Chapter 7

N Important Numbers

7.1 Eigenvalue Problems

In this chapter we shall consider the numerical solution of eigenvalue problems. Such problems occur frequently in engineering, physics, chemistry, economics, and statistics, as well as other areas. In this section we shall discuss a few example problems, classify different types of eigenvalue problems, and provide some mathematical background.

In the *matrix eigenvalue problem* we wish to find a real or complex number λ, an eigenvalue, and a corresponding nonzero vector $\mathbf{x}$, an *eigenvector*, that satisfy the equation

$$A\mathbf{x} = \lambda\mathbf{x}, \tag{7.1.1}$$

where A is a given real or complex $n \times n$ matrix. As discussed in Appendix 2, a solution λ of (7.1.1) is a root of the characteristic polynomial $\det(A - \lambda I) = 0$. This is a polynomial of degree n and therefore has exactly n real or complex roots, $\lambda_1, \cdots, \lambda_n$, provided that the multiplicity of each root is counted. Once an eigenvalue λ_i is known, a corresponding eigenvector $\mathbf{x}_i$ can be determined, in principle, as a solution of the homogeneous system of equations

$$(A - \lambda_i I)\mathbf{x} = 0. \tag{7.1.2}$$

Note that even if the matrix A is real, its eigenvalues – and consequently also its eigenvectors – may be complex. For example (Exercise 7.1.1), the matrix

$$\begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}$$

has eigenvalues $2 \pm i$.

The preceding mathematical procedure form the characteristic polynomial, compute its roots, and solve the homogeneous equations (7.1.2) is not a viable computational procedure except for the most trivial problems. The main purpose of this chapter is to give alternative computational methods.

Differential Equations

As an example of how eigenvalue problems arise, consider the system of ordinary differential equations

$$\frac{d\mathbf{y}}{dt} = A\mathbf{y} \tag{7.1.3}$$

for a given constant $n \times n$ matrix A. If we try a solution of (7.1.3) of the form

$$\mathbf{y}(t) = e^{\lambda t}\mathbf{x} \tag{7.1.4}$$

for some constant unknown vector $\mathbf{x}$ and unknown parameter λ, then we must have

$$\frac{d\mathbf{y}}{dt} = \lambda e^{\lambda t}\mathbf{x} = A(e^{\lambda t}\mathbf{x}),$$

or, since $e^{\lambda t}$ is always nonzero, $A\mathbf{x} = \lambda\mathbf{x}$; that is, (7.1.4) is a solution of (7.1.3) if and only if λ and $\mathbf{x}$ are an eigenvalue and a corresponding eigenvector of A.

An important type of matrix is one that has n linearly independent eigenvectors (see Theorem A.2.1 in Appendix 2 for the definition of linear independence). If this is the case, and $\lambda_1, \ldots, \lambda_n$ and $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are the eigenvalues and corresponding eigenvectors, then

$$\mathbf{y}_1(t) = e^{\lambda_1 t}\mathbf{x}_1, \qquad \mathbf{y}_2(t) = e^{\lambda_2 t}\mathbf{x}_2, \quad \cdots, \quad \mathbf{y}_n(t) = e^{\lambda_n t}\mathbf{x}_n \tag{7.1.5}$$

is a complete set of linearly independent solutions of the differential equation (7.1.3). Hence, any solution of (7.1.3) may be written in the form

$$\mathbf{y}(t) = \sum_{i=1}^{n} c_i \mathbf{y}_i(t) = \sum_{i=1}^{n} c_i e^{\lambda_i t}\mathbf{x}_i, \tag{7.1.6}$$

where the constants $c_1, \ldots, c_n$ may be determined by initial or other conditions. Thus, the general solution of (7.1.3) may be obtained by solving the eigenvalue problem for the matrix A. If A does not have n linearly independent eigenvectors, a similar but more complicated representation of the solution may be given.

Linearly Independent Eigenvectors

We now discuss in more detail the property of a matrix having n linearly independent eigenvectors. As in Appendix 2, a similarity transformation of the matrix A is of the form PAP^{-1}, where P is any nonsingular matrix. A similarity transformation of A arises from a change of variables; for example, consider the system of equations $A\mathbf{x} = \mathbf{b}$ and make the change of variables $\mathbf{y} = P\mathbf{x}$ and $\mathbf{c} = P\mathbf{b}$, where P is a nonsingular matrix. In the new variables the system of equations is $AP^{-1}\mathbf{y} = P^{-1}\mathbf{c}$ or, upon multiplying through by P, $PAP^{-1}\mathbf{y} = \mathbf{c}$. Thus, the coefficient matrix of the system in the new variables is the similarity transform PAP^{-1}.

An important property of similarity transformations is that they preserve the eigenvalues of A: the matrices A and PAP^{-1} have the same eigenvalues. This is easily seen by considering the characteristic polynomial and using the fact that the determinant of a product of matrices is the product of the determinants. Thus

$$\begin{aligned}\det(A - \lambda I) = \det(PP^{-1})\det(A - \lambda I) &= \det(P)\det(A - \lambda I)\det(P^{-1})\\ &= \det(PAP^{-1} - \lambda I),\end{aligned}$$

which shows that the characteristic polynomials, and hence the eigenvalues, of A and PAP^{-1} are identical. However, the eigenvectors change under a similarity transformation. Indeed,

$$PAP^{-1}\mathbf{y} = \lambda\mathbf{y} \quad \text{or} \quad AP^{-1}\mathbf{y} = \lambda P^{-1}\mathbf{y}$$

shows that the eigenvector $\mathbf{y}$ of PAP^{-1} is related to the eigenvector $\mathbf{x}$ of A by $P^{-1}\mathbf{y} = \mathbf{x}$ or $\mathbf{y} = P\mathbf{x}$.

An important question is how "simple" the matrix A may be made under a similarity transformation. A basic result in this regard, which brings us back to linear independence of the eigenvectors, is the following:

> THEOREM 7.1.1 *A matrix A is similar to a diagonal matrix if and only if A has n linearly independent eigenvectors.*

The proof of this theorem is both simple and illustrative. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be n linearly independent eigenvectors of A with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$, and let P be the matrix with columns $\mathbf{x}_1, \ldots, \mathbf{x}_n$: then P is nonsingular since its columns are linearly independent. By the basic definition $A\mathbf{x}_i = \lambda_i\mathbf{x}_i$ applied to each column of P, we have

$$AP = A(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n) = (\lambda_1\mathbf{x}_1, \ldots, \lambda_n\mathbf{x}_n) = PD, \tag{7.1.7}$$

where D is the diagonal matrix $\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Thus (7.1.7) is equivalent to $A = PDP^{-1}$, which shows that A is similar to a diagonal matrix whose diagonal entries are the eigenvalues of A. Conversely, if A is similar to a diagonal matrix, then (7.1.7) shows that the columns of the similarity matrix P must be eigenvectors of A, and they are linearly independent by the nonsingularity of P.

Two important special cases of the preceding result are the following theorems, which we state without proof.

THEOREM 7.1.2 *If A has distinct eigenvalues, then A is similar to a diagonal matrix.*

THEOREM 7.1.3 *If A is a real symmetric matrix (*that is, $A = A^T$*), then A is similar to a diagonal matrix, and the similarity matrix may be taken to be orthogonal (*that is, $PP^T = I$*).*

Symmetric matrices are extremely important in applications. They also have many nice properties as regards their eigenvalues and eigenvectors. In particular, the eigenvalues of a symmetric matrix are always real and are positive if A is positive definite. Moreover, the last part of Theorem 7.1.3 can be rephrased to say that the eigenvectors can be chosen to be orthonormal (Appendix 2).

The Jordan Form

Theorem 7.1.2 shows that if a matrix A does not have n linearly independent eigenvectors, then necessarily it has multiple eigenvalues. (But note that a matrix may have n linearly independent eigenvectors even though it has multiple eigenvalues; the identity matrix is an example.) The matrix

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{7.1.8}$$

is a simple example of a matrix that does not have two linearly independent eigenvectors (see Exercise 7.1.4) and is not similar to a diagonal matrix. However, any $n \times n$ matrix may be made similar to a matrix of the form

$$J = \begin{bmatrix} \lambda_1 & \delta_1 & & & \\ & \lambda_2 & \delta_2 & & \\ & & \ddots & \ddots & \\ & & & & \delta_{n-1} \\ & & & & \lambda_n \end{bmatrix},$$

where the λ_i are the eigenvalues of A and the δ_i are either 0 or 1. If q is the number of δ_i that are nonzero, then A has $n - q$ linearly independent eigenvectors, and whenever a δ_i is nonzero, then the eigenvalues λ_i and λ_{i-1} are identical. Thus the matrix J can be partitioned as

$$J = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_p \end{bmatrix}, \tag{7.1.9a}$$

where p is the number of linearly independent eigenvectors, and each J_i is a matrix of the form

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ & & & \lambda_i \end{bmatrix} \qquad (7.1.9b)$$

with identical eigenvalues and all 1's on the first superdiagonal. The matrix J of (7.1.9) is called the *Jordan canonical form* of A. Note that if A has n linearly independent eigenvectors, then $p = n$; in this case each J_i reduces to a 1×1 matrix, and J is diagonal.

The Jordan canonical form is for useful theoretical purposes but not very useful in practice. For many computational purposes it is very desirable to work with orthogonal or unitary matrices. (A *unitary* matrix U is a complex matrix that satisfies $U^{\star}U = I$, where $U^{\star}$ is the conjugate transpose of U; a real unitary matrix is an orthogonal matrix.) We next state without proof two basic results on similarity transformations with unitary or orthogonal matrices.

> SCHUR'S THEOREM. *For an arbitrary $n \times n$ matrix A, there is a unitary matrix U such that $UAU^{\star}$ is triangular.*
>
> MURNAGHAN-WINTNER THEOREM. *For a real $n \times n$ matrix A, there is an orthogonal matrix P so that*
>
> $$PAP^T = \begin{bmatrix} T_{11} & & \cdots & T_{1m} \\ & T_{22} & & \\ & & \ddots & \vdots \\ & & & T_{mm} \end{bmatrix},$$
>
> *where each T_{ii} is either 2×2 or 1×1.*

In the case of Schur's Theorem, the diagonal elements of $UAU^{\star}$ are the eigenvalues of A since $UAU^{\star}$ is a similarity transformation. If A is real but has some complex eigenvalues, then U is necessarily complex. The Murnaghan-Winter Theorem comes as close to a triangular form as possible with a real orthogonal matrix. In this case, if T_{ii} is 1×1, then it is a real eigenvalue of A, whereas if T_{ii} is 2×2, its two eigenvalues are a complex conjugate pair of eigenvalues of A.

Other Differential Equations

We now return to further examples of eigenvalue problems. Many applications lead in certain simple cases to the ordinary differential equation

$$-y''(x) = \lambda y(x), \qquad y(0) = 0, \qquad y(1) = 0. \qquad (7.1.10)$$

Here we wish to find values – again called eigenvalues – of the scalar λ so that (7.1.10) has corresponding nonzero solutions – called eigenfunctions – that satisfy the given zero boundary conditions. This particularly simple problem can be solved explicitly. There are infinitely many eigenvalues and corresponding eigenfunctions that are given by

$$\lambda_k = k^2\pi^2 \qquad y_k(x) = \sin k\pi x, \qquad k = 1, 2, \ldots, \tag{7.1.11}$$

as is easily checked by substitution into (7.1.10).

Next suppose that (7.1.10) is modified by adding a nonconstant coefficient of y; that is,

$$-y''(x) = \lambda c(x) y(x), \qquad y(0) = 0, \qquad y(1) = 0, \tag{7.1.12}$$

where c is a given positive function. Now it is no longer possible, in general, to obtain the eigenvalues and eigenfunctions of (7.1.12) explicitly, but we can approximate them numerically by the following procedure. Just as in the treatment of boundary-value problems in Chapter 3, we discretize the interval $[0, 1]$ with grid points $x_i = ih$, $i = 0, 1, \ldots, n+1$, $h = 1/(n+1)$, and replace the second derivative in (7.1.12) by the corresponding difference quotient. This gives the discrete equations

$$\frac{1}{h^2}(-y_{i+1} + 2y_i - y_{i-1}) = \lambda c_i y_i, \qquad i = 1, \ldots, n, \tag{7.1.13}$$

where $c_i = c(x_i)$, $y_0 = y_{n+1} = 0$, and y_i is an approximation to $y(x_i)$.

