8. Eigenvector and eigenvalue computations

8.1 Classification

Eigenproblems rooted in mathematical physics are mostly symmetric. In Chapter 6 we unfolded the theoretical richness of the symmetric eigenproblem. In this chapter we turn to the *computational* dilemmas of the problem with most attention still paid to symmetric $Ax = \lambda Bx$ with a positive definite B.

Computation of the symmetric eigenproblem can be divided into three prototypical groups each with its own distinct computational stipulation.

1. The first category consists of those problems $Ax = \lambda Bx$ in which A and B are small (say order < 100) and dense. All eigenvalues and eigenvectors are usually required in this case with good accuracy. Sparse algorithms are irrelevant here and the matrices are stored in tabular form entry by entry.

General eigenproblem $Ax = \lambda Bx$ is first reduced to $Sx' = \lambda x'$, $S = S^T$, with the factorization of $B = LL^T$ and the construction of $S = L^{-1}AL^{-T}$, $x' = L^T x$. If a good complete orthogonal system of eigenvectors X is available for B such that $X^T BX = D^2$ where D is diagonal, then $Ax = \lambda Bx$ may be reduced to $Sx' = \lambda x'$ with $S = D^{-1}X^T AXD^{-1}$ and $x' = DX^{-1}x$.

Computation of all eigenvalues of dense matrix A is invariably based on similarity transformations $P^{-1}AP$ that repeatedly work to transform A into a matrix ever closer to diagonal.

2. The second sort consists of $Ax = \lambda Bx$ with very large (say order > 100) and sparse A and B. Eigenproblems of this kind arise from finite difference and finite element ap-

proximations and are of band form or some other distinct sparseness pattern. Typical to these eigenproblems is the fact that only a small fraction of the eigenvalues and corresponding eigenvectors at the lower end of the spectrum is of interest. Storage of A and B in tabular form is out of the question for such large problems, and suitable solutions rely on matrix-vector product algorithms.

Some of the more sophisticated solution procedures of class 2 require the repeated solution of the general eigenproblem of class 1.

3. Because any symmetric matrix can be similarly reduced to tridiagonal form in a finite number of elementary steps, computation of the eigenvalues and eigenvectors of a tridiagonal matrix constitutes an important class all by itself.

8.2 The characteristic equation

Eigenvalue algorithms based on the computation of the coefficients of the characteristic equation $det(A - \lambda I) = 0$ and its numerical solution are unpopular. Apart from the question of computational efficiency, such methods are inherently unstable. Slight changes in the computed coefficients as resulting from numerical inaccuracies may cause considerably magnified errors in the roots.

In Sec. 6.3 we raised the issue of the change in a repeating root as a result of small changes in the coefficients of the polynomial characteristic equation, and have shown that it can be considerable. The nth order $-\lambda^n = 0$ has n zero roots, but $-\lambda^n + \epsilon = 0,\ \epsilon > 0$ has the n distinct roots $\lambda = \epsilon^{\frac{1}{n}}(\cos\theta + i\sin\theta),\ \cos n\theta = 1,\ \sin n\theta = 0$, of a modulus considerably larger than that of ϵ. Great sensitivity of the roots to variations in the coefficients is, however, a phenomenon not strictly associated with repeating roots.

Suppose that

$$\begin{aligned} p_n(\lambda) &= (\lambda_1 - \lambda)(\lambda_2 - \lambda)\ldots(\lambda_n - \lambda) \\ &= (-\lambda)^n + a_{n-1}(-\lambda)^{n-1} + a_{n-2}(-\lambda)^{n-2} + \ldots + a_0 \end{aligned} \tag{8.1}$$

has no repeating roots. To see what differential changes in the coefficients do to the jth root we differentiate the equation to have

$$d\lambda_j = -\frac{1}{p'(\lambda_j)}\left(\frac{da_0}{a_0}a_0 + \frac{da_1}{a_1}a_1(-\lambda_j) + \frac{da_2}{a_2}a_2(-\lambda_j)^2 + \ldots + \frac{da_{n-1}}{a_{n-1}}a_{n-1}(-\lambda_j)^{n-1}\right) \tag{8.2}$$

where the derivative

$$-p'(\lambda_j) = (\lambda_1 - \lambda_j)(\lambda_2 - \lambda_j)\dots(\lambda_k - \lambda_j)\dots(\lambda_n - \lambda_j),\ k \neq j. \tag{8.3}$$

Close roots are problematic in rendering $|p'(\lambda)|$ small, but the appearance of the very high powers $(-\lambda_j)^k,\ k = n-1, n-2, \dots$ in the expression for $d\lambda_j$ is also a source for concern. It appears in eq.(8.2) that the higher eigenvalues are very sensitive to changes in the coefficients of the high powers of λ in the characteristic equation.

Example. For the modest case of $\lambda_i = i, i = 1, 2, \dots, 7$,

$$\begin{aligned} p_7(\lambda) &= -\lambda^7 + 28\lambda^6 - 322\lambda^5 + 1960\lambda^4 - 6769\lambda^5 + 13132\lambda^2 - 13068\lambda + 5040 = 0 \\ p_7'(\lambda_1) &= p_7'(\lambda_7) = -720, \end{aligned} \tag{8.4}$$

and

$$\begin{aligned} \frac{d\lambda_1}{\lambda_1} &= 7\frac{da_0}{a_0} - 18.15\frac{da_1}{a_1} + 18.23\frac{da_2}{a_2} - 9.40\frac{da_3}{a_3} + 2.72\frac{da_4}{a_4} \\ &\quad -0.45\frac{da_5}{a_5} + 0.039\frac{da_6}{a_6} \\ \frac{d\lambda_7}{\lambda_7} &= \frac{da_0}{a_0} - 18.15\frac{da_1}{a_1} + 127.67\frac{da_2}{a_2} - 460.66\frac{da_3}{a_3} + 933.7\frac{da_4}{a_4} \\ &\quad -1073.78\frac{da_5}{a_5} + 653.51\frac{da_6}{a_6}. \end{aligned} \tag{8.5}$$

For this reason we shall not dwell on algorithms for the computations of the coefficients of the characteristic equation, except for the interesting method of Krylov, and this also we shall present only under the simpler circumstances of a matrix having n separate eigenvalues.

It is founded on

Lemma 8.1. *Let $A = A(n \times n)$ be with n distinct eigenvalues $|\lambda_1| < |\lambda_2| < \dots < |\lambda_n|$, and n corresponding linearly independent eigenvectors $v_1, v_2, \dots, v_n$. If x_0 is such that the expansion*

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \tag{8.6}$$

is with nonvanishing α's, then the n vectors

$$x_0,\ x_1 = Ax_0,\ x_2 = A^2x_0, \dots, x_{n-1} = A^{n-1}x_0 \tag{8.7}$$

are linearly independent.

Proof. We have from eq. (8.6) that

$$x_k = \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 + \cdots + \alpha_n \lambda_n^k v_n \tag{8.8}$$

and shall show that

$$r = \beta_0 x_0 + \beta_1 x_1 + \cdots + \beta_{n-1} x_{n-1} = o \tag{8.9}$$

only if $\beta_0 = \beta_1 = \cdots = \beta_{n-1} = 0$. Indeed,

$$\begin{aligned} r = \alpha_1(\beta_0 + \beta_1\lambda_1 + \beta_2\lambda_1^2 + \cdots + \beta_{n-1}\lambda_1^{n-1})v_1 \\ +\alpha_2(\beta_0 + \beta_1\lambda_2 + \beta_2\lambda_2^2 + \cdots + \beta_{n-1}\lambda_2^{n-1})v_2 \\ \vdots \\ +\alpha_n(\beta_0 + \beta_1\lambda_n + \beta_2\lambda_n^2 + \cdots + \beta_{n-1}\lambda_n^{n-1})v_n \end{aligned} \tag{8.10}$$

and since $v_1, v_2, \ldots, v_n$ are linearly independent and the α's all nonzero, $r = o$ occurs only if

$$\beta_0 + \beta_1\lambda_i + \beta_2\lambda_i^2 + \cdots + \beta_{n-1}\lambda_i^{n-1} = 0 \quad i = 1, 2, \ldots, n. \tag{8.11}$$

But this is impossible with nonvanishing β's since a polynomial equation of degree $n-1$ cannot have n distinct roots. Only when $\beta_0 = \beta_1 = \cdots \beta_{n-1} = 0$ is $r = o$. End of proof.

Theorem 8.2. *Under the assumptions of the previous lemma, vectors $x_0, x_1, \ldots, x_n$, $x_k = A^k x_0$ are linearly dependent so that*

$$\beta_0 x_0 + \beta_1 x_1 + \cdots + \beta_{n-1} x_{n-1} + (-1)^n x_n = o \tag{8.12}$$

with the β's being the coefficients of the characteristic equation

$$(-1)^n \lambda^n + \cdots + \beta_2 \lambda^2 + \beta_1 \lambda + \beta_0 = 0 \tag{8.13}$$

of matrix A.

Proof. Say for simplicity that $A = A(2 \times 2)$. Then x_0, x_1, x_2 are three vectors in R^2 that are certainly linearly dependent; $\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 = o$ for some nonzero $\beta_0, \beta_1, \beta_2$. In terms of the two eigenvectors v_1 and v_2 of A we have that

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2,\ x_1 = \alpha_1 \lambda_1 v_1 + \alpha_2 \lambda_2 v_2,\ x_2 = \alpha_1 \lambda_1^2 v_1 + \alpha_2 \lambda_2^2 v_2 \tag{8.14}$$

and

$$\alpha_1 v_1(\beta_0 + \beta_1 \lambda_1 + \beta_2 \lambda_1^2) + \alpha_2 v_2(\beta_0 + \beta_1 \lambda_2 + \beta_2 \lambda_2^2) = o. \tag{8.15}$$

Since v_1 and v_2 are linearly independent and $\alpha_1 \neq 0, \alpha_2 \neq 0$ it results that

$$\beta_0 + \beta_1 \lambda_1 + \beta_2 \lambda_1^2 = 0 \ \text{ and } \ \beta_0 + \beta_1 \lambda_2 + \beta_2 \lambda_2^2 = 0 \tag{8.16}$$

with $\beta_2 \neq 0$ by virtue of $\lambda_1 \neq \lambda_2$. The two roots fix the quadratic equation, in fact $\beta_0 = \lambda_1 \lambda_2 \beta_2,\ \beta_1 = -(\lambda_1 + \lambda_2)\beta_2$. End of proof.

Example.

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},\ x_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ x_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ x_2 = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \tag{8.17}$$

$\beta_0 = 2,\ \beta_1 = -2$, and $\lambda^2 - 2\lambda + 2 = 0$.

Matters, in this respect, are particularly attractive for the irreducible symmetric tridiagonal matrix, which we know has n distinct eigenvalues. Let

$$A = \begin{bmatrix} a_1 & b_2 & & \\ b_2 & a_2 & b_3 & \\ & b_3 & a_3 & b_4 \\ & & b_4 & a_4 \end{bmatrix} \quad b_i \neq 0 \ \ i = 2, 3, 4 \tag{8.18}$$

be such a matrix, and let $x_0 = e_1$. Then

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ x_1 = \begin{bmatrix} \times \\ b_2 \\ 0 \\ 0 \end{bmatrix},\ x_2 = \begin{bmatrix} \times \\ \times \\ b_2 b_3 \\ 0 \end{bmatrix},\ x_3 = \begin{bmatrix} \times \\ \times \\ \times \\ b_2 b_3 b_4 \end{bmatrix},\ x_4 = \begin{bmatrix} \times \\ \times \\ \times \\ \times \end{bmatrix} \tag{8.19}$$

and the coefficients $\beta_0, \beta_1, \beta_2, \beta_3$ of the characteristic equation of A, here a quartic, are obtained as the solution of the upper-triangular system

$$\begin{bmatrix} 1 & \times & \times & \times \\ & b_2 & \times & \times \\ & & b_2 b_3 & \times \\ & & & b_2 b_3 b_4 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = -x_4 \tag{8.20}$$

which is triangular of type 1 and possesses a unique solution.

Example. Matrix

$$A = \begin{bmatrix} 2 & -1 & \\ -1 & 2 & -1 \\ & -1 & 2 \end{bmatrix} \tag{8.21}$$

is known to have the characteristic equation

$$-\lambda^3 + 6\lambda^2 - 10\lambda + 4 = 0. \tag{8.22}$$

For this matrix

$$\begin{bmatrix} 1 & 2 & 5 \\ & -1 & -4 \\ & & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 14 \\ -14 \\ 6 \end{bmatrix} \tag{8.23}$$

and we find that $\beta_0 = 4,\ \beta_1 = -10,\ \beta_2 = 6$.

Something even simpler comes to mind. Say $A = A(3 \times 3)$ so that $\det(A - \lambda I) = -\lambda^3 + \alpha_2\lambda^2 + \alpha_1\lambda + \alpha_0$. Computing $\delta_i = \det(A - \lambda_i I)$ for the three *arbitrary* values $\lambda_1, \lambda_2, \lambda_3$ of variable λ we can determine the three coefficients $\alpha_0, \alpha_1, \alpha_2$ of the characteristic equation from the set of three linear equations

$$\begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 \\ 1 & \lambda_2 & \lambda_2^2 \\ 1 & \lambda_3 & \lambda_3^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \delta_1 + \lambda_1^3 \\ \delta_2 + \lambda_2^3 \\ \delta_3 + \lambda_3^3 \end{bmatrix} \tag{8.24}$$

with a Vandermonde matrix that is independent of A. It is simple but generally impractical. It is theoretically not easy, but numerical experience readily exposes the great sensitivity of system (8.24) to aberrations in its data. Minute round-off errors in the matrix entries and right-hand sides can be enormously magnified in the solution to produce defective α's; and if the coefficients are flawed, then the roots are useless.

exercises

8.2.1. The characteristic equation of matrix

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

is

$$-\lambda^3 + 6\lambda^2 - 10\lambda + 4 = 0$$

having the three distinct roots $\lambda_1 = 2 - \sqrt{2}$, $\lambda_2 = 2$, $\lambda_3 = 2 + \sqrt{2}$.

Write the perturbed

$$-\lambda^2 + 6(1-\epsilon)\lambda^2 - 10(1+\epsilon)\lambda + 4(1-\epsilon) = 0$$

and study the effect of a small ϵ on the roots. See what happens close to $\epsilon = 0.014$.

8.2.2. Given complex eigenvalue $\lambda = \alpha + i\beta$ of real matrix A, find the corresponding eigenvector $x = u + iv$ using real arithmetic.

8.2.3. Show that if for some $x_0 \neq o$

$$\beta_0 x_0 + \beta_1 A x_0 + \beta_2 A^2 x_0 - A^3 x_0 = o$$

then the roots of

$$\beta_0 + \beta_1 \lambda + \beta_2 \lambda^2 - \lambda^3 = 0$$

contain an eigenvalue of A. Also, that if

$$\beta_0 x_0 + \beta_1 A x_0 + A^2 x_0 = o,\ x_0 \neq o$$

and the roots of

$$\beta_0 + \beta_1 \lambda + \lambda^2 = 0$$

are the complex λ_1 and $\overline{\lambda}_1$, then they are both eigenvalues of A.

8.3 Jacobi methods

The method of Jacobi is one of the oldest (dating from 1846) methods to solve the symmetric algebraic eigenproblem but nonetheless remains one of the most elegant, compact, and practical routines for the computation of all eigenvalues and eigenvectors of small dense symmetric matrices. Its underlying idea is a succession of similarity transformations $A' = Q^{-1}AQ$ with Q designed to make A' closer than A to diagonal form.

To be effective, matrix Q in the similarity transformation should be such that Q^{-1} is easily computed, and the matrix multiplications inexpensive. Matrix Q must also include

convenient parameters to be adjusted to make A' as near to diagonal form as possible, and above all the matrix should not induce excessive errors upon the repeated multiplications.

Jacobi liked the orthogonal

$$Q = J_{kl} = \begin{array}{c} \\ k \\ \\ l \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} \quad & k & \quad & l \end{array} \\ \begin{bmatrix} 1 & & & & \\ & c & & s & \\ & & 1 & & \\ & -s & & c & \\ & & & & 1 \end{bmatrix} \end{array} \tag{8.25}$$

with $c^2 + s^2 = 1$ or $c = \cos\theta,\ s = \sin\theta$, that we have already found so useful in Sec. 5.8. With this matrix, $A' = Q^T A Q$ and the similarity transformation preserves symmetry.

To clearly see the effect of postmultiplying A by Q we write $A = [a_1\ a_2 \dots a_n]$ and have that

$$B = AQ = \begin{array}{c} \begin{array}{ccccc} \quad & \quad & k & \quad & l \end{array} \\ \begin{bmatrix} a_1 & ca_k - sa_l & \dots & ca_l + sa_k & a_n \end{bmatrix} \end{array}. \tag{8.26}$$

In this, columns k and l of A are replaced by their linear combinations $b_k = ca_k - sa_l$, and $b_l = ca_l + sa_k$, respectively. All other columns remain unchanged. Similarly, if we write the rows of B as $b_1^T, b_2^T, \dots, b_n^T$, then

$$A' = Q^T B = \begin{array}{c} \\ k \\ \\ l \\ \\ \end{array} \begin{array}{c} \\ \begin{bmatrix} b_1^T \\ cb_k^T - sb_l^T \\ \vdots \\ cb_l^T + sb_k^T \\ b_n^T \end{bmatrix} \end{array} \tag{8.27}$$

and what AQ did to the columns of A, $Q^T B$ does to the rows of B. The similarity transformation $Q^T A Q$ with the Jacobi rotation matrix $Q = J_{kl}$ affects only the kth and lth rows

and columns of A as is schematically shown below

$$A' = Q^T A Q = \begin{array}{c} \\ k \\ \\ l \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} & k & & l & \end{array} \\ \begin{bmatrix} & \times & & \times & \\ \times & \times & \times & \times & \times \\ & \times & & \times & \\ \times & \times & \times & \times & \times \\ & \times & & \times & \end{bmatrix} \end{array}. \tag{8.28}$$

It is readily verified that since Q is orthogonal

$$b_k^T b_k + b_l^T b_l = a_k^T a_k + a_l^T a_l \tag{8.29}$$

and consequently

$$\sum A_{ij}^2 = \sum B_{ij}^2 = \sum A_{ij}'^2 \tag{8.30}$$

where the three sums extend over all i and j; the sum of A_{ij}^2 over the matrix is invariant under the similarity transformation $Q^T A Q$.

