CHAPTER 20

Numeric Linear Algebra

This chapter deals with two main topics. The first topic is how to solve linear systems of equations numerically. We start with Gauss elimination, which may be familiar to some readers, but this time in an algorithmic setting with partial pivoting. Variants of this method (Doolittle, Crout, Cholesky, Gauss–Jordan) are discussed in Sec. 20.2. All these methods are direct methods, that is, methods of numerics where we know in advance how many steps they will take until they arrive at a solution. However, small pivots and roundoff error magnification may produce nonsensical results, such as in the Gauss method. A shift occurs in Sec. 20.3, where we discuss numeric iteration methods or indirect methods to address our first topic. Here we cannot be totally sure how many steps will be needed to arrive at a good answer. Several factors—such as how far is the starting value from our initial solution, how is the problem structure influencing speed of convergence, how accurate would we like our result to be—determine the outcome of these methods. Moreover, our computation cycle may not converge. Gauss–Seidel iteration and Jacobi iteration are discussed in Sec. 20.3. Section 20.4 is at the heart of addressing the pitfalls of numeric linear algebra. It is concerned with problems that are ill-conditioned. We learn to estimate how "bad" such a problem is by calculating the condition number of its matrix.

The second topic (Secs. 20.6–20.9) is how to solve eigenvalue problems numerically. Eigenvalue problems appear throughout engineering, physics, mathematics, economics, and many areas. For large or very large matrices, determining the eigenvalues is difficult as it involves finding the roots of the characteristic equations, which are high-degree polynomials. As such, there are different approaches to tackling this problem. Some methods, such as Gerschgorin's method and Collatz's method only provide a range in which eigenvalues lie and thus are known as inclusion methods. Others such as tridiagonalization and QR-factorization actually find all the eigenvalues. The area is quite ingeneous and should be fascinating to the reader.

COMMENT. ***This chapter is independent of Chap.* 19 *and can be studied immediately after Chap.* 7 *or* 8.**

Prerequisite: Secs. 7.1, 7.2, 8.1.
Sections that may be omitted in a shorter course: 20.4, 20.5, 20.9.
References and Answers to Problems: App. 1 Part E, App. 2.

20.1 Linear Systems: Gauss Elimination

The basic method for solving systems of linear equations by Gauss elimination and back substitution was explained in Sec. 7.3. If you covered Sec. 7.3, you may wonder why we cover Gauss elimination again. The reason is that *here we cover Gauss elimination in the*

setting of numerics and introduce new material such as pivoting, row scaling, and operation count. Furthermore, we give an algorithmic representation of Gauss elimination in Table 20.1 that can be readily converted into software. We also show when Gauss elimination runs into difficulties with small pivots and what to do about it. The reader should pay close attention to the material as variants of Gauss elimination are covered in Sec. 20.2 and, furthermore, the general problem of solving linear systems is the focus of the first half of this chapter.

A **linear system of *n* equations** in n *unknowns* $x_1, \cdots, x_n$ is a set of equations $\mathrm{E}_1, \cdots, \mathrm{E}_n$ of the form

$$
\begin{aligned}
&\mathrm{E}_1: \quad a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\
&\mathrm{E}_2: \quad a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\
&\qquad \cdots\cdots\cdots\cdots\cdots \\
&\mathrm{E}_n: \quad a_{n1}x_1 + \cdots + a_{nn}x_n = b_n
\end{aligned} \tag{1}
$$

where the **coefficients** a_{jk} and the b_j are given numbers. The system is called **homogeneous** if all the b_j are zero; otherwise it is called **nonhomogeneous.** Using matrix multiplication (Sec. 7.2), we can write (1) as a single vector equation

$$
\mathbf{Ax} = \mathbf{b} \tag{2}
$$

where the **coefficient matrix** $\mathbf{A} = [a_{jk}]$ is the $n \times n$ matrix

$$
\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}
$$

are column vectors. The following matrix $\widetilde{\mathbf{A}}$ is called the **augmented matrix** of the system (1):

$$
\widetilde{\mathbf{A}} = [\mathbf{A} \quad \mathbf{b}] = \begin{bmatrix} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_2 \\ \cdot & \cdots & \cdot & \cdot \\ a_{n1} & \cdots & a_{nn} & b_n \end{bmatrix}.
$$

A **solution** of (1) is a set of numbers $x_1, \cdots, x_n$ that satisfy all the n equations, and a **solution vector** of (1) is a vector $\mathbf{x}$ whose components constitute a solution of (1).

The method of solving such a system by determinants (Cramer's rule in Sec. 7.7) is not practical, even with efficient methods for evaluating the determinants.

A practical method for the solution of a linear system is the so-called *Gauss elimination*, which we shall now discuss (***proceeding independently of Sec. 7.3***).

Gauss Elimination

This standard method for solving linear systems (1) is a systematic process of elimination that reduces (1) to **triangular form** because the system can then be easily solved by **back substitution**. For instance, a triangular system is

$$\begin{aligned} 3x_1 + 5x_2 + 2x_3 &= 8 \\ 8x_2 + 2x_3 &= -7 \\ 6x_3 &= 3 \end{aligned}$$

and back substitution gives $x_3 = \frac{3}{6} = \frac{1}{2}$ from the third equation, then

$$x_2 = \tfrac{1}{8}(-7 - 2x_3) = -1$$

from the second equation, and finally from the first equation

$$x_1 = \tfrac{1}{3}(8 - 5x_2 - 2x_3) = 4.$$

How do we reduce a given system (1) to triangular form? In the first step we ***eliminate*** x_1 from equations E_2 to E_n in (1). We do this by adding (or subtracting) suitable multiples of E_1 to (from) equations $\mathrm{E}_2, \cdots, \mathrm{E}_n$ and taking the resulting equations, call them $\mathrm{E}_2^*, \cdots, \mathrm{E}_n^*$ as the new equations. The first equation, E_1, is called the **pivot equation** in this step, and a_{11} is called the **pivot**. This equation is left unaltered. In the second step we take the new second equation E_2^* (which no longer contains x_1) as the pivot equation and use it to ***eliminate*** $\boldsymbol{x_2}$ from E_3^* to E_n^*. And so on. After $n - 1$ steps this gives a triangular system that can be solved by back substitution as just shown. In this way we obtain precisely all solutions of the *given* system (as proved in Sec. 7.3).

