ax = b$. We want to see how sensitive x is to perturbations of the matrix A . Let $x + \delta x$ be the solution to the perturbed system $(A + \delta A)(x + \delta x) = b$. Expanding the left-hand-side of this equation and neglecting the second order perturbations $\delta A \delta x$, we get

$$\delta A x + A \delta x = 0 \quad \text{or} \quad \delta x = -A^{-1} \delta A x. \quad (2.138)$$

It follows from equation (2.138) that

$$\|\delta x\| \leq \|A^{-1}\| \|\delta A\| \|x\|$$

or

$$\frac{\|\delta x\|/\|x\|}{\|\delta A\|/\|A\|} \leq \|A^{-1}\| \|A\|. \quad (2.139)$$

The quantity $\|A^{-1}\| \|A\|$ is called the *condition number of A* and is denoted by $\kappa(A)$, i.e.,

$$\kappa(A) = \|A^{-1}\| \|A\|.$$

Thus, equation (2.139) can be written

$$\frac{\|\delta x\|/\|x\|}{\|\delta A\|/\|A\|} \leq \kappa(A). \quad (2.140)$$

We have seen previously that $\|A\| = \sigma_1$, the largest singular value. Since A^{-1} has the singular value decomposition $A^{-1} = VS^{-1}U^H$, it follows that $\|A^{-1}\| = 1/\sigma_n$. Therefore, the condition number is given by

$$\kappa(A) = \frac{\sigma_1}{\sigma_n}. \quad (2.141)$$

The condition number is sort of an aspect ratio of the hyper ellipsoid that A maps the unit sphere into.

2.4.6 Computation of the SVD

The methods for calculating the SVD are all variations of methods used to calculate eigenvalues and eigenvectors of Hermitian Matrices. The most natural procedure would be to follow the derivation of the SVD and compute the squares of the singular values and the unitary matrix V by solving the eigenproblem for $A^H A$. The U matrix would then be obtained from AV . Unfortunately, this procedure is not very accurate due to the fact that the singular values of $A^H A$ are the squares of the singular values of A . As a result the ratio of largest to smallest singular value can be much larger for $A^H A$ than for A . There are, however, implicit methods that solve the eigenproblem for $A^H A$ without ever explicitly forming $A^H A$. Most of the SVD algorithms first reduce A to bidiagonal form (all elements zero except the diagonal and first superdiagonal). This can be accomplished using householder reflections alternately on the left and right as shown in figure 2.2.

$$\begin{aligned}
 A_1 = U_1^H A &= \begin{pmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{pmatrix} \longrightarrow A_2 = A_1 V_1 = \begin{pmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{pmatrix} \longrightarrow \\
 A_3 = U_2^H A_2 &= \begin{pmatrix} x & x & 0 & 0 \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{pmatrix} \longrightarrow A_4 = A_3 V_2 = \begin{pmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{pmatrix} \longrightarrow \\
 A_5 = U_3^H A_4 &= \begin{pmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{pmatrix} \longrightarrow A_6 = U_4^H A_5 = \begin{pmatrix} x & x & 0 & 0 \\ 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{pmatrix}.
 \end{aligned}$$

Figure 2.2: Householder reduction of a matrix to bidiagonal form.

Since the application of the Householder reflections on the right don't try to zero all the elements to the right of the diagonal, they don't affect the zeroes already obtained in the columns. We have seen that, even in the complex case, the Householder matrices can be chosen so that the resulting bidiagonal matrix is real. Notice also that when the number of rows m is greater than the number of columns n , the reduction produces zero rows after row n . Similarly, when $n > m$, the reduction produces zero columns after column m . If we replace the products of the Householder reflections by the unitary matrices \hat{U} and \hat{V} , the reduction to a bidiagonal B can be written as

$$B = \hat{U}^H A \hat{V} \quad \text{or} \quad A = \hat{U} B \hat{V}^H. \quad (2.142)$$

If B has the SVD $B = \bar{U} \Sigma \bar{V}^T$, then A has the SVD

$$A = \hat{U}(\bar{U} \Sigma \bar{V}^T) \hat{V}^H = (\hat{U} \bar{U}) \Sigma (\hat{V} \bar{V})^H = U \Sigma V^H,$$

where $U = \hat{U} \bar{U}$ and $V = \hat{V} \bar{V}$. Thus, it is sufficient to find the SVD of the real bidiagonal matrix B . Moreover, it is not necessary to carry along the zero rows or columns of B . For if the square portion B_1 of B has the SVD $B_1 = U_1 \Sigma_1 V_1^T$, then

$$B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix} = \begin{pmatrix} U_1 \Sigma_1 V_1^T \\ 0 \end{pmatrix} = \begin{pmatrix} U_1 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} V_1^T \quad (2.143)$$

or

$$B = (B_1, 0) = (U_1 \Sigma_1 V_1^T, 0) = U_1 (\Sigma_1, 0) \begin{pmatrix} V_1 & 0 \\ 0 & I \end{pmatrix}^T. \quad (2.144)$$

Thus, it is sufficient to consider the computation of the SVD for a real, square, bidiagonal matrix B .

In addition to the implicit methods of finding the eigenvalues of $B^T B$, some methods look instead at the symmetric matrix $\begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$. If the SVD of B is $B = U \Sigma V^T$, then $\begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$ has the eigenequation

$$\begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} V & V \\ U & -U \end{pmatrix} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{pmatrix}. \quad (2.145)$$

In addition, the matrix $\begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix}$ can be reduced to a real tridiagonal matrix T by the relation

$$T = P^T B P \quad (2.146)$$

where $P = (e_1, e_{n+1}, e_2, e_{n+2}, \dots, e_n, e_{2n})$ is a permutation matrix formed by a rearrangement of the columns e_1, e_2, \dots, e_{2n} of the $2n \times 2n$ identity matrix. The matrix P is unitary and is sometimes called the *perfect shuffle* since its operation on a vector mimics a perfect card shuffle of the components. The algorithms based on this double size Symmetric matrix don't actually form the double size matrix, but make efficient use of the symmetries involved in this eigenproblem. For those interested in the details of the various SVD algorithms, I would refer you to the book by Demmel [4].

In Matlab the SVD can be obtained by the call `[U,S,V]=svd(A)`. In LAPACK the general driver routines for the SVD are SGESVD, DGESVD, and CGESVD depending on whether the matrix is real single precision, real double precision, or complex.

Chapter 3

Eigenvalue Problems

Eigenvalue problems occur quite often in Physics. For example, in Quantum Mechanics eigenvalues correspond to certain energy states; in structural mechanics problems eigenvalues often correspond to resonance frequencies of the structure; and in time evolution problems eigenvalues are often related to the stability of the system.

Let A be an $m \times m$ square matrix. A nonzero vector x is an eigenvector of A and λ is its corresponding eigenvalue, if

$$Ax = \lambda x.$$

The set of vectors

$$\mathcal{V}_\lambda = \{x : Ax = \lambda x\}$$

is a subspace called the eigenspace corresponding to λ . The equation $Ax = \lambda x$ is equivalent to $(A - \lambda I)x = 0$. If λ is an eigenvalue, then the matrix $A - \lambda I$ is singular and hence

$$\det(A - \lambda I) = 0.$$

Thus, the eigenvalues of A are roots of a polynomial equation of order n . This polynomial equation is called the characteristic equation of A . Conversely, if $p(z) = a_0 + a_1z + \cdots + a_{n-1}z^{n-1} + a_nz^n$ is an arbitrary polynomial of degree n ($a_n \neq 0$), then the matrix

$$\begin{pmatrix} 0 & & & & -a_0/a_n \\ 1 & 0 & & & -a_1/a_n \\ & 1 & 0 & & -a_2/a_n \\ & & 1 & \cdots & \vdots \\ & & & \ddots & 0 \\ & & & & 1 & -a_{n-1}/a_n \end{pmatrix}$$

has $p(z) = 0$ as its characteristic equation.

In some problems an eigenvalue λ might correspond to a multiple root of the characteristic equation. The multiplicity of the root λ is called its algebraic multiplicity. The dimension of the space

\mathcal{V}_λ is called its geometric multiplicity. If for some eigenvalue λ of A , the geometric multiplicity of λ does not equal its algebraic multiplicity, this eigenvalue is said to be defective. A matrix with one or more defective eigenvalues is said to be a defective matrix. An example of a defective matrix is the matrix

$$\begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}.$$

This matrix has the single eigenvalue 2 with algebraic multiplicity 3. However, the eigenspace corresponding to the eigenvalue 2 has dimension 1. All the eigenvectors are multiples of e_1 . In these notes we will only consider eigenvalue problems involving Hermitian matrices ($A^H = A$). We will see that all such matrices are non defective.

If S is a nonsingular $m \times m$ matrix, then the matrix $S^{-1}AS$ is said to be similar to A . Since

$$\det(S^{-1}AS - \lambda I) = \det(S^{-1}(A - \lambda I)S) = \det(S^{-1}) \det(A - \lambda I) \det(S) = \det(A - \lambda I),$$

it follows that $S^{-1}AS$ and A have the same characteristic equation and hence the same eigenvalues. It can be shown that a Hermitian matrix A always has a complete set of orthonormal eigenvectors. If we form the unitary matrix U whose columns are the eigenvectors belonging to this orthonormal set, then

$$AU = U\Lambda \quad \text{or} \quad U^H AU = \Lambda \tag{3.1}$$

where Λ is a diagonal matrix whose diagonal entries are the eigenvalues. Thus, a Hermitian matrix is similar a diagonal matrix. Since a diagonal matrix is clearly non defective, it follows that all Hermitian matrices are non defective.

If e is a unit eigenvector of the Hermitian matrix A and λ is the corresponding eigenvalue, then

$$Ae = \lambda e \quad \text{and hence} \quad \lambda = e^H Ae.$$

It follows that $\bar{\lambda} = (e^H Ae)^H = e^H A^H e = e^H Ae = \lambda$, i.e., the eigenvalues of a Hermitian matrix are real.