We also readily verify that

$$\begin{aligned} A'_{kk} &= c^2 A_{kk} - 2csA_{kl} + s^2 A_{ll} \\ A'_{ll} &= s^2 A_{kk} + 2csA_{kl} + c^2 A_{ll} \\ A'_{kl} &= (c^2 - s^2)A_{kl} + cs(A_{kk} - A_{ll}) \end{aligned} \qquad \begin{aligned} A'_{kj} &= cA_{kj} - sA_{lj} \\ A'_{lj} &= sA_{kj} + cA_{lj} \end{aligned} \qquad j \neq k, l \tag{8.31}$$

and

$$A_{kk}'^2 + A_{ll}'^2 + 2A_{kl}'^2 = A_{kk}^2 + A_{ll}^2 + 2A_{kl}^2. \tag{8.32}$$

To qualify what we mean by $A' = Q^T A Q$ being closer to diagonal form than A we introduce the three scalar functions $\phi = \phi(A) = \sum A_{ij}^2$, $\delta = \delta(A) = \sum A_{ii}^2$, $\omega = \omega(A) = \phi - \delta$ and use $\omega/\phi = 1 - \delta/\phi$ to measure how near A is to being diagonal. Only when $\delta(A) = \phi(A)$, or $\omega(A) = 0$, is A diagonal.

Here, in view of eq.(8.32),

$$\begin{aligned} \omega(A') &= \omega(A) + 2A_{kl}'^2 - 2A_{kl}^2 \\ \delta(A') &= \delta(A) + 2A_{kl}^2 - 2A_{kl}'^2 \\ \phi(A') &= \phi(A) \end{aligned} \tag{8.33}$$

and $\omega(A')/\phi(A')$ is maximally reduced with the choice of $A'_{kl} = 0$.

Theorem (Jacobi) 8.3. *Let entry A_{kl} of $A = A^T(n \times n)$, be such that $|A_{kl}| \geq |A_{ij}|$ for all $i \neq j$. If in the orthogonal transformation $A' = Q^T AQ$ the rotation angle θ is chosen according to*

$$g = ctn2\theta = \frac{A_{ll} - A_{kk}}{2A_{kl}},\ t = \tan\theta = \frac{sign\ (g)}{|g| + (1 + g^2)^{1/2}} \tag{8.34}$$

so that $c = \cos\theta = (1 + t^2)^{-1/2}$ and $s = \sin\theta = tc$, then

$$\omega(A') \leq (1 - \frac{2}{n^2 - n})\omega(A). \tag{8.35}$$

Proof. To minimize $\omega(A')$ we set

$$A'_{kl} = (c^2 - s^2)A_{kl} + cs(A_{kk} - A_{ll}) = 0 \tag{8.36}$$

and obtain the expression for $ctn\ 2\theta$ using the trigonometric identities $2cs = \sin 2\theta$ and $c^2 - s^2 = \cos 2\theta$. Some more trigonometry leads to

$$t^2 + 2gt = 1 \tag{8.37}$$

and the smaller root of the quadratic equation is used to fix c and s.

With $A'_{kl} = 0,\ \omega(A') = \omega(A) - 2A^2_{kl}$, and since $|A'_{kl}|$ is the largest among the $n^2 - n$ off-diagonal entries

$$\omega(A) \leq (n^2 - n)A^2_{kl} \tag{8.38}$$

from which the inequality of the theorem readily results. End of proof.

It is important that we bear in mind that an off-diagonal entry set to zero does not remain so under successive rotations designed to annul other off-diagonal entries. *Sparseness is not maintained by the Jacobi method.*

Each step of the Jacobi method reduces $\omega(A')$, and the limit of this process is a diagonal matrix. Indeed, if we denote A and A' by A_{k-1} and A_k, respectively, then the equality in Jacobi's theorem becomes

$$\omega(A_k) \leq (1 - \frac{2}{n^2 - n})\omega(A_{k-1}) \tag{8.39}$$

or

$$\omega(A_k) \leq (1 - \frac{2}{n^2 - n})^k \omega(A_0) \tag{8.40}$$

and $\omega(A_k) \to 0$ as $k \to \infty$.

Interestingly enough, convergence of the Jacobi method is proved independently of any intrinsic property of matrix A, except that it be symmetric. The proof gives the impression that convergence is only linear, $\omega_k/\omega_0 = \rho^k$ with $\rho = 1 - 2/(n^2 - n)$, becoming slower and more expensive as n increases. We shall see this to be numerically true at the beginning, but as the similarly reduced matrix nears diagonal form convergence accelerates to become *quadratic*.

In proving Jacobi's theorem we tacitly gave a constructive proof to the spectral theorem that states that for every symmetric matrix A there exists an orthogonal matrix Q such that $Q^T AQ$ is diagonal.

The above is the classical Jacobi method. But searching for the largest $|A_{ij}|$ $i \neq j$ before each rotation is onerous, and an alternative, a serial variant, is devised for the method whereby $A'_{kl} = 0$ is systematically done rowwise. One passage over all the $n(n-1)/2$ off-diagonal entries of the matrix is called a *sweep*. Before A'_{kl} is set to zero $|A_{kl}|/(A_{kk}^2 + A_{ll}^2)^{1/2}$ is compared with a sweep-dependent tolerance to see if the rotation is profitable or should be skipped.

Each sweep brings the matrix closer to diagonal form, and after so many sweeps the matrix is reduced to practically diagonal form with the eigenvalues on the diagonal.

If $Q_0, Q_1, \ldots, Q_m$ are the m rotation matrices used in the succession of orthogonal transformations to reduce A to diagonal form, then $Q = Q_0 Q_1 \cdots Q_m$ holds in its columns the eigenvectors of A.

A theoretical analysis on the convergence of the serial Jacobi method with skips is too technical for presentation here, but we shall look at a substantial numerical example.

A commercially available, professionally written, serial Jacobi computer program is used to compute all eigenvalues of the $n \times n$ matrix

$$\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix} \tag{8.41}$$

for $n = 10$. To measure the closeness of the transformed matrix to diagonal form we use

$$\epsilon_k = [\omega(A_k)/\delta(A_k)]^{1/2} \tag{8.42}$$

and Fig. 8.1 shows $\log(\epsilon_k/\epsilon_0)$ vs. the number of sweeps. During the first two sweeps convergence is nearly linear, but as the matrix nears diagonal form, convergence accelerates considerably. This aspect of the convergence behavior of the Jacobi method saves its practicality and preserves it in the books. The fast final convergence of the Jacobi method means that high accuracy is inexpensive and that typically no more than four sweeps are needed for convergence.

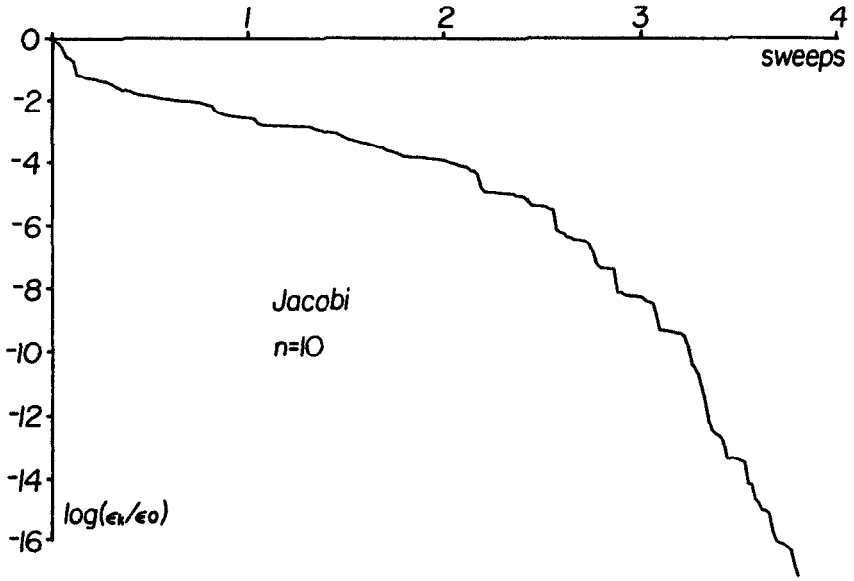


Fig. 8.1

To assess the number of arithmetical operations required by the Jacobi method we observe that there are approximately $n^2/2$ rotations per sweep, and as each rotation requires about $4n$ operations the work per sweep is $2n^3$ operations. Assuming that convergence requires 4 sweeps, a total sum of $8n^2$ operations is required for the computation of all n eigenvalues of the symmetric matrix. Work is nearly doubled if eigenvectors are also sought.

exercises

8.3.1. Carry out one Jacobi sweep over matrix

$$A = \begin{bmatrix} 1 & \alpha & \alpha \\ \alpha & 1 & \alpha \\ \alpha & \alpha & 1 \end{bmatrix}.$$

8.4 The power method

Now that we possess a good algorithm to solve the complete eigenproblem of small dense matrices we turn to the large and sparse $Ax = \lambda Bx$. Working with full matrices is here out of the question for reasons of space and time, and we shall look at routines that take advantage of sparseness. Some of these iterative routines require the solution of a small generalized eigenproblem at each step. Solution of the very large $Ax = \lambda Bx$ is regulated by the fact that only a very small number of eigenvalues (say 10) and their corresponding eigenvectors are usually of any practical interest.

The algebraic eigenproblem $Ax = \lambda Bx$ is often the end product of the mathematical modeling of elastic structure vibrations. In this, $\omega_i = \sqrt{\lambda_i}$ are the approximate natural frequencies of the structure and x_i the corresponding discrete natural vibration modes. Even for the simplest of solids it is physically impossible to isolate and measure vibration modes higher than the tenth.

The *power method* in its various manifestations is the simplest imaginable algorithm, requiring no more than a repeated matrix vector multiplication, yet is one of the most useful methods for the large sparse eigenproblem.

We shall first present the method under the propitious circumstances of $B = I$, and A being symmetric semidefinite. Analysis is simplified with the assumption of non-negative eigenvalues while still displaying the essence of the algorithm. Negative eigenvalues present a slight complication of sign that the reader may easily resolve as an exercise.

Theorem 8.4. *Let $A = A(n \times n)$ be symmetric positive semidefinite with eigenvalues ordered by magnitude as*

$$0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n-2} < \lambda_{n-1} < \lambda_n \tag{8.43}$$

and with corresponding orthonormal eigenvectors $v_1, v_2, \ldots, v_n$. *If*

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n \ , \ \alpha_n \neq 0 \tag{8.44}$$

and

$$x_k = A x_{k-1} \ , \ k = 1, 2, \ldots \tag{8.45}$$

then

$$\lim_{k\to\infty} \| \frac{x_k}{\|x_k\|} - v_n \| / \rho_{n-1}^k = | \frac{\alpha_{n-1}}{\alpha_n} | \tag{8.46}$$

and

$$\lim_{k\to\infty} \Big(\frac{\lambda_k' - \lambda_n}{\lambda_n} \Big) / \rho_{n-1}^{2k} = \Big(\frac{\alpha_{n-1}}{\alpha_n} \Big)^2 \tag{8.47}$$

where

$$\lambda_k' = \frac{x_k^T A x_k}{x_k^T x_k} \ \textit{and} \ \ \rho_j = \frac{\lambda_j}{\lambda_n} \ , \ j = 1, 2, \ldots, n. \tag{8.48}$$

Proof. Since v_n is of arbitrary sign we may assume that $\alpha_n > 0$. At the kth iteration

$$x_k = \alpha_1 \lambda_1^k v_1 + \cdots + \alpha_{n-1} \lambda_{n-1}^k v_{n-1} + \alpha_n \lambda_n^k v_n \tag{8.49}$$

or

$$x_k = \alpha_n \lambda_n^k \Big(\frac{\alpha_1}{\alpha_n} \rho_1^k v_1 + \cdots + \frac{\alpha_{n-1}}{\alpha_n} \rho_{n-1}^k v_{n-1} + v_n \Big) \tag{8.50}$$

where $\rho_j = \lambda_j / \lambda_n$. Because $\rho_j < 1$ if $j < n$, $\rho_j^k \to 0$ as $k \to \infty$, and x_k becomes closer in direction to v_n as iteration proceeds.

Actually

$$\|x_k\| = \alpha_n \lambda_n^k \Big(\Big(\frac{\alpha_1}{\alpha_n} \Big)^2 \rho_1^{2k} + \cdots + \Big(\frac{\alpha_{n-1}}{\alpha_n} \Big)^2 \rho_{n-1}^{2k} + 1 \Big)^{1/2} \tag{8.51}$$

or approximately

$$\|x_k\| = \alpha_n \lambda_n^k \Big(1 + \frac{1}{2} \Big(\frac{\alpha_1}{\alpha_n} \Big)^2 \rho_1^{2k} + \cdots + \frac{1}{2} \Big(\frac{\alpha_{n-1}}{\alpha_n} \Big)^2 \rho_{n-1}^{2k} \Big) \tag{8.52}$$

when $(\alpha_j / \alpha_n)^2 \rho_j^{2k} << 1$, and

$$\frac{\alpha_n \lambda_n^k}{\|x_k\|} = 1 - \frac{1}{2} \Big(\frac{\alpha_1}{\alpha_n} \Big)^2 \rho_1^{2k} - \cdots - \frac{1}{2} \Big(\frac{\alpha_{n-1}}{\alpha_n} \Big)^2 \rho_{n-1}^{2k}. \tag{8.53}$$

We readily verify that

$$\|\frac{x_k}{\|x_k\|} - v_n\|^2 = 2(1 - \frac{x_k^T v_n}{\|x_k\|}) = 2(1 - \frac{\alpha_n \lambda_n^k}{\|x_k\|}) \tag{8.54}$$

whereupon

$$\|\frac{x_k}{\|x_k\|} - v_n\|/\rho_{n-1}^k = \frac{1}{\alpha_n}\Big(\alpha_1^2\Big(\frac{\lambda_1}{\lambda_{n-1}}\Big)^{2k} + \alpha_2^2\Big(\frac{\lambda_2}{\lambda_{n-1}}\Big)^{2k} + \cdots + \alpha_{n-1}^2\Big)^{1/2} \tag{8.55}$$

and since $\lambda_j/\lambda_{n-1} < 1$ if $j < n-1$, as $k \to \infty$ the right-hand side of the above equality tends to $|\alpha_{n-1}/\alpha_n|$. For a sufficiently large k we may write

$$\|\frac{x_k}{\|x_k\|} - v_n\| = |\frac{\alpha_{n-1}}{\alpha_n}|\Big(\frac{\lambda_{n-1}}{\lambda_n}\Big)^k \tag{8.56}$$

meaning that, asymptotically, convergence of $x_k/\|x_k\|$ towards v_n is *linear*.

To prove the second limit we establish that

$$\frac{\lambda_n - \lambda_k'}{\lambda_n} = \frac{\Big(\alpha_1/\alpha_n\Big)^2 \rho_1^{2k}(1-\rho_1) + \cdots + \Big(\alpha_{n-1}/\alpha_n\Big)^2 \rho_{n-1}^{2k}(1-\rho_{n-1})}{\Big(\alpha_1/\alpha_n\Big)^2 \rho_1^{2k} + \cdots + \Big(\alpha_{n-1}/\alpha_n\Big)^2 \rho_{n-1}^{2k} + 1} \tag{8.57}$$

from which the desired result is obtained.

For a sufficiently large k we may write

$$\frac{\lambda_k' - \lambda_n}{\lambda_n} = \Big(\frac{\alpha_{n-1}}{\alpha_n}\Big)^2 \Big(\frac{\lambda_{n-1}}{\lambda_n}\Big)^{2k} \tag{8.58}$$

meaning that, asymptotically, convergence of λ_k' to λ_n is linear with an ultimate rate of convergence that is $(\lambda_{n-1}/\lambda_n)^2$. End of proof.

Application of the power method is best for matrices for which $|\lambda_{n-1}/\lambda_n| << 1$. This is most dramatically illustrated on matrices

$$A = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix} \text{ and } A^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}. \tag{8.59}$$

We know that the n eigenvalues of the positive definite and symmetric A are

$$\lambda_j = 4\sin^2\frac{\pi}{2}\Big(\frac{2j-1}{2n+1}\Big) \quad j = 1, 2, \ldots, n \tag{8.60}$$

and those of A^{-1} are $\lambda_n^{-1} < \lambda_{n-1}^{-1} < \ldots < \lambda_1^{-1}$. When n is large

$$\lambda_{n-1} = 4\Big(1 - \frac{\pi^2}{n^2}\Big),\ \lambda_n = 4\Big(1 - \frac{\pi^2}{4n^2}\Big), \rho_{n-1} = \frac{\lambda_{n-1}}{\lambda_n} = 1 - \frac{3\pi^2}{4n^2} \tag{8.61}$$

and the rate of convergence of the power method applied to A slumps as n increases. Convergence of λ_n can be painfully slow. On the other hand, for matrix A^{-1}

$$\frac{\lambda_n(A^{-1})}{\lambda_{n-1}(A^{-1})} = \frac{\lambda_2(A)}{\lambda_1(A)} = \frac{1}{9} \tag{8.62}$$

independently of n, and a few steps produce accurate approximations to $\lambda_n(A^{-1}) = \lambda_1^{-1}(A)$ for any matrix size.

Figure 8.2 numerically substantiates the theory. Computations also demonstrate the importance of a good initial guess x_0. With an iterative process that converges only linearly, accurate results are expensive and a good initial guess can result in substantial computational savings. Computation of Ax_k is the most expensive item in the power method if A is large, and any reduction in the number of iterations is welcome.