The pivot a_{kk} (in step k) ***must be*** different from zero and ***should be*** large in absolute value to avoid roundoff magnification by the multiplication in the elimination. For this we choose as our pivot equation one that has the absolutely largest a_{jk} in column k on or below the main diagonal (actually, the uppermost if there are several such equations). This popular method is called **partial pivoting**. It is used in CASs (e.g., in Maple).

Partial pivoting distinguishes it from **total pivoting**, which involves both row and column interchanges but is hardly used in practice.

Let us illustrate this method with a simple example.

EXAMPLE 1 **Gauss Elimination. Partial Pivoting**

Solve the system

$$\begin{aligned} \mathrm{E}_1: \quad 8x_2 + 2x_3 &= -7 \\ \mathrm{E}_2: \quad 3x_1 + 5x_2 + 2x_3 &= 8 \\ \mathrm{E}_3: \quad 6x_1 + 2x_2 + 8x_3 &= 26. \end{aligned}$$

Solution. We must pivot since E_1 has no x_1-term. In Column 1, equation E_3 has the largest coefficient. Hence we interchange E_1 and E_3,

$$\begin{aligned} 6x_1 + 2x_2 + 8x_3 &= 26 \\ 3x_1 + 5x_2 + 2x_3 &= 8 \\ 8x_2 + 2x_3 &= -7. \end{aligned}$$

Step 1. Elimination of x_1

It would suffice to show the augmented matrix and operate on it. We show both the equations and the augmented matrix. In the first step, the first equation is the pivot equation. Thus

$$\begin{array}{rl} \text{Pivot } 6 \longrightarrow & \boxed{6x_1} + 2x_2 + 8x_3 = 26 \\ \text{Eliminate} \longrightarrow & \boxed{3x_1} + 5x_2 + 2x_3 = 8 \\ & 8x_2 + 2x_3 = -7 \end{array} \qquad \left[\begin{array}{cccc} 6 & 2 & 8 & 26 \\ 3 & 5 & 2 & 8 \\ 0 & 8 & 2 & -7 \end{array}\right].$$

To eliminate x_1 from the other equations (here, from the second equation), do:

$$\text{Subtract } \tfrac{3}{6} = \tfrac{1}{2} \text{ times the pivot equation from the second equation.}$$

The result is

$$\begin{array}{rl} 6x_1 + 2x_2 + 8x_3 = 26 \\ 4x_2 - 2x_3 = -5 \\ 8x_2 + 2x_3 = -7 \end{array} \qquad \left[\begin{array}{cccc} 6 & 2 & 8 & 26 \\ 0 & 4 & -2 & -5 \\ 0 & 8 & 2 & -7 \end{array}\right].$$

Step 2. Elimination of x_2

The largest coefficient in Column 2 is 8. Hence we take the *new* third equation as the pivot equation, interchanging equations 2 and 3,

$$\begin{array}{rl} & 6x_1 + 2x_2 + 8x_3 = 26 \\ \text{Pivot } 8 \longrightarrow & \boxed{8x_2} + 2x_3 = -7 \\ \text{Eliminate} \longrightarrow & \boxed{4x_2} - 2x_3 = -5 \end{array} \qquad \left[\begin{array}{cccc} 6 & 2 & 8 & 26 \\ 0 & 8 & 2 & -7 \\ 0 & 4 & -2 & -5 \end{array}\right].$$

To eliminate x_2 from the third equation, do:

$$\text{Subtract } \tfrac{1}{2} \text{ times the pivot equation from the third equation.}$$

The resulting triangular system is shown below. This is the end of the forward elimination. Now comes the back substitution.

Back substitution. ***Determination of x_3, x_2, x_1***

The triangular system obtained in Step 2 is

$$\begin{array}{rl} 6x_1 + 2x_2 + 8x_3 = 26 \\ 8x_2 + 2x_3 = -7 \\ -3x_3 = -\tfrac{3}{2} \end{array} \qquad \left[\begin{array}{cccc} 6 & 2 & 8 & 26 \\ 0 & 8 & 2 & -7 \\ 0 & 0 & -3 & -\tfrac{3}{2} \end{array}\right].$$

From this system, taking the last equation, then the second equation, and finally the first equation, we compute the solution

$$\begin{aligned} x_3 &= \tfrac{1}{2} \\ x_2 &= \tfrac{1}{8}(-7 - 2x_3) = -1 \\ x_1 &= \tfrac{1}{6}(26 - 2x_2 - 8x_3) = 4. \end{aligned}$$

This agrees with the values given above, before the beginning of the example. ■

The general algorithm for the Gauss elimination is shown in Table 20.1. To help explain the algorithm, we have numbered some of its lines. b_j is denoted by $a_{j,n+1}$, for uniformity. In lines 1 and 2 we look for a possible pivot. [For $k = 1$ we can always find one; otherwise x_1 would not occur in (1).] In line 2 we do pivoting if necessary, picking an a_{jk} of greatest absolute value (the one with the smallest j if there are several) and interchange the

corresponding rows. If $|a_{kk}|$ is greatest, we do no pivoting. m_{jk} in line 4 suggests *multiplier*, since these are the factors by which we have to multiply the pivot equation E_k^* in Step k before subtracting it from an equation E_j^* below E_k^* from which we want to eliminate x_k. Here we have written E_k^* and E_j^* to indicate that after Step 1 these are no longer the equations given in (1), but these underwent a change in each step, as indicated in line 5. Accordingly, a_{jk} etc. in all lines refer to the most recent equations, and $j \geqq k$ in line 1 indicates that we leave untouched all the equations that have served as pivot equations in previous steps. For $p = k$ in line 5 we get 0 on the right, as it should be in the elimination,

$$a_{jk} - m_{jk}a_{kk} = a_{jk} - \frac{a_{jk}}{a_{kk}}a_{kk} = 0.$$

In line 3, if the last equation in the *triangular* system is $0 = b_n^* \neq 0$, we have no solution. If it is $0 = b_n^* = 0$, we have no unique solution because we then have fewer equations than unknowns.