The equations (7.1.13) constitute a matrix eigenvalue problem of the form

$$A\mathbf{y} = \lambda B\mathbf{y}, \tag{7.1.14}$$

where A is the $(2, -1)$ tridiagonal matrix of (3.1.10) and B is a diagonal matrix with elements $h^2 c_i$. Equation (7.1.14) is an example of a *generalized eigenvalue problem* in which the matrix B on the right-hand side of the equation is not the identity matrix. In the present case we have assumed that the function $c(x)$ is positive; therefore B is non-singular and we can multiply (7.1.14) by B^{-1} to convert it to the standard eigenvalue problem $B^{-1}A\mathbf{y} = \lambda\mathbf{y}$.

It is not always advisable to convert (7.1.14) back to a standard eigenvalue problem even if B is non-singular (see the Supplementary Discussion of Section 7.2). Moreover, if A and B are symmetric, the product $B^{-1}A$ is not symmetric, in general. However, if B is also positive definite, we can convert (7.1.14) to a standard eigenvalue problem for a symmetric matrix as follows. First, compute the Cholesky decomposition $B = LL^T$ (Section 4.5). Then, multiply (7.1.14) by L^{-1} so that (7.1.14) becomes

$$L^{-1}AL^{-T}L^T y = \lambda L^T y \quad \text{or} \quad \hat{A}\mathbf{z} = \lambda\mathbf{z},$$

where $\mathbf{z} = L^T\mathbf{y}$ and $\hat{A} = L^{-1}AL^{-T}$ is symmetric.

Although the main purpose of this chapter is to describe methods for computing eigenvalues and eigenvectors, it is important to note that many problems require information only about the location of eigenvalues and not their precise values. As an example of this we return to the system (7.1.3) of ordinary differential equations. An important property of this system is whether all solutions tend to zero as t tends to infinity. If so, the zero solution is said to be *asymptotically stable.* Assuming again that the matrix A has n linearly independent eigenvectors, all solutions will go to zero as t goes to infinity if and only if each of the solutions (7.1.5) does, and since the vectors $\mathbf{x}_i$ are constant, this will be the case if and only if $e^{\lambda_i t}$ approaches zero as t approaches infinity for each i. If λ_i is real, this will be the case if and only if $\lambda_i < 0$, and if λ_i is complex, the real part of λ_i, denoted by $\mathrm{Re}(\lambda_i)$, must be negative. Thus the zero solution of (7.1.3) is asymptotically stable if and only if

$$\mathrm{Re}(\lambda_i) < 0, \qquad i = 1, \ldots, n, \tag{7.1.15}$$

so that all the eigenvalues lie in the left-half of the complex plane. A related example is an iterative method of the form

$$\mathbf{x}^{k+1} = A\mathbf{x}^k + \mathbf{d}, \qquad k = 0, 1, \ldots \,. \tag{7.1.16}$$

As we will see in Chapter 9, the iterates $\mathbf{x}^k$ will converge for any starting vector $\mathbf{x}^0$ if and only if all the eigenvalues of A satisfy $|\lambda_i| < 1$. This will be true if $||A|| < 1$ for some norm, but the following approach is sometimes more easily applied.

Gerschgorin's Theorem

Let $A = (a_{ij})$ be a real or complex $n \times n$ matrix and let

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad i = 1, \ldots, n.$$

That is, r_i is the sum of the absolute values of the off-diagonal elements in the ith row of A. Next define disks in the complex plane centered at a_{ii} and with radius r_i:

$$\Lambda_i = \{z : |z - a_{ii}| \leq r_i\}, \qquad i = 1, \ldots, n.$$

We then have the following:

GERSCHGORIN'S THEOREM *All the eigenvalues of A lie in the union of the disks $\Lambda_1, \ldots, \Lambda_n$. Moreover, if S is a union of m disks such that S is disjoint from all the other disks, then S contains exactly m eigenvalues of A (counting multiplicities).*

As a simple example of the use of Gerschgorin's theorem, consider the matrix

$$A = \frac{1}{16}\begin{bmatrix} -8 & -2 & 4 \\ -1 & -4 & 2 \\ 2 & 2 & -10 \end{bmatrix}, \qquad (7.1.17)$$

for which the Gerschgorin disks are illustrated in Figure 7.1. Note that we can immediately conclude that all eigenvalues of A have negative real part; hence if A were the coefficient matrix of the system of differential equations (7.1.3), the zero solution of that system would be asymptotically stable. Similarly, we can immediately conclude that the eigenvalues of A are all less than 1 in absolute value, so the vectors $\mathbf{x}^k$ defined by (7.1.16) converge.

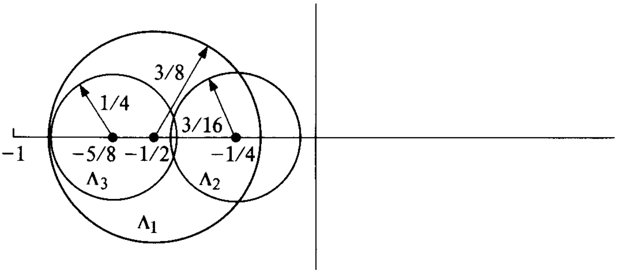

Figure 7.1: *Gerschgorin's Disks in the Complex Plane*

To illustrate the second part of Gerschgorin's Theorem, suppose that the second row of the matrix of (7.1.17) is changed to $\frac{1}{16}(-1, 6, 2)$. Then the disk Λ_2 is centered at $+\frac{3}{8}$, again with radius $\frac{3}{16}$. Since Λ_2 is now disjoint from the other two disks, it contains exactly one eigenvalue of A. Moreover, since any complex eigenvalues of A must occur in conjugate pairs, this eigenvalue must be real and therefore lies in the interval $[\frac{3}{16}, \frac{9}{16}]$.

The proof of the first part of Gerschgorin's Theorem is very easy. Let λ be any eigenvalue of A, and $\mathbf{x}$ a corresponding eigenvector. Then, by (7.1.1),

$$(\lambda - a_{ii})x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j, \qquad i = 1, \ldots, n.$$

If we let x_k be the component of largest absolute value in the vector $\mathbf{x}$, then

$$|\lambda - a_{kk}| \leq \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{kj}|\frac{|x_j|}{|x_k|} \leq \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{kj}|.$$

Thus λ is in the disk centered at a_{kk} and therefore in the union of all the disks. The proof of the second part of the theorem is more complicated and relies on the fact that the eigenvalues of a matrix are continuous functions of the elements of the matrix.

By a simple similarity transformation, it is sometimes possible to use Gerschgorin's Theorem to extract additional information about the eigenvalues. For example, consider the matrix

$$A = \begin{bmatrix} 8 & 1 & 0 \\ 1 & 12 & 1 \\ 0 & 1 & 10 \end{bmatrix}.$$

Since A is symmetric its eigenvalues are real, and by Gerschgorin's Theorem we conclude that they lie in the union of the intervals $[7,9]$, $[10,14]$, $[9,11]$. Since these intervals are not disjoint we cannot yet conclude that any of them contains an eigenvalue. However, if we do a similarity transformation with the matrix $D = \text{diag}(d,1,1)$ we obtain

$$DAD^{-1} = \begin{bmatrix} 8 & d & \\ d^{-1} & 12 & 1 \\ & 1 & 10 \end{bmatrix}.$$

By Gerschgorin's Theorem, the eigenvalues of this matrix (which are the same as those of A) lie in the union of the intervals $[8-d, 8+d]$, $[11-d^{-1}, 13+d^{-1}]$, $[9,11]$. As long as $1 > d > \frac{1}{2}[3-\sqrt{5}]$, the first interval is disjoint from the others and thus contains exactly one eigenvalue. In particular, the interval $[7.6, 8.4]$ contains one eigenvalue.

Another important use of Gerschgorin's Theorem is in ascertaining the change in the eigenvalues of a matrix due to changes in the coefficients. Let A be a given $n \times n$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and suppose that E is a matrix whose elements are small compared to those of A; for example, E may be the rounding errors committed in entering the matrix A into a computer. Suppose that $\mu_1, \ldots, \mu_n$ are the eigenvalues of $A + E$. Then, what can one say about the changes $|\lambda_i - \mu_i|$? We next give a relatively simple result in the case that A has n linearly independent eigenvectors. Recall, from Appendix 2, that the infinity norm of a matrix is the maximum value of the sums of the absolute values of the elements in each row.

THEOREM 7.1.4 *Assume that $A = PDP^{-1}$, where D is the diagonal matrix of eigenvalues of A, and let $d = ||P^{-1}EP||_\infty$. Then every eigenvalue of $A + E$ is within d of an eigenvalue of A.*

The proof of this theorem is a simple consequence of Gerschgorin's Theorem. Set $C = P^{-1}(A+E)P$. Then C has the same eigenvalues $\mu_1, \ldots, \mu_n$ as

$A+E$. Let $B = P^{-1}EP$. Then $C = D + B$, and the diagonal elements of C are $\lambda_i + b_{ii}$, $i = 1, \ldots, n$. Hence, by Gerschgorin's Theorem, the eigenvalues $\mu_1, \ldots, \mu_n$ are in the union of the disks

$$\{z : |z - \lambda_i - b_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |b_{ij}|\}.$$

Therefore, given any μ_k, there is an i such that

$$|\mu_k - \lambda_i - b_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |b_{ij}|,$$

or

$$|\mu_k - \lambda_i| \leq \sum_{j=1}^{n} |b_{ij}| \leq d,$$

which was to be shown.

Ill-conditioned Eigenvalues

Note that the quantity d need not be small even though $||E||_\infty$ is small; this will depend on P. In general, the more ill-conditioned the matrix P (in the sense of Chapter 4), the more ill-conditioned will be the eigenvalues of A, and the more the eigenvalues may change because of small changes in the coefficients of A. We give a simple example of this. Let

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1+10^{-10} \end{bmatrix}, \qquad A+E = \begin{bmatrix} 1 & 1 \\ 10^{-10} & 1+10^{-10} \end{bmatrix}.$$

Then the eigenvalues of A are 1 and $1 + 10^{-10}$, and those of $A+E$ are approximately 1 ± 10^{-5}. Thus a change of 10^{-10} in one element of A has caused a change 10^5 times as large in the eigenvalues. The reason for this is that the matrix P of eigenvectors of A is very ill-conditioned. It is easy to verify that

$$P = \begin{bmatrix} 1 & 1 \\ 0 & 10^{-10} \end{bmatrix}, \qquad P^{-1} = \begin{bmatrix} 1 & -10^{10} \\ 0 & 10^{10} \end{bmatrix}.$$

Therefore the matrix $P^{-1}EP$ of Theorem 7.1.4 is

$$P^{-1}EP = \begin{bmatrix} 1 & -10^{10} \\ 0 & 10^{10} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 10^{-10} & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 10^{-10} \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix},$$

and thus $d = ||P^{-1}EP||_\infty = 2$. Note that the actual change in the eigenvalues is far smaller than this bound.

It is an interesting and important fact that the eigenvalues of a symmetric matrix are always well-conditioned; this is the interpretation of the following theorem, stated without proof.

THEOREM 7.1.5 *Let A and B be real symmetric $n \times n$ matrices with eigenvalues $\lambda_1, \ldots, \lambda_n$ and $\mu_1, \ldots, \mu_n$, respectively. Then given any μ_j, there is a λ_i such that*

$$|\lambda_i - \mu_j| \leq ||A - B||_2.$$

Note that in this theorem it is the 2-norm (see Appendix 2) that is used, and hence the result does not follow directly from Theorem 7.1.4.

In this section we have given various examples of eigenvalue problems and some of the basic mathematical theory. In the remainder of this chapter we will discuss the foundation of various methods for computing eigenvalues and eigenvectors.