It was shown by Abel, Galois and others in the nineteenth century that there can be no algebraic expression for the roots of a polynomial equation whose order is greater than four. Since eigenvalues are roots of the characteristic equation and since the roots of any polynomial are the eigenvalues of some matrix, there can be no purely algebraic method for computing eigenvalues. Thus, algorithms for finding eigenvalues must at some stage be iterative in nature. The methods to be discussed here first reduce the Hermitian matrix A to a real, symmetric, tridiagonal matrix T by means of a unitary similarity transformation. The eigenvalues of T are then found using certain iterative procedures. The most common iterative procedures are the QR algorithm and the divide-and-conquer algorithm.

3.1 Reduction to Tridiagonal Form

The reduction to tridiagonal form can be done with Householder reflectors. I will illustrate the procedure with a 5×5 matrix A , i.e.,

$$A = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}.$$

We can zero out the elements in the first column from row three to the end using a Householder reflector of the form

$$U_1 = \begin{pmatrix} 1 & 0 \\ 0 & Q_1 \end{pmatrix}.$$

This reflector does not alter the elements of the first row. Thus, multiplying $U_1 A$ on the right by U_1^H zeros out the elements of the first row from column three on and doesn't affect the first column. Hence,

$$Q_1 A Q_1^H = \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{pmatrix}.$$

Moreover, the Householder reflector can be chosen so that the 12 element and the 21 element are real. We can continue in this manner to zero out the elements below the first subdiagonal and above the first superdiagonal. Furthermore, the Householder reflectors can be chosen so that the super and sub diagonals are real. The diagonals of the resulting tridiagonal matrix are real since the transformations have preserved the Hermitian property. Collecting the products of the Householder reflectors into a unitary matrix U , we have

$$U A U^H = T \quad \text{or} \quad A = U^H T U$$

where T is a real, symmetric, tridiagonal matrix. Since A and T are similar, they have the same eigenvalues. Thus, we only need eigenvalue routines for real symmetric matrices. In the following sections we will assume that the matrix A is real and symmetric

3.2 The Power Method

The power method is one of the oldest methods for obtaining the eigenvectors of a matrix. It is no longer used for this purpose because of its slow convergence, but it does underlie some of the practical algorithms. Let v_1, v_2, \dots, v_n be an orthonormal basis of eigenvectors of the matrix A

and let $\lambda_1, \dots, \lambda_n$ be the corresponding eigenvalues. We will assume that the eigenvalues and eigenvectors are so ordered that

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|.$$

We will assume further that $|\lambda_1| > |\lambda_2|$. Let v be an arbitrary vector with $\|v\| = 1$. Then there exist constants c_1, \dots, c_n such that

$$v = c_1 v_1 + \dots + c_n v_n. \quad (3.2)$$

We will make the further assumption that $c_1 \neq 0$. Successively applying A to equation (3.2), we obtain

$$A^k v = c_1 A^k v_1 + \dots + c_n A^k v_n = c_1 \lambda_1^k v_1 + \dots + c_n \lambda_n^k v_n. \quad (3.3)$$

You can see from equation (3.3) that the term $c_1 \lambda_1^k v_1$ will eventually dominate and thus $A^k v$, if properly scaled at each step to prevent overflow, will approach a multiple of the eigenvector v_1 . This convergence can be slow if there are other eigenvalues close in magnitude to λ_1 . The condition $c_1 \neq 0$ is equivalent to the condition

$$\langle v \rangle \cap \langle v_2, \dots, v_n \rangle = \{0\}.$$

3.3 The Rayleigh Quotient

The Rayleigh quotient of a vector x is the real number

$$r(x) = \frac{x^T A x}{x^T x}.$$

If x is an eigenvector of A corresponding to the eigenvalue λ , then $r(x) = \lambda$. If x is any nonzero vector, then

$$\begin{aligned} \|Ax - \alpha x\|^2 &= (x^T A^T - \alpha x^T)(Ax - \alpha x) \\ &= x^T A^T A x - 2\alpha x^T A x + \alpha^2 x^T x \\ &= x^T A^T A x - 2\alpha r(x) x^T x + \alpha^2 x^T x + r^2(x) x^T x - r^2(x) x^T x \\ &= x^T A^T A x + x^T x (\alpha - r(x))^2 - r^2(x) x^T x. \end{aligned}$$

Thus, $\alpha = r(x)$ minimizes $\|Ax - \alpha x\|$. If x is an approximate eigenvector, then $r(x)$ is an approximate eigenvalue.

3.4 Inverse Iteration with Shifts

For any μ that is not an eigenvalue of A , the matrix $(A - \mu I)^{-1}$ has the same eigenvectors as A and has eigenvalues $(\lambda_j - \mu)^{-1}$ where $\{\lambda_j\}$ are the eigenvalues of A . Suppose μ is close to the

eigenvalue λ_i . Then $(\lambda_i - \mu)^{-1}$ will be large compared to $(\lambda_j - \mu)^{-1}$ for $j \neq i$. If we apply power iteration to $(A - \mu I)^{-1}$, the process will converge to a multiple of the eigenvector v_i corresponding to λ_i . This procedure is called inverse iteration with shifts. Although the power method is not used in practice, the inverse power method with shifts is frequently used to compute eigenvectors once an approximate eigenvalue has been obtained.

3.5 Rayleigh Quotient Iteration

The Rayleigh quotient can be used to obtain the shifts at each stage of inverse iteration. The procedure can be summarized as follows.

1. Choose a starting vector $v^{(0)}$ of unit magnitude.
2. Let $\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$ be the corresponding Rayleigh quotient.
3. For $k = 1, 2, \dots$
 - Solve $(A - \lambda^{(k-1)})w = v^{(k-1)}$ for w , i.e., compute $(A - \lambda^{(k-1)})^{-1} v^{(k-1)}$.
 - Normalize w to obtain $v^{(k)} = w / \|w\|$.
 - Let $\lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ be the corresponding Rayleigh quotient.

It can be shown that the convergence of Rayleigh quotient iteration is ultimately cubic. Cubic convergence triples the number of significant digits on each iteration.

3.6 The Basic QR Method

The QR method was discovered independently by Francis [6] and Kublanovskaya [11] in 1961. It is one of the standard methods for finding eigenvalues. The discussion in this section is based largely on the paper *Understanding the QR Algorithm* by Watkins [13]. As before, we will assume that the matrix A is real and symmetric. Therefore, there is an orthonormal basis v_1, \dots, v_n such that $A v_j = \lambda_j v_j$ for each j . We will assume that the eigenvalues λ_j are ordered so that $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

The QR algorithm can be summarized as follows:

1. Choose $A_0 = A$
2. For $m = 1, 2, \dots$

$$A_{m-1} = Q_m R_m \quad \text{QR factorization}$$

$$A_m = R_m Q_m$$
3. Stop when A_m is approximately diagonal.

It is probably not obvious what this algorithm has to do with eigenvalues. We will show that the QR method is a way of organizing simultaneous iteration, which in turn is a multivector generalization of the power method.

We can apply the power method to subspaces as well as to single vectors. Suppose S is a k -dimensional subspace. We can compute the sequence of subspaces S, AS, A^2S, \dots . Under certain conditions this sequence will converge to the subspace spanned by the eigenvectors v_1, v_2, \dots, v_k corresponding to the k largest eigenvalues of A . We will not provide a rigorous convergence proof, but we will attempt to make this result seem plausible. Assume that $|\lambda_k| > |\lambda_{k+1}|$ and define the subspaces

$$T = \langle v_1, \dots, v_k \rangle \quad U = \langle v_{k+1}, \dots, v_n \rangle .$$

We will first show that all the null vectors of A lie in U . Suppose v is a null vector of A , i.e., $Av = 0$. We can expand v in terms of the basis v_1, \dots, v_n giving

$$v = c_1 v_1 + \dots + c_k v_k + c_{k+1} v_{k+1} + \dots + c_n v_n .$$

Thus,

$$Av = c_1 \lambda_1 v_1 + \dots + c_k \lambda_k v_k + c_{k+1} \lambda_{k+1} v_{k+1} + \dots + c_n \lambda_n v_n = 0 .$$

Since the vectors $\{v_j\}$ are linearly independent and $|\lambda_1| \geq \dots \geq |\lambda_k| > 0$, it follows that $c_1 = c_2 = \dots = c_k = 0$, i.e., v belongs to the subspace U . We will now make the additional assumption $S \cap U = \{0\}$. This assumption is analogous to the assumption $c_1 \neq 0$ in the power method. If x is a nonzero vector in S , then we can write

$$x = c_1 v_1 + c_2 v_2 + \dots + c_k v_k \quad \text{(component in } T)$$

$$+ c_{k+1} v_{k+1} + \dots + c_n v_n . \quad \text{(component in } U)$$

Thus,

$$A^m x / (\lambda_k)^m = c_1 (\lambda_1 / \lambda_k)^m v_1 + \dots + c_{k-1} (\lambda_{k-1} / \lambda_k)^m v_{k-1} + c_k v_k$$

$$+ c_{k+1} (\lambda_{k+1} / \lambda_k)^m v_{k+1} + \dots + c_n (\lambda_n / \lambda_k)^m v_n .$$

Since x doesn't belong to U , at least one of the coefficients c_1, \dots, c_k must be nonzero. Notice that the first k terms on the right-hand-side do not decrease in absolute value as $m \rightarrow \infty$ whereas the remaining terms approach zero. Thus, $A^m x$, if properly scaled, approaches the subspace T as $m \rightarrow \infty$. In the limit $A^m S$ must approach a subspace of T . Since $S \cap U = \{0\}$, A can have no null

vectors in S . Thus, A is invertible on S . It follows that all of the subspaces $A^m S$ have dimension k and hence the limit can not be a proper subspace of T , i.e., $A^m S \rightarrow T$ as $m \rightarrow \infty$.