EXAMPLE 2 Gauss Elimination in Table 20.1, Sample Computation

In Example 1 we had $a_{11} = 0$, so that pivoting was necessary. The greatest coefficient in Column 1 was a_{31}. Thus $\tilde{j} = 3$ in line 2, and we interchanged E_1 and E_3. Then in lines 4 and 5 we computed $m_{21} = \frac{3}{6} = \frac{1}{2}$ and

$$a_{22} = 5 - \tfrac{1}{2} \cdot 2 = 4, \qquad a_{23} = 2 - \tfrac{1}{2} \cdot 8 = -2, \qquad a_{24} = 8 - \tfrac{1}{2} \cdot 26 = -5,$$

and then $m_{31} = \frac{0}{6} = 0$, so that the third equation $8x_2 + 2x_3 = -7$ did not change in Step 1. In Step 2 ($k = 2$) we had 8 as the greatest coefficient in Column 2, hence $\tilde{j} = 3$. We interchanged equations 2 and 3, computed $m_{32} = -\frac{4}{8} = -\frac{1}{2}$ in line 5, and the $a_{33} = -2 - \frac{1}{2} \cdot 2 = -3$, $a_{34} = -5 - \frac{1}{2}(-7) = -\frac{3}{2}$. This produced the triangular form used in the back substitution. ■

If $a_{kk} = 0$ in Step k, ***we must pivot***. If $|a_{kk}|$ is small, ***we should pivot*** because of roundoff error magnification that may seriously affect accuracy or even produce nonsensical results.

EXAMPLE 3 Difficulty with Small Pivots

The solution of the system

$$0.0004x_1 + 1.402x_2 = 1.406$$
$$0.4003x_1 - 1.502x_2 = 2.501$$

is $x_1 = 10, x_2 = 1$. We solve this system by the Gauss elimination, using four-digit floating-point arithmetic. (4D is for simplicity. Make an 8D-arithmetic example that shows the same.)

(a) Picking the first of the given equations as the pivot equation, we have to multiply this equation by $m = 0.4003/0.0004 = 1001$ and subtract the result from the second equation, obtaining

$$-1405x_2 = -1404.$$

Hence $x_2 = -1404/(-1405) = 0.9993$, and from the first equation, instead of $x_1 = 10$, we get

$$x_1 = \frac{1}{0.0004}(1.406 - 1.402 \cdot 0.9993) = \frac{0.005}{0.0004} = 12.5.$$

This failure occurs because $|a_{11}|$ is small compared with $|a_{12}|$, so that a small roundoff error in x_2 leads to a large error in x_1.

(b) Picking the second of the given equations as the pivot equation, we have to multiply this equation by $0.0004/0.4003 = 0.0009993$ and subtract the result from the first equation, obtaining

$$1.404x_2 = 1.404.$$

Hence $x_2 = 1$, and from the pivot equation $x_1 = 10$. This success occurs because $|a_{21}|$ is not very small compared to $|a_{22}|$, so that a small roundoff error in x_2 would not lead to a large error in x_1. Indeed, for instance, if we had the value $x_2 = 1.002$, we would still have from the pivot equation the good value $x_1 = (2.501 + 1.505)/0.4003 = 10.01$. ■

Table 20.1 Gauss Elimination

ALGORITHM GAUSS ($\tilde{\mathbf{A}} = [a_{jk}] = [\mathbf{A} \quad \mathbf{b}]$)

This algorithm computes a unique solution $\mathbf{x} = [x_j]$ of the system (1) or indicates that (1) has no unique solution.

INPUT: Augmented $n \times (n + 1)$ matrix $\tilde{\mathbf{A}} = [a_{jk}]$, where $a_{j,n+1} = b_j$

OUTPUT: Solution $\mathbf{x} = [x_j]$ of (1) or message that the system (1) has no unique solution

For $k = 1, \cdots, n - 1$, do:

1 $m = k$

For $j = k + 1, \cdots, n$, do:

If $(|a_{mk}| < |a_{jk}|)$ then $m = j$

End

If $a_{mk} = 0$ then OUTPUT "No unique solution exists"
Stop
[*Procedure completed unsuccessfully*]

2 Else exchange row k and row m

3 If $a_{nn} = 0$ then OUTPUT "No unique solution exists."
Stop
Else

4 For $j = k + 1, \cdots, n$, do:

$$m_{jk} := \frac{a_{jk}}{a_{kk}}$$

5 For $p = k + 1, \cdots, n + 1$, do:

$$a_{jp} := a_{jp} - m_{jk}a_{kp}$$

End

End

End

6 $$x_n = \frac{a_{n,n+1}}{a_{nn}} \qquad [\textit{Start back substitution}]$$

For $i = n - 1, \cdots, 1$, do:

7 $$x_i = \frac{1}{a_{ii}}\left(a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right)$$

End

OUTPUT $\mathbf{x} = [x_j]$. Stop

End GAUSS

Error estimates for the Gauss elimination are discussed in Ref. [E5] listed in App. 1.