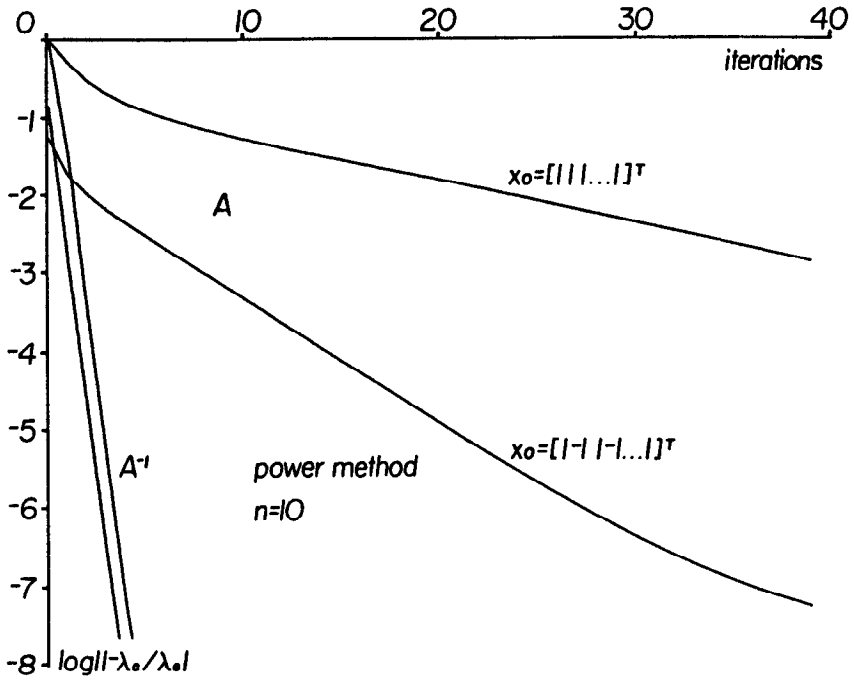


Fig. 8.2

Some more remarks on the power method:

Suppose that λ_{n-1} is close to λ_n so that λ_{n-1}/λ_n is only slightly less than 1, but that λ_{n-2} is well separated from the top two eigenvalues. Under these circumstances convergence of the power method to λ_n may be impractically slow. But we must recognize that the high computational cost to obtain a precise v_n is a result of our determination to accurately separate the two close eigenvalues. If we agree to disregard the difference between λ_{n-1} and λ_n and look upon them as one repeating eigenvalue, then the rate of convergence becomes $(\lambda_{n-2}/\lambda_n)^2$ and x_k tends to an eigenvector, depending on x_0, that lies in the plane of v_n and v_{n-1}.

A decidedly more efficient way to separate close eigenvalues is discussed in Sec. 8.6.

We easily perceive that the requirement that matrix A in Theorem 8.4 be positive semidefinite is not essential to the success of the power method. But if A is merely symmetric we must brace for the eventuality of eigenvalues equal in magnitude but different in sign.

Application of the power method to the general symmetric eigenproblem $Ax = \lambda Bx$, where B is positive definite, is in principle as simple as for the particular case of $B = I$. Factorization of B reduces the general eigenproblem to the particular $L^{-1}AL^{-T}x' = \lambda x'$, where $x' = L^T x$, and where $B = LL^T$. The power method becomes

$$Ax_{k-1} = LL^T x_k \tag{8.63}$$

and two systems of triangular equations need be solved at each step. Sparseness is taken advantage of both in the operation Ax_k and in the LL^T factorization of B, but this factorization is expensive and space-consuming. Another possibility that does not require the LL^T decomposition and can lead to considerable space saving in case of highly structured B is to write $Ax_{k-1} = Bx_k$ and solve the linear system $Bx_k = (Ax_{k-1})$ with the conjugate gradient method. Such a scheme is often feasible since it is common for B to be very well-conditioned.

Let $A = A^T$ have eigenvalues $|\lambda_1| \leq |\lambda_2| \leq |\lambda_{n-1}| < |\lambda_n|$ with corresponding orthonormal eigenvectors $v_1, v_2, \ldots, v_n$. Then

$$A^k = \lambda_n^k(\rho_1^k v_1 v_1^T + \rho_2^k v_2 v_2^T + \ldots + \rho_{n-1}^k v_{n-1} v_{n-1}^T + v_n v_n^T),\ \rho_j = \lambda_j/\lambda_n \tag{8.64}$$

and as $k \to \infty$ matrix A^k approaches the rank one matrix $\lambda_n^k v_n v_n^T$. After so many iterations vector x_k remains nearly colinear with v_n,

$$x_k = A^k x_0 = \lambda_n^k (v_n^T x_0) v_n \tag{8.65}$$

approximately, and

$$|\lambda_n| = \|x_{k+1}\|/\|x_k\|, \ |\lambda_n| = (\|x_k\|/(x_0^T v_n))^{1/k} \tag{8.66}$$

for very large k.

Say symmetric matrix A happens to have two dominant eigenvalues equal in magnitude but opposite in sign, λ and $\lambda' = -\lambda$, respectively, with corresponding orthonormal eigenvectors v and v'. After a good number of power iterations vector x_k becomes nearly confined to the two-dimensional space spanned by v and v'. Then

$$x_k = \alpha v + \alpha' v', \ x_{k+1} = \lambda(\alpha v - \alpha' v'), \ x_{k+2} = \lambda^2(\alpha v + \alpha' v') \tag{8.67}$$

and

$$\lambda = \pm(\|x_{k+2}\|/\|x_k\|)^{1/2}. \tag{8.68}$$

Any matrix with distinct eigenvalues has n linearly independent eigenvectors, expansion (8.44) exists for its eigenvectors, and eq. (8.49) is true for this matrix as it is true for the symmetric. Let the dominant eigenvalues of A be λ and $\overline{\lambda}$ with corresponding eigenvectors v and $\overline{v}$. After a great number of iterations essentially

$$x_k = A^k x_0 = \alpha v + \alpha' \overline{v} \tag{8.69}$$

where $\alpha' = \overline{\alpha}$ by dint of x_k being real.

Hence

$$x_k = \alpha v + \overline{\alpha v}, \ x_{k+1} = \alpha\lambda v + \overline{\alpha}\overline{\lambda}\overline{v}, \ x_{k+2} = \alpha\lambda^2 v + \overline{\alpha}\overline{\lambda}^2\overline{v} \tag{8.70}$$

are linearly dependent,

$$x_{k+2} + \beta_1 x_{k+1} + \beta_0 x_k = \alpha v(\lambda^2 + \beta_1\lambda + \beta_0) + \overline{\alpha v}(\overline{\lambda}^2 + \beta_1\overline{\lambda} + \beta_0) = o \tag{8.71}$$

and eigenvalues λ and $\overline{\lambda}$ are the two conjugate roots of the quadratic equation $\lambda^2+\beta_1\lambda+\beta_0 = 0$.

In reality, vectors x_{k+2}, x_{k+1}, x_k are never truly colinear and eq. (8.71) is solved for β_0 and β_1 in some approximate sense.

In many matrix iterative processes only the modulus of the dominant eigenvalue–the spectral radius of A–is desired. Then

$$\rho = \max_j |\lambda_j| = \lim_{k\to\infty} \|A^k x_0\|^{\frac{1}{k}} = \lim_{k\to\infty} \|x_{k+1}\|/\|x_k\| \tag{8.72}$$

for any random x_0, whether A is diagonalizable or not, but convergence can be painfully slow. Applying this to

$$A = \begin{bmatrix} 3 & 4 & 4 & 4 \\ -3 & -5 & -8 & -9 \\ 3 & 6 & 10 & 10 \\ -1 & -2 & -3 & -2 \end{bmatrix}, \; x_0 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \tag{8.73}$$

that has eigenvalues $\lambda_1 = \lambda_2 = 1, \lambda_3 = \lambda_4 = 2$ with two linearly independent eigenvectors, we obtain $\rho_k = 2.39, 2.20, 2.10, 2.05$ for $k = 5, 10, 20, 40$.

One last remark. If $|\lambda_n| > 1$, $\|x_k\|$ increases with k to the point of causing overflow failure, hence it is prudent to normalize x_k at each step of the power method.

exercises

8.4.1. Apply the power method to

$$A = \begin{bmatrix} & 1 & & & \\ -1 & & 1 & & \\ & -1 & & 1 & \\ & & -1 & & 1 \\ & & & -1 & \end{bmatrix}, \; x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$A^2 = \begin{bmatrix} -1 & & 1 & & \\ & -2 & & 1 & \\ 1 & & -2 & & 1 \\ & 1 & & -2 & \\ & & 1 & & -1 \end{bmatrix}, \; x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

8.4.2. Apply the power method to

$$J = \begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & 1 & 1 & \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix}, \; x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \; x_0 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

8.4.3. For matrix

$$A = \begin{bmatrix} 1 & -1 & & & \\ -1 & 4 & -3 & & \\ & -3 & 8 & -5 & \\ & & -5 & 12 & -7 \\ & & & -7 & 16 \end{bmatrix}$$

carry out one inverse iteration starting with $x_0 = [5\ 4\ 3\ 2\ 1]^T$. Having found $x_1 = A^{-1}x_0$, assume that $x_1 = v$ and form diagonal matrix $D, D_{ii} = v_i$. Apply the similarity transformation $D^{-1}AD$ to A and use Gerschgorin's theorem to establish a nontrivial lower bound on λ_1, the smallest eigenvalue of A. Use Rayleigh's quotient to establish an upper bound on λ_1.

Apply the power method starting with $x_0 = [1\ -1\ 1\ -1\ 1]^T$ and do the same to λ_5. Improve the bounds with more power and inverse iterations.

8.5 Shifted inverse iteration

Inverse iteration is the name given to the application of the power method to A^{-1}. From our last numerical example, we can understand when and why this version of the power method holds special appeal.

Let A be symmetric positive definite with eigenvalues $0 < \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$ and corresponding eigenvectors $v_1, v_2, \ldots, v_n$. The eigenvalues of A^{-1} are then $0 < \lambda_n^{-1} \leq \lambda_{n-1}^{-1} \leq \cdots \leq \lambda_2^{-1} < \lambda_1^{-1}$, with corresponding eigenvectors $v_n, v_{n-1}, \ldots, v_1$. Applied to A^{-1}, the power method, starting with $x_0, x_0^T v_1 \neq 0$, produces a vector sequence $x_0, x_1, \ldots, x_k = A^{-1}x_{k-1}$ that converges to v_1 at the asymptotic rate λ_1/λ_2. For finite difference and finite element stiffness matrices this rate is independent of n.

In practice, inverse iteration is carried out as $Lx' = x_{k-1},\ L^T x_k = x'$ requiring only two back-substitutions per step. Extension of inverse iteration to $Ax = \lambda Bx$ is obvious.

The subtraction $A - \lambda' I$ causes every eigenvalue λ_j of A to shift to $\lambda_j - \lambda'$, and the jth eigenvalues of $(A - \lambda' I)^{-1}$ become $(\lambda_j - \lambda')^{-1}$. If λ' is well near λ_m for some m, then $(\lambda_m - \lambda')^{-1}$ is the largest eigenvalue in magnitude of $(A - \lambda' I)^{-1}$, it is clearly separated from the rest of the eigenvalues, and convergence of inverse iteration to the eigenvector corresponding to λ_m is fast. With a good choice of λ' no more than two or three cycles should be needed for convergence.

In the extreme the shift may be performed at each step of inverse iteration,

$$\lambda'_k = \frac{x_k^T A x_k}{x_k^T x_k}, \ (A - \lambda'_{k-1} I) x_k = x_{k-1} \tag{8.74}$$

for which convergence becomes asymptotically *cubic*. Consider

$$A = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix}, \ x_0 = \begin{bmatrix} 1 \\ \eta_2 \epsilon \\ \eta_3 \epsilon \end{bmatrix} \tag{8.75}$$

with $\lambda_1 < \lambda_2 < \lambda_3$, and parameter $|\epsilon| << 1$. For this initial vector

$$\lambda' = \lambda'(x_0) = \frac{\lambda_1 + \epsilon^2(\lambda_2 \eta_2^2 + \lambda_3 \eta_3^2)}{1 + \epsilon^2(\eta_2^2 + \eta_3^2)} \tag{8.76}$$

and since ϵ^2 is minute

$$\lambda_1 - \lambda' = -\epsilon^2((\lambda_2 - \lambda_1)\eta_2^2 + (\lambda_3 - \lambda_1)\eta_3^2), \lambda_2 - \lambda' = \lambda_2 - \lambda_1, \lambda_3 - \lambda' = \lambda_3 - \lambda_1. \tag{8.77}$$

The difference between the Rayleigh approximation $\lambda'(x_0)$ and λ_1 is already proportional to ϵ^2. Next

$$x_1 = \frac{1}{\lambda_1 - \lambda'} \begin{bmatrix} 1 \\ \epsilon(\lambda_1 - \lambda')/(\lambda_2 - \lambda') \\ \epsilon(\lambda_1 - \lambda')/(\lambda_3 - \lambda') \end{bmatrix} = \frac{1}{\lambda_1 - \lambda'} \begin{bmatrix} 1 \\ \eta_2(\lambda)\epsilon^3 \\ \eta_3(\lambda)\epsilon^3 \end{bmatrix} \tag{8.78}$$

where $\eta_2(\lambda)$ and $\eta_3(\lambda)$ are rational functions of $\lambda_1, \lambda_2, \lambda_3$. Now

$$\lambda_1 - \lambda'(x_1) = -\epsilon^6 \frac{(\lambda_2 - \lambda_1)\eta_2^2 + (\lambda_3 - \lambda_1)\eta_3^2}{1 + \epsilon^6(\eta_2^2 + \eta_3^2)} \tag{8.79}$$

and the error in $\lambda'(x_1)$ is nearly proportional to ϵ^6.

The price for the accelerated convergence: repeated factorization of $A - \lambda'_{k-1} I$.

Notice that in any event $A - \lambda' I$ may not be positive semidefinite, causing difficulties in the triangular factorization.

One of the more important uses of inverse iteration is to compute an eigenvector for an approximate eigenvalue λ'. In this case $A - \lambda' I$ is nearly singular and it becomes more so as the approximation is improved. It appears that this would doom inverse iteration, for at each step we need to solve a nearly singular system of linear equations that is highly susceptible to round-off errors. But it does not.

To understand why this is so we shall start by considering the nearly singular 2×2 system

$$\begin{bmatrix} 1 & -1 \\ -1 & 1+\epsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ , \ Ax = f, \ |\epsilon| << 1 \tag{8.80}$$

with matrix A that has two eigenvalues, $\lambda_1 = \frac{1}{2}\epsilon$ and $\lambda_2 = 2 + \frac{1}{2}\epsilon$, and corresponding eigenvectors $v_1 = [1 \ 1 - \frac{1}{2}\epsilon]^T$, $v_2 = [1 \ \ -1 - \frac{1}{2}\epsilon]^T$. When $\epsilon = 0, \lambda_1 = 0$, $\lambda_2 = 2$, $v_1 = [1 \ 1]^T$, $v_2 = [1 \ -1]^T$, and A is singular.

We readily solve the system for $x = [1 + \epsilon \ 1]^T/\epsilon$ and as ϵ decreases in magnitude vector x grows in length, accurately enough $\|x\| = \sqrt{2}/\epsilon$. Small changes in ϵ cause large changes in the magnitude of x, but they barely change the eigenvectors of $A = A(\epsilon)$, and the near colinearity of x and v_1. Indeed

$$\frac{x^T v_1}{\|x\| \ \|v_1\|} = 1 - \frac{1}{16}\epsilon^2, \ \frac{x^T v_2}{\|x\| \ \|v_2\|} = \frac{1}{4}\epsilon \tag{8.81}$$

and this is what we want: x to be nearly colinear with v_1. Magnitudes do not count in the eigenproblem, it is direction that is important.

Graphically, the solution of our 2×2 nearly singular system can be described as the intersection of nearly parallel lines. Small changes in the coefficient matrix A or right-hand side vector f cause large movements of the intersection point, but these movements are nearly parallel to v_1; the error in the solution is overwhelmingly in the v_1 direction.

For an analytic explanation of the success of inverse iteration applied to a nearly singular symmetric matrix, we assume the eigenvectors $v_1, v_2, \ldots, v_n$ of A normalized, and write

$$A^{-1} = \frac{1}{\lambda_1} v_1 v_1^T + \frac{1}{\lambda_2} v_2 v_2^T + \cdots + \frac{1}{\lambda_n} v_n v_n^T \tag{8.82}$$

to have one inverse iteration produce

$$x_1 = A^{-1} x_0 = \frac{v_1^T x_0}{\lambda_1} v_1 + \frac{v_2^T x_0}{\lambda_2} v_2 + \cdots + \frac{v_n^T x_0}{\lambda_n} v_n. \tag{8.83}$$

Round-off error effects can be looked upon as causing a slight change in A. The eigenvalues and eigenvectors of A barely change (assuming well-separated eigenvalues) due to this perturbation, and even if $\lambda_1/\lambda_n << 1$, x_1 remains nearly colinear with v_1, provided that $v_1^T x_0 \neq 0$.

One might hope to circumvent inverse iteration for the computation of v_1, the eigenvector belonging to the smallest eigenvalue of the positive definite and symmetric A, by doing this: Suppose that a good approximation λ_n' is found to the largest eigenvalue λ_n of A. Suppose even that λ_n' is so near λ_n that for all practical purposes we may take $\lambda_n' = \lambda_n$. The shifted matrix $\lambda_n I - A$ has eigenvalues $0, \lambda_n - \lambda_{n-1}, \ldots, \lambda_n - \lambda_1$ with corresponding eigenvectors $v_n, v_{n-1}, \ldots, v_1$. Since $\lambda_n - \lambda_1$ is largest in magnitude the power method applied to $\lambda_n I - A$ will converge to v_1. This shifting device looks attractive inasmuch as it does not require factorizations or back substitutions, but it is, unfortunately, inefficient. For matrix A with $\lambda_i = i^2$, for instance,

$$\frac{\lambda_n - \lambda_2}{\lambda_n - \lambda_1} = \frac{n^2 - 4}{n^2 - 1} = 1 - \frac{3}{n^2} \tag{8.84}$$

if n is large, intimating a dismal convergence rate.