Supplementary Discussion and References: 7.1

Further discussion of the use of eigenvalues for solving linear ordinary differential equations can be found in most elementary differential equation textbooks. See also Ortega [1987]. Discussions of the theory of matrix eigenvalue problems in a form most suitable for scientific computing are given in Golub and Van Loan [1989], Ortega [1987], and Ortega [1990]. See also Wilkinson [1965], Householder [1964], Stewart [1973], and Parlett [1980].

EXERCISES 7.1

7.1.1. Compute the characteristic equations $\det(A - \lambda I)$ for the matrices

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \qquad A = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}.$$

Next compute the eigenvalues of A by obtaining the roots of these polynomials, and then compute the eigenvectors by solving the homogeneous equations (7.1.2).

7.1.2. Give the solution of the initial-value problem

$$\mathbf{y}'(t) = A\mathbf{y}(t), \qquad \mathbf{y}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

in terms of the eigenvalues and eigenvectors that were computed in Exercise 7.1.1.

7.1.3. If A is the matrix

$$A = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix},$$

determine whether the zero solution of $\mathbf{y}' = A\mathbf{y}$ is asymptotically stable.

7.1.4. Compute an eigenvector of the matrix (7.1.8) and show that there are no other linearly independent eigenvectors.

7.1.5. Assume that a matrix A has two eigenvalues $\lambda_1 = \lambda_2$ and corresponding linearly independent eigenvectors $\mathbf{x}_1$, $\mathbf{x}_2$. Show that any linear combination $c_1\mathbf{x}_1 + c_2\mathbf{x}_2$ is also an eigenvector.

7.1.6. Suppose that A has eigenvalues $\lambda_1, \ldots, \lambda_n$ and eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Show that for any constants α and β, $\alpha A + \beta I$ has eigenvalues $\alpha\lambda_i + \beta$ and corresponding eigenvectors $\mathbf{x}_i$. Use this result in combination with Exercise 4.4.5 to show that the matrix

$$A = \begin{bmatrix} a & b & & & & \\ b & a & b & & & \\ & b & a & b & & \\ & & \ddots & \ddots & \ddots & \\ & & & b & a & b \\ & & & & b & a \end{bmatrix}$$

has eigenvalues $\lambda_k = a + 2b\cos[k\pi/(n+1)]$, $k = 1, \ldots, n$. What are the eigenvectors?

7.1.7. A polynomial in a matrix A is $p(A) = \alpha_0 + \alpha_1 A + \cdots + \alpha_m A^m$. If A has an eigenvalue λ and corresponding eigenvector $\mathbf{x}$, show that $p(\lambda) = \alpha_0 + \alpha_1\lambda + \cdots + \alpha_m\lambda^m$ is an eigenvalue of $p(A)$ with corresponding eigenvector $\mathbf{x}$. Formulate and prove the corresponding result for a rational function of a matrix.

7.1.8. If A and B are $n \times n$ matrices at least one of which is nonsingular, show that AB and BA have the same eigenvalues.

7.1.9. Find the Gerschgorin disks for the matrix

$$A = \begin{bmatrix} 4 & 2 & 2 \\ 1 & 8 & 1 \\ 1 & 1 & 12 \end{bmatrix}.$$

Use the fact that A and A^T have the same eigenvalues to conclude that A has an eigenvalue that satisfies $|\lambda - 4| \leq 2$ by applying Gerschgorin's Theorem to A^T.

7.1.10. Use Gerschgorin's Theorem to prove that a symmetric strictly diagonally dominant matrix with positive diagonal elements is positive definite.

7.1.11. If $p(\lambda) = a_0 + a_1\lambda + \cdots + a_{n-1}\lambda^{n-1} + \lambda^n$, the matrix

$$A = \begin{bmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ -a_0 & -a_1 & & \cdots & -a_{n-1} \end{bmatrix}$$

is called the *companion matrix* (or *Frobenius matrix*) of p. Show that $p(\lambda)$ is the characteristic polynomial of A. Then apply Gerschgorin's Theorem to both A and A^T to obtain bounds for the roots of p.

7.1.12. Let A be a real, symmetric matrix. Show that Schur's Theorem implies that there exists an orthogonal matrix Q such that $Q^TAQ = D$, where D is a diagonal matrix.

7.1.13. A matrix A is *skew-symmetric* if $A^T = -A$. Let A be a real, skew-symmetric matrix and $PAP^T = T$, where T is given by the Murnaghan-Wintner Theorem. Describe the structure of T in this case.

7.1.14. Show how to write the differential equation

$$y''(t) + ay'(t) + by(t) = 0$$

as a system of first-order equations in the form (7.1.3).Then give conditions on a and b so that $y(t) \to 0$ as $t \to \infty$ for any initial conditions.

7.1.15. Suppose that the matrix A has p zero eigenvalues and corresponding linearly independent eigenvectors. Show how to obtain the solution of the differential equation $A\mathbf{y}' = \mathbf{y}$ even though A^{-1} does not exist. What does this imply about initial or boundary conditions?

7.1.16. Compare the eigenvalues of (7.1.10) with those of the matrix

$$B = h^{-2}A, \quad h = \frac{\pi}{n+1},$$

where A is the $(2, -1)$ tridiagonal matrix of (3.1.10). Which eigenvalues of B are accurate approximations of those of the differential equation?

7.1.17. Consider the equation (7.1.14) where A and B are symmetric and B is positive definite. Show that we can construct matrices F and D such that

$$A = FDF^T \text{ and } B = FF^T,$$

where D is a diagonal matrix whose entries are the eigenvalue of (7.1.14).

7.1.18. Consider a matrix J_i of the form (7.1.9b). Show that there exists a diagonal matrix D so that DJ_iD^{-1} is the same as J_i except that the off-diagonal 1's are replaced by ϵ.

7.1.19. Assume that $A = A^T$. In Theorem 7.1.4, give an upper bound for $d = \|P^{-1}EP\|_\infty$.

7.2 The QR Method

We now begin the study of methods to compute the eigenvalues and eigenvectors of an $n \times n$ matrix A. We will assume that A is real but, in general, it may have complex eigenvalues and eigenvectors. We will first consider a method that applies to such matrices, and then specialize to the important special case in which A is symmetric and thus has real eigenvalues.

We begin with the QR factorization of Section 4.5:

$$A = QR, \tag{7.2.1}$$

where Q is orthogonal and R is upper triangular. Now form a new matrix by multiplying these factors in reverse order:

$$A_1 = RQ. \tag{7.2.2}$$

Since

$$A = QR = QRQQ^{-1} = QA_1Q^{-1} = QA_1Q^T,$$

A_1 is similar to A and has the same eigenvalues (see also Exercise 7.1.8). We then compute the QR factorization of A_1 and reverse the order of the factors to obtain another matrix A_2:

$$A_1 = Q_1R_1, \qquad A_2 = R_1Q_1.$$

Again, A_2 is similar to A_1, and hence to A. We continue this process, alternately doing a QR factorization and then reversing the order of the factors to generate a sequence of matrices

$$A_k = Q_kR_k, \qquad A_{k+1} = R_kQ_k, \qquad k = 0, 1, \ldots, \tag{7.2.3}$$

where $A_0 = A$. All of these matrices are similar and thus have the same eigenvalues as A. The generation of the matrices A_k of (7.2.3) is called the *QR algorithm.* For this algorithm we have the following basic convergence theorem, which we state without proof.

THEOREM 7.2.1. *(QR Convergence) If the eigenvalues of A satisfy*

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|, \tag{7.2.4}$$

then the matrices A_k of (7.2.3) converge to an upper triangular matrix whose diagonal elements are the eigenvalues of A. Moreover, if $A = PDP^{-1}$, where $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, and if P^{-1} has an LU decomposition, then

$$A_k \to T = \begin{bmatrix} \lambda_1 & * & \cdots & * \\ & \ddots & & \vdots \\ & & & * \\ & & & \lambda_n \end{bmatrix}, \quad as \quad k \to \infty, \tag{7.2.5}$$

and the rate of convergence to zero of the off-diagonal elements $a_{ij}^{(k)}$ of A_k is given by

$$a_{ij}^{(k)} = 0\left(\frac{|\lambda_i|^k}{|\lambda_j|^k}\right), \qquad k \to \infty, \qquad i > j. \qquad (7.2.6)$$

The technical condition that P^{-1} have an LU decomposition ensures that the eigenvalues appear on the main diagonal of T in descending order of magnitude. This is the usual situation, although if the condition is not satisfied the order of the eigenvalues may be different. The more stringent condition is (7.2.4), which precludes not only multiple eigenvalues but also complex conjugate pairs of eigenvalues. If the matrix A is real, then all the factors Q_k and R_k are also real, and there is, of course, no possibility that the A_k could converge to a triangular matrix with complex eigenvalues. However, what does occur – which is the best that one could hope – is that the A_k will "converge" to an almost-triangular form illustrated by the matrix

$$\begin{bmatrix} \boxed{} & * & \cdots & & * \\ & \lambda_3 & & & \vdots \\ & & \boxed{} & & \\ & & & \boxed{} & \\ & & & \lambda_8 & * \\ & & & & \lambda_9 \end{bmatrix}. \qquad (7.2.7)$$

In this example we have assumed that there are three real eigenvalues λ_3, λ_8, λ_9 with distinct absolute values and three complex conjugate pairs of eigenvalues, again with distinct absolute values. The latter eigenvalues are determined by the three 2×2 matrices indicated by the blocks on the main diagonal. Actually, the elements of these 2×2 matrices do not converge, but their eigenvalues do converge to eigenvalues of A. Hence the computation of complex eigenvalues of real matrices does not present any problem. Note that (7.2.7) is the Murnaghan-Wintner form. Thus the QR algorithm attempts to obtain the Schur triangular form of the matrix when possible, and the Murnaghan-Wintner form otherwise.

Hessenberg Form

The QR algorithm as described so far is too inefficient to be effective, and two important modifications must be made. The first problem is that each step of (7.2.3) requires $0(n^3)$ operations, which makes the process very slow. We can circumvent this difficulty by making a preliminary reduction of the matrix A to a form for which the decomposition can be more rapidly computed. This is the *Hessenberg form*

$$\begin{bmatrix} * & * & & \cdots & * \\ * & & \ddots & & \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & & * & \\ 0 & \cdots & 0 & * & * \end{bmatrix}, \tag{7.2.8}$$

which has one non-zero diagonal below the main diagonal, while the elements above the main diagonal are, in general, non-zero . The reduction by similarity transformations of the original matrix A to Hessenberg form can be effected by the Householder transformations used in Section 4.5, as we now discuss.

Let $P_2 = I - 2\mathbf{w}_2\mathbf{w}_2^T$ be a Householder transformation such that P_2A has zeros in its first column below the second position:

$$P_2A = \begin{bmatrix} * & * & \cdots & * \\ * & * & & \\ 0 & * & & \vdots \\ \vdots & \vdots & & \\ 0 & * & \cdots & * \end{bmatrix}. \tag{7.2.9}$$

The vector $\mathbf{w}_2$ has a zero in the first component; otherwise, it is defined analogously to (4.5.13) by

$$\mathbf{w}_2 = \mu_2\mathbf{u}_2, \ \mathbf{u}_2^T = (0, a_{21} - s_2, a_{31}, \ldots, a_n), \tag{7.2.10}$$

where

$$s_2 = \pm(\sum_{j=2}^{n} a_{ji}^2)^{1/2}, \ \mu_2 = (2s_2^2 - 2a_{21}s_2)^{-1/2}.$$

Since we are performing similarity transformations, we must also multiply on the right by P_2^T:

$$A_2 = P_2AP_2^T = A - 2\mathbf{w}_2\mathbf{w}_2^TA - 2A\mathbf{w}_2\mathbf{w}_2^T + 4\mathbf{w}_2^TA\mathbf{w}_2\mathbf{w}_2\mathbf{w}_2^T. \tag{7.2.11}$$

Since the first component of $\mathbf{w}_2$ is zero, the multiplication on the right by P_2^T does not change the zeros introduced in the first column by P_2A. Thus A_2 is similar to A and has the form shown in (7.2.9). We now continue this process. P_3 is determined by a vector $\mathbf{w}_3$ whose first two components are zero and is otherwise chosen analogously to (7.2.10) to produce zeros in the third column below the third element. And so on. After $n-2$ Householder transformations, the matrix

$$PAP^T = H, \qquad P = P_{n-1} \cdots P_2, \tag{7.2.12}$$

will have the Hessenberg form (7.2.8). Since (7.2.12) is a similarity transformation, A and H will have the same eigenvalues. This reduction requires $0(n^3)$ operations; more precisely, it requires roughly twice as many operations as a QR factorization.