Row scaling means the multiplication of each Row j by a suitable scaling factor s_j. It is done in connection with partial pivoting to get more accurate solutions. Despite much research (see Refs. [E9], [E24] in App. 1) and the proposition of several principles, scaling is still not well understood. As a possibility, one can scale for pivot choice only (not in the calculation, to avoid additional roundoff) and take as first pivot the entry a_{j1} for which $|a_{j1}|/|A_j|$ is largest; here A_j is an entry of largest absolute value in Row j. Similarly in the further steps of the Gauss elimination.

For instance, for the system

$$4.0000x_1 + 14020x_2 = 14060$$
$$0.4003x_1 - 1.502x_2 = 2.501$$

we might pick 4 as pivot, but dividing the first equation by 10^4 gives the system in Example 3, for which the second equation is a better pivot equation.

Operation Count

Quite generally, important factors in judging the quality of a numeric method are

Amount of storage

Amount of time ($\equiv$ number of operations)

Effect of roundoff error

For the Gauss elimination, the operation count for a full matrix (a matrix with relatively many nonzero entries) is as follows. In Step k we eliminate x_k from $n - k$ equations. This needs $n - k$ divisions in computing the m_{jk} (line 3) and $(n - k)(n - k + 1)$ multiplications and as many subtractions (both in line 4). Since we do $n - 1$ steps, k goes from 1 to $n - 1$ and thus the total number of operations in this forward elimination is

$$f(n) = \sum_{k=1}^{n-1} (n - k) + 2\sum_{k=1}^{n-1} (n - k)(n - k + 1) \qquad \text{(write } n - k = s\text{)}$$
$$= \sum_{s=1}^{n-1} s + 2\sum_{s=1}^{n-1} s(s + 1) = \tfrac{1}{2}(n - 1)n + \tfrac{2}{3}(n^2 - 1)n \approx \tfrac{2}{3}n^3$$

where $2n^3/3$ is obtained by dropping lower powers of n. We see that $f(n)$ grows about proportional to n^3. We say that $f(n)$ is of *order* n^3 and write

$$f(n) = O(n^3)$$

where O suggests **order**. The general definition of O is as follows. We write

$$f(n) = O(h(n))$$

if the quotients $|f(n)/h(n)|$ and $|h(n)/f(n)|$ remain bounded (do not trail off to infinity) as $n \to \infty$. In our present case, $h(n) = n^3$ and, indeed, $f(n)/n^3 \to \frac{2}{3}$ because the omitted terms divided by n^3 go to zero as $n \to \infty$.

In the back substitution of x_i we make $n - i$ multiplications and as many subtractions, as well as 1 division. Hence the number of operations in the back substitution is

$$b(n) = 2\sum_{i=1}^{n}(n - i) + n = 2\sum_{s=1}^{n} s + n = n(n + 1) + n = n^2 + 2n = O(n^2).$$

We see that it grows more slowly than the number of operations in the forward elimination of the Gauss algorithm, so that it is negligible for large systems because it is smaller by a factor n, approximately. For instance, if an operation takes 10^{-9} sec, then the times needed are:

Algorithm	$n = 1000$	$n = 10000$
Elimination	0.7 sec	11 min
Back substitution	0.001 sec	0.1 sec

PROBLEM SET 20.1

APPLICATIONS of linear systems see Secs. 7.1 and 8.2.

1–3 GEOMETRIC INTERPRETATION

Solve graphically and explain geometrically.

1. $x_1 - 4x_2 = 20.1$

$3x_1 + 5x_2 = 5.9$

2. $-5.00x_1 + 8.40x_2 = 0$

$10.25x_1 - 17.22x_2 = 0$

3. $7.2x_1 - 3.5x_2 = 16.0$

$-14.4x_1 + 7.0x_2 = 31.0$

4–16 GAUSS ELIMINATION

Solve the following linear systems by Gauss elimination, with partial pivoting if necessary (but without scaling). Show the intermediate steps. Check the result by substitution. If no solution or more than one solution exists, give a reason.

4. $6x_1 + x_2 = -3$

$4x_1 - 2x_2 = 6$

5. $2x_1 - 8x_2 = -4$

$3x_1 + x_2 = 7$

6. $25.38x_1 - 15.48x_2 = 30.60$

$-14.10x_1 + 8.60x_2 = -17.00$

7. $-3x_1 + 6x_2 - 9x_3 = -46.725$

$x_1 - 4x_2 + 3x_3 = 19.571$

$2x_1 + 5x_2 - 7x_3 = -20.073$

8. $5x_1 + 3x_2 + x_3 = 2$

$-4x_2 + 8x_3 = -3$

$10x_1 - 6x_2 + 26x_3 = 0$

9. $6x_2 + 13x_3 = 137.86$

$6x_1 - 8x_3 = -85.88$

$13x_1 - 8x_2 = 178.54$

10. $4x_1 + 4x_2 + 2x_3 = 0$

$3x_1 - x_2 + 2x_3 = 0$

$3x_1 + 7x_2 + x_3 = 0$

11. $3.4x_1 - 6.12x_2 - 2.72x_3 = 0$

$-x_1 + 1.80x_2 + 0.80x_3 = 0$

$2.7x_1 - 4.86x_2 + 2.16x_3 = 0$

12. $5x_1 + 3x_2 + x_3 = 2$

$-4x_2 + 8x_3 = -3$

$10x_1 - 6x_2 + 26x_3 = 0$

13.
$$\begin{aligned} 3x_2 + 5x_3 &= 1.20736 \\ 3x_1 - 4x_2 &= -2.34066 \\ 5x_1 + 6x_3 &= -0.329193 \end{aligned}$$

14.
$$\begin{aligned} -47x_1 + 4x_2 - 7x_3 &= -118 \\ 19x_1 - 3x_2 + 2x_3 &= 43 \\ -15x_1 + 5x_2 &= -25 \end{aligned}$$