8.6 Ritz separation of close eigenvalues

In case of the largest eigenvalues being nearly equal in magnitude, the power method, or for that matter inverse iteration, brings x_k into the subspace spanned by the eigenvectors corresponding to the dominant eigenvalues, but convergence to one of them may be irritatingly slow and wasteful. To be specific, suppose that inverse iteration is carried out on matrix A with close λ_1 and λ_2 but with eigenvalues that are otherwise well-separated from λ_2. After a sufficiently large number of iterations the method turns out

$$x_k = \alpha_1 v_1 + \alpha_2 v_2,\ x_{k+1} = \lambda_1 \alpha_1 v_1 + \lambda_2 \alpha_2 v_2,\ x_{k+2} = \lambda_1^2 \alpha_1 v_1 + \lambda_2^2 \alpha_2 v_2 \tag{8.85}$$

that are linearly dependent.

If so then there are two nonzero scalars β_0 and β_1 so that

$$x_{k+2} + \beta_1 x_{k+1} + \beta_0 x_k = o \tag{8.86}$$

or in terms of v_1 and v_2

$$\alpha_1 v_1(\lambda_1^2 + \beta_1\lambda_1 + \beta_0) + \alpha_2 v_2(\lambda_2^2 + \beta_1\lambda_2 + \beta_0) = o \tag{8.87}$$

and the two eigenvalues are obtained from the quadratic equation

$$\lambda^2 + \beta_0\lambda + \beta_0 = 0. \tag{8.88}$$

The Ritz separation method starts with the trial vector

$$x = \beta_0 x_0 + \beta_1 x_1 \tag{8.89}$$

where x_0 and x_1 are written for x_k and x_{k+1}, then proceeds to minimize $\lambda(x) = x^T Ax/x^T x$ over β_0 and β_1. This leads to the 2×2 general eigenproblem

$$\begin{bmatrix} x_0^T Ax_0 & x_0^T Ax_1 \\ x_1^T Ax_0 & x_1^T Ax_1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \lambda \begin{bmatrix} x_0^T x_0 & x_0^T x_1 \\ x_1^T x_0 & x_1^T x_1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \tag{8.90}$$

for which the characteristic equation is

$$\lambda^2 \begin{vmatrix} Z_{00} & Z_{01} \\ Z_{10} & Z_{11} \end{vmatrix} - \lambda \begin{vmatrix} Z_{00} & Z_{01} \\ Z_{11} & Z_{12} \end{vmatrix} + \begin{vmatrix} Z_{01} & Z_{11} \\ Z_{11} & Z_{12} \end{vmatrix} = 0 \tag{8.91}$$

where $Z_{ij} = x_i^T x_j$, and $x_2 = Ax_1$.

The smaller root of the characteristic equation approximates λ_1, and the larger, λ_2. Each root comes with a pair β_0, β_1, which when put back into $x = \beta_0 x_0 + \beta_1 x_1$ yields an approximation to the corresponding eigenvector.

Clearly, the procedure may be extended to higher-dimensional vector subspaces to separate more than two close eigenvalues. Extension of the Ritz separation technique to the general $Ax = \lambda Bx$ is also obvious, except that in this case the right-hand 2×2 matrix does not include $x_i^T x_j$ but rather $x_i^T Bx_j$.

Example. Consider

$$A = \begin{bmatrix} 1 & & \\ & 1.99 & \\ & & 2.01 \end{bmatrix}, \; x_0 = \begin{bmatrix} 0.01 \\ 1 \\ 1 \end{bmatrix}, \; x_1 = \begin{bmatrix} 0.01 \\ 1.99 \\ 2.01 \end{bmatrix}, \; x_2 = \begin{bmatrix} 0.01 \\ 1.99^2 \\ 2.01^2 \end{bmatrix} \tag{8.92}$$

for which we obtain

$$\lambda(x_0) = 1.99995, \;\; \lambda(x_1) = 2.0000875, \;\; \lambda(x_2) = 2.0001969 \tag{8.93}$$

where $\lambda(x) = x^T Ax/x^T x$. In trying to separate the highest eigenvalues by means of eq. (8.86) we write $\beta_0 x_0 + \beta_1 x_1 = -x_2$ and seek the least squares solution to it by forming

$$\begin{bmatrix} x_0^T x_0 & x_0^T x_1 \\ x_1^T x_0 & x_1^T x_1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = - \begin{bmatrix} x_0^T x_2 \\ x_1^T x_2 \end{bmatrix} \tag{8.94}$$

which is numerically

$$\begin{bmatrix} 2.0001 & 4.0001 \\ 4.0001 & 8.0003 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = - \begin{bmatrix} 8.003 \\ 16.0013 \end{bmatrix}. \tag{8.95}$$

We solve the 2×2 system for $\beta_0 = 3.333365$, $\beta_1 = -3.6666667$, and obtain from $\lambda^2 + \beta_1 \lambda + \beta_0 = 0$ the two roots $\lambda_1 = 1.66676$ and $\lambda_2 = 1.9999$.

The Ritz separation produces the characteristic equation

$$0.0006\lambda^2 - 0.0022001\lambda + 0.002 = 0 \tag{8.96}$$

from which we extract $\lambda_1 = 1.6663$ and $\lambda_2 = 2.0004$, implying the need for more power iterations. After three more steps Ritz separates the eigenvalues into $\lambda_1 = 1.9853$ and $\lambda_2 = 2.00699$, and three additional steps produce $\lambda_1 = 1.98993$ and $\lambda_2 = 2.00994$.

8.7 Rayleigh quotient minimization

Instead of applying the Ritz method at the end of power iteration to separate close eigenvalues we may apply it at each iterative step to improve convergence. In this section we describe a combination of the Ritz method and the conjugate gradient algorithm applied to the minimization of Rayleigh's quotient.

In Chapter 7 we gave the conjugate gradient algorithm for the solution of the linear system $Ax = f$ with a positive definite and symmetric matrix A the variational interpretation of minimizing the quadratic functional $\phi(x) = x^T f - \frac{1}{2} x^T Ax$. We shall see now how to adapt the algorithm to find the minimum of

$$\lambda(x) = \frac{x^T Ax}{x^T Bx} \tag{8.97}$$

with a symmetric A and a positive definite B.

One of the vectors involved in the conjugate gradient algorithm is the residual $r = f - Ax$, which is the gradient of the quadratic functional $\phi(x)$. For the Rayleigh quotient

$$grad\lambda(x) = r = \frac{2}{x^T Bx}(Ax - \lambda Bx) \tag{8.98}$$

and having selected the initial guess x_0 we set, as in the quadratic case,

$$p_0 = r_0 = \frac{2}{x_0^T B x_0}(Ax_0 - \lambda_0 B x_0), \quad \lambda_0 = \lambda(x_0). \tag{8.99}$$

The first step of the conjugate gradient algorithm consists of writing $x = x_0 + \alpha_0 p_0$ and fixing α_0 by the condition that α_0 minimizes $\phi(x)$. To do the same to $\lambda(x)$ we write $x = \alpha_0' x_0 + \alpha_0 p_0$ and minimize $\lambda(x)$ with respect to α_0' and α_0. This leads to the 2×2 eigenproblem

$$\begin{bmatrix} x_0^T A x_0 & x_0^T A p_0 \\ p_0^T A x_0 & p_0^T A p_0 \end{bmatrix} \begin{bmatrix} \alpha_0' \\ \alpha_0 \end{bmatrix} = \lambda \begin{bmatrix} x_0^T B x_0 & x_0^T B p_0 \\ p_0^T B x_0 & p_0^T B p_0 \end{bmatrix} \begin{bmatrix} \alpha_0' \\ \alpha_0 \end{bmatrix} \tag{8.100}$$

from which we obtain, by setting $\alpha_0' = 1$ and eliminating λ_1,

$$\alpha_0^2 \begin{vmatrix} x_0^T A p_0 & x_0^T B p_0 \\ p_0^T A p_0 & p_0^T B p_0 \end{vmatrix} + \alpha_0 \begin{vmatrix} x_0^T A x_0 & x_0^T B x_0 \\ p_0^T A p_0 & p_0^T B p_0 \end{vmatrix} + \begin{vmatrix} x_0^T A x_0 & x_0^T B x_0 \\ p_0^T A x_0 & p_0^T B x_0 \end{vmatrix} = 0. \tag{8.101}$$

The largest root of the quadratic equation corresponds to the maximization of $\lambda(x)$, and the lowest root to the minimization of $\lambda(x)$.

With α_0 secured we have the next approximation $x_1 = x_0 + \alpha_0 p_0$ with which we compute the new residual vector r_1, and in complete analogy with the conjugate gradients minimization algorithm for quadratic functions we choose the new search direction $p_1 = r_1 + \beta_0 p_0$ with $\beta_0 = r_1^T r_1 / r_0^T r_0$, and repeat the minimization. Going through the iterative cycles we produce better and better approximations to the lowest (highest) eigenvalue and its corresponding eigenvector.

It is beyond us to do a theoretical study on the convergence behavior of the method just proposed, but a numerical example should give us a fair indication of its worth.

Figure 8.3 shows the convergence of the first eigenvalue of diagonal matrix

$$D_{ii} = 4 \sin^2 \frac{\pi i}{(n+1)} \quad i = 1, 2, \ldots, n = 19 \tag{8.102}$$

for the initial guess $x_0 = [1/1 \;\; 1/2 \;\; \ldots 1/n]^T$, with computations done in single ($u = 0.5\, 10^{-6}$) precision and double ($u = 10^{-16}$) precision.

The algorithm appears to be surprisingly effective, extremely easy to program, and does not require the matrices in tabular form. The fact that the algorithm can be applied to the general eigenproblem $Ax = \lambda Bx$ without factoring B or inverting it, is of considerable

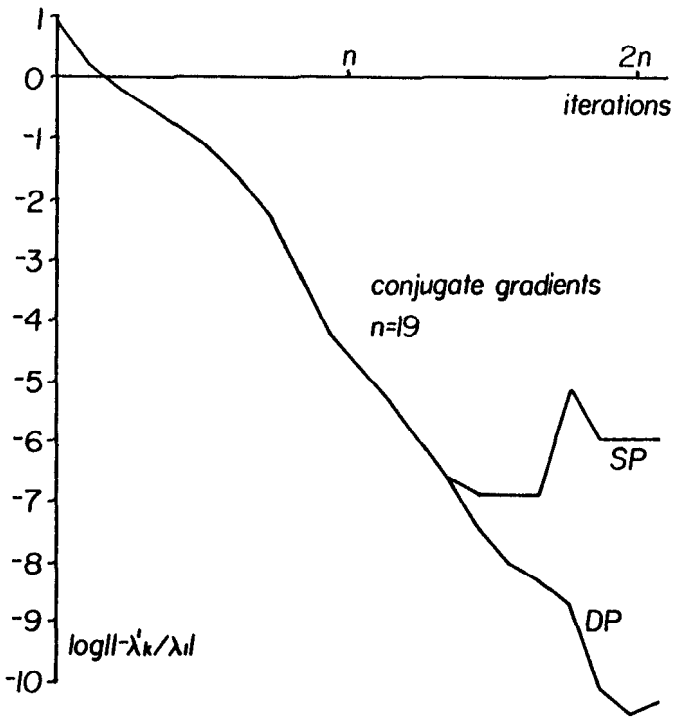


Fig. 8.3

advantage in physical applications where the general problem rather than the special $Ax = \lambda x$ is the rule.

In the basic power method each multiplication $x_{k+1} = Ax_k$ of the iterated vector by the matrix strengthens the prominence of the v_n component of x_k at the expense of the other eigenvectors. To the contrary, minimization methods that seek the lowest eigenvalue of A work to suppress the eigenvectors corresponding to the high eigenvalues.

Let A by symmetric, and

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n,\ x_1 = Ax_0 = \alpha_1\lambda_1 v_1 + \alpha_2\lambda_2 v_2 + \ldots + \alpha_n\lambda_n v_n$$

$$x_2 = Ax_1 = \alpha_1\lambda_1^2 v_1 + \alpha_2\lambda_2^2 v_2 + \ldots + \alpha_n\lambda_n^2 v_n \tag{8.103}$$

so that

$$x_k = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 = \alpha_1(\beta_0 + \beta_1\lambda_1 + \beta_2\lambda_1^2)v_1 + \alpha_2(\beta_0 + \beta_1\lambda_2 + \beta_2\lambda_2^2)v_2$$

$$+ \ldots + \alpha_n(\beta_0 + \beta_1\lambda_n + \beta_2\lambda_n^2)v_n. \tag{8.104}$$

An iterative method that seeks to approach x_k towards v_1 in some way selects $\beta_0, \beta_1, \beta_2$ to diminish $\beta_0 + \beta_1\lambda_j + \beta_2\lambda_j^2$ for the higher eigenvalues.

Minimization of the Rayleigh quotient $\lambda(x) = x^TAx/x^TBx$ over $\beta_0, \beta_1, \beta_2$ produces a Krylov-Rayleigh-Ritz alternative to conjugate gradients. This leads to the solution of the, here 3×3, eigenproblem

$$\begin{bmatrix} x_0^TAx_0 & x_0^TAx_1 & x_0^TAx_2 \\ x_1^TAx_0 & x_1^TAx_1 & x_1^TAx_2 \\ x_2^TAx_0 & x_2^TAx_1 & x_2^TAx_2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \lambda \begin{bmatrix} x_0^TBx_0 & x_0^TBx_1 & x_0^TBx_2 \\ x_1^TBx_0 & x_1^TBx_1 & x_1^TBx_2 \\ x_2^TBx_0 & x_2^TBx_1 & x_2^TBx_2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \tag{8.105}$$

and the procedure is repeated with $\beta_0, \beta_1, \beta_2$ that correspond to the least among the three eigenvalues of eq. (8.105).

Figure 8.4 shows the convergence of the iterated λ_k' towards the correct λ_1 for the diagonal matrix of eq. (8.102) and the same initial guess as for the conjugate gradients method. All computations are done in double precision. Convergence is shown for a three-dimensional search over the Krylov subspace x_0, Ax_0, A^2x_0, entailing the solution of a 3×3 eigenproblem at each step, and a five-dimensional search over $x_0, Ax_0, \ldots, A^4x_0$, entailing the solution of a 5×5 eigenproblem at each iterative step. In both cases convergence is evidently linear.

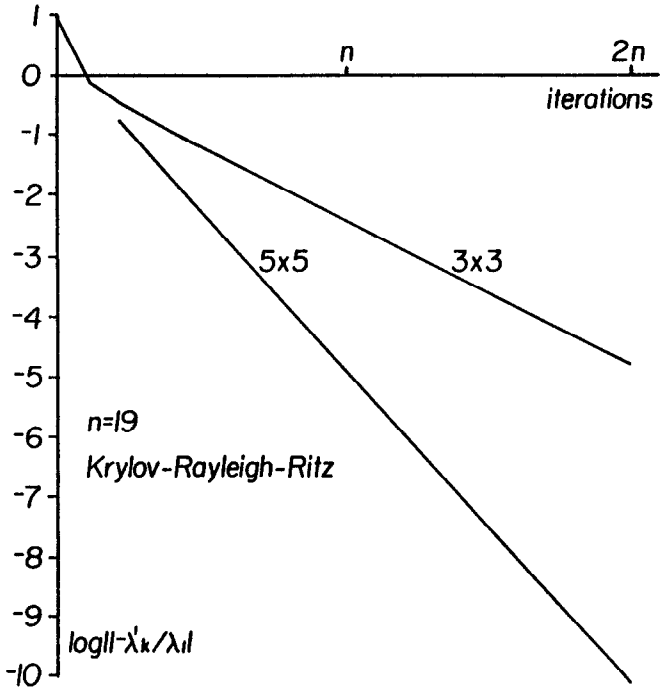


Fig. 8.4

The number of iterations in Fig. 8.4 is for the 3×3 scheme. Each step of the 5×5 scheme is counted as 5/3 iterations to account for the extra number of matrix vector multiplications at each step.

As iteration progresses, vectors $x_0, x_1 = Ax_0, \ldots, x_{m-1} = A^{m-1}x_0$ that form the Krylov search subspace become nearly linearly dependent, and the right-hand matrix in eq. (8.104) becomes nearly singular. Great care must be exercised and high accuracy employed in the solution of the $m\times m$ generalized eigenproblem $\det(A'-\lambda B') = 0$, particularly with methods that assume B' positive definite. Not stopping the iteration and proceeding with a practically singular B' causes loss of numerical stability and jumps between eigenvectors.

Use of large Krylov subspaces may also cause the right-hand matrix B' to become nearly singular with similar effects. In our example, 9 appears to be the highest possible number of Krylov vectors. The method of conjugate gradients is optimal in the quadratic case and appears to be nearly so here as well.

8.8 Higher eigenvalues

Suppose that the power method or the conjugate gradient algorithm has been applied to $Ax = \lambda Bx$ to compute the lowest eigenvalue λ_1 with its corresponding eigenvector v_1. A second application of one of these algorithms with an initial guess x_0 such that $x_0^T Bv_1 = 0$, $x_0^T Bv_2 \neq 0$, should assure in principle convergence to the next eigenvector v_2.

To clear an arbitrary x_0' of its v_1 component we write the projection matrix

$$P = I - v_1 v_1^T B \ , \ v_1^T Bv_1 = 1 \tag{8.106}$$

and have with it that for any x_0'

$$x_0 = Px_0' = x_0' - (v_1^T Bx_0')v_1 \tag{8.107}$$

is B-orthogonal to x_0. Indeed,

$$v_1^T Bx_0 = v_1^T Bx_0' - (v_1^T Bv_1)(v_1^T Bx_0') = 0 \tag{8.108}$$

since $v_1^T Bv_1 = 1$.

In reality v_1 is known only approximately, and also round-off errors intervene to spoil the orthogonality condition $x_k^T B v_1 = 0$. It is advisable, therefore, to project x_k periodically during iteration.

Once v_2 is also available, the search for the next eigenvector starts with x_0 such that $x_0^T B v_1 = x_0^T B v_2 = 0$, which is achieved with

$$x_0 = x_0' - (v_1^T B x_0')v_1 - (v_2^T B x_0')v_2, \;\; v_1^T B v_1 = v_2^T B v_2 = 1 \tag{8.109}$$

and so on.

After the approximations $v_1', v_2', \ldots, v_m'$ to the first m eigenvectors are computed, a better Ritz approximation to λ_1 is extracted from

$$x = \alpha_1 v_1' + \alpha_2 v_2' + \cdots + \alpha_m v_m' \tag{8.110}$$

by the minimizing of $\lambda(x) = x^T Ax / x^T Bx$ over $\alpha_1, \alpha_2, \ldots, \alpha_m$, a minimization that entails the solution of an $m \times m$ eigenproblem.