Numerically, we can't iterate on an entire subspace. Therefore, we pick a basis of this subspace and iterate on this basis. Let q_1^0, \dots, q_k^0 be a basis of S . Since A is invertible on S , Aq_1^0, \dots, Aq_k^0 is a basis of AS . Similarly, $A^m q_1^0, \dots, A^m q_k^0$ is a basis of $A^m S$ for all m . Thus, in principle we can iterate on a basis of S to obtain bases for AS, A^2S, \dots . However, for large m these bases become ill-conditioned since all the vectors tend to point in the direction of the eigenvector corresponding to the eigenvalue of largest absolute value. To avoid this we orthonormalize the basis at each step. Thus, given an orthonormal basis q_1^m, \dots, q_k^m of $A^m S$, we compute Aq_1^m, \dots, Aq_k^m and then orthonormalize these vectors (using something like the Gram-Schmidt process) to obtain an orthonormal basis $q_1^{m+1}, \dots, q_k^{m+1}$ of $A^{m+1}S$. This process is called simultaneous iteration. Notice that this process of orthonormalization has the property

$$\langle Aq_1^m, \dots, Aq_i^m \rangle = \langle q_1^{m+1}, \dots, q_i^{m+1} \rangle \quad \text{for } i = 1, \dots, k.$$

Let us consider now what happens when we apply simultaneous iteration to the complete set of orthonormal vectors e_1, \dots, e_n where e_k is the k -th column of the identity matrix. Let us define

$$S_k = \langle e_1, \dots, e_k \rangle, \quad T_k = \langle v_1, \dots, v_k \rangle, \quad U_k = \langle v_{k+1}, \dots, v_n \rangle$$

for $k = 1, 2, \dots, n-1$. We also assume that $S_k \cap U_k = \{0\}$ and $|\lambda_k| > |\lambda_{k+1}| > 0$ for each $1 \leq k \leq n-1$. It follows from our previous discussion that $A^m S_k \rightarrow T_k$ as $m \rightarrow \infty$. In terms of bases, the orthonormal vectors q_1^m, \dots, q_k^m will converge to an orthonormal basis q_1, \dots, q_n such that $T_k = \langle q_1, \dots, q_k \rangle$ for each $k = 1, \dots, n-1$. Each of the subspaces T_k is invariant under A , i.e., $AT_k \subset T_k$. We will now look at a property of invariant subspaces. Suppose T is an invariant subspace of A . Let $Q = (Q_1, Q_2)$ be an orthogonal matrix such that the columns of Q_1 is a basis of T . Then

$$Q^T A Q = \begin{pmatrix} Q_1^T A Q_1 & Q_1^T A Q_2 \\ Q_2^T A Q_1 & Q_2^T A Q_2 \end{pmatrix} = \begin{pmatrix} Q_1^T A Q_1 & 0 \\ 0 & Q_2^T A Q_2 \end{pmatrix}$$

, i.e., the basis consisting of the columns of Q block diagonalizes A . Let Q be the matrix with columns q_1, \dots, q_n . Since each T_k is invariant under A , the matrix $Q^T A Q$ has the block diagonal form

$$Q^T A Q = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \quad \text{where } A_1 \text{ is } k \times k$$

for each $k = 1, \dots, n-1$. Therefore, $Q^T A Q$ must be diagonal. The diagonal entries are the eigenvalues of A . If we define $A_m = Q_m^T A Q_m$ where $Q_m = \langle q_1^m, \dots, q_n^m \rangle$, then A_m will become approximately diagonal for large m .

We can summarize simultaneous iteration as follows:

1. We start with the orthogonal matrix $Q_0 = I$ whose columns form a basis of n -space

2. For $k = 1, 2, \dots$ we compute

$$Z_m = AQ_{m-1} \quad \text{Power iteration step} \quad (3.4a)$$

$$Z_m = Q_m R_m \quad \text{Orthonormalize columns of } Z_m \quad (3.4b)$$

$$A_m = Q_m^T A Q_m \quad \text{Test for diagonal matrix.} \quad (3.4c)$$

The QR algorithm is an efficient way to organize these calculations. Equations (3.4a) and (3.4b) can be combined to give

$$AQ_{m-1} = Q_m R_m. \quad (3.5)$$

Combining equations (3.4c) and (3.5), we get

$$A_{m-1} = Q_{m-1}^T A Q_{m-1} = Q_{m-1}^T (Q_m R_m) = (Q_{m-1}^T Q_m) R_m = \hat{Q}_m R_m \quad (3.6)$$

where $\hat{Q}_m = Q_{m-1}^T Q_m$. Equation (3.5) can be rewritten as

$$Q_m^T A Q_{m-1} = R_m. \quad (3.7)$$

Combining equations (3.4c) and (3.7), we get

$$A_m = Q_m^T A Q_m = (Q_m^T A Q_{m-1}) Q_{m-1}^T Q_m = R_m (Q_{m-1}^T Q_m) = R_m \hat{Q}_m. \quad (3.8)$$

Equation (3.6) is a QR factorization of A_{m-1} . Equation (3.8) shows that A_m has the same Q and R factors but with their order reversed. Thus, the QR algorithm generates the matrices A_m recursively without having to compute Z_m and Q_m at each step. Note that the orthogonal matrices \hat{Q}_m and Q_m satisfy the relation

$$\hat{Q}_1 \hat{Q}_2 \cdots \hat{Q}_k = (Q_0^T Q_1)(Q_1^T Q_2) \cdots (Q_{k-1}^T Q_k) = Q_k.$$

We have now seen that the QR method can be considered as a generalization of the power method. We will see that the QR algorithm is also related to inverse power iteration. In fact we have the following duality result.

Theorem 3. *If A is an $n \times n$ symmetric nonsingular matrix and if S and S^\perp are orthogonal complementary subspaces. Then $A^m S$ and $A^{-m} S^\perp$ are also orthogonal complements.*

Proof. If x and y are n -vectors, then

$$x \cdot y = x^T y = x^T A^T (A^T)^{-1} y = (Ax)^T (A^T)^{-1} y = (Ax)^T A^{-1} y = Ax \cdot A^{-1} y.$$

Applying this result repeatedly, we obtain

$$x \cdot y = A^m x \cdot A^{-m} y.$$

It is clear from this relation that every element in $A^m S$ is orthogonal to every element in $A^{-m} S^\perp$. Let q_1, \dots, q_k be a basis of S and let q_{k+1}, \dots, q_n be a basis of S^\perp . Then $A^m q_1, \dots, A^m q_k$ is a basis of $A^m S$ and $A^{-m} q_{k+1}, \dots, A^{-m} q_n$ is a basis of $A^{-m} S^\perp$. Suppose there exist scalars c_1, \dots, c_n such that

$$c_1 A^m q_1 + \dots + c_k A^m q_k + c_{k+1} A^{-m} q_{k+1} + \dots + c_n A^{-m} q_n = 0. \quad (3.9)$$

Taking the dot product of this relation with $c_1 A^m q_1 + \dots + c_k A^m q_k$, we obtain

$$\|c_1 A^m q_1 + \dots + c_k A^m q_k\|^2 = 0$$

and hence $c_1 A^m q_1 + \dots + c_k A^m q_k = 0$. Since $A^m q_1, \dots, A^m q_k$ are linearly independent, it follows that $c_1 = c_2 = \dots = c_k = 0$. In a similar manner we obtain $c_{k+1} = \dots = c_n = 0$. Therefore, $A^m q_1, \dots, A^m q_k, A^{-m} q_{k+1}, \dots, A^{-m} q_n$ are linearly independent and hence form a basis for n -space. Thus, $A^m S$ and $A^{-m} S^\perp$ are orthogonal complements. \square

It can be seen from this theorem that performing power iteration on subspaces S_k is also performing inverse power iteration on S_k^\perp . Since

$$\langle q_1^m, \dots, q_k^m \rangle = \langle A^m e_1, \dots, A^m e_k \rangle,$$

Theorem 3 implies that

$$\langle q_{k+1}^m, \dots, q_n^m \rangle = \langle A^{-m} e_{k+1}, \dots, A^{-m} e_n \rangle.$$

For $k = n - 1$ we have $\langle q_n^m \rangle = \langle A^{-m} e_n \rangle$. Thus, q_n^m is the result at the m -th step of applying the inverse power method to e_n . It follows that q_n^m should converge to an eigenvector corresponding to the smallest eigenvalue λ_n . Moreover, the element in the n -th row and n -th column of $A_m = Q_m^T A Q_m$ should converge to the smallest eigenvalue λ_n .

The convergence of the QR method, like that of the power method, can be quite slow. To make the method practical, the convergence is accelerated using shifts as in the inverse power method.

3.6.1 The QR Method with Shifts

Suppose we apply a shift μ_m at the m -th step, i.e., we replace A by $A - \mu_m I$. Then the algorithm becomes

1. Set $A_0 = A$.

2. for $k = 1, 2, \dots$

$$A_{k-1} - \mu_k I = \hat{Q}_k \hat{R}_k \quad \text{QR factorization}$$

$$A_k = \hat{R}_k \hat{Q}_k + \mu_k I.$$

3. Deflate when eigenvalue converges

It follows from the QR factorization of $A_{k-1} - \mu_k I$ that

$$\hat{Q}_k^T A_{k-1} \hat{Q}_k - \mu_k I = \hat{Q}_k^T (A_{k-1} - \mu_k I) \hat{Q}_k = \hat{Q}_k^T \hat{Q}_k \hat{R}_k \hat{Q}_k = \hat{R}_k \hat{Q}_k. \quad (3.10)$$

Equation (3.10) implies that

$$A_k = \hat{Q}_k^T A_{k-1} \hat{Q}_k. \quad (3.11)$$

It follows by induction on equation (3.11) that

$$A_k = \hat{Q}_k^T \cdots \hat{Q}_1^T A \hat{Q}_1 \cdots \hat{Q}_k. \quad (3.12)$$

If we define

$$Q_k = \hat{Q}_1 \cdots \hat{Q}_k,$$

then equation (3.12) can be written

$$A_k = Q_k^T A Q_k. \quad (3.13)$$

Thus, each A_k has the same eigenvalues as A .

Theorem 4. For each $k \geq 1$ we have the relation

$$(A - \mu_k I) \cdots (A - \mu_1 I) = \hat{Q}_1 \cdots \hat{Q}_k \hat{R}_k \cdots \hat{R}_1 = Q_k R_k$$

where $Q_k = \hat{Q}_1 \cdots \hat{Q}_k$ and $R_k = \hat{R}_k \cdots \hat{R}_1$.