A particularly important special case of (7.2.12) is when A is symmetric. In this case H must also be symmetric (Exercise 7.2.1); thus it is tridiagonal. We summarize the discussion above as:

> THEOREM 7.2.2. *An $n \times n$ real matrix can be reduced to Hessenberg form* (7.2.8) *by $n-2$ Householder similarity transformations. If A is symmetric, the Hessenberg form is tridiagonal.*

We now apply the QR method (7.2.3) to the Hessenberg matrix H. The QR factorization of H can be carried out by Householder transformations, as before, but since there is only one non-zero element below the main diagonal in each column it is slightly easier to use Givens transformations (see Section 4.5). Each Givens transformation will eliminate one zero below the main diagonal and thus $n-1$ Givens transformations produce the QR factorization. As discussed in Section 4.5, the first Givens transformation modifies the first two rows of H and requires $4n$ multiplications and $2n$ additions. At each stage the length of the rows decreases by one and hence the total number of operations for the row modifications is

$$4\sum_{k=2}^{n} k \text{ multiplications } + 2\sum_{k=2}^{n} k \text{ additions } = 0(n^2) \text{ operations.}$$

In addition to these row modifications it is necessary to obtain the multipliers but the overall operation count is still $0(n^2)$, as opposed to $0(n^3)$ for a full matrix. This is the advantage of using the Hessenberg form.

The initial reduction of A to Hessenberg form would not be effective if the QR method itself did not preserve the Hessenberg form. But it does. Let $Q^T = Q_{n-1} \cdots Q_1$ be the product of the Givens transformations so that $Q = Q_1^T \cdots Q_{n-1}^T$. Each Q_i has off-diagonal elements only in the $(i+1, i)$ and $(i, i+1)$ positions and hence Q itself is a Hessenberg matrix (Exercise 7.2.2). Then since R is upper triangular, the product RQ is a Hessenberg matrix and can be formed in $0(n^2)$ operations (Exercise 7.2.3). Thus all of the matrices generated by the QR method retain the Hessenberg form and each complete QR step requires $0(n^2)$ operations.

Shifting

Even with the initial reduction of the matrix to Hessenberg form, the QR method is still inefficient due to the possibly slow rate of convergence to zero of the subdiagonal elements. This rate of convergence is indicated by (7.2.6), which shows that if two eigenvalues, say λ_i and λ_{i+1}, are very close in absolute value, then the off-diagonal element in position $(i+1, i)$ will converge to zero very slowly.

We will attempt to mitigate this convergence problem by *shifting* the eigenvalues of H. Suppose that $\hat{\lambda}_n$ is a good approximation to the smallest eigenvalue, λ_n (assumed real), and consider the matrix $\hat{H} = H - \hat{\lambda}_n I$, which has eigenvalues $\lambda_1 - \hat{\lambda}_n, \ldots, \lambda_n - \hat{\lambda}_n$. If we apply the QR method to $\hat{H}$, then the off-diagonal element in the last row of the matrices H_k will converge to zero as powers of the quotient $(\lambda_n - \hat{\lambda}_n)/(\lambda_{n-1} - \hat{\lambda}_n)$, as opposed to the quotient λ_n/λ_{n-1}. For example, suppose that $\lambda_n = 0.99$, $\lambda_{n-1} = 1.1$, and $\hat{\lambda}_n = 1.0$. Then, $\lambda_n/\lambda_{n-1} = 0.9$ while $|\lambda_n - \hat{\lambda}_n|/|\lambda_{n-1} - \hat{\lambda}_n| = 0.1$, so that the convergence of the $(n, n-1)$ element is approximately 20 times faster for the matrix $\hat{H}$.

Of course, we usually will not know a good approximation $\hat{\lambda}_n$ to use as the shift parameter. However, as the QR process proceeds, if the (n, n) elements $h_{nn}^{(k)}$ of the matrices H_k are converging to the eigenvalue λ_n, we can use them as the shift parameters; that is, at the kth stage do the next QR step on the matrix $\hat{H}_k = H_k - h_{n,n}^{(k)} I$. Then we continue using the (n, n) element of the current matrix to make a shift at each stage. Each shift changes the eigenvalues of the original matrix by the amount of the shift, so we need to keep track of the accumulation of shifts that are made; indeed, it is this accumulation that converges to the eigenvalue λ_n. The convergence is signaled by the off-diagonal element in the last row becoming sufficiently small. When this occurs the last row and column of the matrix may be dropped, and to determine the eigenvalue λ_{n-1} we proceed with the resulting $(n-1) \times (n-1)$ submatrix. Note that the eigenvalues of this submatrix, and hence of the original matrix, have been changed by the total accumulation of shifts (which is the approximation to λ_n), and this must be added back to the other computed eigenvalues at the end of the computation. Alternatively, the shifts may be added back in at each stage, as illustrated by (7.2.13) in a different context, so that all of the matrices H_k retain the same eigenvalues.

The preceding discussion has been predicated on the assumption that the smallest eigenvalue, λ_n, is real. Now suppose that λ_n is complex. Then shifting by $h_{nn}^{(k)}$, which remains real, is not a particularly good strategy since the imaginary part of the eigenvalue cannot be approximated. Instead, as was discussed earlier, the eigenvalues of the lower right 2×2 submatrices of the matrices H_k produced by the unshifted QR algorithm will converge to the eigenvalue pair $\lambda_n, \lambda_{n-1} = \bar{\lambda}_n$. Hence we use the eigenvalues of these 2×2 submatrices as shift parameters. Consider the first step applied to the matrix H_1 and let $k_1, k_2 = \bar{k}_1$ be the eigenvalues of the lower right 2×2 submatrix. If we add back the shifts, we obtain

$$\begin{aligned} H_1 - k_1 I &= Q_1 R_1, \qquad & H_2 &= R_1 Q_1 + k_1 I, \\ H_2 - k_2 I &= Q_2 R_2, \qquad & H_3 &= R_2 Q_2 + k_2 I. \end{aligned} \tag{7.2.13}$$

If k_1 and k_2 are complex, the matrices H_1, H_2, Q_1, Q_2, R_1, and R_2 will generally be complex, and consequently the QR steps need to be carried out in complex arithmetic. However, an interesting fact is that H_3 is real (Exercise 7.2.5). Indeed, it is possible to carry out the transformation from H_1 to H_3 entirely in real arithmetic, although we will not go into the details of this here. This procedure is called the *double-shift QR method.* Even if the eigenvalues are real, it is a good strategy to shift twice using the eigenvalues of the lower right 2×2 submatrix. With this choice of shifts, as with shifting by the (n, n) element, the rate of convergence is usually at least quadratic.

There is another possibility that enhances the speed of the QR method. Suppose that the subdiagonal element $a_{i+1,i}$ of the Hessenberg matrix H is zero. Then H can be written in block form

$$H = \begin{bmatrix} H_1 & * \\ 0 & H_2 \end{bmatrix}, \tag{7.2.14}$$

and the eigenvalues of H are those of the matrices H_1 and H_2 (Exercise 7.2.6). Thus the QR method can be applied to these smaller matrices, which reduces the operation count. This observation can also be applied during the QR method: if it should happen that the elements in position $(i+1, i)$ converge to zero more rapidly than other off-diagonal elements, then the problem can be decomposed into two smaller problems.

Householder's Method

We now return to the important special case in which A is symmetric. In this case the Hessenberg matrix is tridiagonal (Theorem 7.2.2) and the reduction of the original matrix by Householder transformations is known as *Householder's method.* The QR method can again be used to compute the eigenvalues of the tridiagonal matrix T (see the Supplementary Discussion for alternative methods). In this case there are two simplifications. Since T is symmetric, its eigenvalues are necessarily real and therefore there is no need to be concerned with complex shifts or convergence of 2×2 submatrices. Secondly, the QR steps are very rapid; each requires only $0(n)$ operations and the tridiagonal form is preserved (Exercise 7.2.4).

Computation of Eigenvectors

We next discuss the computation of eigenvectors, if they are desired. Assume that an approximate eigenvalue has been computed. Then there are two steps to obtain the corresponding approximate eigenvector. First, compute the approximate eigenvector of the Hessenberg (or tridiagonal) matrix. We postpone the discussion of this until the following section since it is a special case of methods to be given there. Second, transform this eigenvector back to an eigenvector of the original matrix A. We now consider this second step.

Let $\mathbf{y}$ be an eigenvector of H corresponding to the eigenvalue λ. Let $H = PAP^T$, where $P = P_{n-1} \cdots P_2$ is the product of the Householder transformations $P_i = I - 2\mathbf{w}_i\mathbf{w}_i^T$ used to obtain the Hessenberg form. Then $\mathbf{x} = P^T\mathbf{y}$ is the corresponding eigenvector of A since

$$A\mathbf{x} = AP^T\mathbf{y} = P^TPAP^T\mathbf{y} = P^TH\mathbf{y} = \lambda P^T\mathbf{y} = \lambda\mathbf{x}.$$

If $\mathbf{y}$ is only an approximate eigenvector of H, we still use the same transformation, $P^T\mathbf{y}$, to obtain an approximate eigenvector of A. Thus

$$\mathbf{x} = P^T\mathbf{y} = (P_{n-1} \cdots P_2)^T\mathbf{y} = P_2^T \cdots P_{n-1}^T\mathbf{y}.$$

This is very easy to carry out. The first step is

$$P_{n-1}^T\mathbf{y} = (I - 2\mathbf{w}_{n-1}\mathbf{w}_{n-1}^T)\mathbf{y} = \mathbf{y} - 2(\mathbf{w}_{n-1}^T\mathbf{y})\mathbf{w}_{n-1}.$$

Then $P_{n-2}^T = I - 2\mathbf{w}_{n-2}\mathbf{w}_{n-2}^T$ is applied to this vector, and so on. Note that we need to retain the vectors $\mathbf{w}_i$ that were used to produce the Hessenberg matrix H from A. The non-zero components of the $\mathbf{w}_i$ can be stored in the corresponding subdiagonal positions of A that are set to zero, if desired.

We now summarize briefly the main points of this section. For the QR method to be efficient we must first reduce the original matrix A to Hessenberg form (tridiagonal if A is symmetric), and then incorporate shifts into the basic QR algorithm applied to this Hessenberg matrix. As the iteration proceeds, the eigenvalues are obtained one by one (or two at a time in the case of a complex conjugate pair), the matrix is reduced in size, and the iteration proceeds toward the remaining eigenvalues. Properly implemented, the QR algorithm is the best general-purpose method for nonsymmetric matrices. We have not been able to give all of the details necessary for such an implementation and have tried only to present the basic flavor of the method. The Supplementary Discussion gives references for further reading.

Supplementary Discussion and References: 7.2

It is tempting to try to find an orthogonal matrix so that P^TAP is diagonal if A is symmetric. Unfortunately, this cannot be done with a finite number of operations, except in trivial cases, but there is a classical algorithm that attempts to find P as a limit of a sequence of products $P_1 \cdots P_k$. This is *Jacobi's method*, in which each P_i is a Givens matrix and, in the simplest case, the elements of A are zeroed in the order $(2,1), (3,1), \ldots, (n,1), (3,2), (4,2), \ldots$. Ideally, after all subdiagonal elements have been zeroed we would have a diagonal matrix but non-zero elements generally will appear in positions that had previously been zeroed. The process is then repeated and, under mild assumptions, the matrices $A_k = P_k^T \cdots P_1^T A P_1 \cdots P_k$ converge as $k \to \infty$ to a diagonal matrix containing the eigenvalues of A, and $P_1 \cdots P_k$ converges to a matrix P whose columns are the eigenvectors. Although Jacobi's method is slow relative to the methods discussed in this section, it has been enjoying a recent revival due to its good properties on parallel computers (see, e.g., Golub and Van Loan [1989]).