The decisive advantage of iterative methods based on the minimization of Rayleigh's quotient is that neither A nor B are needed explicitly in these algorithms, an advantage that can result in considerable space and overhead savings. These savings come, however, at the expense of time and efficiency. Once λ_1 is computed all the information gained in the computation is discarded, and computation of λ_2 starts with a clean slate.

exercises

8.8.1. Compute the second eigenvalue of

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

8.8.2. Let $x_1' = (1 - \frac{1}{2}\epsilon^2)v_1 + \epsilon v_2$, where $|\epsilon| << 1$ and where v_1 and v_2 are two orthonormal eigenvectors of $A = A^T$. Show that if $x_2' = \alpha_1 v_1 + \alpha_2 v_2, \alpha_1^2 + \alpha_2^2 = 1$, is orthogonal to x_1', then

$$\lambda_2' = x_2'^T A x_2' = \lambda_2 + \epsilon^2(\lambda_1 - \lambda_2).$$

8.9 Hessenberg-Givens tridiagonalization

In the next three sections we shall discuss ways to bring a symmetric matrix in a finite number of similarity transformations to tridiagonal form. Then we shall discuss eigenvalue solution methods for such matrices.

Theorem 6.13 guarantees that any square matrix can be brought to an almost upper-triangular, Hessenberg form in a finite sequence of elementary similarity transformations. For a symmetric matrix, symmetric similarity operations reduce the matrix to tridiagonal form.

Theorem (Hessenberg-Givens) 8.5. *Any symmetric matrix can be brought to tridiagonal form by means of a finite sequence of elementary similarity transformations.*

Proof. The orthogonal similarity transformation $A' = QAQ^T$ maintains symmetry. Different choices of Q produce different tridiagonalization algorithms. In the Givens version, Jacobi rotation matrix $Q = J_{kl}$ is used to annul entry $(k-1, l)$. Shown below is the sequence of rotations designed to put zeros in the first row of matrix A

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ 2 \\ 3 \\ \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} & & 2 & 3 & \end{array} \\ \begin{bmatrix} \times & \times & 0 & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ 2 \\ \\ 4 \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} & & 2 & & 4 \end{array} \\ \begin{bmatrix} \times & \times & 0 & 0 & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \end{array}$$

$$\rightarrow \begin{array}{c} \\ 2 \\ \\ \\ 5 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} & 2 & & & 5 \end{array} \\ \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \end{array}. \tag{8.111}$$

The first rotation replaces rows and columns 2 and 3 by their linear combination, and A_{13} is annulled by

$$cA_{13} - sA_{12} = 0, \quad c^2 + s^2 = 1 \tag{8.112}$$

or

$$c = A_{12}(A_{12}^2 + A_{13}^2)^{-\frac{1}{2}},\ s = A_{13}(A_{12}^2 + A_{13}^2)^{-\frac{1}{2}}. \tag{8.113}$$

Subsequent entries in the first row are annulled by moving the second column to the right and the second row down. In this way the created zeros remain unaffected.

Having cleared the entries in the first row we descend to the row below it and set out to annul entry (2,4) with a (3,4) rotation

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ \\ 3 \\ 4 \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} & & 3 & 4 & \end{array} \\ \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \end{array} \tag{8.114}$$

Because the new 3rd and 4th columns are linear combinations of the old 3rd and 4th columns, the zeroes in the first row add up again to zero, and the similarity transformations continue row by row to finally create a tridiagonal matrix. End of proof.

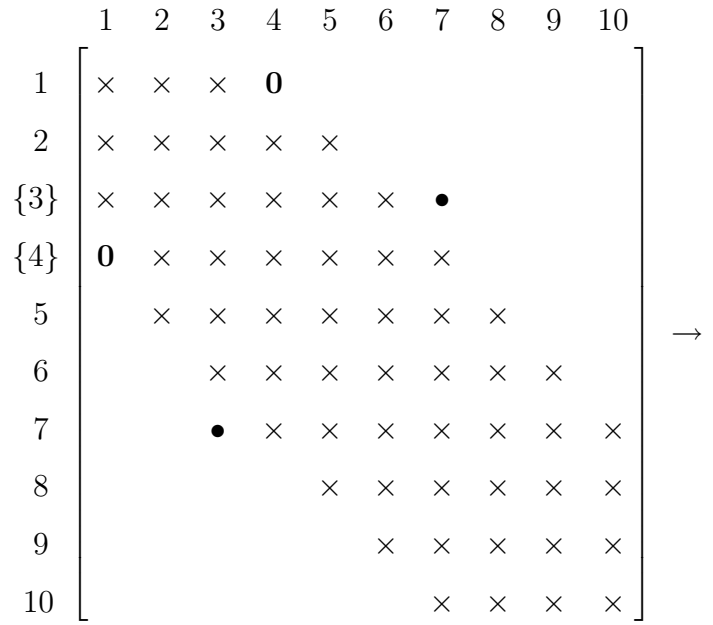
To estimate the work involved in the Givens tridiagonalization assume that $A = A(n \times n)$ is large enough for $n-1$ to be confounded with n. Treatment of the first row requires about n rotations, each requiring $4n$ operations, altogether $4n^2$ operations. The second row requires only $n-1$ rotations with $4(n-1)$ operations per rotation, and so on, so that the total work is nearly $\frac{4}{3}n^3$ operations.

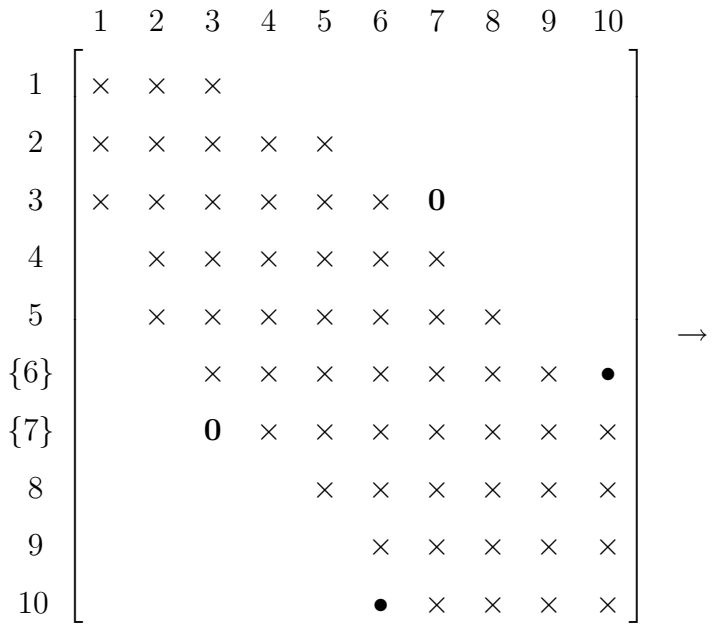
As in the Jacobi method, also here $\sum A_{ij}^2$ remains constant throughout the tridiagonalization process and consequently the size of all matrix elements remains bounded; the method is stable.

It is worth noting that the Givens tridiagonalization method can be adapted to the reduction of a band matrix with no creation of nonzero entries outside the band.

The adaptation is due to Rutishouser and is schematically described below on a 10×10

matrix of bandwidth 7.





$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ \{9\} \\ \{10\} \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\left[\begin{array}{cccccccccc}
\times & \times & \times & & & & & & & \\
\times & \times & \times & \times & \times & & & & & \\
\times & \times & \times & \times & \times & \times & & & & \\
 & \times & \times & \times & \times & \times & \times & & & \\
 & \times & \times & \times & \times & \times & \times & \times & & \\
 & & \times & \times & \times & \times & \times & \times & \times & \mathbf{0} \\
 & & & \times & \times & \times & \times & \times & \times & \times \\
 & & & & \times & \times & \times & \times & \times & \times \\
 & & & & & \times & \times & \times & \times & \times \\
 & & & & & \mathbf{0} & \times & \times & \times & \times
\end{array}\right]
\end{array}
\rightarrow
$$

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \{4\} \\ \{5\} \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\
\left[\begin{array}{cccccccccc}
\times & \times & \times & & & & & & & \\
\times & \times & \times & \times & \mathbf{0} & & & & & \\
\times & \times & \times & \times & \times & \times & & & & \\
 & \times & \times & \times & \times & \times & \times & \bullet & & \\
 & \mathbf{0} & \times & \times & \times & \times & \times & \times & & \\
 & & \times & \times & \times & \times & \times & \times & \times & \\
 & & & \times & \times & \times & \times & \times & \times & \times \\
 & & & \bullet & \times & \times & \times & \times & \times & \times \\
 & & & & & \times & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times
\end{array}\right]
\end{array}
. \qquad (8.115)
$$

Entry (1,4) is the first targeted for annulment with a linear combination of rows and columns 3 and 4. Removal of entry (1,4) creates, however, a new nonzero entry at (3,7), which is in turn annulled by a linear combination of rows and columns 6 and 7. This latest annulment creates a nonzero entry at (6,10) which is finally set to zero with a linear combination of rows and columns 9 and 10. We return now up the diagonal to entry (2,5) and chase it down as before.

For a matrix of dimension n and half bandwidth $k,\ 2 < k < n$, Rutishouser's reduction requires some $4n^2(k-2)$ arithmetical operations.

8.10 Householder tridiagonalization

Whereas in the Givens and Jacobi methods one entry is annulled at a time, the Householder method combines rotations to clear at once all the desired entries in one row. The result is a more efficient algorithm for the tridiagonalization of a symmetric matrix in a finite number of similarity transformations.

Householder's similarity transformation is done with the symmetric orthogonal reflection matrix

$$Q = I - 2uu^T,\ u^T u = 1 \tag{8.116}$$

for which we verify that $Q(u) = Q(-u)$, $Q^2 = I$ and $Qu = -u$.

As in Sec. 5.8, also here we are essentially given the two vectors a and b, $a \neq b$, with $a^T a = b^T b$, and are required to find u in Q so that b is the reflection of $a, b = Qa$. To see that unit vector

$$u = \frac{1}{\|a-b\|}(a-b),\ \|a-b\| = \sqrt{2(a^T a - a^T b)} \tag{8.117}$$

does it look at Fig. 8.5 and at

$$\begin{aligned} Qa = a - 2u(a^T u) &= a - 2\frac{(a^T a - a^T b)}{\|a-b\|^2}(a-b) \\ &= a - (a-b) = b. \end{aligned} \tag{8.118}$$

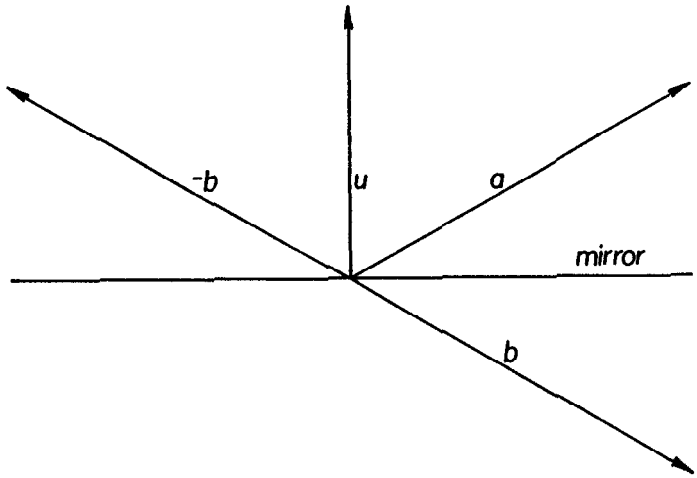


Fig. 8.5

As a first step in the tridiagonalization we seek Q to achieve

$$Q\begin{bmatrix} A_{11} & A_{12} & \times & \times \\ A_{12} & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} Q = \begin{bmatrix} C_{11} & C_{21} & 0 & 0 \\ C_{21} & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \tag{8.119}$$

with unit vector u having a zero first component, $u = [0\ u_2\ u_3\ \ldots\ u_n]^T$. Because of the first zero in u, matrix $B = QA$ is with an unchanged first row, and matrix $C = BQ$ with an unchanged first column:

$$B = QA = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix},\ C = BQ = QAQ = \begin{bmatrix} A_{11} & B_{21} & B_{31} \\ B_{21} & B_{22} & B_{32} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \tag{8.120}$$

since C is symmetric. Hence we need Q to perform the transformation

$$Q\begin{bmatrix} A_{11} \\ A_{12} \\ \vdots \\ A_{1n} \end{bmatrix} = \begin{bmatrix} A_{11} \\ B_{21} \\ 0 \\ 0 \end{bmatrix} \tag{8.121}$$

for

$$A_{11}^2 + A_{12}^2 + \ldots + A_{1n}^2 = A_{11}^2 + B_{21}^2 \tag{8.122}$$

or

$$B_{21} = \pm\gamma,\ \gamma = (A_{12}^2 + A_{13}^2 + \ldots + A_{1n}^2)^{1/2}. \tag{8.123}$$

According to the opening arguments of this section

$$u = \beta\begin{bmatrix} 0 \\ A_{12} - B_{21} \\ A_{13} \\ \vdots \\ A_{1n} \end{bmatrix},\ \beta = ((A_{12} - B_{21})^2 + A_{13}^2 + \ldots + A_{1n}^2)^{-1/2} \tag{8.124}$$

and we make the sign choice

$$B_{21} = -sign\ (A_{12})\gamma. \tag{8.125}$$

Some algebra leads to

$$\beta = \frac{\sqrt{2}}{2\gamma}(1 + |A_{12}|/\gamma)^{-\frac{1}{2}} \tag{8.126}$$

and this is all we need to create

$$Q^T A Q = \begin{bmatrix} \times & \times & o^T \\ \times & & \\ & A' & \\ o & & \end{bmatrix}. \tag{8.127}$$

To continue the tridiagonalization we write

$$Q = \begin{bmatrix} 1 & o^T \\ o & Q' \end{bmatrix} \tag{8.128}$$

and have that

$$\begin{bmatrix} 1 & o^T \\ o & Q' \end{bmatrix} \begin{bmatrix} A_{11} & a_1'^T \\ a_1' & A' \end{bmatrix} \begin{bmatrix} 1 & o^T \\ o & Q' \end{bmatrix} \begin{bmatrix} A_{11} & a_1'^T \\ a_1' & Q'A'Q' \end{bmatrix} \tag{8.129}$$

since u' in $Q' = I - 2u'u'^T$ is orthogonal to $a_1' = [\times\ 0\ 0\ 0]^T$ by virtue of its null first component. The procedure is now repeated on the smaller A' and so on until tridiangulization is completed.

The arithmetical work involved with the Householder method is about $2n^3/3$ operations, better than with the Givens method.

8.11 Lanczos tridiagonalization

The Lanczos method can be thought of as the Ritz method with trial vectors being the residuals generated by the conjugate gradient algorithm.

We recall from Chapter 7 that for a positive definite symmetric matrix A the conjugate gradient algorithm produces the sequence of vectors $r_0, r_1, \ldots, r_{m-1}$ that satisfy the orthogonality and A-orthogonality conditions $r_i^T r_j = 0$ if $i \neq j$, and $r_i^T A r_j = 0$ if $|i-j| \geq 2$. Hence with the trial vector

$$x = \xi_1 r_0 + \xi_2 r_1 + \cdots + \xi_m r_{m-1} \tag{8.130}$$

the Ritz method results in the $m \times m$ eigenproblem

$$\begin{bmatrix} r_0^T A r_0 & r_0^T A r_1 & & & & \\ r_1^T A r_0 & r_1^T A r_1 & r_1^T A r_2 & & & \\ & r_2^T A r_1 & r_2^T A r_2 & r_2^T A r_3 & & \\ & & & \ddots & & \\ & & r_{m-2}^T A r_{m-3} & r_{m-2}^T A r_{m-2} & r_{m-2}^T A r_{m-1} \\ & & & r_{m-1}^T A r_{m-2} & r_{m-1}^T A r_{m-1} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \\ \xi_{m-1} \\ \xi_m \end{bmatrix}$$

$$= \lambda \begin{bmatrix} r_0^T r_0 & & & & \\ & r_1^T r_1 & & & \\ & & r_3^T r_3 & & \\ & & & \ddots & \\ & & & r_{m-2}^T r_{m-2} & \\ & & & & r_{m-1}^T r_{m-1} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \\ \xi_{m-1} \\ \xi_m \end{bmatrix}. \tag{8.131}$$

Using the conjugate gradient relationships

$$\alpha_j = \frac{r_j^T r_j}{p_j^T A p_j},\ \beta_j = \frac{r_{j+1}^T r_{j+1}}{r_j^T r_j},\ p_j = r_j + \beta_{j-1} p_{j-1},\ \beta_j = \frac{r_{j+1}^T r_{j+1}}{r_j^T r_j}, p_i^T A p_j = 0,\ i \neq j \tag{8.132}$$

and the transformation $\xi_j = (r_{j-1}^T r_{j-1})^{-\frac{1}{2}} \eta_j$ we reduce the eigenproblem to

$$\begin{bmatrix} 1/\alpha_0 & -\beta_0^{1/2}/\alpha_0 & & & \\ -\beta_0^{1/2}/\alpha_0 & 1/\alpha_1 + \beta_0/\alpha_0 & -\beta_1^{1/2}/\alpha_1 & & \\ & -\beta_1^{1/2}/\alpha_1 & 1/\alpha_2 + \beta_1/\alpha_1 & -\beta_2^{1/2}/\alpha_2 & \\ & & \ddots & & \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \\ \\ \eta_m \end{bmatrix} = \lambda \begin{bmatrix} \eta_1 \\ \eta_2 \\ \\ \\ \eta_m \end{bmatrix}. \tag{8.133}$$

If r_0 is confined to a subspace spanned by m eigenvectors of A, then the conjugate gradient algorithm generates no more than m nonzero r vectors and the Lanczos method yields only part of the spectrum of A. A random r_0 raises the expectation for a complete tridiagonal Lanczos eigenproblem including all eigenvalues of A but without their multiplicities.