Proof. For $k = 1$ the result is just the $k = 1$ step. Assume that the result holds for some k , i.e.,

$$(A - \mu_k I) \cdots (A - \mu_1 I) = Q_k R_k. \quad (3.14)$$

From the $k + 1$ step we have

$$A_k - \mu_{k+1} I = \hat{Q}_{k+1} \hat{R}_{k+1}. \quad (3.15)$$

Combining equations (3.13) and (3.15), we get

$$A_k - \mu_{k+1} I = Q_k^T A Q_k - \mu_{k+1} I = Q_k^T (A - \mu_{k+1} I) Q_k = \hat{Q}_{k+1} \hat{R}_{k+1},$$

and hence

$$A - \mu_{k+1} I = Q_k \hat{Q}_{k+1} \hat{R}_{k+1} Q_k^T = Q_{k+1} \hat{R}_{k+1} Q_k^T. \quad (3.16)$$

Combining equations (3.14) and (3.16), we get

$$(A - \mu_{k+1} I)(A - \mu_k I) \cdots (A - \mu_1 I) = Q_{k+1} \hat{R}_{k+1} Q_k^T Q_k R_k = Q_{k+1} R_{k+1},$$

which is the result for $k + 1$. This completes the proof by induction \square

It follows from Theorem 4 that

$$(A - \mu_k I) \cdots (A - \mu_1 I) e_1 = Q_k R_k e_1.$$

Since R_k is upper triangular, $Q_k R_k e_1$ is proportional to the first column of Q_k . Thus, the first column of Q_k , apart from a constant multiplier, is the result of applying the power method with shifts to e_1 . Taking the inverse of the result in Theorem 4, we obtain

$$(A - \mu_1 I)^{-1} \cdots (A - \mu_k I)^{-1} = R_k^{-1} Q_k^T. \quad (3.17)$$

Since for each j the factor $A - \mu_j I$ is symmetric, its inverse $(A - \mu_j I)^{-1}$ is also symmetric. Taking the transpose of equation (3.17), we get

$$(A - \mu_k I)^{-1} \cdots (A - \mu_1 I)^{-1} = Q_k (R_k^{-1})^T. \quad (3.18)$$

Applying equation (3.18) to e_n , we get

$$(A - \mu_k I)^{-1} \cdots (A - \mu_1 I)^{-1} e_n = Q_k (R_k^{-1})^T e_n.$$

Since $(R_k^{-1})^T$ is lower triangular, $(R_k^{-1})^T e_n$ is a multiple of the last column of Q_k . Therefore, the last column of Q_k , apart from a constant multiplier, is the result of applying the inverse power method with shifts to e_n . We have yet to say how the shifts are to be chosen. One choice is to choose μ_k to be the Rayleigh quotient corresponding the last column of Q_{k-1} . This is readily available to us since, by equation (3.13), it is equal to the (n, n) element of A_{k-1} . By our remarks on Rayleigh quotient iteration, we should expect cubic convergence to the eigenvalue λ_n . This choice of shifts generally leads to convergence, but there are a few matrices for which the process fails to converge. For example, consider the matrix

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

The unshifted QR algorithm doesn't converge since

$$\begin{aligned} A &= \hat{Q}_1 \hat{R}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ A_1 &= \hat{R}_1 \hat{Q}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = A. \end{aligned}$$

Thus, all the iterates are equal to A . The Rayleigh quotient shift doesn't help since $A_{22} = 0$. A shift that does work all the time is the *Wilkinson Shift*. This shift is obtained by considering the lower-rightmost 2×2 submatrix of A_{k-1} and choosing μ_k to be the eigenvalue of this 2×2 submatrix that is closest to the (n, n) element of A_{k-1} . When there is sufficient convergence to the eigenvalue λ_n , the off-diagonal elements in the last row and column of the A_k matrices will be very small. We can deflate these matrices by removing the first and last columns, and then λ_{n-1} can be obtained using the deflated matrices. Continuing in this manner we can obtain all of the eigenvalues.

Until recently, the QR method with shifts (or one of its variants) was the primary method for computing eigenvalues and eigenvectors. Recently a competitor has emerged called the *Divide-and-Conquer* algorithm.

3.7 The Divide-and-Conquer Method

The Divide-and-Conquer algorithm was first introduced by Cuppen [3] in 1981. As first introduced, the algorithm suffered from certain accuracy and stability problems. These were not overcome until a stable algorithm was introduced in 1993 by Gu and Eisenstat [8]. The divide-and-conquer algorithm is faster than the shifted QR algorithm if the size is greater than about 25 and both eigenvalues and eigenvectors are required. Let us begin by discussing the basic theory underlying the method. Let T denote a symmetric tridiagonal matrix for which we desire the eigenvalues and eigenvectors, i.e., T has the form

$$T = \left(\begin{array}{cccc|cc} a_1 & b_1 & & & & \\ b_1 & \ddots & \ddots & & & \\ & \ddots & a_{m-1} & b_{m-1} & & \\ & & b_{m-1} & a_m & b_m & \\ \hline & & & b_m & a_{m+1} & b_{m+1} \\ & & & & b_{m+1} & \ddots \\ & & & & & \ddots & b_{n-1} \\ & & & & & & b_{n-1} & a_n \end{array} \right). \quad (3.19)$$

The matrix T can be split into the sum of two matrices as follows:

$$\begin{aligned}
T &= \left(\begin{array}{ccc|cc} a_1 & b_1 & & & \\ b_1 & \ddots & \ddots & & \\ & \ddots & a_{m-1} & b_{m-1} & \\ & & b_{m-1} & a_m - b_m & \\ \hline & & & a_{m+1} - b_m & b_{m+1} \\ & & & b_{m+1} & \ddots \\ & & & & \ddots & b_{n-1} \\ & & & & b_{n-1} & a_n \end{array} \right) \\
&+ \left(\begin{array}{c|c} & \\ \hline b_m & b_m \\ \hline b_m & b_m \end{array} \right) \\
&= \left(\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array} \right) + b_m \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} (0, \dots, 0, 1, 1, 0, \dots, 0) = \left(\begin{array}{c|c} T_1 & 0 \\ \hline 0 & T_2 \end{array} \right) + b_m v v^T \quad (3.20)
\end{aligned}$$

where m is roughly one half of n , T_1 and T_2 are tridiagonal, and v is the vector $v = e_m + e_{m+1}$. Suppose we have the following eigen decompositions of T_1 and T_2

$$T_1 = Q_1 \Lambda_1 Q_1^T \quad T_2 = Q_2 \Lambda_2 Q_2^T \quad (3.21)$$

where Λ_1 and Λ_2 are diagonal matrices of eigenvalues. Then T can be written

$$\begin{aligned}
T &= \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + b_m v v^T \\
&= \begin{pmatrix} Q_1 \Lambda_1 Q_1^T & 0 \\ 0 & Q_2 \Lambda_2 Q_2^T \end{pmatrix} + b_m v v^T \\
&= \begin{pmatrix} Q_1 & 0 \\ 0 & Q_2 \end{pmatrix} \left[\begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} + b_m u u^T \right] \begin{pmatrix} Q_1^T & 0 \\ 0 & Q_2^T \end{pmatrix} \quad (3.22)
\end{aligned}$$

where

$$u = \begin{pmatrix} Q_1^T & 0 \\ 0 & Q_2^T \end{pmatrix} v.$$

Therefore, T is similar to a matrix of the form $D + \rho uu^T$ where $D = \text{diag}(d_1, \dots, d_n)$. Thus, it suffices to look at the eigen problem for matrices of the form $D + \rho uu^T$. Let us assume first that λ is an eigenvalue of $D + \rho uu^T$, but is not an eigenvalue of D . Let x be an eigenvector of $D + \rho uu^T$ corresponding to λ . Then

$$(D + \rho uu^T)x = Dx + \rho(u^T x)u = \lambda x.$$

and hence

$$x = -\rho(u^T x)(D - \lambda I)^{-1}u. \quad (3.23)$$

Multiplying equation (3.23) by u^T and collecting terms, we get

$$(u^T x)(1 + \rho u^T (D - \lambda I)^{-1}u) = (u^T x) \left(1 + \rho \sum_{k=1}^n \frac{u_k^2}{d_k - \lambda} \right) = 0. \quad (3.24)$$

Since λ is not an eigenvalue of D , we must have $u^T x \neq 0$. Thus,

$$f(\lambda) = 1 + \rho \sum_{k=1}^n \frac{u_k^2}{d_k - \lambda} = 0. \quad (3.25)$$

Equation (3.25) is called the *secular equation* and $f(\lambda)$ is called the *secular function*. The eigenvalues of $D + \rho uu^T$ that are not eigenvalues of D are roots of the secular equation. It follows from equation (3.23) that the eigenvector corresponding to the eigenvalue λ is proportional to $(D - \lambda I)^{-1}u$. Figure 3.1 shows a plot of an example secular function.

The slope of $f(\lambda)$ is given by

$$f'(\lambda) = \rho \sum_{k=1}^n \frac{u_k^2}{(d_k - \lambda)^2}.$$

Thus, the slope (when it exists) is positive if $\rho > 0$ and negative if $\rho < 0$. Suppose the d_i are such that $d_1 > d_2 > \dots > d_n$ and that all the components of u are nonzero. Then there must be a root between each pair (d_i, d_{i+1}) . This gives $n - 1$ roots. Since $f(\lambda) \rightarrow 1$ as $\lambda \rightarrow \infty$ or as $\lambda \rightarrow -\infty$, there is another root greater than d_1 if $\rho > 0$ and a root less than d_n if $\rho < 0$. This gives n roots. The only way the secular equation will have less than n roots is if one or more of the components of u are zero or if one or more of the d_i are equal. Suppose λ is a root of the secular equation. We will show that $x = (D - \lambda I)^{-1}u$ is an eigenvector of $D + \rho uu^T$ corresponding to the eigenvalue λ . Since λ is a root of the secular equation, we have

$$f(\lambda) = 1 + \rho u^T (D - \lambda I)^{-1}u = 1 + \rho u^T x = 0$$

or $\rho u^T x = -1$. Since $x = (D - \lambda I)^{-1}u$, we have

$$(D - \lambda I)x = Dx - \lambda x = u \quad \text{or} \quad Dx - u = \lambda x.$$

It follows that

$$(D + \rho uu^T)x = Dx + \rho(u^T x)u = Dx - u = \lambda x$$

as was to be proved.