The idea of reducing the original symmetric matrix to tridiagonal form, rather than attempting to obtain a diagonal matrix as in Jacobi's method, is due to J. W. Givens in the early 1950's. He used the plane rotation matrices now associated with his name. Shortly thereafter A. Householder noted that the reduction could be done more efficiently using elementary reflection matrices, now called Householder matrices.

The QR method was introduced independently by Francis [1961, 1962] and Kublanovskaya [1961]. It was preceeded by the corresponding algorithm based on the LU decomposition and called the LR algorithm by H. Rutishauser in 1958. This method proceeds as in (7.2.3), but the QR factorizations are replaced by LR (i.e. LU) factorizations. Although the LU factorizations are faster (see Section 4.5), the QR method in general enjoys better numerical stability properties and has been the method of choice.

Excellent codes for the QR method, and the special case of Householder's method for symmetric matrices, are contained in EISPACK (Garbow et al. [1979]), which is now being transformed to the new LAPACK package (Dongarra and Anderson et al. [1990]).

J. Wilkinson contributed immensely to the understanding and extension of all of the methods of this section. A wealth of material, including the proof of Theorem 7.2.1, detailed rounding error analyses, and further discussions of the practicalities of different methods may be found in his classic book (Wilkinson [1965]). See also Householder [1964] for a more mathematical treatment of some of the topics of this chapter, Parlett [1980] for results pertaining primarily to symmetric matrices, and Stewart [1973]. In particular, this latter book gives an implicitly shifted version of the QR method as well as relationships between the QR method and the power, inverse power, and Rayleigh quotient methods to be discussed in the next section. For a more recent review of all of the methods in this chapter, see Golub and Van Loan [1989].

In Section 7.1 it was mentioned that the generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$ can be converted to a standard problem $B^{-1}A\mathbf{x} = \lambda\mathbf{x}$ if B is nonsingular. An alternative is the QZ algorithm (Moler and Stewart [1973]; see also Golub and Van Loan [1989]), which is an extension of the QR algorithm to the generalized eigenvalue problem.

Although the QR algorithm is probably the best method, in general, for computing the eigenvalues of a symmetric tridiagonal matrix T, there are two attractive alternatives, each of which is sometimes very useful. The first is based on a *Sturm sequence*, which for a tridiagonal matrix with diagonal elements a_i and off-diagonal elements b_i (assumed to be non-zero), is a sequence of polynomials defined by

$$p_i(\lambda) = (a_i - \lambda)p_{i-1}(\lambda) - b_{i-1}^2 p_{i-2}(\lambda), \qquad i = 2, \ldots, n, \tag{7.2.15}$$

with $p_0(\lambda) \equiv 1$ and $p_1(\lambda) = a_1 - \lambda$. The polynomial p_k is the characteristic polynomial of the $k \times k$ leading principal submatrix of T. In particular, p_n is the characteristic polynomial of T and its roots are the eigenvalues of T. These polynomials have the remarkable property that the number of agreements in sign between consecutive terms in the sequence $1, p_1(\hat{\lambda}), \ldots, p_n(\hat{\lambda})$ is equal to the number of roots of p_n greater than or equal to $\hat{\lambda}$. (See Exercise 7.2.9 for a related result.) This property then allows a bisection type algorithm. In particular, by using two test points $\hat{\lambda}_1$ and $\hat{\lambda}_2$ it is possible to know the number of the roots in the interval $[\hat{\lambda}_1, \hat{\lambda}_2]$, which can be very useful. For further information see Parlett [1980], which also discusses the *spectrum splicing* method (which is essentially equivalent for tridiagonal matrices to Sturm sequences). This is based on an LDL^T decomposition of $T - \hat{\lambda}I$ and application of the Inertia Theorem to ascertain the number of eigenvalues of T greater than $\hat{\lambda}$ (for the Inertia Theorem see, e.g., Ortega [1987]).

The second alternative for symmetric tridiagonal matrices is an iterative method for the characteristic polynomial $p_n(\lambda)$ of T. Consider Newton's method (Section 5.2). We can differentiate the sequence (7.2.15) to obtain the corresponding sequence for the derivatives

$$p_i'(\lambda) = -p_{i-1}(\lambda) + (a_i - \lambda)p_{i-1}'(\lambda) - b_{i-1}^2 p_{i-2}'(\lambda), \qquad i = 2, \ldots, n, \tag{7.2.16}$$

where $p_0' = 0$ and $p_1' = -1$. The two sequences (7.2.15) and (7.2.16) can be evaluated together to yield $p_n(\lambda)$ and $p_n'(\lambda)$ to use in Newton's method. This can be combined with the Sturm sequence property to ascertain an interval in which a root is known to lie before applying Newton's method to achieve rapid convergence to the root. See, for example, Wilkinson [1965] for further details. Other root-finding techniques could also be used in place of Newton's method.

EXERCISES 7.2

7.2.1. If A is symmetric, show that PAP^T is also symmetric for any matrix P. Apply this, in particular, to (7.2.12) to conclude that the Hessenberg matrix H is tridiagonal if A is symmetric.

7.2.2. Show that if Q_i is a diagonal matrix except for non-zero elements in the $(i+1, i)$ and $(i, i+1)$ positions, then the product $Q_1 \cdots Q_{n-1}$ is a Hessenberg matrix.

7.2.3. Show that if Q is a Hessenberg matrix and R is upper-triangular, then the product RQ is a Hessenberg matrix. Show that this multiplication requires $0(n^2)$ operations.

7.2.4. Let A be a symmetric banded matrix. Show that the QR method (7.2.3) preserves the bandwidth of A. What is the operation count for one step? Specialize this to show that if the QR method is applied to a symmetric tridiagonal matrix, the tridiagonal form is preserved and each QR step requires $0(n)$ operations. What happens for nonsymmetric banded matrices?

7.2.5. Let H_1 be real and $k_1, k_2 = \bar{k}_1$ be complex scalars. Show that the matrix H_3 defined by (7.2.13) is real.

7.2.6. Suppose that the matrix A has the block form

$$A = \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}.$$

Show that $\det(A - \lambda I) = \det(A_1 - \lambda I)\det(A_2 - \lambda I)$ so that the eigenvalues of A are those of A_1 and A_2.

7.2.7. Let A be skew-symmetric ($A^T = -A$). Show that if H is the Hessenberg matrix obtained by Householder reduction, then H is tridiagonal and skew-symmetric. Note that a skew-symmetric matrix has zero main diagonal elements. Can you use this to simplify the QR algorithm?

7.2.8. Let

$$A = \begin{pmatrix} a_1 & & & b_1 \\ & \ddots & & \vdots \\ & & & b_{n-1} \\ c_1 & \cdots & c_{n-1} & a_n \end{pmatrix}.$$

Show that if $b_i c_i > 0$, then there exists a diagonal matrix D so that DAD^{-1} is symmetric.

7.2.9. Let $p_k(\lambda)$ be the characteristic polynomial of the leading principal $k \times k$ submatrix of A. Suppose that A is symmetric with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$. Show that $p_k(\lambda) > 0$, $k = 1, \ldots, n$ if $\lambda < \lambda_1$, and that the $p_k(\lambda)$ alternate in sign if $\lambda > \lambda_n$.

7.2.10. Let A and B be symmetric $n \times n$ matrices with B positive definite. Show how to find a matrix S so that $A = STS^T$ and $B = SS^T$, where T is a tridiagonal matrix whose eigenvalues are the same as those of $B^{-1}A$.

7.2.11. Let A be a real $n \times n$ matrix. Suppose that we wish to solve the linear systems $(A + \mu I)\mathbf{x} = \mathbf{b}$ for several values of the real parameter μ. How may we use the decomposition $A = PHP^T$, where P is orthogonal and H is upper Hessenberg? How many operations does your algorithm take?

7.3 Other Iterative Methods

The Householder and QR methods of the previous section are of primary value when the matrix A is not particularly sparse, and all or a large number of the eigenvalues are desired. Conversely, they are not very useful for very large sparse matrices for which only a few eigenvalues are desired. Problems such as this arise in partial differential equations, discussed in Chapters 8 and 9, as well as in other areas. A typical problem of this type might involve a $5,000 \times 5,000$ matrix with only ten or fewer nonzero elements in each row, and for which only a few eigenvalues, perhaps four or five, are desired. For such a problem the QR method is unsatisfactory because the QR factorization may change zero elements of A into non-zero elements as the factorization proceeds. (See Section 9.2 for further discussion of "fill-in.") The purpose of the present section is to describe some alternative methods.

The Power Method

A classical method that has a certain usefulness – but also serious defects – for large sparse problems is the *power method.* Let A have eigenvalues $\lambda_1, \ldots, \lambda_n$, which we assume for the moment are real and satisfy

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|. \tag{7.3.1}$$

For a given vector $\mathbf{x}^0$, consider the sequence of vectors generated by

$$\mathbf{x}^{k+1} = A\mathbf{x}^k, \qquad k = 0, 1, \ldots . \tag{7.3.2}$$

To analyze this sequence, assume that A has n linearly independent eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$, and expand $\mathbf{x}^0$ in terms of these eigenvectors:

$$\mathbf{x}^0 = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n. \tag{7.3.3}$$

Then, since $\mathbf{x}^k = A^k\mathbf{x}^0$ and $A^k\mathbf{v}_i = \lambda_i^k\mathbf{v}_i$,

$$\begin{aligned} \mathbf{x}^k &= c_1\lambda_1^k\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 + \cdots + c_n\lambda_n^k\mathbf{v}_n \qquad (7.3.4) \\ &= \lambda_1^k\left[c_1\mathbf{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{v}_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\mathbf{v}_n\right]. \end{aligned}$$

Because of (7.3.1) the terms $(\lambda_i/\lambda_1)^k$, $i = 2, \ldots, n$ all tend to zero as k goes to infinity. Therefore, if $c_1 \neq 0$,

$$\lambda_1^{-k}\mathbf{x}^k \to c_1\mathbf{v}_1, \qquad \text{as } k \to \infty, \tag{7.3.5}$$

which shows that the vectors $\mathbf{x}^k$ tend to the direction of the eigenvector $\mathbf{v}_1$. The magnitude of the vectors $\mathbf{x}^k$, however, will tend to zero if $|\lambda_1| < 1$, or become unbounded if $|\lambda_1| > 1$. Therefore scaling of the vectors $\mathbf{x}^k$ is required, and the scaling process will also give approximations to the eigenvalue λ_1.

One way to choose the scaling factors is based on the observation that as $\mathbf{x}^k$ approaches the direction $\mathbf{v}_1$, then $A\mathbf{x}^k \doteq \lambda_1\mathbf{x}^k$. Hence ratios of the components of $\mathbf{x}^k$ and $A\mathbf{x}^k$ are approximations to λ_1. To avoid choosing components that are too small, let x_i^k be a component of maximum absolute value of $\mathbf{x}^k$ and define

$$\hat{\mathbf{x}}^{k+1} = A\mathbf{x}^k, \qquad \gamma_k = \frac{\hat{x}_i^{k+1}}{x_i^k}, \qquad \mathbf{x}^{k+1} = \frac{\hat{\mathbf{x}}^{k+1}}{\gamma_k}. \tag{7.3.6}$$

Then γ_k is an approximation to λ_1 and scaling $\hat{\mathbf{x}}^{k+1}$ by γ_k prevents the $\mathbf{x}^k$ from going to zero or infinity. In fact, it can be shown that

$$\gamma_k \to \lambda_1 \text{ and } \mathbf{x}^k \to c\mathbf{v}_1 \text{ as } k \to \infty, \tag{7.3.7}$$

where the last relation says that $\mathbf{x}^k$ tends to some multiple of the eigenvector $\mathbf{v}_1$.