The appeal of the Lanczos tridiagonalization method is considerable for the large sparse eigenproblems of finite differences and finite elements since it does not require A itself but only Ap_k. Unfortunately, what looks so promising theoretically is flawed in practice. The algorithm is sensitive to round-off errors that spoil the orthogonality and A-orthogonality of the r vectors. Matrices that are in theory tridiagonal and diagonal are not so in practice.

Each approximate eigenvalue of the tridiagonal matrix comes with the m components $\xi_1, \xi_2, \ldots, \xi_m$ of the Lanczos-Ritz eigenvector and the approximate eigenvector may be reconstructed from $x = \xi_1 r_0 + \xi_2 r_1 + \cdots + \xi_m r_{m-1}$. If storage of all $r_0, r_1, \ldots, r_{m-1}$ is undesirable, then the eigenvector may be found by inverse iteration.

A numerical example should clearly reveal the ill effects of round-off errors. We undertake to apply the Lanczos method to the computation of all eigenvalues of the diagonal matrix $D_{ii} = i^2,\ i = 1, 2, \ldots, 20$, with an initial $r_0 = [\epsilon\ 1\ \ldots\ 1]^T$ for different ϵ values.

Table 8.1 below lists the result of this computation done in single ($u = 0.5\ 10^{-6}$) precision. The eigenvalues computed with $\epsilon = 1$ are inaccurate and the highest eigenvalue $\lambda = 400$ seems to repeat. With $\epsilon = 0.1$ the lowest computed eigenvalue is closer to 4 than to 1 and the highest eigenvalue still repeats. It appears as though the algorithm cannot make up its mind as to whether r_0 is in a 19-or a 20-dimensional vector space. Matters are settled, in this respect, with $\epsilon = 0.01$. The smallest eigenvalue $\lambda = 1$ is out and the higher eigenvalues are computed with greater accuracy than for $\epsilon = 1$, except that the computed highest eigenvalue repeats.

Table 8.1: Accuracy of eigenvalue computation, single precision, Lanczos method for the 20×20 diagonal $D_{ii} = i^2$ and $r_0 = [\epsilon\ 1\ 1\ \ldots\ 1]^T$.

	$\epsilon = 1$	eigenvalues	
		computed	exact
k	$r_k^T r_k$		
1	$0.2000E+02$	$0.3999873E+03$	400
2	$0.1509E+02$	$0.3999834E+03$	361
3	$0.1391E+02$	$0.3609873E+03$	324
4	$0.1286E+02$	$0.3239895E+03$	289
5	$0.1185E+02$	$0.2889873E+03$	256
6	$0.1086E+02$	$0.2559999E+03$	225
7	$0.9882E+01$	$0.2250003E+03$	196
8	$0.8925E+01$	$0.1960012E+03$	169
9	$0.7988E+01$	$0.1690070E+03$	144
10	$0.7077E+01$	$0.1440390E+03$	121
11	$0.6193E+01$	$0.1211608E+03$	100
12	$0.5340E+01$	$0.1004460E+03$	81
13	$0.4523E+01$	$0.8173644E+02$	64
14	$0.3746E+01$	$0.6448241E+02$	49
15	$0.3014E+01$	$0.4838033E+02$	36
16	$0.2350E+01$	$0.3379993E+02$	25
17	$0.2402E+01$	$0.2135901E+02$	16
18	$0.3017E+01$	$0.1157512E+02$	9
19	$0.1129E+01$	$0.4785712E+01$	4
20	$0.6548E+00$	$0.1071318E+01$	1

	$\epsilon = 0.1$	eigenvalues	
		computed	exact
k	$r_k^T r_k$		
1	$0.1901E+02$	$0.3999861E+03$	400
2	$0.1272E+02$	$0.3988076E+03$	361
3	$0.1012E+02$	$0.3609885E+03$	324
4	$0.7972E+01$	$0.3239873E+03$	289
5	$0.6180E+01$	$0.2889863E+03$	256
6	$0.4709E+01$	$0.2559994E+03$	225
7	$0.3523E+01$	$0.2249996E+03$	196
8	$0.2585E+01$	$0.1960012E+03$	169
9	$0.1857E+01$	$0.1690108E+03$	144
10	$0.1303E+01$	$0.1440694E+03$	121
11	$0.8900E+00$	$0.1213227E+03$	100
12	$0.5894E+00$	$0.1010604E+03$	81
13	$0.3763E+00$	$0.8319557E+02$	64
14	$0.2300E+00$	$0.6655870E+02$	49
15	$0.1334E+00$	$0.5050694E+02$	36
16	$0.7271E-01$	$0.3574448E+02$	25
17	$0.3726E-01$	$0.2313942E+02$	16
18	$0.2890E-01$	$0.1334122E+02$	9
19	$0.2988E-01$	$0.6970556E+01$	4
20	$0.9669E-02$	$0.3607424E+01$	1

	$\epsilon = 0.01$	eigenvalues	
		computed	exact
k	$r_k^T r_k$		
1	$0.1900E+02$	$0.3999858E+03$	400
2	$0.1269E+02$	$0.3999766E+03$	361
3	$0.1008E+02$	$0.3609863E+03$	324
4	$0.7929E+01$	$0.3239858E+03$	289
5	$0.6132E+01$	$0.2889844E+03$	256
6	$0.4658E+01$	$0.2559991E+03$	225
7	$0.3473E+01$	$0.2249997E+03$	196
8	$0.2537E+01$	$0.1960002E+03$	169
9	$0.1812E+01$	$0.1690032E+03$	144
10	$0.1262E+01$	$0.1440241E+03$	121
11	$0.8538E+00$	$0.1211228E+03$	100
12	$0.5578E+00$	$0.1004622E+03$	81
13	$0.3492E+00$	$0.8225214E+02$	64
14	$0.2072E+00$	$0.6623778E+02$	49
15	$0.1145E+00$	$0.5153043E+02$	36
16	$0.5768E-01$	$0.3791283E+02$	25
17	$0.3010E-01$	$0.2599255E+02$	16
18	$0.4244E-01$	$0.1631502E+02$	9
19	$0.7678E-02$	$0.9040213E+01$	4
20	$0.1657E-02$	$0.3998635E+01$	1

Various suggestions have been made to ward-off the harm of round-off errors by repeated reorthogonalizations during the conjugate gradient generation of the r vectors, but we shall not deal with these highly technical issues here.

Increasing the accuracy of the computation to double ($u = 10^{-16}$) precision dramatically improves the accuracy of the computed eigenvalues as seen in Table 8.2 below. The table shows also the convergence of the eigenvalues with more residuals and larger tridiagonal matrices. Convergence of the lowest eigenvalue is at best linear and to have accurate eigenvalues the Lanczos tridiagonalization must be carried out in high precision to completion.

Table 8.2: Accuracy and convergence of eigenvalue computation, double precision Lanczos method for the 20×20 diagonal matrix $D_{ii} = i^2$ and $r_0 = [1\ 1\ \ldots\ 1]^T$.

k	$r_k^T r_k$
1	0.2000D+02
2	0.1509D+02
3	0.1391D+02
4	0.1286D+02
5	0.1185D+02
6	0.1086D+02
7	0.9882D+01
8	0.8925D+01
9	0.7988D+01
10	0.7077D+01
11	0.6193D+01
12	0.5340D+01
13	0.4523D+01
14	0.3746D+01
15	0.3013D+01
16	0.2329D+01
17	0.1702D+01
18	0.1140D+01
19	0.6545D+00
20	0.2629D+00

	eigenvalues	
	computed	exact
$k = 2$		
	0.31167138D+03	4
	0.51088020D+02	1
$k = 3$		
	0.36376036D+03	9
	0.18355971D+03	4
	0.25777375D+02	1
$k = 4$		
	0.38414872D+03	16
	0.26379530D+03	9
	0.11571509D+03	4
	0.15583254D+02	1
$k = 5$		
	0.39323176D+03	5
	0.30836236D+03	6
	0.19060881D+03	9
	0.78409809D+02	4
	0.10490873D+02	1
$k = 6$		
	0.39736734D+03	36
	0.33358566D+03	25
	0.24194824D+03	16
	0.14129117D+03	9
	0.56184482D+02	4
	0.75818773D+01	1

	eigenvalues	
	computed	exact
$k = 7$		
	0.39912370D+03	49
	0.34783047D+03	36
	0.27582469D+03	25
	0.19071109D+03	16
	0.10767313D+03	9
	0.41996569D+02	4
	0.57614296D+01	1
$k = 8$		
	0.39976381D+03	64
	0.35546709D+03	49
	0.29769979D+03	36
	0.22690807D+03	25
	0.15215764D+03	16
	0.84079887D+02	9
	0.32424647D+02	4
	0.45445367D+01	1
$k = 9$		
	0.39995050D+03	81
	0.35909351D+03	64
	0.31121668D+03	49
	0.25243579D+03	36
	0.18727720D+03	25
	0.12301475D+03	16
	0.67009092D+02	9
	0.25674683D+02	4
	0.36895455D+01	1

$k = 10$

0.39999212D+03	10
0.36049685D+03	81
0.31880956D+03	64
0.26964229D+03	49
0.21364657D+03	36
0.15551041D+03	25
0.10067814D+03	16
0.54306165D+02	9
0.20740451D+02	4
0.30650171D+01	1

$k = 11$

0.39999906D+03	121
0.36090388D+03	100
0.32237989D+03	81
0.28032651D+03	64
0.23247770D+03	49
0.18105909D+03	36
0.12997995D+03	25
0.83274046D+02	16
0.44614759D+02	9
0.17024657D+02	4
0.25944401D+01	1

$k = 12$

0.39999992D+03	144
0.36098713D+03	121
0.32364396D+03	100
0.28599766D+03	81
0.24490462D+03	64
0.20004542D+03	49
0.15381235D+03	36
0.10929081D+03	25
0.69484427D+02	16
0.37055771D+02	9
0.14155463D+02	4
0.22308079D+01	1

$k = 13$

0.39999999D+03	169
0.36099881D+03	144
0.32394839D+03	121
0.28828388D+03	100

$k = 14$

0.40000000D+03	196
0.36099993D+03	169
0.32399522D+03	144
0.28889380D+03	121
0.25502211D+03	100
0.22067116D+03	81
0.18467767D+03	64
0.14777217D+03	49
0.11168693D+03	36
0.78288689D+02	25
0.49294273D+02	16
0.26171560D+02	9
0.10072202D+02	4
0.17138312D+01	1

$k = 15$

0.40000000D+03	225
0.36100000D+03	196
0.32399972D+03	169
0.28899076D+03	144
0.25586224D+03	121
0.22400306D+03	100
0.19212310D+03	81
0.15948389D+03	64
0.12673615D+03	49
0.95249901D+02	36
0.66479325D+02	25
0.41744305D+02	16
0.22158727D+02	9
0.85846394D+01	4
0.15267430D+01	1

$k = 16$

0.40000000D+03	256
0.36100000D+03	225
0.32399999D+03	196
0.28899955D+03	169
0.25598972D+03	144
0.22487893D+03	121
0.19522484D+03	100
0.16612038D+03	81
0.13700021D+03	64

0.25197292D+03	81
0.21303963D+03	64
0.17200238D+03	49
0.13096955D+03	36
0.92342460D+02	25
0.58379433D+02	16
0.31042375D+02	9
0.11891875D+02	4
0.19439551D+01	1

$k = 17$

0.40000000D+03	289
0.36100000D+03	256
0.32400000D+03	225
0.28899999D+03	196
0.25599962D+03	169
0.22499320D+03	144
0.19592917D+03	121
0.16856042D+03	100
0.14231867D+03	81
0.11671349D+03	64
0.91903912D+02	49
0.68597380D+02	36
0.47655146D+02	25
0.29877015D+02	16
0.15929131D+02	9
0.63098053D+01	4
0.12465586D+01	1

$k = 18$

0.40000000D+03	324
0.36100000D+03	289
0.32400000D+03	256
0.28900000D+03	225
0.25599999D+03	196
0.22499984D+03	169
0.19599755D+03	144
0.16897560D+03	121
0.14384218D+03	100
0.12033103D+03	81
0.98075664D+02	64
0.10830110D+03	49
0.81059443D+02	36
0.56406256D+02	25
0.35366551D+02	16
0.18796790D+02	9
0.73497662D+01	4
0.13731803D+01	1

$k = 19$

0.40000000D+03	361
0.36100000D+03	324
0.32400000D+03	289
0.28900000D+03	256
0.25600000D+03	225
0.22500000D+03	196
0.19599998D+03	169
0.16899966D+03	144
0.14399652D+03	121
0.12097527D+03	100
0.99876175D+02	81
0.80559946D+02	64
0.62866601D+02	49
0.46791046D+02	36
0.32568373D+02	25
0.20573037D+02	16
0.11177703D+02	9
0.46555721D+01	4
0.10594370D+01	1

$k = 20$

0.40000000D+03	400
0.36100000D+03	361
0.32400000D+03	324
0.28900000D+03	289
0.25600000D+03	256
0.22500000D+03	225
0.19600000D+03	196
0.16900000D+03	169
0.14400000D+03	144
0.12100000D+03	121

0.76974015D+02	49	0.10000000D+03	100
0.57369340D+02	36	0.81000000D+02	81
0.39852433D+02	25	0.64000000D+02	64
0.25032625D+02	16	0.49000000D+02	49
0.13429174D+02	9	0.36000000D+02	36
0.54220041D+01	4	0.25000000D+02	25
0.11425974D+01	1	0.16000000D+02	16
		0.90000000D+01	9
		0.40000000D+01	4
		0.10000000D+01	1

Application of the Lanczos method to $Ax = \lambda Bx$ requires the LL^T factorization, detracting more from its appeal.

8.12 Sturm's theory for tridiagonals

Givens, Householder and Lanczos make it amply clear how important the tridiagonal eigenproblem is.

Let $T = T(m \times m)$ be a symmetric, irreducible tridiagonal matrix, and write

$$p_m(\lambda) = \det(T - \lambda I) = \det \begin{bmatrix} a_1 - \lambda & b_2 & & & & & \\ b_2 & a_2 - \lambda & b_3 & & & & \\ & b_3 & a_3 - \lambda & b_4 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & b_{m-1} & a_{m-1} - \lambda & b_m & \\ & & & & b_m & a_m - \lambda \end{bmatrix}. \quad (8.134)$$

Symmetric elementary row and column operations reduce the computation of $\det(T - \lambda I)$ to that of

$$\det \begin{bmatrix} p_1/p_0 & & & & & & \\ & p_2/p_1 & & & & & \\ & & p_3/p_2 & & & & \\ & & & \ddots & & & \\ & & & & p_{m-1}/p_{m-2} & b_m & \\ & & & & b_m & a_m - \lambda \end{bmatrix}, \; p_0 = 1 \quad (8.135)$$

from which we obtain the recursive formula

$$p_m = (a_m - \lambda)p_{m-1} - b_m^2 p_{m-2}, \; p_0 = 1, \; p_1 = a_1 - \lambda. \quad (8.136)$$

Example. if

$$T = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix} \tag{8.137}$$

then

$$p_m = (2-\lambda)p_{m-1} - p_{m-2} \tag{8.138}$$

and

$$p_0 = 1,\ p_1 = 2-\lambda,\ p_2 = (2-\lambda)^2 - 1,\ p_3 = (2-\lambda)\left((2-\lambda)^2 - 2\right) \tag{8.139}$$

$$p_4 = (2-\lambda)^4 - 3(2-\lambda) - 1.$$

Characteristic polynomials p_m are important for the following interlace

Theorem 8.6. *If symmetric tridiagonal $T = T(n \times n)$ is irreducible, $b_j^2 > 0$ for all j, then the roots of characteristic equation $p_n = 0$ are distinct, and between any two roots of p_n there is a root of p_{n-1}.*

Proof. By induction. The first $p_1 = a_1 - \lambda$ has a single root $\lambda = a_1$. For $p_2 = (a_1 - \lambda)(a_2 - \lambda) - b_2^2$ we have that $p_2(a_1) = -b_2^2 < 0$, $p_2(-\infty) > 0,\ p_2(+\infty) > 0$, and hence $p_2 = 0$ has two distinct roots, with the root of $p_1 = 0$ strictly between them as in figure 8.6.

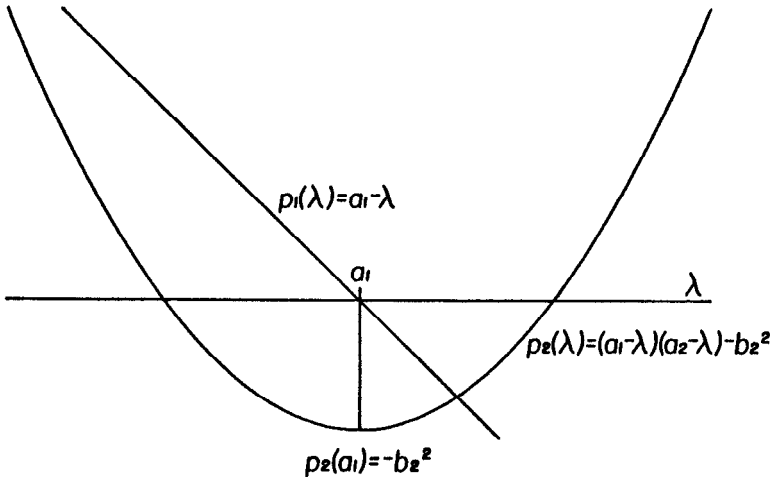


Fig. 8.6

Next assume that p_{m-1} has $m-1$ distinct roots, and that between each pair there is a root of p_{m-2}. Let λ_1 and λ_2 be two consecutive roots of p_{m-1}. Then

$$p_m(\lambda_1) = -b_m^2 p_{m-2}(\lambda_1),\ p_m(\lambda_2) = -b_m^2 p_{m-2}(\lambda_2) \tag{8.140}$$

and since p_{m-2} changes sign in the interval $\lambda_1 \leq \lambda \leq \lambda_2$, then so does p_m, and between two consecutive roots of p_{m-1} there is a root of p_m. We verify that $p_m(-\infty) > 0$ for any m, and hence there is a real root of p_m less than the smallest root of p_{m-1}. A similar argument shows that p_m has one root larger than the last root of p_{m-1}. See figure 8.7. By induction the proof is done.