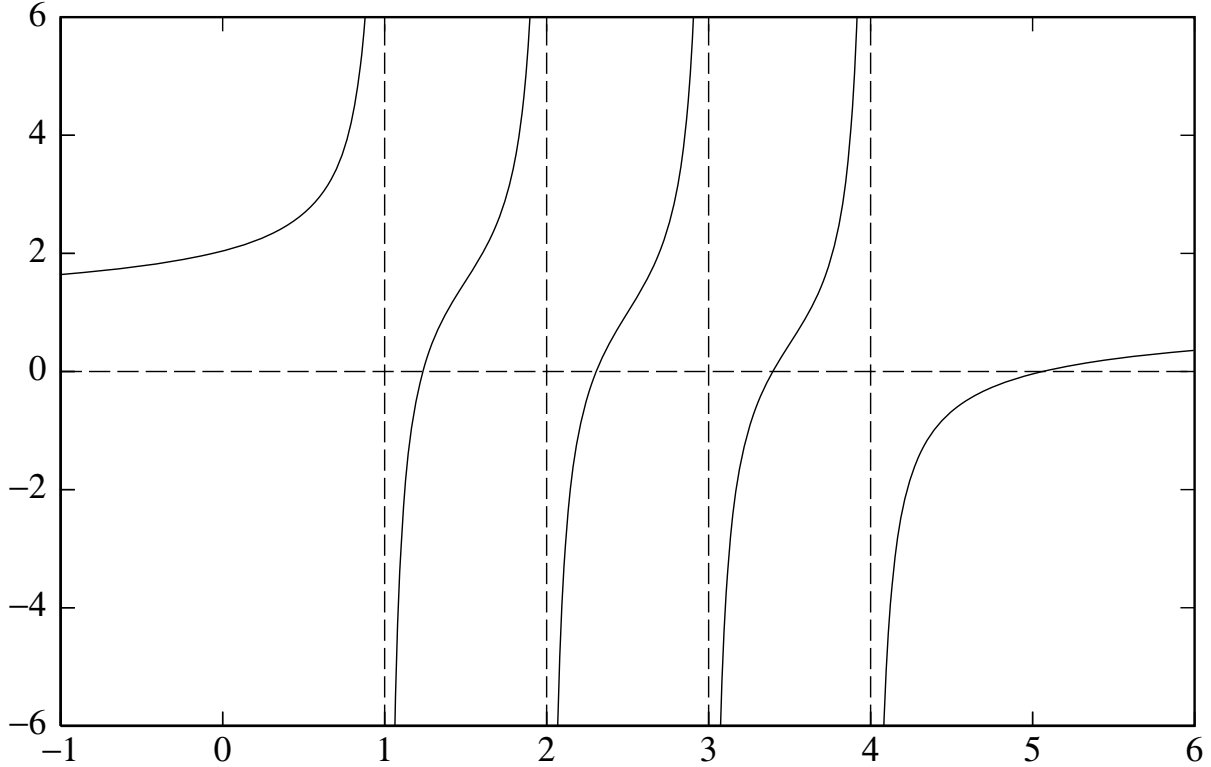


Figure 3.1: Graph of $f(\lambda) = 1 + \frac{.5}{1-\lambda} + \frac{.5}{2-\lambda} + \frac{.5}{3-\lambda} + \frac{.5}{4-\lambda}$

Let us now look at the special cases where there are less than n roots of the secular equation. If $u_i = 0$, then

$$(D + \rho uu^T)e_i = De_i + \rho(u^T e_i)u = De_i + \rho u_i u = De_i = d_i e_i,$$

i.e., e_i is an eigenvector of $D + \rho uu^T$ corresponding to the eigenvalue d_i .

If $d_i = d_j$ for $i \neq j$ and either u_i or u_j is nonzero, then the vector $x = \alpha e_i + \beta e_j$ is an eigenvector of D corresponding to the eigenvalue d_i for any α and β that are not both zero. We can choose α and β so that

$$u^T x = \alpha u_i + \beta u_j = 0.$$

For example, $\alpha = u_j$ and $\beta = -u_i$ would work. With this choice of α and β , the vector $x = \alpha e_i + \beta e_j$ is an eigenvector of $D + \rho uu^T$ corresponding to the eigenvalue d_i . In this way we can obtain n eigenvalues and vectors even when the secular equation has less than n roots.

Finding Roots of the Secular Equation The first thought would be to use Newton's method to find the roots of $f(\lambda)$. However, when one or more of the u_i are small but not small enough to neglect, the function $f(\lambda)$ behaves pretty much like it would if the terms corresponding to the small u_i were not present until λ is very close to one of the corresponding d_i where it abruptly approaches $\pm\infty$. Thus, almost any initial guess will lead away from the desired root. This is

illustrated in Figure 3.2 where the .5 factor multiplying $1/(2 - \lambda)$ in the previous example is replaced by 0.01. Notice that the curve is almost vertical at the zero crossing near 2. To solve this

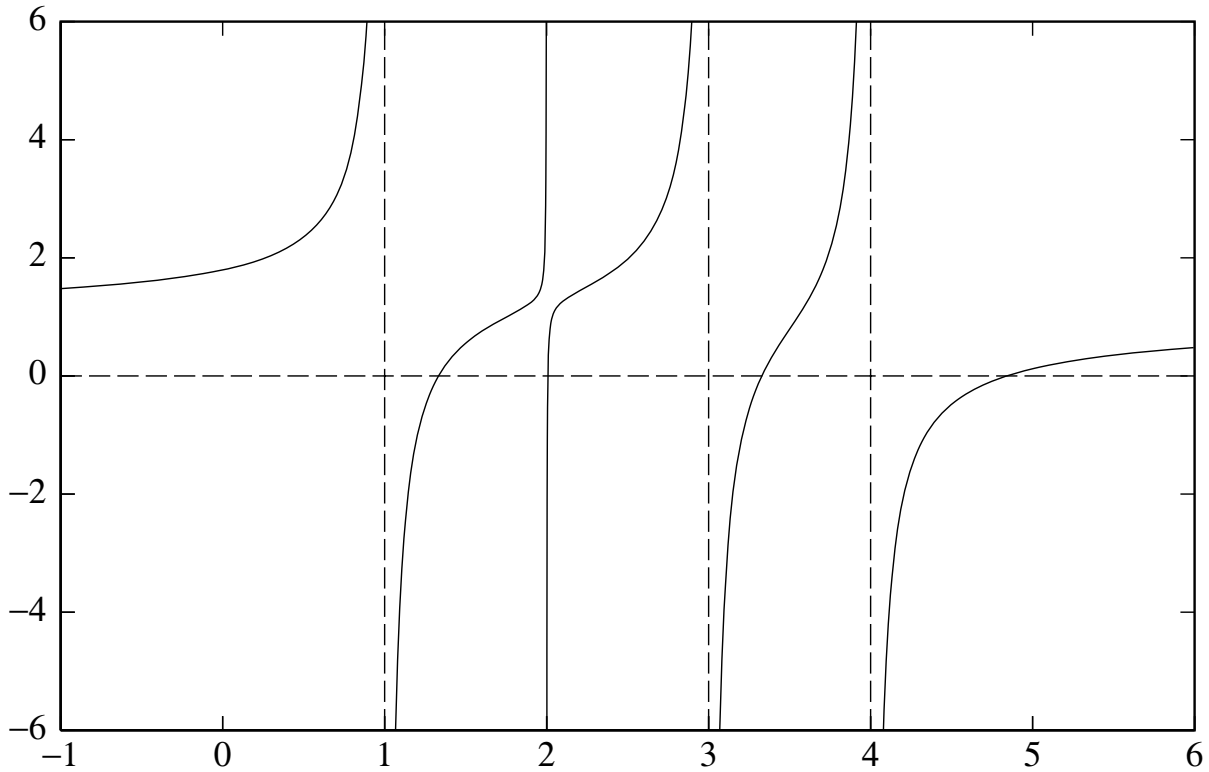


Figure 3.2: Graph of $f(\lambda) = 1 + \frac{.5}{1-\lambda} + \frac{.01}{2-\lambda} + \frac{.5}{3-\lambda} + \frac{.5}{4-\lambda}$

problem, a modified form of Newton's method is used. Newton's method approximates the curve near the guess by the tangent line at the guess and then finds the place where this line crosses zero. Alternatively, we could approximate $f(\lambda)$ near the guess by another curve that is tangent to $f(\lambda)$ at the guess as long as we can find the nearby zero crossing of this curve. If we are looking for a root between d_i and d_{i+1} , we could use a function of the form

$$g(\lambda) = c_1 + \frac{c_2}{d_i - \lambda} + \frac{c_3}{d_{i+1} - \lambda} \quad (3.26)$$

to approximate $f(\lambda)$. Once c_1 , c_2 and c_3 are chosen, the roots of $g(\lambda)$ can be found by solving the quadratic equation

$$c_1(d_i - \lambda)(d_{i+1} - \lambda) + c_2(d_{i+1} - \lambda) + c_3(d_i - \lambda) = 0. \quad (3.27)$$

Let us write $f(\lambda)$ as follows

$$f(\lambda) = 1 + \rho\psi_1(\lambda) + \rho\psi_2(\lambda) \quad (3.28)$$

where

$$\psi_1(\lambda) = \sum_{k=1}^i \frac{u_k^2}{d_k - \lambda} \quad \text{and} \quad \psi_2(\lambda) = \sum_{k=i+1}^n \frac{u_k^2}{d_k - \lambda}. \quad (3.29)$$

Notice that ψ_1 has only positive terms and ψ_2 has only negative terms for $d_{i+1} < \lambda < d_i$. If λ_j is our initial guess, then we approximate ψ_1 near λ_j by the function g_1 given by

$$g_1(\lambda) = \alpha_1 + \frac{\alpha_2}{d_i - \lambda} \quad (3.30)$$

where α_1 and α_2 are chosen so that

$$g_1(\lambda_j) = \psi_1(\lambda_j) \quad \text{and} \quad g_1'(\lambda_j) = \psi_1'(\lambda_j). \quad (3.31)$$

It is easily shown that $\alpha_1 = \psi_1(\lambda_j) - (d_i - \lambda_j)\psi_1'(\lambda_j)$ and $\alpha_2 = (d_i - \lambda_j)^2\psi_1'(\lambda_j)$. Similarly, we approximate ψ_2 near λ_j by the function g_2 given by

$$g_2(\lambda) = \alpha_3 + \frac{\alpha_4}{d_{i+1} - \lambda} \quad (3.32)$$

where α_3 and α_4 are chosen so that

$$g_2(\lambda_j) = \psi_2(\lambda_j) \quad \text{and} \quad g_2'(\lambda_j) = \psi_2'(\lambda_j). \quad (3.33)$$

Again it is easily shown that $\alpha_3 = \psi_2(\lambda_j) - (d_{i+1} - \lambda_j)\psi_2'(\lambda_j)$ and $\alpha_4 = (d_{i+1} - \lambda_j)^2\psi_2'(\lambda_j)$.