There is a relationship between the power method and the QR method of the previous section (see Exercise 7.3.1). However, an advantage of the power method is that the vectors $\mathbf{x}^k$ can be generated by only matrix-vector multiplications (plus the work needed to compute the scaling factors); operations on the matrix A itself are unnecessary. The main disadvantage is the possibly slow rate of convergence, which, as shown by (7.3.4), is determined primarily by the ratio λ_2/λ_1. If this ratio is close to 1, as is typical for many problems, the convergence will be slow. One way to attempt to mitigate this problem is to use shifts as was done with the QR algorithm. If the power method is applied to the matrix $A-\sigma I$, whose eigenvalues are $\lambda_1-\sigma, \ldots, \lambda_n-\sigma$ (Exercise 7.1.6), then the rate of convergence will be determined by the ratio $|\lambda_2-\sigma|/|\lambda_1-\sigma|$, provided that $\lambda_1-\sigma$ remains the dominant eigenvalue. But even with this shift the convergence may still be painfully slow. For example, suppose that a $1,000 \times 1,000$ matrix has the eigenvalues 1,000, 999, $\ldots$, 1. Then, after a shift by $\sigma = 500$ the ratio is 0.998, which is barely better than the unshifted ratio of 0.999.

The power method also has other disadvantages. If there is more than one dominant eigenvalue, for example, $|\lambda_1| = |\lambda_2| > |\lambda_3|$, which would be the case for a real matrix with a dominant complex conjugate pair of roots, the sequence (7.3.6) may not converge. There are ways to circumvent this difficulty, but in the case of complex roots acceleration of the convergence by shifts is even less satisfactory. Another problem concerns computing the subdominant eigenvalues. Once we have approximated λ_1, we need to remove it in some fashion from the matrix or subsequent iterations will again converge to λ_1 rather than λ_2. We next show how to accomplish this by a process known as *deflation*. Deflation will also be useful in other methods to be discussed shortly.

Deflation

Assume that A is symmetric so that its eigenvalues λ_i are real and, by Theorem 7.1.3, the associated eigenvectors $\mathbf{v}_i$ can be assumed to be orthonormal. Again, let $\mathbf{x}^0$ be given by (7.3.3). Then by the orthonormality of the $\mathbf{v}_i$, we have $\mathbf{v}_1^T\mathbf{x}^0 = c_1$. Thus the vector

$$\hat{\mathbf{x}}^0 = \mathbf{x}^0 - (\mathbf{v}_1^T\mathbf{x}^0)\mathbf{v}_1 = c_2\mathbf{v}_2 + \ldots + c_n\mathbf{v}_n \qquad (7.3.8)$$

is a linear combination of only $\mathbf{v}_2, \ldots, \mathbf{v}_n$, and the same will be true for the sequence (7.3.2) starting from $\hat{\mathbf{x}}^0$. If $|\lambda_2| > |\lambda_i|$, $i \geq 3$, then the power method will produce iterates converging to λ_2 and a multiple of $\mathbf{v}_2$. This idea extends to any number of eigenvectors (Exercise 7.3.2).

The above deflation procedure allows us, in principle, to remove the effect of λ_1 and $\mathbf{v}_1$ from the subsequent calculation of the remaining eigenvalues and eigenvectors. In practice, however, we will not know $\mathbf{v}_1$ exactly so that the vector $\hat{\mathbf{x}}^0$, formed with an approximation to $\mathbf{v}_1$, will still have a component in the direction $\mathbf{v}_1$, and the power method will still give convergence to λ_1 rather than λ_2. Even if $\mathbf{v}_1$ were known exactly, rounding error in the formation of $\hat{\mathbf{x}}^0$ and the power method computations would have the same effect. Therefore it is necessary to apply (7.3.8) periodically to the current iterates in order to keep the effect of $\mathbf{v}_1$ suppressed. That is, if $\mathbf{x}^k$ is the current power method iterate and $\hat{\mathbf{v}}_1$ our approximation to $\mathbf{v}_1$, we would form

$$\hat{\mathbf{x}}^k = \mathbf{x}^k - (\hat{\mathbf{v}}_1^T\mathbf{x}_k)\hat{\mathbf{v}}_1,$$

and then continue the iteration with $\hat{\mathbf{x}}^k$. This would be done only occasionally. Another way to carry out a deflation process is given in Exercise 7.3.3.

Inverse Iteration and Computation of Eigenvectors

We next consider a variation of the power method, called *inverse iteration* or the *inverse power method*, whose rate of convergence is potentially much faster than that of the power method. Consider the sequence $\{\mathbf{x}^k\}$ defined by

$$(A - \sigma I)\mathbf{x}^k = \mathbf{x}^{k-1}, \qquad k = 1, 2, \ldots, \qquad (7.3.9)$$

for some parameter σ; that is, $\mathbf{x}^k$ is the solution of the linear system (7.3.9). This is the power method for the matrix $(A - \sigma I)^{-1}$. If A again has eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$, then $(A - \sigma I)^{-1}$ has eigenvalues $(\lambda_i - \sigma)^{-1}$ and eigenvectors $\mathbf{v}_i$, and the sequence $\{\mathbf{x}^k\}$ of (7.3.9) obeys the relationship (7.3.4) with the λ_i replaced by $(\lambda_i - \sigma)^{-1}$:

$$\mathbf{x}^k = \frac{c_1}{(\lambda_1 - \sigma)^k}\mathbf{v}_1 + \ldots + \frac{c_n}{(\lambda_n - \sigma)^k}\mathbf{v}_n. \qquad (7.3.10)$$

We will return to (7.3.9) shortly as the basis for a method of computing both the eigenvalues and eigenvectors of A, but we first note that inverse iteration is the standard way to compute the eigenvectors of a matrix once the eigenvalues have already been computed by, for example, the QR method. Suppose that σ is an approximation to λ_j and rewrite (7.3.10) for $k = 1$ as

$$\mathbf{x}^1 = \frac{c_j}{\lambda_j - \sigma}\mathbf{v}_j + \sum_{i \neq j} \frac{c_i}{\lambda_i - \sigma}\mathbf{v}_i. \tag{7.3.11}$$

Now suppose that $|\lambda_j - \sigma|$ is small (say, $0(10^{-6})$), λ_j is not particularly close to another eigenvalue λ_i, and c_j is not small. Then the dominant term in (7.3.11) will be $c_j(\lambda_j - \sigma)^{-1}\mathbf{v}_j$. Only the direction of $\mathbf{v}_j$ needs to be computed since we can scale this to any desired length. Thus the effect of solving the system $(A - \sigma I)\mathbf{x}^1 = \mathbf{x}^0$ is to approximate the direction of the desired eigenvector. Note that the better σ approximates λ_j, the closer to singular is the matrix $A - \sigma I$. This ill-conditioning of $A - \sigma I$ is not deleterious in this case, however, since any error in solving the system will be primarily in the direction $\mathbf{v}_j$ that we are approximating.

Two factors will affect the accuracy of this approximation to the eigenvector. First, if λ_j is very close to another eigenvalue, say λ_{j+1}, then $\lambda_{j+1} - \sigma$ will also be small, and the first term of (7.3.11) will no longer be dominant; we will then approximate some linear combination of $\mathbf{v}_j$ and $\mathbf{v}_{j+1}$. Closeness of the eigenvalues is an intrinsic property of the matrix and hampers any numerical method in the calculation of the eigenvectors. The second factor is the possibility that c_j is very small, and if this is the case, then again the first term of (7.3.11) may not be sufficiently dominant to give a good approximation to the desired eigenvector. In principle, we can insure that this will not happen by choosing the vector $\mathbf{x}^0$ so that c_j is not small. However, we can do that with certainty only if the eigenvectors are known, and of course that is not the case. It has been found that choosing $\mathbf{x}^0$ to be the vector with components all equal to 1 usually works very well. A similar strategy that sometimes works even better is to do Gaussian elimination on $A - \sigma I$ to produce the upper-triangular matrix U, and then solve the system $U\mathbf{y} = \mathbf{z}$ where $\mathbf{z}$ is a vector all of whose components are equal to 1. In this case the vector $\mathbf{x}^0$ is not specified explicitly: it is the vector that would give rise to a vector of all 1's under the Gaussian elimination calculation. Obviously, there is a great deal of flexibility in choosing the vector $\mathbf{x}^0$. Indeed, any "randomly" chosen vector would be very unlikely to yield a particularly small c_j. We note that one of the worst possible strategies would be to attempt to solve the homogeneous system $(A - \sigma I)\mathbf{x}^1 = 0$, which would be the mathematical definition of the eigenvector if $\sigma = \lambda_j$.

It is usually worthwhile to do another iteration using the approximate eigenvector $\mathbf{x}^1$ just computed. Even if the original choice of $\mathbf{x}^0$ is such that c_j is very small, $\mathbf{x}^1$ will have a $\hat{c}_j$ that is larger, and another iteration may then give a suitable approximation. This could be repeated as many times as desired, but one extra iteration is generally sufficient.

In the context of the QR method of the previous section, the above inverse iteration procedure would be applied to the Hessenberg matrix H (or the tridiagonal matrix T in case A is symmetric). Once the desired eigenvectors of H have been computed, they are transformed back to eigenvectors of the original matrix as discussed in Section 7.2. Note that the eigenvalues, and consequently the eigenvectors, may be complex. This does not affect the inverse iteration procedure except that complex arithmetic must be performed. However, a side benefit of complex eigenvalues of a real matrix is that the eigenvectors occur in complex conjugate pairs, as do the eigenvalues, so that if $\mathbf{u} + i\mathbf{v}$ is an eigenvector for $a + ib$, then $\mathbf{u} - i\mathbf{v}$ is an eigenvector for $a - ib$, and no further computation is needed for this second eigenvector (Exercise 7.3.9).

Computation of Eigenvalues

As we have seen, each step of the inverse iteration (7.3.4) can greatly improve an approximation to an eigenvector if σ is a good approximation to a corresponding eigenvalue. However, there remains the problem of approximating the eigenvalue itself for matrices that are not suitable for the QR method. Since (7.3.9) is the power method for $(A - \sigma I)^{-1}$, we can proceed as in (7.3.6):

$$(A - \sigma I)\hat{\mathbf{x}}^{k+1} = \mathbf{x}^k, \qquad k = 0, 1, \ldots \tag{7.3.12a}$$

$$\mathbf{x}^{k+1} = \frac{\hat{\mathbf{x}}^{k+1}}{\gamma_k}, \qquad k = 0, 1, \ldots \tag{7.3.12b}$$

where γ_k is defined as in (7.3.6). Then

$$\gamma_k \to \gamma = (\sigma - \lambda_j)^{-1}, \qquad \mathbf{x}^k \to c_j \mathbf{v}_j, \qquad \text{as } k \to \infty, \tag{7.3.13}$$

provided that

$$|(\sigma - \lambda_j)^{-1}| > |(\sigma - \lambda_i)^{-1}|, \qquad i \neq j. \tag{7.3.14}$$

The eigenvalue λ_j is then given by

$$\lambda_j = \sigma - \frac{1}{\gamma}. \tag{7.3.15}$$

If $|(\sigma - \lambda_i)^{-1}| = \max\{|(\sigma - \lambda_m)^{-1}|;\ m \neq j\}$, then the rate of convergence is governed by the ratio

$$\frac{|(\sigma - \lambda_i)^{-1}|}{|(\sigma - \lambda_j)^{-1}|} = \frac{|\sigma - \lambda_j|}{|\sigma - \lambda_i|}. \tag{7.3.16}$$

The closer σ is to λ_j, the smaller this ratio. Therefore it is reasonable to replace a fixed σ by estimates of λ_j depending on γ_k. From (7.3.13),

$$\sigma_k = \sigma - \frac{1}{\gamma_k} \to \lambda_j, \qquad \text{as } k \to \infty, \tag{7.3.17}$$

and thus we modify (7.3.12a) to

$$(A - \sigma_k I)\hat{\mathbf{x}}^{k+1} = \mathbf{x}^k, \qquad k = 0, 1, \ldots, \tag{7.3.18}$$

where σ_k is given by (7.3.17). The value of σ in (7.3.17) would be our best estimate of the eigenvalue we wish to approximate. For example, if we want the smallest eigenvalue in absolute value we might choose $\sigma = 0$, whereas if we want the largest we could choose $\sigma = ||A||$ for some norm. Once an eigenvalue has been approximated, we could use the same deflation procedure discussed for the power method to minimize the effect of that eigenvalue on further computations. However, the use of the shifts σ_k allow us to circumvent the need for deflation if we have good estimates for the eigenvalues to be computed. Of course, if we are solving (7.3.18) by LU factorization, each time we change σ_k we must refactor $A - \sigma_k I$.