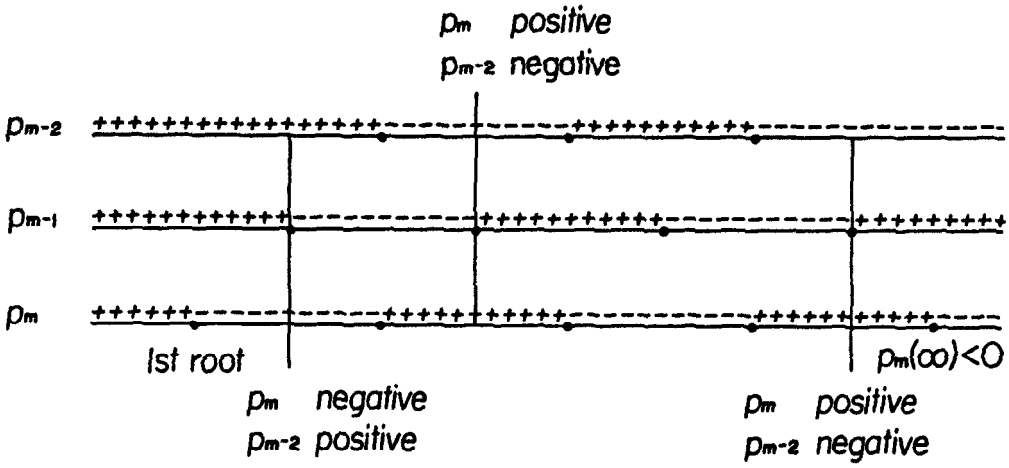


Fig. 8.7

The root stack of $p_0, p_1, p_2, p_3, p_4, p_5$ is shown in figure 8.8.

Polynomials $p_0, p_0, \ldots, p_n$ are seen to possess the following three preponderant properties:

1. $p_0(\lambda) \neq 0$,

2. when $p_m = 0,\ p_{m-1}$ and p_{m+1} are both nonzero and of opposite sign,

3. as λ crosses a zero of p_m, the quotient p_m/p_{m-1} changes from positive to negative.

A sequence of polynomials $p_0, p_1, \ldots, p_n$ that possesses the three properties forms a *Sturm sequence.*

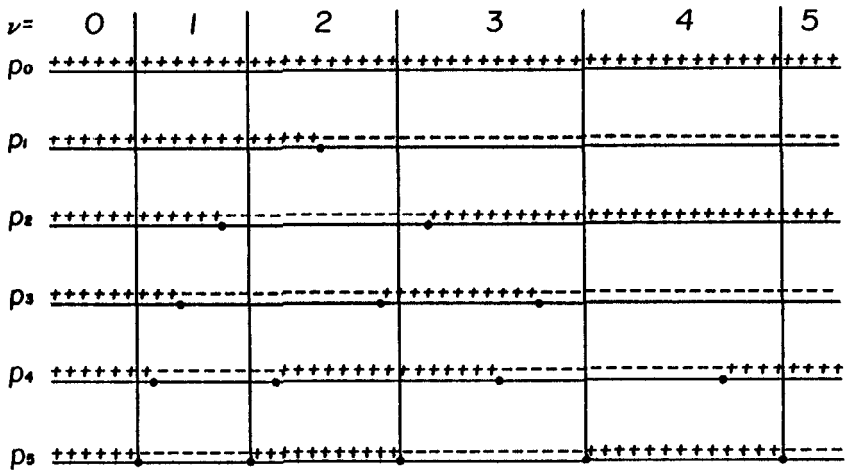


Fig. 8.8

The following theorem provides a useful device for counting the eigenvalues of a tridiagonal matrix in an interval.

Theorem 8.7 *Let $p_0, p_1, \ldots, p_n$ be a Sturm sequence on the interval $\alpha < \lambda < \beta$, $p_n(\alpha) \neq 0, p_n(\beta) \neq 0$, and let $\nu(\lambda)$ be the number of sign changes in the ordered array $p_0(\lambda), p_1(\lambda), \ldots, p_n(\lambda)$. Then the number of zeroes of p_n in the interval $\alpha < \lambda < \beta$ is $\nu(\beta) - \nu(\alpha)$.*

Proof. We know that $p_m(-\infty) > 0$ for any m so that for a sufficiently small $\lambda,\ \nu(\lambda) = 0$. As λ increases it crosses the first root of $p_n = 0$ and $\nu(\lambda)$ jumps from zero to one. Look now at figure 8.9. Further change in the value of $\nu(\lambda)$ can occur only if λ crosses a root of one of the polynomials. But when $p_m(\lambda) = 0$, p_{m-1} and p_{m+1} have opposite signs. The value of ν is not changed when a root of $p_m,\ m < n$ is crossed. Only when a root of $p_n = 0$ is crossed does $\nu(\lambda)$ change. End of proof.

Example. For

$$T - \lambda I = \begin{bmatrix} 2-\lambda & -1 & \\ -1 & 2-\lambda & -1 \\ & -1 & 2-\lambda \end{bmatrix} \tag{8.141}$$

we have that if $\lambda = 2$, then $p_0(\lambda) = 1,\ p_1(\lambda) = 0,\ p_2(\lambda) = -1,\ p_3(\lambda) = 0$ and hence $\lambda = 2$ is an eigenvalue of T. Also, since $\nu(2) = 1$, $\nu(-\infty) = 0,\ \nu(+\infty) = 3$, we conclude that one root of $p_n = 0$ is less than 2, and two roots are larger or equal to 2. See figure 8.9.

Notice that for $\lambda = 2,\ T - \lambda I$ does not admit an LU factorization.

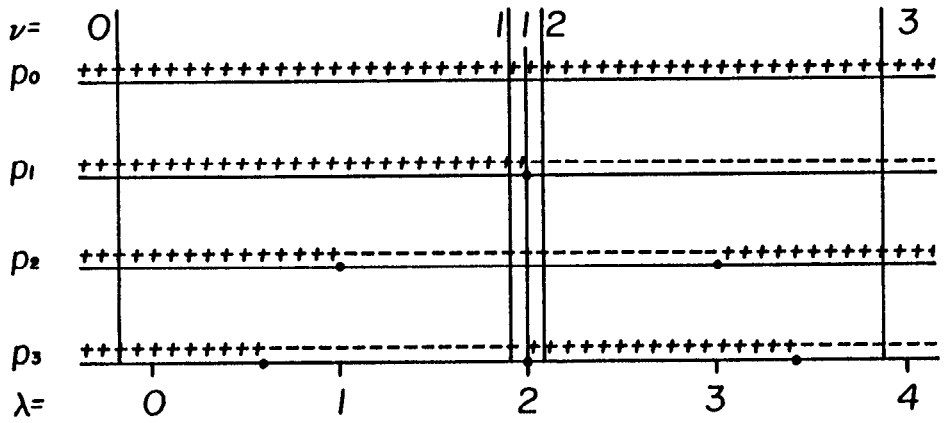


Fig. 8.9

Solution of $p_n(\lambda) = 0$ to find the eigenvalues of the tridiagonal T can be done with any procedure that accurately locates roots of polynomials. Bisection, or the Newton-Raphson methods are good candidates. We recall that $\det(T)$ equals the product of all eigenvalues of T, and hence it is conceivable that $p_n(\lambda)$ may become large to overflow or small to underflow. The difficulty is avoided by replacing $p_m(\lambda)$ by the ratio

$$q_m(\lambda) = p_m(\lambda)/p_{m-1}(\lambda) \quad m = 1, 2, \ldots, n \tag{8.142}$$

so that $\nu(\lambda)$ becomes the number of negative q's. The recurrence formula becomes now

$$q_m = a_m - \lambda - b_m^2/q_{m-1} \quad m = 2, 3, \ldots, n, \quad q_1 = a_1 - \lambda \tag{8.143}$$

and if $q_{m-1}(\lambda)$ is zero it is replaced by a small number.

Sturm's theory is particularly useful in the process of finding few eigenvalues of T in a given interval.

exercises

8.12.1. Matrix

$$A = \begin{bmatrix} 1 & -1 & & & \\ -1 & 4 & -3 & & \\ & -3 & 8 & -5 & \\ & & -5 & 12 & -7 \\ & & & -7 & 16 \end{bmatrix}$$

gives rise to the Sturm sequence

$$p_1 = 1$$
$$p_1 = 1 - \lambda$$
$$p_2 = (4 - \lambda)p_1 - p_0$$
$$p_3 = (8 - \lambda)p_2 - 9p_1$$
$$p_4 = (12 - \lambda)p_3 - 25p_2$$
$$p_5 = (16 - \lambda)p_4 - 49p_3.$$

Use theorem 8.6 to determine that interval $0.39 < \lambda < 0.40$ contains the lowest eigenvalue of A only. Bisect the interval to get tighter enclosures on λ_1.

Use Sturm's theory to locate all eigenvalues of A.

8.12.2. Prove that if λ is an eigenvalue of tridiagonal matrix T of multiplicity m, then at least $m - 1$ subdiagonal entries of T are zero.

8.12.3. For matrix

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

$p_n(\lambda) = (-\lambda)^n + \ldots - a_3(n)\lambda^3 + a_2(n)\lambda^2 - a_1(n)\lambda + a_0(n)$. Show that

$$a_0(n) = 2a_0(n-1) - a_0(n-2)$$
$$a_1(n) = 2a_1(n-1) + a_0(n-1) - a_1(n-2)$$
$$a_2(n) = 2a_2(n-1) + a_1(n-1) - a_2(n-2)$$
$$a_3(n) = 2a_3(n-1) + a_2(n-1) - a_3(n-2).$$

Store

$$-a_3(n)\lambda^3 + a_2(n)\lambda^2 - a_1(n)\lambda + a_0(n) = 0$$

for $n > 3$, and compare the smallest root of this equation with

$$\lambda_1(n) = 4\sin^2 \frac{\pi}{2(n+1)}.$$

8.12.4. Show that if tridiagonal

$$T = \begin{bmatrix} a_1 & b_2 & & \\ c_2 & a_2 & b_3 & \\ & c_3 & a_3 & b_4 \\ & & c_4 & a_4 \end{bmatrix}$$

is such that $b_i/c_i > 0$, then real diagonal D exists so that $D^{-1}TD$ is symmetric and hence with real eigenvalues.

8.12.5. Show that if A and B are symmetric and B is positive definite, then the zeroes of $\det(A_k - \lambda B_k) = 0$ separate those of $\det\,(A_{k+1} - \lambda B_{k+1}) = 0$, where A_k and B_k are leading principal submatrices of order k.

8.12.6. Write

$$A = \begin{bmatrix} a_1 & b_2 & & \\ b_2 & a_2 & b_3 & \\ & b_3 & a_3 & b_4 \\ & & b_4 & a_4 \end{bmatrix}\ ,\ B = \begin{bmatrix} a'_1 & b'_2 & & \\ b'_2 & a'_2 & b'_3 & \\ & b'_3 & a'_3 & b'_4 \\ & & b'_4 & a'_4 \end{bmatrix}$$

and argue that the polynomials

$$p_0(\lambda) = 1\ ,\ p_1(\lambda) = a_1 - \lambda a'_1$$

$$p_k(\lambda) = (a_k - \lambda'_k a'_k)p_{k-1}(\lambda) - (b_k - \lambda b'_k)^2 p_{k-2}(\lambda)$$

form a Sturm sequence.

Apply this to the location of all roots of

$$\det(\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} - \lambda \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & 1 & 4 & 1 & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix}) = 0.$$

8.12.7. Find c and s, $c^2 + s^2 = 1$, so that

$$\begin{bmatrix} c & & s \\ & 1 & \\ -s & & c \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -2 \\ -1 & -2 & 3 \end{bmatrix} \begin{bmatrix} c & & -s \\ & 1 & \\ s & & c \end{bmatrix}$$

is tridiagonal.

8.12.8. Use Sturm's theory to locate all eigenvalues of matrix

$$A = \begin{bmatrix} 3 & 1 & & & & & & \\ 1 & 2 & 1 & & & & & \\ & 1 & 1 & 1 & & & & \\ & & 1 & 0 & 1 & & & \\ & & & 1 & 1 & 1 & & \\ & & & & 1 & 2 & 1 & \\ & & & & & 1 & 3 & \end{bmatrix}.$$

Use inverse iteration to compute the corresponding eigenvectors.

8.13 The Francis-Kublanovskaya QR algorithm

The Francis-Kublanovskaya QR algorithm consists of simultaneous power iteration carried to the extreme of n vectors iterated at once. Let matrix candidate $A = A_0$ for the eigenvalue computations be symmetric. In the QR algorithm the matrix is factored as $A_0 = Q_0 R_0$ with an orthogonal Q_0 and an upper-triangular R_0, and the orthonormal columns of Q_0 are taken as the initial guess for the n eigenvectors of A. Then matrix $A_1 = Q_0^T A_0 Q_0$ is formed and QR factored to yield the next eigenvector approximation Q_1, and the process is repeated to make A_k approach diagonal form.

We shall write the QR algorithm as

$$Q_0^T A_0 = R_0 \ , \quad A_1 = R_0 Q_0 = Q_0^T A_0 Q_0 \tag{8.144}$$

or in a lower-triangular version as

$$Q_0^T A_0 = L_0 \ , \ A_1 = L_0 Q_0 = Q_0^T A_0 \tag{8.145}$$

in which form it is called the QL algorithm.

What renders the QL and QR algorithms so attractive for band matrices is

Theorem 8.8. *If in the QR algorithm $A_0 = A_0^T$ is a band matrix of half bandwidth k, then so is $A_1 = Q_0^T A_0 Q_0$.*

Proof. Consider the schematic formation of A_1

$$
\underset{A_0}{\begin{bmatrix}
\times & \times & \times & & & & \\
\times & \times & \times & \times & & & \\
\times & \times & \times & \times & \times & & \\
 & \times & \times & \times & \times & \times & \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times
\end{bmatrix}}
\underset{R_0^{-1}}{\begin{bmatrix}
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times \\
 & & & & & \times & \times \\
 & & & & & & \times
\end{bmatrix}}
=
\underset{Q_0}{\begin{bmatrix}
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times
\end{bmatrix}}
$$

$$
\underset{R_0}{\begin{bmatrix}
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times \\
 & & & & & \times & \times \\
 & & & & & & \times
\end{bmatrix}}
\underset{Q_0}{\begin{bmatrix}
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times
\end{bmatrix}}
=
$$

$$
\underset{A_1}{\begin{bmatrix}
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times
\end{bmatrix}}
=
\begin{bmatrix}
\times & \times & \times & & & & \\
\times & \times & \times & \times & & & \\
\times & \times & \times & \times & \times & & \\
 & \times & \times & \times & \times & \times & \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times
\end{bmatrix}
\tag{8.146}
$$

Because $A_1 = A_1^T$, A_1 is again a band matrix of the same bandwidth as A_0. End of proof.

Since a symmetric matrix can be tridiagonalized by elementary operations, we are greatly interested in the application of the QR or QL algorithm to tridiagonal matrices. Factorization to have $Q^T A = L$ is done here by a succession of Jacobi rotations

$$
J_1 J_2 \cdots J_{n-1} A = L \tag{8.147}
$$

with

$$J_i = \begin{array}{c} \\ i \\ i+1 \end{array} \begin{array}{c} \begin{array}{cccc} i & & i+1 & \end{array} \\ \begin{bmatrix} 1 & & & \\ & c_i & -s_i & \\ & s_i & c_i & \\ & & & 1 \end{bmatrix} \end{array} \tag{8.148}$$

that combine two neighboring rows as described below.

$$\begin{array}{ccccc} A & & J_4A & & J_3J_4A \\ \begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} & \rightarrow & \begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & 0 \\ & & & \times & \times \end{bmatrix} & \rightarrow & \begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & 0 & \\ & \times & \times & \times & \\ & & \times & \times & \times \end{bmatrix} \end{array}$$

$$\begin{array}{ccccccc} & & J_2J_3J_4A & & J_1J_2J_3J_4A & & L \\ \rightarrow & \begin{bmatrix} \times & \times & & & \\ \times & \times & 0 & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \end{bmatrix} & \rightarrow & \begin{bmatrix} \times & 0 & & & \\ \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \end{bmatrix} & = & \begin{bmatrix} \times & & & & \\ \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \end{bmatrix} \end{array} \tag{8.149}$$

In this way

$$J_1 J_2 \cdots J_{n-1} A_0 = L_0 \ , \quad \text{and} \quad A_1 = L_0 J_{n-1}^T \cdots J_2^T J_1^T . \tag{8.150}$$

It appears in the above factorization that an extra subdiagonal is added, but we know that if A_0 is tridiagonal then so is A_1 and the second subdiagonal can be neglected.

For the tridiagonal matrix with diagonal elements $a_1, a_2, \ldots, a_n$, and subdiagonal elements $b_2, b_3, \ldots, b_n$ the pre-multiplication of A_0 by Q_0^T and the post-multiplication by Q_0

can be combined into the following algorithm for a QL step:

$$
\begin{aligned}
p_n &= a_n \ , \ c_n = 1 \ , \ s_n = 0 \\
r_{i+1} &= (p_{i+1}^2 + b_{i+1}^2)^{1/2} \\
g_{i+1} &= c_{i+1} \ b_{i+1} \\
h_{i+1} &= c_{i+1} p_{i+1} \\
b'_{i+2} &= s_{i+1} r_{i+1} \\
c_i &= p_{i+1}/r_{i+1} \\
s_i &= b_{i+1}/r_{i+1} \\
p_i &= c_i a_i - s_i g_{i+1} \\
a'_{i+1} &= h_{i+1} + s_i(c_i g_{i+1} + s_i a_i) \\
b'_2 &= s_1 p_1, \ a'_1 = c_1 p_1
\end{aligned}
\qquad i = n-1, \ldots, 1 \qquad (8.151)
$$

in which $a'_1, a'_2, \ldots, a'_n$, and $b'_2, b'_3, \ldots, b'_n$ are on the diagonal and super-diagonal of $A_1 = Q_0^T A_0 Q_0$.

If a super-diagonal entry b'_i in the iterated matrix becomes small as in

$$
A' = \begin{bmatrix} \lambda_1 & \epsilon & & \\ \epsilon & \times & \times & \times \\ & \times & \times & \times \\ & \times & \times & \times \end{bmatrix} \qquad (8.152)
$$

then λ_1 is recorded as an eigenvalue and iteration continues on a smaller matrix.