15.
$$\begin{aligned} 2.2x_2 + 1.5x_3 - 3.3x_4 &= -9.30 \\ 0.2x_1 + 1.8x_2 + 4.2x_4 &= 9.24 \\ -x_1 - 3.1x_2 + 2.5x_3 &= -8.70 \\ 0.5x_1 - 3.8x_3 + 1.5x_4 &= 11.94 \end{aligned}$$

16.
$$\begin{aligned} 3.2x_1 + 1.6x_2 &= -0.8 \\ 1.6x_1 - 0.8x_2 + 2.4x_3 &= 16.0 \\ 2.4x_2 - 4.8x_3 + 3.6x_4 &= -39.0 \\ 3.6x_3 + 2.4x_4 &= 10.2 \end{aligned}$$

17. CAS EXPERIMENT. Gauss Elimination. Write a program for the Gauss elimination with pivoting. Apply it to Probs. 13–16. Experiment with systems whose coefficient determinant is small in absolute value. Also investigate the performance of your program for larger systems of your choice, including sparse systems.

18. TEAM PROJECT. Linear Systems and Gauss Elimination. (a) Existence and uniqueness. Find a and b such that $ax_1 + x_2 = b, x_1 + x_2 = 3$ has (i) a unique solution, (ii) infinitely many solutions, (iii) no solutions.

(b) Gauss elimination and nonexistence. Apply the Gauss elimination to the following two systems and compare the calculations step by step. Explain why the elimination fails if no solution exists.

$$\begin{aligned} x_1 + x_2 + x_3 &= 3 \\ 4x_1 + 2x_2 - x_3 &= 5 \\ 9x_1 + 5x_2 - x_3 &= 13 \end{aligned}$$

$$\begin{aligned} x_1 + x_2 + x_3 &= 3 \\ 4x_1 + 2x_2 - x_3 &= 5 \\ 9x_1 + 5x_2 - x_3 &= 12. \end{aligned}$$

(c) Zero determinant. Why may a computer program give you the result that a homogeneous linear system has only the trivial solution although you know its coefficient determinant to be zero?

(d) Pivoting. Solve System (A) (below) by the Gauss elimination first without pivoting. Show that for any fixed machine word length and sufficiently small $\epsilon > 0$ the computer gives $x_2 = 1$ and then $x_1 = 0$. What is the exact solution? Its limit as $\epsilon \to 0$? Then solve the system by the Gauss elimination with pivoting. Compare and comment.

(e) Pivoting. Solve System (B) by the Gauss elimination and three-digit rounding arithmetic, choosing (i) the first equation, (ii) the second equation as pivot equation. (Remember to round to 3S after each operation before doing the next, just as would be done on a computer!) Then use four-digit rounding arithmetic in those two calculations. Compare and comment.

(A)
$$\begin{aligned} \epsilon x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned}$$

(B)
$$\begin{aligned} 4.03x_1 + 2.16x_2 &= -4.61 \\ 6.21x_1 + 3.35x_2 &= -7.19 \end{aligned}$$

20.2 Linear Systems: LU-Factorization, Matrix Inversion

We continue our discussion of numeric methods for solving linear systems of n equations in n unknowns $x_1, \cdots, x_n$,

(1) $$\mathbf{Ax} = \mathbf{b}$$

where $\mathbf{A} = [a_{jk}]$ is the $n \times n$ given coefficient matrix and $\mathbf{x}^\mathsf{T} = [x_1, \cdots, x_n]$ and $\mathbf{b}^\mathsf{T} = [b_1, \cdots, b_n]$. We present three related methods that are modifications of the Gauss

elimination, which require fewer arithmetic operations. They are named after Doolittle, Crout, and Cholesky and use the idea of the LU-factorization of **A**, which we explain first.

An **LU-factorization** of a given square matrix **A** is of the form

$$\mathbf{A} = \mathbf{LU} \tag{2}$$

where **L** is ***lower triangular*** and **U** is ***upper triangular***. For example,

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 8 & 5 \end{bmatrix} = \mathbf{LU} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 0 & -7 \end{bmatrix}.$$

It can be proved that for any nonsingular matrix (see Sec. 7.8) the rows can be reordered so that the resulting matrix **A** has an LU-factorization (2) in which **L** turns out to be the matrix of the ***multipliers*** m_{jk} of the Gauss elimination, with main diagonal $1, \cdots, 1$, and **U** is the matrix of the triangular system at the end of the Gauss elimination. (See Ref. [E5], pp. 155–156, listed in App. 1.)

The ***crucial idea*** now is that **L** and **U** in (2) can be computed directly, without solving simultaneous equations (thus, without using the Gauss elimination). As a count shows, this needs about $n^3/3$ operations, about half as many as the Gauss elimination, which needs about $2n^3/3$ (see Sec. 20.1). And once we have (2), we can use it for solving $\mathbf{Ax} = \mathbf{b}$ in two steps, involving only about n^2 operations, simply by noting that $\mathbf{Ax} = \mathbf{LUx} = \mathbf{b}$ may be written

$$\text{(3)} \qquad \text{(a)} \quad \mathbf{Ly} = \mathbf{b} \qquad \text{where} \qquad \text{(b)} \quad \mathbf{Ux} = \mathbf{y}$$

and solving first (3a) for **y** and then (3b) for **x**. Here we can require that **L** have main diagonal $1, \cdots, 1$ as stated before; then this is called **Doolittle's method**.[1] Both systems (3a) and (3b) are triangular, so we can solve them as in the back substitution for the Gauss elimination.

A similar method, **Crout's method**,[2] is obtained from (2) if **U** (instead of **L**) is required to have main diagonal $1, \cdots, 1$. In either case the factorization (2) is unique.

EXAMPLE 1 **Doolittle's Method**

Solve the system in Example 1 of Sec. 20.1 by Doolittle's method.