Putting these approximations together, we have the following approximation for f near λ_j

$$f(\lambda) \doteq 1 + \rho g_1(\lambda) + \rho g_2(\lambda) = (1 + \rho\alpha_1 + \rho\alpha_3) + \frac{\rho\alpha_2}{d_i - \lambda} + \frac{\rho\alpha_4}{d_{i+1} - \lambda} \equiv c_1 + \frac{c_2}{d_i - \lambda} + \frac{c_3}{d_{i+1} - \lambda}. \quad (3.34)$$

This modified Newton's method generally converges very fast.

Recursive Procedure We have shown how the eigenvalues and eigenvectors of T can be obtained from the eigenvalues and eigenvectors of the smaller matrices T_1 and T_2 . The procedure we have applied to T can also be applied to T_1 and T_2 . Continuing in this manner we can reduce the original eigen problem to the solution of a series of 1-dimensional eigen problems and the solution of a series of secular equations. In practice the recursive procedure is not carried all the way down to 1-dimensional problems, but stops at some size where the QR method can be applied effectively. We saw previously that the eigenvector corresponding to the eigenvalue λ is proportional to $(D - \lambda I)^{-1}u$ as in equation (3.23). There are a number of subtle issues involved in computing the eigenvectors this way when there are closely spaced pairs of eigenvalues. The interested reader should consult the book by Demmel [4] for a discussion of these issues.

Chapter 4

Iterative Methods

Direct methods for solving systems of equations $Ax = b$ or computing eigenvalues/eigenvectors of a matrix A become very expensive when the size n of A becomes large. These methods generally involve order n^3 operations and order n^2 storage. For large problems iterative methods are often used. Each step of an iterative method generally involves the multiplication of the matrix A by a vector v to obtain Av . Since the matrix A is not modified in this process it is often possible to take advantage of special structure of the matrix in forming Av . The special structure most often exploited is sparseness (many elements of A zero). Taking advantage of the structure of A can often drastically reduce the cost of each iteration. The cost of iterative methods also depends on the rate of convergence. Convergence is usually better when the matrix A is well conditioned. Therefore, preconditioning of the matrix is often employed prior to the start of iteration. There are many iterative methods. In this section we will discuss only two: the Lanczos method for eigen problems and the conjugate gradient method for equation solution.

4.1 The Lanczos Method

As before, we will restrict our attention here to real symmetric matrices. We saw previously that the power method is an iterative method whose m -th iterate $x^{(m)}$ is given by $x^{(m)} = Ax^{(m-1)}$. Lanczos had the idea that better convergence could be obtained if we made use of all the iterates $x^{(0)}, Ax^{(0)}, A^2x^{(0)}, \dots, A^m x^{(0)}$ at the m -th step instead of just the final iterate $x^{(m)}$. The subspace generated by $x^{(0)}, Ax^{(0)}, \dots, A^{m-1}x^{(0)}$ is called the m -th Krylov subspace and is denoted by \mathcal{K}_m . Lanczos showed that you could generate an orthonormal basis q_1, \dots, q_m of the Krylov subspace \mathcal{K}_m recursively. He then showed that the eigen problem restricted to this subspace is equivalent to finding the eigenvalues/eigenvectors of the tridiagonal matrix $T_m = Q_m^T A Q_m$ where Q_m is the matrix whose columns are q_1, \dots, q_m . As m becomes larger some of the eigenvalues of T_m converge to eigenvalues of A .

Let q_1 be defined by

$$q_1 = x^{(0)} / \|x^{(0)}\| \quad (4.1)$$

and let q_2 be given by

$$q_2 = r_1 / \|r_1\| \quad \text{where} \quad r_1 = Aq_1 - (Aq_1 \cdot q_1)q_1. \quad (4.2)$$

It is easily verified that $r_1 \cdot q_1 = q_2 \cdot q_1 = 0$. We generate the remaining vectors q_k recursively. Suppose q_1, \dots, q_p have been generated. We form

$$r_p = Aq_p - (Aq_p \cdot q_p)q_p - (Aq_p \cdot q_{p-1})q_{p-1} \quad (4.3)$$

$$q_{p+1} = r_p / \|r_p\|. \quad (4.4)$$

Clearly, $r_p \cdot q_p = r_p \cdot q_{p-1} = 0$ by construction. For $s \leq p-2$ we have

$$r_p \cdot q_s = Aq_p \cdot q_s = q_p \cdot Aq_s. \quad (4.5)$$

But, it follows from equations (4.3)–(4.4) that

$$\begin{aligned} Aq_s &= r_s + (Aq_s \cdot q_s)q_s + (Aq_s \cdot q_{s-1})q_{s-1} \\ &= \|r_s\|q_{s+1} + (Aq_s \cdot q_s)q_s + (Aq_s \cdot q_{s-1})q_{s-1}. \end{aligned} \quad (4.6)$$

Thus, $r_p \cdot q_s = q_p \cdot Aq_s = 0$ since Aq_s is a linear combination of vectors q_k with $k < p$. It follows that q_{p+1} is orthogonal to all of the preceding q_k vectors. We will now show that q_1, \dots, q_m is a basis for the space \mathcal{K}_m . It follows from equations (4.3) and (4.4) that

$$\langle x^{(0)} \rangle = \langle q_1 \rangle \quad \text{and} \quad \langle x^{(0)}, Ax^{(0)} \rangle = \langle q_1, q_2 \rangle.$$

Suppose for some k we have

$$\langle x^{(0)}, Ax^{(0)}, \dots, A^{k-1}x^{(0)} \rangle = \langle q_1, q_2, \dots, q_k \rangle.$$

Then, $A^k x^{(0)}$ can be written as a linear combination of Aq_1, \dots, Aq_k . It follows from equations (4.3) and (4.4) that Aq_i can be written as a linear combination of q_{i-1}, q_i, q_{i+1} . Therefore, $A^k x^{(0)}$ can be written as a linear combination of q_1, \dots, q_{k+1} and hence

$$\langle x^{(0)}, Ax^{(0)}, \dots, A^k x^{(0)} \rangle = \langle q_1, q_2, \dots, q_{k+1} \rangle.$$

It follows by induction that q_1, \dots, q_m is a basis for $\mathcal{K}_m = \langle x^{(0)}, Ax^{(0)}, \dots, A^{m-1}x^{(0)} \rangle$.

Define $\alpha_p = Aq_p \cdot q_p$ and $\beta_p = Aq_p \cdot q_{p-1}$. Then

$$\begin{aligned} \beta_p &= Aq_p \cdot q_{p-1} = q_p \cdot Aq_{p-1} \\ &= q_p \cdot [\|r_{p-1}\|q_p + (Aq_{p-1} \cdot q_{p-1})q_{p-1} + (Aq_{p-1} \cdot q_{p-2})q_{p-2}] \\ &= \|r_{p-1}\|. \end{aligned} \quad (4.7)$$

It follows from equations (4.3), (4.4), and (4.7) that

$$Aq_p = \beta_{p+1}q_{p+1} + \alpha_p q_p + \beta_p q_{p-1}. \quad (4.8)$$

In view of equation (4.8), the matrix $T_m = Q_m^T A Q_m$ has the tridiagonal form

$$T_m = Q_m^T A Q_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}. \quad (4.9)$$

The original eigenvalue problem can be given a variational interpretation. Let a function ϕ be defined by

$$\phi(x) = \frac{Ax \cdot x}{x \cdot x}. \quad (4.10)$$

We will show that $\phi(x)$ is an eigenvalue of A if and only if x is a stationary point of ϕ , i.e., $\delta_h \phi(x) = 0$ for all h . Since

$$\begin{aligned} \delta_h \phi(x) &\equiv \left. \frac{d}{d\lambda} \phi(x + \lambda h) \right|_{\lambda=0} = \frac{(x \cdot x)(2Ax \cdot h) - (Ax \cdot x)(2x \cdot h)}{(x \cdot x)^2} \\ &= \frac{2}{x \cdot x} \left[Ax - \frac{Ax \cdot x}{x \cdot x} x \right] \cdot h, \end{aligned} \quad (4.11)$$

we have

$$\delta_h \phi(x) = 0 \quad \text{for all } h \quad \Leftrightarrow \quad \left[Ax - \frac{Ax \cdot x}{x \cdot x} x \right] \cdot h = 0 \quad \text{for all } h \quad (4.12a)$$

$$\Leftrightarrow \quad Ax = \frac{Ax \cdot x}{x \cdot x} x = \phi(x)x. \quad (4.12b)$$

Suppose in this variational principle we restrict both x and h to the subspace \mathcal{K}_m . Then x and h can be expressed in the form $x = Q_m y$ and $h = Q_m w$ for some $y, w \in \mathbb{R}^m$. With these relations equation (4.12a) becomes

$$\left[(AQ_m)y - \frac{(AQ_m)y \cdot Q_m y}{Q_m y \cdot Q_m y} x \right] \cdot Q_m w = 0 \quad \text{for all } w$$

or

$$\left[(Q_m^T A Q_m)y - \frac{(Q_m^T A Q_m)y \cdot y}{y \cdot y} y \right] \cdot w = 0 \quad \text{for all } w. \quad (4.13)$$

Thus the variational principle restricted to \mathcal{K}_m leads to the reduced eigenvalue problem

$$T_m y = (Q_m^T A Q_m)y = \mu y. \quad (4.14)$$

It has been found that the extreme eigenvalues usually converge the fastest with this method. The biggest numerical problem with this method is that round-off errors cause the vectors $\{q_k\}$ generated in this way tend to lose their orthogonality as the number of steps increases. It has been found that this loss of orthogonality increases rapidly whenever one of the eigenvalues of T_m approaches

an eigenvalue of A . There are a number of methods that counteract this loss of orthogonality by periodically reorthogonalizing the vectors $\{q_k\}$ based on the convergence of the eigenvalues.