The Rayleigh Quotient Method

We next describe another way to choose the shift parameters σ_k in the case that A is a symmetric matrix. For a given vector $\mathbf{v} \neq 0$, the *Rayleigh quotient* is the quantity

$$\sigma(\mathbf{v}) = \frac{\mathbf{v}^T A \mathbf{v}}{\mathbf{v}^T \mathbf{v}}. \tag{7.3.19}$$

The Rayleigh quotient has two basic properties (Exercise 7.4.5): If $\lambda_1 \leq \ldots \leq \lambda_n$ and $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are the eigenvalues and corresponding orthonormal eigenvectors of A, then for any $\mathbf{v}$,

$$\lambda_1 \leq \sigma(\mathbf{v}) \leq \lambda_n \tag{7.3.20}$$

and if $\mathbf{v} = \mathbf{v}_i$, then

$$\sigma(\mathbf{v}) = \lambda_i. \tag{7.3.21}$$

Another basic property of the Rayleigh quotient is that if $\mathbf{v}$ is a good approximation to an eigenvector, then the Rayleigh quotient is a much better approximation to the corresponding eigenvalue. We make this precise in the following theorem.

THEOREM 7.3.1 *Let* $\mathbf{v} = \gamma \mathbf{v}_j + \mathbf{w}$, *where*

$$\mathbf{w} = \sum_{i \neq j} c_i \mathbf{v}_i.$$

If $\gamma = 1 + 0(\varepsilon)$ *and* $||\mathbf{w}||_2 = 0(\varepsilon)$, *then*

$$|\sigma(\mathbf{v}) - \lambda_j| = 0(\varepsilon^2). \tag{7.3.22}$$

PROOF: By assumption, $\mathbf{v}_j$ is orthogonal to $\mathbf{w}$ so that

$$\mathbf{v}^T\mathbf{v} = (\gamma\mathbf{v}_j + \mathbf{w})^T(\gamma\mathbf{v}_j + \mathbf{w}) = \gamma^2 + \mathbf{w}^T\mathbf{w}.$$

Since

$$A(\gamma\mathbf{v}_j + \mathbf{w}) = \gamma\lambda_j\mathbf{v}_j + \sum_{i\neq j}\lambda_i c_i\mathbf{v}_i = \gamma\lambda_j\mathbf{v}_j + \hat{\mathbf{w}},$$

$\hat{\mathbf{w}}$ is also orthogonal to $\mathbf{v}_j$. Thus

$$\mathbf{v}^T A\mathbf{v} = (\gamma\mathbf{v}_j + \mathbf{w})^T(\gamma\lambda_j\mathbf{v}_j + \hat{\mathbf{w}}) = \gamma^2\lambda_j + \mathbf{w}^T\hat{\mathbf{w}},$$

so that

$$\sigma(\mathbf{v}) = \frac{\gamma^2\lambda_j + \mathbf{w}^T\hat{\mathbf{w}}}{\gamma^2 + \mathbf{w}^T\mathbf{w}} = \frac{\lambda_j + \gamma^{-2}\mathbf{w}^T\hat{\mathbf{w}}}{1 + \gamma^{-2}\mathbf{w}^T\mathbf{w}}. \tag{7.3.23}$$

If $\lambda = \max\{|\lambda_i| : i \neq j\}$, then

$$\mathbf{w}^T\hat{\mathbf{w}} = \sum_{i\neq j}\lambda_i c_i^2 \leq \lambda^2\sum_{i\neq j}c_i^2 = \lambda^2 0(\varepsilon^2) = 0(\varepsilon^2),$$

since λ is fixed. By the estimates of Exercise 7.3.6, we then conclude from (7.3.23) that

$$\sigma(\mathbf{v}) = \frac{\lambda_j + [1 + 0(\varepsilon)]^2 0(\varepsilon^2)}{1 + [1 + 0(\varepsilon)]^2 0(\varepsilon^2)} = \lambda_j + 0(\varepsilon^2), \tag{7.3.24}$$

which was to be proved.

As a consequence of Theorem 7.3.1, if an approximate eigenvector $\mathbf{v}$ is known to m digits of accuracy, then $\sigma(\mathbf{v})$ is an eigenvalue to approximately $2m$ digits of accuracy. This forms the basis of a combined Rayleigh quotient/inverse iteration method of the form:

$$\sigma_k = \sigma(\mathbf{x}^k), \qquad (A - \sigma_k I)\mathbf{x}^{k+1} = \mathbf{x}^k, \qquad k = 0, 1, \ldots\ . \tag{7.3.25}$$

Thus starting from an initial $\mathbf{x}^0$, at each stage we compute a new Rayleigh quotient approximation σ_k to the eigenvalue, and then a new approximation to the eigenvector by an inverse iteration step. This procedure can be very rapidly convergent when it is successful; indeed, it can be shown that there is a cubic rate of convergence of the sequence $\{\sigma_k\}$ to a simple eigenvalue. However, there are several drawbacks. The first is that of obtaining a satisfactory starting vector $\mathbf{x}^0$, since if $\mathbf{x}^0$ is not a reasonable approximation to the direction $\mathbf{v}_j$ of interest, the sequences (7.3.25) will not necessarily converge to the eigenvalue and eigenvector pair $\lambda_j, \mathbf{v}_j$. One way to obtain a suitable $\mathbf{x}^0$ is to use the power method for several steps to obtain an approximation to the eigenvector corresponding to the largest eigenvalue in absolute value. One then could switch to (7.3.25). For the other eigenvalue/eigenvector pairs, one would need to use deflation to obtain approximate eigenvectors by the power method to use in (7.3.25).

Another drawback of (7.3.25), or of inverse iteration in general, is the necessity of solving the linear system of equations at each stage. For large sparse matrices such as arise in the solution of partial differential equations, this is a major problem and is the subject of Chapter 9.

Lanczos' Method

We end this chapter by considering one more method for large sparse symmetric matrices. As with the other methods of this section, Lanczos' method is usually used to approximate only a few eigenvalues. In contrast to the other methods, it approximates both the largest and smallest eigenvalues simultaneously, although the rate of convergence to the smallest eigenvalues may be slower.

Recall from the previous section that a symmetric matrix can be reduced to a tridiagonal matrix T by an orthogonal similarity transformation:

$$A = QTQ^T. \tag{7.3.26}$$

The orthogonal matrix Q was constructed by means of Householder transformations, but we will now obtain T by an entirely different approach which does not destroy the sparsity of A. Suppose that (7.3.26) holds and

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}. \tag{7.3.27}$$

If we multiply (7.3.26) on the right by Q,

$$AQ = QT, \tag{7.3.28}$$

and equate the columns of the two sides of (7.3.28), we have

$$A\mathbf{q}_1 = \alpha_1\mathbf{q}_1 + \beta_1\mathbf{q}_2, \tag{7.3.29a}$$

$$A\mathbf{q}_i = \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i\mathbf{q}_i + \beta_i q_{i+1}, \qquad i = 2, \ldots, n-1, \tag{7.3.29b}$$

$$A\mathbf{q}_n = \beta_{n-1}\mathbf{q}_{n-1} + \alpha_n\mathbf{q}_n, \tag{7.3.29c}$$

where $\mathbf{q}_i$ is the ith column of Q. Since the orthogonality of Q implies that $\mathbf{q}_i^T\mathbf{q}_j = 0$, $i \neq j$ and $\mathbf{q}_i^T\mathbf{q}_i = 1$, if we multiply (7.3.29) by $\mathbf{q}_i^T$ we see that

$$\alpha_i = \mathbf{q}_i^T A\mathbf{q}_i, \qquad i = 1, \ldots, n. \tag{7.3.30}$$

To characterize the β_i we write (7.3.29a) as

$$\beta_1 \mathbf{q}_2 = A\mathbf{q}_1 - \alpha_1 \mathbf{q}_1,$$

and take norms of both sides to obtain

$$\beta_1 = \pm ||A\mathbf{q}_1 - \alpha_1 \mathbf{q}_1||_2. \qquad (7.3.31)$$

The sign of β_1 is immaterial and we shall elect to take the β_i as positive. Then in a similar way we obtain from (7.3.29b)

$$\beta_i = ||A\mathbf{q}_i - \alpha_i \mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}||_2, \qquad i = 2, \ldots, n-1. \qquad (7.3.32)$$

We can now use the above characterizations as the basis for an algorithm to obtain Q and T. Let $\mathbf{q}_1$ be an arbitrary vector such that $\mathbf{q}_1^T\mathbf{q}_1 = 1$. Form $A\mathbf{q}_1$ and then α_1 and β_1 from (7.3.30) and (7.3.31). Next, from (7.3.29a),

$$\mathbf{q}_2 = \frac{1}{\beta_1}(A\mathbf{q}_1 - \alpha_1 \mathbf{q}_1).$$

Now we can obtain α_2 and β_2 from (7.3.30) and (7.3.32) and then $\mathbf{q}_3$ from (7.3.29b). Continuing in this way, we may compute all of the α_i, β_i, and $\mathbf{q}_i$. The algorithm is summarized in Figure 7.2. One of the main strengths of the Lanczos algorithm is that the matrix A is never modified, as it is in the Householder reduction to tridiagonal form. As in the power method, A need not even be known explicitly as long as the matrix-vector product $A\mathbf{q}$ can be formed.

```
Choose $\mathbf{q}_1$ with $\mathbf{q}_1^T\mathbf{q}_1 = 1$. Set $\beta_0 = 0$
For $i = 1$ to $n-1$
    $\mathbf{p}_i = A\mathbf{q}_i$
    $\alpha_i = \mathbf{q}_i^T\mathbf{p}_i$
    $\mathbf{w}_i = \mathbf{p}_i - \alpha_i\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}$
    $\beta_i = ||\mathbf{w}_i||_2$
    $\mathbf{q}_{i+1} = \beta_i^{-1}\mathbf{w}_i$
$\alpha_n = \mathbf{q}_n^T A\mathbf{q}_n$
```

Figure 7.2: *Lanczos Algorithm*

We have to verify that the $\mathbf{q}_i$ generated by the Lanczos algorithm of Figure 7.2 are indeed orthonormal. Clearly, the choice of β_i guarantees that $||\mathbf{q}_{i+1}||_2 = 1$, provided that $\beta_i \neq 0$; we shall return to this point shortly. For now we assume that all $\beta_i \neq 0$ and show that the $\mathbf{q}_i$ are orthogonal. First,

$$\mathbf{q}_1^T\mathbf{q}_2 = \beta_1^{-1}\mathbf{q}_1^T(A\mathbf{q}_1 - \alpha_1\mathbf{q}_1) = \beta_1^{-1}(\alpha_1 - \alpha_1) = 0, \qquad (7.3.33)$$

and, by induction,

$$\mathbf{q}_i^T\mathbf{q}_{i+1} = \beta_i^{-1}\mathbf{q}_i^T(A\mathbf{q}_i - \alpha_i\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}) = \beta_i^{-1}(\alpha_i - \alpha_i) = 0, \qquad (7.3.34)$$

for $i = 2, \ldots, n-1$. We now show by induction that all of the $\mathbf{q}_i$ are orthogonal. We make the induction hypothesis that

$$\mathbf{q}_j^T\mathbf{q}_{i+1} = 0, \qquad j = 1, \ldots, i, \qquad (7.3.35)$$

which we have shown to be true for $i = 1$. We then wish to prove that

$$\mathbf{q}_j^T\mathbf{q}_{i+2} = 0, \qquad j = 1, \ldots, i+1. \qquad (7.3.36)$$

By (7.3.34) we have shown this for $j = i + 1$, so we assume that $j \leq i$. Then, by (7.3.35),

$$\begin{aligned}\mathbf{q}_j^T\mathbf{q}_{i+2} &= \beta_{i+1}^{-1}\mathbf{q}_j^T(A\mathbf{q}_{i+1} - \alpha_{i+1}\mathbf{q}_{i+1} - \beta_i\mathbf{q}_i) \qquad (7.3.37)\\ &= \beta_{i+1}^{-1}(\mathbf{q}_j^TA\mathbf{q}_{i+1} - \beta_i\mathbf{q}_j^T\mathbf{q}_i).\end{aligned}$$

Now by (7.3.29b) and the symmetry of A,

$$\mathbf{q}_j^TA\mathbf{q}_{i+1} = \mathbf{q}_{i+1}^TA\mathbf{q}_j = \mathbf{q}_{i+1}^T(\beta_{j-1}\mathbf{q}_{j-1} + \alpha_j\mathbf{q}_j + \beta_j\mathbf{q}_{j+1}). \qquad (7.3.38)$$

If $j < i$, then all of the inner products in (7.3.38) are zero by (7.3.35), as is $\mathbf{q}_j^T\mathbf{q}_i$ in (7.3.37); hence $\mathbf{q}_j^T\mathbf{q}_{i+2} = 0$. If $j = i$, then (7.3.38) shows that $\mathbf{q}_i^TA\mathbf{q}_{i+1} = \beta_i$, which cancels the β_i in (7.3.37) so that, again, $\mathbf{q}_j^T\mathbf{q}_{i+2} = 0$ and the induction is complete.