Solution. The decomposition (2) is obtained from

$$\mathbf{A} = [a_{jk}] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

[1] MYRICK H. DOOLITTLE (1830–1913). American mathematician employed by the U.S. Coast and Geodetic Survey Office. His method appeared in *U.S. Coast and Geodetic Survey*, 1878, 115–120.

[2] PRESCOTT DURAND CROUT (1907–1984), American mathematician, professor at MIT, also worked at General Electric.

by determining the m_{jk} and u_{jk}, using matrix multiplication. By going through **A** row by row we get successively

$a_{11} = 3 = 1 \cdot u_{11} = u_{11}$	$a_{12} = 5 = 1 \cdot u_{12} = u_{12}$	$a_{13} = 2 = 1 \cdot u_{13} = u_{13}$
$a_{21} = 0 = m_{21}u_{11}$	$a_{22} = 8 = m_{21}u_{12} + u_{22}$	$a_{23} = 2 = m_{21}u_{13} + u_{23}$
$m_{21} = 0$	$u_{22} = 8$	$u_{23} = 2$
$a_{31} = 6 = m_{31}u_{11}$	$a_{32} = 2 = m_{31}u_{12} + m_{32}u_{22}$	$a_{33} = 8 = m_{31}u_{13} + m_{32}u_{23} + u_{33}$
$= m_{31} \cdot 3$	$= 2 \cdot 5 + m_{32} \cdot 8$	$= 2 \cdot 2 - 1 \cdot 2 + u_{33}$
$m_{31} = 2$	$m_{32} = -1$	$u_{33} = 6$

Thus the factorization (2) is

$$\begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 6 & 2 & 8 \end{bmatrix} = \mathbf{LU} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{bmatrix}.$$

We first solve $\mathbf{Ly} = \mathbf{b}$, determining $y_1 = 8$, then $y_2 = -7$, then y_3 from $2y_1 - y_2 + y_3 = 16 + 7 + y_3 = 26$; thus (note the interchange in **b** because of the interchange in **A**!)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -7 \\ 26 \end{bmatrix}. \qquad \text{Solution} \qquad \mathbf{y} = \begin{bmatrix} 8 \\ -7 \\ 3 \end{bmatrix}.$$

Then we solve $\mathbf{Ux} = \mathbf{y}$, determining $x_3 = \frac{3}{6}$ then x_2, then x_1, that is,

$$\begin{bmatrix} 3 & 5 & 2 \\ 0 & 8 & 2 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -7 \\ 3 \end{bmatrix}. \qquad \text{Solution} \qquad \mathbf{x} = \begin{bmatrix} 4 \\ -1 \\ \frac{1}{2} \end{bmatrix}.$$

This agrees with the solution in Example 1 of Sec. 20.1. ■

Our formulas in Example 1 suggest that for general n the entries of the matrices $\mathbf{L} = [m_{jk}]$ (with main diagonal $1, \cdots, 1$ and m_{jk} suggesting "multiplier") and $\mathbf{U} = [u_{jk}]$ in the **Doolittle method** are computed from

$$u_{1k} = a_{1k} \qquad k = 1, \cdots, n$$

$$m_{j1} = \frac{a_{j1}}{u_{11}} \qquad j = 2, \cdots, n$$

(4)
$$u_{jk} = a_{jk} - \sum_{s=1}^{j-1} m_{js}u_{sk} \qquad k = j, \cdots, n; \quad j \geqq 2$$

$$m_{jk} = \frac{1}{u_{kk}} \left(a_{jk} - \sum_{s=1}^{k-1} m_{js}u_{sk} \right) \qquad j = k + 1, \cdots, n; \quad k \geqq 2.$$

Row Interchanges. Matrices, such as

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

have no LU-factorization (try!). This indicates that for obtaining an LU-factorization, row interchanges of $\mathbf{A}$ (and corresponding interchanges in $\mathbf{b}$) may be necessary.

Cholesky's Method

For a ***symmetric, positive definite*** matrix $\mathbf{A}$ (thus $\mathbf{A} = \mathbf{A}^T$, $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$) we can in (2) even choose $\mathbf{U} = \mathbf{L}^T$, thus $u_{jk} = m_{kj}$ (but cannot impose conditions on the main diagonal entries). For example,

$$\text{(5)} \qquad \mathbf{A} = \begin{bmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{bmatrix} = \mathbf{L}\mathbf{L}^T = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{bmatrix} \begin{bmatrix} 2 & 1 & 7 \\ 0 & 4 & -3 \\ 0 & 0 & 5 \end{bmatrix}.$$

The popular method of solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ based on this factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ is called **Cholesky's method**.[3] In terms of the entries of $\mathbf{L} = [l_{jk}]$ the formulas for the factorization are

$$\text{(6)} \qquad \begin{aligned} l_{11} &= \sqrt{a_{11}} \\ l_{j1} &= \frac{a_{j1}}{l_{11}} && j = 2, \cdots, n \\ l_{jj} &= \sqrt{a_{jj} - \sum_{s=1}^{j-1} l_{js}^2} && j = 2, \cdots, n \\ l_{pj} &= \frac{1}{l_{jj}} \left(a_{pj} - \sum_{s=1}^{j-1} l_{js} l_{ps} \right) && p = j+1, \cdots, n; \quad j \geqq 2. \end{aligned}$$

If $\mathbf{A}$ is symmetric but not positive definite, this method could still be applied, but then leads to a *complex* matrix $\mathbf{L}$, so that the method becomes impractical.