Table 8.3 describes the convergence of the QL algorithm for all eigenvalues of tridiagonal $T = T(20 \times 20)$ with $a_i = 2$ and $b_i = -1$. Figure 8.10 is a drawing made from table 8.3 for the first seven eigenvalues, and convergence is clearly seen to be no better than linear, as is expected of a procedure that is essentially a power method. Careful analysis demonstrates that quickness of convergence for λ_j depends on how isolated it is from the other eigenvalues, and that b'_i/b_i is proportional to λ_{i-1}/λ_i. Improvement in the rate of convergence is achieved with an appropriate shift and an algorithm that assumes the form

$$
Q_0(A_0 - \lambda'_0 I) = L_0 \quad A_1 = L_0 Q_0^T . \qquad (8.153)
$$

Ways to determine a good universal shifting strategy are too technical for discussion here.

Table 8.3: Convergence of the QL algorithm for a symmetric 20×20 tridiagonal matrix with $a_i = 2, b_i = -1$.

		1 iteration	
i	λ_i	a_i	b_i
1	0.22338348D−01	0.14634146D+00	−0.36365191D+00
2	0.88854388D−01	0.20075047D+01	−0.99605263D+00
3	0.19806226D+00	0.20083160D+01	−0.99561403D+00
4	0.34752245D+00	0.20092664D+01	−0.99509803D+00
5	0.53389626D+00	0.20103896D+01	−0.99448528D+00
6	0.75302040D+00	0.20117302D+01	−0.99374998D+00
7	0.10000000D+01	0.20133482D+01	−0.99285711D+00
8	0.12693180D+01	0.20153257D+01	−0.99175819D+00
9	0.15549581D+01	0.20177778D+01	−0.99038454D+00
10	0.18505398D+01	0.20208696D+01	−0.98863624D+00
11	0.21494602D+01	0.20248447D+01	−0.98636342D+00
12	0.24450419D+01	0.20300752D+01	−0.98333294D+00
13	0.27306820D+01	0.20371517D+01	−0.97916590D+00
14	0.30000000D+01	0.20470588D+01	−0.97321265D+00
15	0.32469796D+01	0.20615385D+01	−0.96428180D+00
16	0.34661037D+01	0.20839161D+01	−0.94998913D+00
17	0.36524775D+01	0.21212121D+01	−0.92496246D+00
18	0.38019377D+01	0.21904762D+01	−0.87481777D+00
19	0.39111456D+01	0.23428571D+01	−0.74833148D+00
20	0.39776617D+01	0.28000000D+01	

5 iterations

i	λ_i	a_i	b_i
1	0.22338348D−01	0.22338610D−01	−0.13244722D−03
2	0.88854388D−01	0.89426143D−01	−0.81937862D−02
3	0.19806226D+00	0.22052880D+00	−0.73184130D−01
4	0.34752245D+00	0.62558654D+00	−0.42050819D+00
5	0.53389626D+00	0.17250467D+01	−0.90080118D+00
6	0.75302040D+00	0.21375860D+01	−0.92762972D+00
7	0.10000000D+01	0.21524602D+01	−0.91958808D+00
8	0.12693180D+01	0.21698842D+01	−0.91012710D+00
9	0.15549581D+01	0.21904762D+01	−0.89889184D+00
10	0.18505398D+01	0.22150538D+01	−0.88540895D+00
11	0.21494602D+01	0.22447164D+01	−0.86903589D+00
12	0.24450419D+01	0.22809706D+01	−0.84888265D+00
13	0.27306820D+01	0.23259259D+01	−0.82368777D+00
14	0.30000000D+01	0.23826087D+01	−0.79161485D+00
15	0.32469796D+01	0.24554865D+01	−0.74990551D+00
16	0.34661037D+01	0.25513784D+01	−0.69425821D+00
17	0.36524775D+01	0.26811146D+01	−0.61764706D+00
18	0.38019377D+01	0.28627451D+01	−0.50787450D+00
19	0.39111456D+01	0.31282051D+01	−0.34154440D+00
20	0.39776617D+01	0.35384615D+01	

10 iterations

i	λ_i	a_i	b_i
1	0.22338348D−01	0.22338348D−01	−0.13267844D−06
2	0.88854388D−01	0.88854569D−01	−0.14025979D−03
3	0.19806226D+00	0.19812562D+00	−0.30896518D−02
4	0.34752245D+00	0.34938567D+00	−0.19289789D−01
5	0.53389626D+00	0.55131250D+00	−0.69434478D−01
6	0.75302040D+00	0.84269651D+00	−0.19262441D+00
7	0.10000000D+01	0.13224174D+01	−0.44666285D+00
8	0.12693180D+01	0.19998549D+01	−0.71338520D+00
9	0.15549581D+01	0.23930413D+01	−0.76943939D+00
10	0.18505398D+01	0.24758756D+01	−0.74996425D+00
11	0.21494602D+01	0.25253283D+01	−0.72364953D+00
12	0.24450419D+01	0.25821206D+01	−0.69293278D+00
13	0.27306820D+01	0.26486486D+01	−0.65678958D+00
14	0.30000000D+01	0.27272727D+01	−0.61385914D+00
15	0.32469796D+01	0.28211144D+01	−0.56232293D+00
16	0.34661037D+01	0.29343715D+01	−0.49970265D+00
17	0.36524775D+01	0.30727969D+01	−0.42253699D+00
18	0.38019377D+01	0.32444444D+01	−0.32581259D+00
19	0.39111456D+01	0.34608696D+01	−0.20160040D+00
20	0.39776617D+01	0.37391304D+01	

20 iterations

i	λ_i	a_i	b_i
1	0.22338348D−01	0.22338348D−01	−0.13382018D−12
2	0.88854388D−01	0.88854388D−01	−0.46303437D−07
3	0.19806226D+00	0.19806227D+00	−0.11119173D−04
4	0.34752245D+00	0.34752280D+00	−0.25515758D−03
5	0.53389626D+00	0.53391397D+00	−0.19900317D−02
6	0.75302040D+00	0.75329823D+00	−0.85635598D−02
7	0.10000000D+01	0.10021026D+01	−0.25587013D−01
8	0.12693180D+01	0.12791546D+01	−0.60238450D−01
9	0.15549581D+01	0.15880060D+01	−0.12057984D+00
10	0.18505398D+01	0.19378197D+01	−0.21439573D+00
11	0.21494602D+01	0.23358227D+01	−0.33809229D+00
12	0.24450419D+01	0.27356368D+01	−0.43726852D+00
13	0.27306820D+01	0.29992152D+01	−0.45053972D+00
14	0.30000000D+01	0.31217432D+01	−0.41549665D+00
15	0.32469796D+01	0.32133728D+01	−0.36901848D+00
16	0.34661037D+01	0.33125241D+01	−0.31635942D+00
17	0.36524775D+01	0.34242293D+01	−0.25676050D+00
18	0.38019377D+01	0.35508274D+01	−0.18886693D+00
19	0.39111456D+01	0.36950904D+01	−0.11064579D+00
20	0.39776617D+01	0.38604651D+01	

40 iterations

i	λ_i	a_i	b_i
1	0.22338348D−01	0.22338348D−01	−0.13613345D−24
2	0.88854388D−01	0.88854388D−01	−0.50484270D−14
3	0.19806226D+00	0.19806226D+00	−0.14536155D−09
4	0.34752245D+00	0.34752245D+00	−0.47567598D−07
5	0.53389626D+00	0.53389626D+00	−0.20494549D−05
6	0.75302040D+00	0.75302040D+00	−0.29333865D−04
7	0.10000000D+01	0.10000002D+01	−0.21431910D−03
8	0.12693180D+01	0.12693213D+01	−0.10030325D−02
9	0.15549581D+01	0.15549944D+01	−0.34287830D−02
10	0.18505398D+01	0.18507880D+01	−0.92851379D−02
11	0.21494602D+01	0.21506569D+01	−0.21006103D−01
12	0.24450419D+01	0.24494064D+01	−0.41184220D−01
13	0.27306820D+01	0.27433266D+01	−0.71784877D−01
14	0.30000000D+01	0.30298355D+01	−0.11255362D−01
15	0.32469796D+01	0.33017893D+01	−0.15401729D+00
16	0.34661037D+01	0.35260234D+01	−0.16727419D+00
17	0.36524775D+01	0.36634213D+01	−0.14224835D+00
18	0.38019377D+01	0.37520436D+01	−0.10241765D+00
19	0.39111456D+01	0.38369881D+01	−0.58138692D−01
20	0.39776617D+01	0.39277108D+01	

60 iterations

i	λ_i	a_i	b_i
1	0.22338348D−01	0.22338348D−01	−0.13848870D−36
2	0.88854388D−01	0.88854388D−01	−0.55042599D−21
3	0.19806226D+00	0.19806226D+00	−0.19003211D−14
4	0.34752245D+00	0.34752245D+00	−0.88681095D−11
5	0.53389626D+00	0.53389626D+00	−0.21117431D−08
6	0.75302040D+00	0.75302040D+00	−0.10081008D−06
7	0.10000000D+01	0.10000000D+01	−0.18182350D−05
8	0.12693180D+01	0.12693180D+01	−0.17311133D−04
9	0.15549581D+01	0.15549582D+01	−0.10553403D−03
10	0.18505398D+01	0.18505405D+01	−0.46395444D−03
11	0.21494602D+01	0.21494680D+01	−0.15878264D−02
12	0.24450419D+01	0.24451026D+01	−0.44500619D−02
13	0.27306820D+01	0.27310264D+01	−0.10566173D−01
14	0.30000000D+01	0.30014861D+01	−0.21745719D−01
15	0.32469796D+01	0.32519918D+01	−0.39333400D−01
16	0.34661037D+01	0.34791695D+01	−0.62110999D−01
17	0.36524775D+01	0.36743285D+01	−0.78328629D−01
18	0.38019377D+01	0.38100301D+01	−0.67904389D−01
19	0.39111456D+01	0.38876780D+01	−0.39368515D−01
20	0.39776617D+01	0.39512083D+01	

Having secured an approximate eigenvalue λ' the corresponding approximate eigenvector x' is best computed by inverse iteration of $A - \lambda' I$. It may appear that solution of $(A - \lambda' I)x' = o$ is a more attractive proposition for a tridiagonal A, but this is not the case on practical grounds. As is so often the case in numerical methods, what appears to be

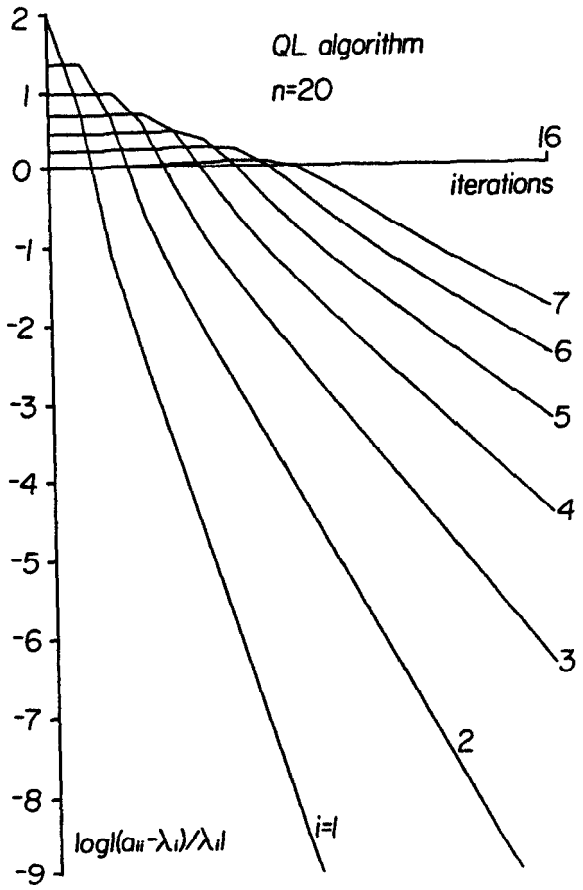


Fig. 8.10

theoretically perfect may be mortally tainted by imperfections. Consider

$$\begin{bmatrix} 2-\lambda & -1 & & & \\ -1 & 2-\lambda & -1 & & \\ & -1 & 2-\lambda & -1 & \\ & & -1 & 2-\lambda & -1 \\ & & & -1 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = o, \quad (A-\lambda I)x = o, \tag{8.154}$$

for eigenvalue λ and chose x_1 so as to get the components

$$\begin{aligned} x_2 =& (2-\lambda) \\ x_3 =& (2-\lambda)^2 - 1 \\ x_4 =& (2-\lambda)^3 - 2(2-\lambda) \\ x_5 =& (2-\lambda)^4 - 3(2-\lambda)^2 + 1 \end{aligned} \tag{8.155}$$

with $-x_4 + (1-\lambda)x_5 = 0$. If λ is an eigenvalue of matrix A, x_2, x_3, x_4, x_5 are correct, but for λ not truly an eigenvalue of A, x_2, x_3, x_4, x_5 computed from eq.(8.155) can be in serious

error. Theoretically, $\lambda_1 = 4sin^2(\pi/22) = 0.08$ for which $x_1 = 1, x_2 = 1.919, x_3 = 2.683, x_4 = 3.229, x_5 = 3.513$. For $\lambda = 0$ eq.(8.155) yields $x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 4, x_5 = 5$, not an appealing approximation.

exercises

8.13.1. For matrices

$$A = \begin{bmatrix} 1 & \epsilon & \\ \epsilon & 1 & \epsilon \\ & \epsilon & 1 \end{bmatrix}, \; A' = \begin{bmatrix} 1 & \epsilon & \\ \epsilon & 4 & \epsilon \\ & \epsilon & 9 \end{bmatrix}, \; |\epsilon| << 1$$

show that the eigenvalues of A are $\lambda_1 = 1 - \sqrt{2}\epsilon, \; \lambda_2 = 1, \; \lambda_3 = 1 + \sqrt{2}\epsilon$, and that those of A' are in the intervals $1 \pm \epsilon^2, \; 4 \pm 2\epsilon^2, \; 9 \pm \epsilon^2$. For the latter use Gerschgorin's theorem preceded by a similarity transformation $D^{-1}A'D$ with a diagonal D.

8.13.2. Use the elementary similarity transformations of Sec. 6.7 to reduce

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

to tridiagonal form. Interchange rows and columns to have a nonzero (1,2) pivot entry in A.

8.13.3. Find the sequence of elementary similarity transformations that affect

$$\begin{bmatrix} 1 & -1 & \\ -1 & 2 & -1 \\ & -1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} & -1 & \\ 1/3 & & -1 \\ & 5/3 & 3 \end{bmatrix}.$$

8.13.4. Write out all eigenvalues of the $n \times n$ matrix

$$A = \begin{bmatrix} & 1 & 1 & \\ & & 1 & 1 \\ & & & 1 \\ \epsilon & & & \end{bmatrix}.$$

8.13.5. For matrices

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 3 \\ 3 & 3 & 3 \end{bmatrix}, \; B = \begin{bmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{bmatrix}$$

fix α and β in the elementary matrix

$$E = \begin{bmatrix} 1 & & \alpha \\ & 1 & \beta \\ & & 1 \end{bmatrix},\ E^{-1} = \begin{bmatrix} 1 & & -\alpha \\ & 1 & -\beta \\ & & 1 \end{bmatrix}$$

so that both $A'_{13} = 0$ and $B'_{13} = 0$, $A' = E^{-1}AE, B' = E^{-1}AE$.

8.13.6. Consider the linear system $Ax = f$ with positive definite and symmetric A. Write $A = L + D + L^T$

$$\begin{matrix} A \end{matrix} \quad \begin{matrix} L \end{matrix} \quad \begin{matrix} D \end{matrix} \quad \begin{matrix} L^T \end{matrix}$$

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} = \begin{bmatrix} 0 & & \\ \times & 0 & \\ \times & \times & 0 \end{bmatrix} + \begin{bmatrix} \times & & \\ & \times & \\ & & \times \end{bmatrix} + \begin{bmatrix} 0 & \times & \times \\ & 0 & \times \\ & & 0 \end{bmatrix}$$

so that $D_{ii} = A_{ii}$, and prove that

$$|\lambda(G)| < 1,\ G = -(D + L)^{-1}L^T.$$

Hint: Use

$$G' = D^{1/2}GD^{-1/2} = -(I + L')L'^{-1},\ L' = D^{-1/2}LD^{-1/2}.$$

Prove that the iterative scheme

$$(D + L)x_1 = -L^T x_0 + f$$

$$x_1 = x_0 + (D + L)^{-1}r_0,\ r_0 = f - Ax_0$$

converges for any initial guess x_0.

Apply the *Gauss-Seidel* iteration to the solution of

$$\begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & -1 & 2 & -1 & \\ & & & -1 & 2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and study its convergence to $x_0 = [5\ 4\ 3\ 2\ 1]^T$.

8.13.7. *Successive Over-Relaxation* is a refinement of the Gauss-Seidel method whereby

$$M(\omega)x_1 = N(\omega)x_0 + \omega f$$

for the relaxation parameter ω in

$$M(\omega) = D + \omega L \quad \text{and} \quad N(\omega) = (1-\omega)D - \omega L^T.$$

Finding the best ω is no simple matter, but program the algorithm for the system in the previous exercise and study the effect of $0 < \omega < 2$ on convergence.

8.13.8. Consider matrix A and its inverse

$$A = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix}, \; A^{-1} = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Looking for a *sparse* approximate M to A^{-1} such that

$$\|MA - I\| \leq \rho < 1$$

we propose the three candidates

$$M_1 = \begin{bmatrix} 1 & & & \\ & 1/2 & & \\ & & 1/2 & \\ & & & 1/2 \end{bmatrix}, \; M_2 = \begin{bmatrix} 4 & & & \\ & 3 & & \\ & & 2 & \\ & & & 1 \end{bmatrix}, \text{ and } M_3 = \begin{bmatrix} 4 & 3 & & \\ 3 & 3 & 2 & \\ & 2 & 2 & 1 \\ & & 1 & 1 \end{bmatrix}.$$

Compute the spectral radius of $M_i A - I$ for $i = 1, 2, 3$, and determine the best choice.