We now return to the assumption that $\beta_i \neq 0$. Suppose that $\beta_1 = ||\mathbf{w}_1||_2 = 0$. Then $A\mathbf{q}_1 = \alpha_1\mathbf{q}_1$, so that $\mathbf{q}_1$ and α_1 are an eigenvector and a corresponding eigenvalue. More generally, if $\beta_i = ||\mathbf{w}_i||_2 = 0$, then the Lanczos process stops and we have at this point

$$AQ_i = Q_iT_i, \qquad (7.3.39)$$

where $Q_i = (\mathbf{q}_1, \ldots, \mathbf{q}_i)$ and T_i is the $i \times i$ leading principal submatrix of T. If $\hat{\lambda}_1, \ldots, \hat{\lambda}_i$ are the eigenvalues of T_i, then we can write

$$T_i = \hat{Q}_iD_i\hat{Q}_i^T, \qquad (7.3.40)$$

where $D = \mathrm{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_i)$ and the columns of $\hat{Q}_i$ are the corresponding orthonormal eigenvectors of T_i. Putting (7.3.40) into (7.3.39) we obtain

$$AQ_i\hat{Q}_i = Q_i\hat{Q}_iD_i. \qquad (7.3.41)$$

The columns of $Q_i\hat{Q}_i$ are orthonormal (Exercise 7.3.9), and thus (7.3.41) shows that the $\hat{\lambda}_i$ are eigenvalues of A with corresponding eigenvectors which are the columns of $Q_i\hat{Q}_i$. Thus the emergence of a $\beta_i = 0$ signals that we can find i eigenvalues of A by computing the eigenvalues of T_i. The process can then be restarted by choosing a $\mathbf{q}_{i+1}$ that is orthogonal to $\mathbf{q}_1, \ldots, \mathbf{q}_i$. It is easy to show (Exercise 7.3.10) that $\beta_i = 0$ if and only if $\mathbf{q}_1$ is a linear combination of i eigenvectors of A; this is highly unlikely to happen in practice.

Assuming that no $\beta_i = 0$ we could carry the Lanczos process to completion and obtain the tridiagonal matrix T of (7.3.27). In practice this is rarely done for the large sparse matrices for which the Lanczos method is most useful. It turns out that if we stop the process at the kth step for some moderate size of k, the largest and smallest eigenvalues of the corresponding tridiagonal matrix T_k may be surprisingly good approximations to the corresponding eigenvalues of A . Moreover, other eigenvalues of T_k will approximate the corresponding small or large eigenvalues of A, although generally not as well as the extremal eigenvalue approximations.

Summary

In this section we have discussed several approaches to obtaining at least a few eigenvalues and corresponding eigenvectors of large sparse matrices. None of these methods is completely reliable and their usefulness depends in large measure on the location of the eigenvalues of A. In general, the symmetric eigenvalue problem is much easier than the nonsymmetric, but even for symmetric matrices there are no foolproof methods yet known.

Supplementary Discussion and References: 7.3

As with Section 7.2, the books by Golub and Van Loan [1989], Stewart [1973], and Wilkinson [1965] provide excellent information on more advanced aspects of the methods considered. See also Parlett [1980] for symmetric matrices. The analysis of the algorithms of this section was restricted to matrices with n linearly independent eigenvectors, but it can be extended to general matrices; see Golub and Van Loan [1989] and Wilkinson [1965], in particular.

The Rayleigh quotient and Lanczos methods were presented in the text only for symmetric matrices, but extensions to nonsymmetric matrices are possible; see Golub and Van Loan [1989].

Many of the methods of this section have extensions to "block" methods in which several eigenvalues and eigenvectors are approximated simultaneously. For example, "subspace" iteration is an extension of the power or inverse power iterations. For further discussion of block methods, see Golub and Van Loan [1989] and Parlett [1980].

As we have shown, the vectors $\mathbf{q}_i$ generated by the Lanczos algorithm are orthogonal. One serious problem with the algorithm is that due to rounding error this orthogonality is lost and with relatively few steps can be lost so significantly that the algorithm is no longer satisfactory. For ways to circumvent this problem by "reorthogonalization;" see Golub and Van Loan [1989].

EXERCISES 7.3

7.3.1. (Stewart [1973]). Let A_k be the sequence of matrices generated by the QR algorithm and let $\hat{Q}_k = Q_0 \cdots Q_{k-1}$, $\hat{R}_k = R_{k-1} \dots R_0$. Show by induction that $A_k = \hat{Q}_k \hat{R}_k$. Conclude from this that the first column of $\hat{Q}_k$ is a multiple of $A^k \mathbf{e}_1$.

7.3.2. Let A be symmetric with eigenvalues λ_i and corresponding orthonormal eigenvectors $\mathbf{v}_i$. If $\mathbf{x} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n$, show that

$$\mathbf{x} - (\mathbf{v}_1^T\mathbf{x})\mathbf{v}_1 - \cdots - (\mathbf{v}_m^T\mathbf{x})\mathbf{v}_m = c_{m+1}\mathbf{v}_{m+1} + \cdots + c_n\mathbf{v}_n.$$

7.3.3. Let A be symmetric with eigenvalues λ_i and corresponding orthonormal eigenvectors $\mathbf{v}_i$. Show that the matrix $A_2 = A - \lambda_1\mathbf{v}_1\mathbf{v}_1^T$ has eigenvalues $0, \lambda_2, \dots, \lambda_n$ and corresponding eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$. Discuss how to compute $A_2\mathbf{x}$ without forming A_2 explicitly.

7.3.4. Let A be a real matrix with complex eigenvalue $a + ib$ and corresponding eigenvector $\mathbf{u} + i\mathbf{v}$. Show that $\mathbf{u} - i\mathbf{v}$ is the eigenvector corresponding to $a - ib$.

7.3.5. Let A be a symmetric matrix with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$. Show that if $\mathbf{v} = \mathbf{v}_i$, the Rayleigh quotient of (7.3.19) is $\sigma = \lambda_i$. Use the fact that any vector $\mathbf{v}$ can be written as $\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n$ to show that (7.3.20) holds.

7.3.6. From the geometric series

$$\frac{1}{1-\alpha} = 1 + \alpha + \alpha^2 + \cdots$$

conclude that

$$\frac{1}{1+0(\varepsilon)} = 1 + 0(\varepsilon), \qquad \frac{1}{1+0(\varepsilon^2)} = 1 + 0(\varepsilon^2).$$

Show also that $[1 + 0(\varepsilon)]^2 = 1 + 0(\varepsilon)$.

7.3.7. The matrix

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

has eigenvalues 1 and 3 with corresponding eigenvectors $(1, -1)$ and $(1, 1)$. Apply several steps of the power method (7.3.6) to this matrix, starting with the vector $\mathbf{x}_0 = (1, 0)^T$. Carry the iteration far enough for the rate of convergence to become apparent.

7.3.8. Apply the power method to the shifted matrix $A - \frac{1}{2}I$, where A is given in Exercise 7.3.7. Discuss the improvement in the rate of convergence.

7.3.9. Let Q_i be an $n \times i$ matrix whose columns are orthonormal and P_i an $i \times i$ orthogonal matrix. Show that the columns of Q_iP_i are orthonormal.

7.3.10. Use (7.3.41) to show that if $\beta_i = 0$ in the Lanczos algorithm (and $\beta_i \neq 0$, $j < i$), then $\mathbf{q}_1$ is a linear combination of i eigenvectors of A.

7.3.11. Let $\mathbf{q}$ be an arbitrary vector with $\|\mathbf{q}\|_2 = 1$ and assume that $A = A^T$. Set $\sigma = \mathbf{q}^T A\mathbf{q}$ and $\mathbf{z} = A\mathbf{q} - \sigma\mathbf{q}$. Show that the interval $|\lambda - \sigma| \leq \|\mathbf{z}\|_2$ must contain at least one eigenvalue of A.

7.3.12. Develop the Lanczos algorithm for skew-symmetric matrices.

7.3.13. Let A be an arbitrary real matrix with eigenvalues λ_i and corresponding eigenvectors $\mathbf{v}_i$. Assume $\lambda_2 = \bar{\lambda}_1$, so that $\mathbf{v}_2 = \bar{\mathbf{v}}_1$ (Exercise 7.3.4). Given a vector $\mathbf{r}$ such that $\mathbf{r} = c\mathbf{v}_1 + \bar{c}\bar{\mathbf{v}}_1$, show how $\lambda_1, \bar{\lambda}_1, \mathbf{v}_1$ and $\bar{\mathbf{v}}_1$ can be calculated from $\mathbf{r}, \mathbf{s} = A\mathbf{r}$, and $\mathbf{t} = A\mathbf{s}$. Does this suggest an acceleration scheme for computing eigenvalues of non-symmetric matrices?

7.3.14. (Steepest Descent) Let A be a real symmetric matrix, $\mathbf{q}$ be an arbitrary vector with $\|\mathbf{q}\|_2 = 1$, and

$$\gamma^{-1}\mathbf{z} = A\mathbf{q} - \sigma\mathbf{q}, \quad \sigma = \mathbf{q}^T A\mathbf{q},$$

where γ is choosen so that $\|\mathbf{z}\|_2 = 1$. Show that $\mathbf{q}^T\mathbf{z} = 0$. In order to compute the smallest eigenvalue of A, let $\mathbf{w} = a\mathbf{q} + b\mathbf{z}$.

a. Show how to compute a and b so that $\mathbf{w}^T A\mathbf{w}/\mathbf{w}^T\mathbf{w}$ is a minimum.

b. Consider the sequence $\mathbf{w}_k = a_k\mathbf{q}_k + b_k\mathbf{z}_k$ and

$$\mu_k = \min\{\mathbf{w}_k^T A\mathbf{w}_k/\mathbf{w}_k^T\mathbf{w}_k : \mathbf{w}_k \neq 0\}.$$

Show that $\mu_{k+1} \leq \mu_k$.

7.3.15. Let $A = D+B$ and $A(\alpha) = D+\alpha B$, where D contains the diagonal elements of A and B the off-diagonal. Use the continuation method described in the Supplementary Discussion of Section 5.3 to give an algorithm for computing the smallest eigenvalue in magnitude of $A(1) = A$.

7.3.16. Let T be a symmetric tridiagonal matrix and $K = T + E$, where $E = (e_{ij})$ is a symmetric tridiagonal matrix with $|e_{ij}| \leq \varepsilon$. Use Exercise 7.3.11 to give bounds on how far each eigenvalue of K is from an eigenvalue of T.