EXAMPLE 2 **Cholesky's Method**

Solve by Cholesky's method:

$$\begin{aligned} 4x_1 + 2x_2 + 14x_3 &= 14 \\ 2x_1 + 17x_2 - 5x_3 &= -101 \\ 14x_1 - 5x_2 + 83x_3 &= 155. \end{aligned}$$

[3] ANDRÉ-LOUIS CHOLESKY (1875–1918), French military officer, geodecist, and mathematician. Surveyed Crete and North Africa. Died in World War I. His method was published posthumously in *Bulletin Géodésique* in 1924 but received little attention until JOHN TODD (1911–2007) — Irish-American mathematician, numerical analysist, and early pioneer of computer methods in numerics, professor at Caltech, and close personal friend and collaborator of ERWIN KREYSZIG, see [E20]—taught Cholesky's method in his analysis course at King's College, London, in the 1940s.

Solution. From (6) or from the form of the factorization

$$\begin{bmatrix} 4 & 2 & 14 \\ 2 & 17 & -5 \\ 14 & -5 & 83 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

we compute, in the given order,

$$l_{11} = \sqrt{a_{11}} = 2 \qquad l_{21} = \frac{a_{21}}{l_{11}} = \frac{2}{2} = 1 \qquad l_{31} = \frac{a_{31}}{l_{11}} = \frac{14}{2} = 7$$

$$l_{22} = \sqrt{a_{22} - l_{21}^2} = \sqrt{17 - 1} = 4$$

$$l_{32} = \frac{1}{l_{23}}(a_{32} - l_{31}l_{21}) = \frac{1}{4}(-5 - 7 \cdot 1) = -3$$

$$l_{33} = \sqrt{a_{33} - l_{31}^2 - l_{32}^2} = \sqrt{83 - 7^2 - (-3)^2} = 5.$$

This agrees with (5). We now have to solve $\mathbf{Ly} = \mathbf{b}$, that is,

$$\begin{bmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 7 & -3 & 5 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 14 \\ -101 \\ 155 \end{bmatrix}. \qquad \text{Solution} \qquad \mathbf{y} = \begin{bmatrix} 7 \\ -27 \\ 5 \end{bmatrix}.$$

As the second step, we have to solve $\mathbf{Ux} = \mathbf{L}^\mathsf{T}\mathbf{x} = \mathbf{y}$, that is,

$$\begin{bmatrix} 2 & 1 & 7 \\ 0 & 4 & -3 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -27 \\ 5 \end{bmatrix}. \qquad \text{Solution} \qquad \mathbf{x} = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}.$$

■

THEOREM 1

Stability of the Cholesky Factorization

The Cholesky LL^T-factorization is numerically stable (as defined in Sec. 19.1).

PROOF We have $a_{jj} = l_{j1}^2 + l_{j2}^2 + \cdots + l_{jj}^2$ by squaring the third formula in (6) and solving it for a_{jj}. Hence for all l_{jk} (note that $l_{jk} = 0$ for $k > j$) we obtain (the inequality being trivial)

$$l_{jk}^2 \leqq l_{j1}^2 + l_{j2}^2 + \cdots + l_{jj}^2 = a_{jj}.$$

That is, l_{jk}^2 is bounded by an entry of $\mathbf{A}$, which means stability against rounding. ■

Gauss–Jordan Elimination. Matrix Inversion

Another variant of the Gauss elimination is the **Gauss–Jordan elimination**, introduced by W. Jordan in 1920, in which back substitution is avoided by additional computations that reduce the matrix to diagonal form, instead of the triangular form in the Gauss elimination. But this reduction from the Gauss triangular to the diagonal form requires more operations than back substitution does, so that the method is *disadvantageous* for solving systems $\mathbf{Ax} = \mathbf{b}$. But it may be used for matrix inversion, where the situation is as follows.

The **inverse** of a nonsingular square matrix $\mathbf{A}$ may be determined in principle by solving the n systems

$$\mathbf{Ax} = \mathbf{b}_j \qquad (j = 1, \cdots, n) \tag{7}$$

where $\mathbf{b}_j$ is the jth column of the $n \times n$ unit matrix.

However, it is preferable to produce $\mathbf{A}^{-1}$ by operating on the unit matrix $\mathbf{I}$ in the same way as the Gauss–Jordan algorithm, reducing $\mathbf{A}$ to $\mathbf{I}$. A typical illustrative example of this method is given in Sec. 7.8.

PROBLEM SET 20.2

1–5 DOOLITTLE'S METHOD

Show the factorization and solve by Doolittle's method.

1.
$$4x_1 + 5x_2 = 14$$
$$12x_1 + 14x_2 = 36$$

2.
$$2x_1 + 9x_2 = 82$$
$$3x_1 - 5x_2 = -62$$

3.
$$5x_1 + 4x_2 + x_3 = 6.8$$
$$10x_1 + 9x_2 + 4x_3 = 17.6$$
$$10x_1 + 13x_2 + 15x_3 = 38.4$$

4.
$$2x_1 + x_2 + 2x_3 = 0$$
$$-2x_1 + 2x_2 + x_3 = 0$$
$$x_1 + 2x_2 - 2x_3 = 18$$

5.
$$3x_1 + 9x_2 + 6x_3 = 4.6$$
$$18x_1 + 48x_2 + 39x_3 = 27.2$$
$$9x_1 - 27x_2 + 42x_3 = 9.0$$

6. TEAM PROJECT. Crout's method factorizes $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ is lower triangular and $\mathbf{U}$ is upper triangular with diagonal entries $u_{jj} = 1, j = 1, \cdots, n$.

(a) Formulas. Obtain formulas for Crout's method similar to (4).

(b) Examples. Solve Prob. 5 by Crout's method.

(c) Factor the following matrix by the Doolittle, Crout, and Cholesky methods.

$$\begin{bmatrix} 1 & -4 & 2 \\ -4 & 25 & 4 \\ 2 & 4 & 24 \end{bmatrix}$$

(d) Give the formulas for factoring a tridiagonal matrix by Crout's method.

(e) When can you obtain Crout's factorization from Doolittle's by transposition?

7–12 CHOLESKY'S METHOD

Show the factorization and solve.