We can give another way of looking at the Lanczos algorithm. Let K_m denote the matrix whose columns are $x^{(0)}, Ax^{(0)}, \dots, A^{m-1}x^{(0)}$. We will show that K_m has a reduced QR factorization $K_m = Q_m R_m$ where Q_m is the matrix occurring in the Lanczos method with columns q_1, \dots, q_m . We have shown previously that

$$\begin{aligned} \langle x^{(0)} \rangle &= \langle q_1 \rangle \\ \langle x^{(0)}, Ax^{(0)} \rangle &= \langle q_1, q_2 \rangle \\ &\vdots \\ \langle x^{(0)}, Ax^{(0)}, \dots, A^{m-1}x^{(0)} \rangle &= \langle q_1, q_2, \dots, q_m \rangle. \end{aligned}$$

We can express this result in matrix form as

$$K_m = (x^{(0)}, Ax^{(0)}, \dots, A^{m-1}x^{(0)}) = (q_1, \dots, q_m)R_m = Q_m R_m \quad (4.15)$$

where R_m is an upper triangular matrix. This is the reduced QR factorization that we set out to establish. Of course we don't want to determine Q_m and R_m directly since the matrix K_m becomes poorly conditioned for large m .

4.2 The Conjugate Gradient Method

The conjugate gradient (CG) method is a widely used iterative method for solving a system of equations $Ax = b$ when A is symmetric and positive definite. It was first introduced in 1952 by Hestenes and Stiefel [9]. Although this was not the original motivation, the CG method can be considered as a Krylov subspace method related to the Lanczos method. We assume that q_1, q_2, \dots are orthonormal vectors generated using the Lanczos recursion starting with the initial vector b . As before we let $Q_k = (q_1, \dots, q_k)$ and $T_k = Q_k^T A Q_k$. Since A is positive definite, we can define an A-norm by

$$\|x\|_A^2 = x^T A x. \quad (4.16)$$

We will show that each iterate x_m in the CG method is the unique element of the Krylov subspace \mathcal{K}_m that minimizes the error $\|x - x_m\|_A$ where x is the solution of $Ax = b$.

Let r_k denote the residual $r_k = b - Ax_k$. Since $q_1 = b/\|b\|$, it follows that

$$\begin{aligned}
Q_k^T r_k &= Q_k^T (b - Ax_k) = Q_k^T b - Q_k^T Ax_k \\
&= \begin{pmatrix} q_1^T \\ \vdots \\ q_k^T \end{pmatrix} b - T_k Q_k^T x_k \\
&= \|b\| e_1 - T_k Q_k^T x_k \\
&= T_k Q_k^T (\|b\| Q_k T_k^{-1} e_1 - x_k).
\end{aligned} \tag{4.17}$$

If x_k is chosen to be

$$x_k = \|b\| Q_k T_k^{-1} e_1, \tag{4.18}$$

then $Q_k^T r_k = 0$, i.e., r_k is orthogonal to each of the vectors q_1, \dots, q_k and hence to every vector in \mathcal{K}_k . It follows from equation (4.18) that x_k is a linear combination of q_1, \dots, q_k and hence is a member of \mathcal{K}_k . If \hat{x} is an arbitrary element of \mathcal{K}_k , then

$$\hat{x} = x_k + \delta \quad \text{for some } \delta \text{ in } \mathcal{K}_k.$$

Since r_k is orthogonal to every vector in \mathcal{K}_k , we have

$$\begin{aligned}
\|x - \hat{x}\|_A^2 &= (x - \hat{x})^T A (x - \hat{x}) \\
&= (x - x_k - \delta)^T A (x - x_k - \delta) \\
&= \|x - x_k\|_A^2 + \|\delta\|_A^2 - 2\delta^T A (x - x_k) \\
&= \|x - x_k\|_A^2 + \|\delta\|_A^2 - 2\delta^T r_k \\
&= \|x - x_k\|_A^2 + \|\delta\|_A^2.
\end{aligned} \tag{4.19}$$

Thus $\|x - \hat{x}\|_A^2$ is minimized for $\delta = 0$, i.e., when $\hat{x} = x_k$. We will now develop a simple recursive method to generate the iterates x_k .

The matrix $T_k = Q_k^T A Q_k$ is also positive definite and hence has a Cholesky factorization

$$T_k = L_k D_k L_k^T \tag{4.20}$$

where L_k is unit lower triangular and D_k is diagonal with positive diagonals. Combining equations (4.18) and (4.20), we get

$$\begin{aligned}
x_k &= \|b\| Q_k (L_k^{-T} D_k^{-1} L_k^{-1}) e_1 \\
&= \tilde{P}_k y_k
\end{aligned} \tag{4.21}$$

where $\tilde{P}_k = Q_k L_k^{-T}$ and $y_k = \|b\| D_k^{-1} L_k^{-1} e_1$. We denote the columns of \tilde{P}_k by $\tilde{p}_1, \dots, \tilde{p}_k$ and the components of y_k by η_1, \dots, η_k . We will show that the columns of \tilde{P}_{k-1} are $\tilde{p}_1, \dots, \tilde{p}_{k-1}$ and the components of y_{k-1} are $\eta_1, \dots, \eta_{k-1}$. It follows from equation (4.20) and the definition of \tilde{P}_k that

$$\begin{aligned}
\tilde{P}_k^T A \tilde{P}_k &= L_k^{-1} Q_k^T A Q_k L_k^{-T} = L_k^{-1} T_k L_k^{-T} \\
&= L_k^{-1} (L_k D_k L_k^T) L_k^{-T} = D_k.
\end{aligned}$$

Thus

$$\tilde{p}_i^T A \tilde{p}_j = 0 \quad \text{for all } i \neq j. \quad (4.22)$$

It is easy to see from equation (4.9) that T_{k-1} is the leading $(k-1) \times (k-1)$ submatrix of T_k . Equation (4.20) can be written

$$\begin{aligned} T_k &= \begin{pmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{k-1} & 1 \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & \ddots & & \\ & & d_{k-1} & \\ & & & d_k \end{pmatrix} \begin{pmatrix} 1 & & & \\ l_1 & \ddots & & \\ & \ddots & \ddots & \\ & & l_{k-1} & 1 \end{pmatrix}^T \\ &= \begin{pmatrix} L_{k-1} & 0 \\ l_{k-1} e_{k-1}^T & 1 \end{pmatrix} \begin{pmatrix} D_{k-1} & 0 \\ 0 & d_k \end{pmatrix} \begin{pmatrix} L_{k-1} & 0 \\ l_{k-1} e_{k-1}^T & 1 \end{pmatrix}^T \\ &= \begin{pmatrix} L_{k-1} D_{k-1} L_{k-1}^T & \star \\ \star & \star \end{pmatrix} \end{aligned}$$

where \star denotes terms that are not significant to the argument. Thus, L_{k-1} and D_{k-1} are the leading $(k-1) \times (k-1)$ submatrices of L_k and D_k respectively. Since L_k has the form

$$L_k = \begin{pmatrix} L_{k-1} & 0 \\ \star & 1 \end{pmatrix},$$

the inverse L_k^{-1} must have the form

$$L_k^{-1} = \begin{pmatrix} L_{k-1}^{-1} & 0 \\ \star & 1 \end{pmatrix}.$$

Therefore, it follows from the definition of y_k that

$$\begin{aligned} y_k &\equiv \|b\| D_k^{-1} L_k^{-1} e_1 \\ &= \|b\| \begin{pmatrix} D_{k-1}^{-1} & 0 \\ 0 & 1/d_k \end{pmatrix} \begin{pmatrix} L_{k-1}^{-1} & 0 \\ \star & 1 \end{pmatrix} e_1 \\ &= \|b\| \begin{pmatrix} D_{k-1}^{-1} L_{k-1}^{-1} & 0 \\ \star & 1/d_k \end{pmatrix} e_1 \\ &= \|b\| \begin{pmatrix} D_{k-1}^{-1} L_{k-1}^{-1} & 0 \\ \star & 1/d_k \end{pmatrix} \begin{pmatrix} e_1 \\ 0 \end{pmatrix} \quad e_1 \text{ is here a } (k-1)\text{-vector} \\ &= \begin{pmatrix} \|b\| D_{k-1}^{-1} L_{k-1}^{-1} e_1 \\ \eta_k \end{pmatrix} = \begin{pmatrix} y_{k-1} \\ \eta_k \end{pmatrix}, \end{aligned}$$

i.e., y_{k-1} consists of the first $k-1$ components of y_k . It follows from the definition of \tilde{P}_k that

$$\begin{aligned} \tilde{P}_k &\equiv Q_k L_k^{-T} \\ &= (Q_{k-1}, q_k) \begin{pmatrix} L_{k-1}^{-T} & \star \\ 0 & 1 \end{pmatrix} \\ &= (Q_{k-1} L_{k-1}^{-T}, \tilde{p}_k) = (\tilde{P}_{k-1}, \tilde{p}_k), \end{aligned}$$

i.e., \tilde{P}_{k-1} consists of the first $k - 1$ columns of \tilde{P}_k .

We now develop a recursion relation for x_k . It follows from equation (4.21) that

$$\begin{aligned}
x_k &= \tilde{P}_k y_k \\
&= (\tilde{P}_{k-1}, \tilde{p}_k) \begin{pmatrix} y_{k-1} \\ \eta_k \end{pmatrix} \\
&= \tilde{P}_{k-1} y_{k-1} + \eta_k \tilde{p}_k \\
&= x_{k-1} + \eta_k \tilde{p}_k.
\end{aligned} \tag{4.23}$$

We now develop a recursion relation for \tilde{p}_k . It follows from the definition of \tilde{P}_k that

$$\tilde{P}_k L_k^T = Q_k$$

or

$$(\tilde{p}_1, \dots, \tilde{p}_k) \begin{pmatrix} 1 & l_1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & l_{k-1} & \\ & & & & 1 \end{pmatrix} = (q_1, \dots, q_k). \tag{4.24}$$

Equating the k -th columns in equation (4.24), we get

$$l_{k-1} \tilde{p}_{k-1} + \tilde{p}_k = q_k$$

or

$$\tilde{p}_k = q_k - l_{k-1} \tilde{p}_{k-1}. \tag{4.25}$$

Next we develop a recursion relation for the residuals r_k . Multiplying equation (4.23) by A and subtracting from b , we obtain

$$r_k = r_{k-1} - \eta_k A \tilde{p}_k. \tag{4.26}$$

Since x_{k-1} belongs to \mathcal{K}_{k-1} , it follows that Ax_{k-1} belongs to \mathcal{K}_k . Since b also belongs to \mathcal{K}_k , it is clear that $r_{k-1} = b - Ax_{k-1}$ is a member of \mathcal{K}_k . Since r_{k-1} and q_k both belong to \mathcal{K}_k and both are orthogonal to \mathcal{K}_{k-1} , they must be parallel. Thus,

$$q_k = \frac{r_{k-1}}{\|r_{k-1}\|}. \tag{4.27}$$

We now define p_k by

$$p_k = \|r_{k-1}\| \tilde{p}_k. \tag{4.28}$$

Substituting equations (4.27) and (4.28) into equations (4.23), (4.26), and (4.25), we get

$$x_k = x_{k-1} + \frac{\eta_k}{\|r_{k-1}\|} p_k = x_{k-1} + v_k p_k \tag{4.29a}$$

$$r_k = r_{k-1} - \frac{\eta_k}{\|r_{k-1}\|} p_k = r_{k-1} - v_k A p_k \tag{4.29b}$$

$$p_k = \|r_{k-1}\| q_k - \frac{l_{k-1} \|r_{k-1}\|}{\|r_{k-2}\|} p_{k-1} = r_{k-1} + \mu_k p_{k-1}. \tag{4.29c}$$

Here we have used the definitions

$$v_k = \frac{\eta_k}{\|r_{k-1}\|}$$

and

$$\mu_k = -\frac{l_{k-1}\|r_{k-1}\|}{\|r_{k-2}\|}.$$

Equations (4.29a), (4.29b), and (4.29c) are our three basic recursion relations. We next develop a formula for v_k . Since $r_{k-1} = \|r_{k-1}\|q_k$ and r_k is orthogonal to \mathcal{K}_k , multiplication of equation (4.29b) by r_{k-1}^T gives

$$0 = r_{k-1}^T r_k = \|r_{k-1}\|^2 - v_k r_{k-1}^T A p_k.$$

Thus

$$v_k = \frac{\|r_{k-1}\|^2}{r_{k-1}^T A p_k}. \quad (4.30)$$

Multiplying equation (4.29c) by $p_k^T A$, we get

$$p_k^T A p_k = p_k^T A r_{k-1} + 0 = r_{k-1}^T A p_k. \quad (4.31)$$

Combining equations (4.30) and (4.31), we obtain the desired formula

$$v_k = \frac{\|r_{k-1}\|^2}{p_k^T A p_k}. \quad (4.32)$$

We next develop a formula for μ_k . In view of equations (4.22) and (4.28), multiplication of equation (4.29c) by $p_{k-1}^T A$, gives

$$0 = p_{k-1}^T A p_k = p_{k-1}^T A r_{k-1} + \mu_k p_{k-1}^T A p_{k-1}$$

or

$$\mu_k = -\frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}}. \quad (4.33)$$

Multiplying equation (4.29b) by r_k^T , we get

$$r_k^T r_k = 0 - v_k r_k^T A p_k$$

or

$$v_k = -\frac{r_k^T r_k}{r_k^T A p_k} = -\frac{\|r_k\|^2}{r_k^T A p_k}. \quad (4.34)$$

Combining equations (4.32) and (4.34), we get

$$-\frac{\|r_k\|^2}{r_k^T A p_k} = \frac{\|r_{k-1}\|^2}{p_k^T A p_k}. \quad (4.35)$$

Evaluating equation (4.35) for $k = k - 1$ and combining the result with equation (4.33), we obtain the desired formula

$$\mu_k = -\frac{p_{k-1}^T A r_{k-1}}{p_{k-1}^T A p_{k-1}} = \frac{\|r_{k-1}\|^2}{\|r_{k-2}\|^2}. \quad (4.36)$$

We can now summarize the CG algorithm

1. Compute the initial values $x_0 = 0$, $r_0 = b$, and $p_1 = b$.
2. For $k = 1, 2, \dots$ compute

| | |
|-----------------------------------------|-------------------------|
| $z = Ap_k$ | Save Ap_k |
| $v_k = \ r_{k-1}\ ^2 / p_k^T z$ | New step length |
| $x_k = x_{k-1} + v_k p_k$ | Update approximation |
| $r_k = r_{k-1} - v_k z$ | New residual |
| $\mu_{k+1} = \ r_k\ ^2 / \ r_{k-1}\ ^2$ | Improvement of residual |
| $p_{k+1} = r_k + \mu_{k+1} p_k$ | New search direction |
3. Stop when $\|r_k\|$ is small enough.

Notice that the algorithm at each step only involves one matrix vector product, two dot products (by saving $\|r_k\|^2$ at each step), and three linear combinations of vectors. The storage required is only four vectors (current values of z , r , x , and p) in addition to the matrix A . As with all iterative methods, the convergence is fastest when the matrix is well conditioned. The convergence also depends on the distribution of eigenvalues.

4.3 Preconditioning

The convergence of iterative methods often depends on the condition of the underlying matrix as well as the distribution of its eigenvalues. The convergence can often be improved by applying a preconditioner M^{-1} to A , i.e., we consider the matrix $M^{-1}A$ in place of A . If we are solving a system of equations $Ax = b$, this system can be replaced by $M^{-1}Ax = M^{-1}b$. The matrix $M^{-1}A$ might be better suited for an iterative method. Of course M must be fairly simple to compute, or the advantage might be lost. We often try to choose M so that it approximates A in some sense. If the original A was symmetric and positive definite, we generally choose M to be symmetric and positive definite. However, $M^{-1}A$ is generally not symmetric and positive definite even when both A and M are. If M is symmetric and positive definite, then $M = EE^T$ for some E (possibly obtained by a Cholesky factorization). The system of equations $Ax = b$ can be replaced by $(E^{-1}AE^{-T})\hat{x} = E^{-1}b$ where $\hat{x} = E^T x$. The matrix $E^{-1}AE^{-T}$ is symmetric and positive definite. Since

$$E^{-T}(E^{-1}AE^{-T})E^T = M^{-1}A, \quad \text{Similarity Transformation}$$

$E^{-1}AE^{-T}$ has the same eigenvalues as $M^{-1}A$.

The choice of a good preconditioner is more of an art than a science. The following are some of

the ways M might be chosen:

1. M can be chosen to be the diagonal of A , i.e., $M = \text{diag}(a_{11}, a_{22}, \dots, a_{nn})$.
2. M can be chosen on the basis of an incomplete Cholesky or LU factorization of A . If A is sparse, then the Cholesky factorization $A = LL^T$ will generally produce an L that is not sparse. Incomplete Cholesky factorization uses Cholesky-like formulas, but only fills in those positions that are nonzero in the original A . If \hat{L} is the factor obtained in this manner, we take $M = \hat{L}\hat{L}^T$.
3. If a system of equations is obtained by a discretization of a differential or integral equation, it is sometimes possible to use a coarser discretization and interpolation to approximate the system obtained using a fine discretization.
4. If the underlying physical problem involves both short-range and long-range interactions, a preconditioner can sometimes be obtained by neglecting the long-range-interactions.
5. If the underlying physical problem can be broken up into nonoverlapping domains, then a preconditioner might be obtained by neglecting interactions between domains. In this way M becomes a block diagonal matrix.
6. Sometimes the inverse operator A^{-1} can be expressed as a matrix power series. An approximate inverse can be obtained by truncating this series. For example, we might approximate A^{-1} by a few terms of the Neumann series $A^{-1} = I + (I - A) + (I - A)^2 + \dots$.

There are many more preconditioners designed for particular types of problems. The user should survey the literature to find a preconditioner appropriate to the problem at hand.

Bibliography

- [1] Beltrami, E., *Sulle Funzioni Bilineari*, Giornale di Matematiche **11**, pp. 98–106 (1873).
- [2] Cayley, A., *A Memoir on the Theory of Matrices*, Phil. Trans. **148**, pp. 17–37 (1858).
- [3] Cuppen, J., *A divide and conquer algorithm for the symmetric tridiagonal eigenproblem*, Numer. Math. **36**, pp. 177–195 (1981).
- [4] Demmel, J.W., *Applied Numerical Linear Algebra*, SIAM (1997).
- [5] Eckart, C. and Young, G., *A Principal Axis Transformation for Non-Hermitian Matrices*, Bull. Amer. Math. Soc. **45**, pp. 118–121 (1939).
- [6] Francis, J., *The QR transformation: A unitary analogue to the LR transformation*, parts I and II, Computer J. **4**, pp. 256–272 and 332–345 (1961).
- [7] Golub, G. and Van Loan, C., *Matrix Computations*, Johns Hopkins University Press (1996)
- [8] Gu, M. and Eisenstat, S., *A stable algorithm for the rank-1 modification of the symmetric eigenproblem*, Computer Science Dept. Report YaleU/DCS/RR-967, Yale University (1993).
- [9] Hestenes, M. and Stiefel, E., *Methods of Conjugate Gradients for Solving Linear Systems*, J. Res. Nat. Bur. Stand. **49**, pp. 409–436 (1952).
- [10] Jordan, C., *Sur la réduction des formes bilinéaires*, Comptes Rendus de l'Académie des Sciences, Paris **78**, pp. 614–617 (1874).
- [11] Kublanovskaya, V., *On some algorithms for the solution of the complete eigenvalue problem*, USSR Comp. Math. Phys. **3**, pp. 637–657 (1961).
- [12] Trefethen, L. and Bau, D., *Numerical Linear Algebra*, SIAM (1997).
- [13] Watkins, D., *Understanding the QR Algorithm*, SIAM Review, vol. 24, No. 4 (1982).