7.
$$9x_1 + 6x_2 + 12x_3 = 17.4$$
$$6x_1 + 13x_2 + 11x_3 = 23.6$$
$$12x_1 + 11x_2 + 26x_3 = 30.8$$

8.
$$4x_1 + 6x_2 + 8x_3 = 0$$
$$6x_1 + 34x_2 + 52x_3 = -160$$
$$8x_1 + 52x_2 + 129x_3 = -452$$

9.
$$0.01x_1 + 0.03x_3 = 0.14$$
$$0.16x_2 + 0.08x_3 = 0.16$$
$$0.03x_1 + 0.08x_2 + 0.14x_3 = 0.54$$

10.
$$4x_1 + 2x_3 = 1.5$$
$$4x_2 + x_3 = 4.0$$
$$2x_1 + x_2 + 2x_3 = 2.5$$

11.
$$x_1 - x_2 + 3x_3 + 2x_4 = 15$$
$$-x_1 + 5x_2 - 5x_3 - 2x_4 = -35$$
$$3x_1 - 5x_2 + 19x_3 + 3x_4 = 94$$
$$2x_1 - 2x_2 + 3x_3 + 21x_4 = 1$$

12.
$$4x_1 + 2x_2 + 4x_3 = 20$$
$$2x_1 + 2x_2 + 3x_3 + 2x_4 = 36$$
$$4x_1 + 3x_2 + 6x_3 + 3x_4 = 60$$
$$2x_2 + 3x_3 + 9x_4 = 122$$

13. Definiteness. Let $\mathbf{A}, \mathbf{B}$ be $n \times n$ and positive definite. Are $-\mathbf{A}, \mathbf{A}^{\mathsf{T}}, \mathbf{A} + \mathbf{B}, \mathbf{A} - \mathbf{B}$ positive definite?

14. CAS PROJECT. Cholesky's Method. (a) Write a program for solving linear systems by Cholesky's method and apply it to Example 2 in the text, to Probs. 7–9, and to systems of your choice.

(b) Splines. Apply the factorization part of the program to the following matrices (as they occur in (9), Sec. 19.4 (with $c_j = 1$), in connection with splines).

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \quad \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}.$$

15–19 INVERSE

Find the inverse by the Gauss–Jordan method, showing the details.

15. In Prob. 1 **16.** In Prob. 4

17. In Team Project 6(c) **18.** In Prob. 9

19. In Prob. 12

20. Rounding. For the following matrix **A** find det **A**. What happens if you roundoff the given entries to **(a)** 5S, **(b)** 4S, **(c)** 3S, **(d)** 2S, **(e)** lS? What is the practical implication of your work?

$$\mathbf{A} = \begin{bmatrix} \frac{1}{3} & \frac{1}{4} & 2 \\ -\frac{1}{9} & 1 & \frac{1}{7} \\ \frac{4}{63} & -\frac{3}{28} & \frac{13}{49} \end{bmatrix}$$

20.3 Linear Systems: Solution by Iteration

The Gauss elimination and its variants in the last two sections belong to the **direct methods** for solving linear systems of equations; these are methods that give solutions after an amount of computation that can be specified in advance. In contrast, in an **indirect** or **iterative method** we start from an approximation to the true solution and, if successful, obtain better and better approximations from a computational cycle repeated as often as may be necessary for achieving a required accuracy, so that the amount of arithmetic depends upon the accuracy required and varies from case to case.

We apply iterative methods if the convergence is rapid (if matrices have large main diagonal entries, as we shall see), so that we save operations compared to a direct method. We also use iterative methods if a large system is **sparse**, that is, has very many zero coefficients, so that one would waste space in storing zeros, for instance, 9995 zeros per equation in a potential problem of 10^4 equations in 10^4 unknowns with typically only 5 nonzero terms per equation (more on this in Sec. 21.4).

Gauss–Seidel Iteration Method[4]

This is an iterative method of great practical importance, which we can simply explain in terms of an example.

EXAMPLE 1 Gauss–Seidel Iteration

We consider the linear system

$$\begin{aligned} x_1 - 0.25x_2 - 0.25x_3 &= 50 \\ -0.25x_1 + x_2 - 0.25x_4 &= 50 \\ -0.25x_1 + x_3 - 0.25x_4 &= 25 \\ -0.25x_2 - 0.25x_3 + x_4 &= 25. \end{aligned} \tag{1}$$

[4]PHILIPP LUDWIG VON SEIDEL (1821–1896), German mathematician. For Gauss see footnote 5 in Sec. 5.4.

(Equations of this form arise in the numeric solution of PDEs and in spline interpolation.) We write the system in the form

$$
\begin{aligned}
x_1 &= 0.25x_2 + 0.25x_3 + 50 \\
x_2 &= 0.25x_1 + 0.25x_4 + 50 \\
x_3 &= 0.25x_1 + 0.25x_4 + 25 \\
x_4 &= 0.25x_2 + 0.25x_3 + 25.
\end{aligned} \tag{2}
$$

These equations are now used for iteration; that is, we start from a (possibly poor) approximation to the solution, say $x_1^{(0)} = 100, x_2^{(0)} = 100, x_3^{(0)} = 100, x_4^{(0)} = 100$, and compute from (2) a perhaps better approximation

Use "old" values
("New" values here not yet available)

$$
\begin{aligned}
x_1^{(1)} &= \phantom{0.25x_1^{(1)} +\ } 0.25x_2^{(0)} + 0.25x_3^{(0)} \phantom{+\ 0.25x_4^{(0)}} + 50.00 = 100.00 \\
x_2^{(1)} &= 0.25x_1^{(1)} \phantom{+\ 0.25x_2^{(1)} +\ } + 0.25x_4^{(0)} + 50.00 = 100.00 \\
x_3^{(1)} &= 0.25x_1^{(1)} \phantom{+\ 0.25x_2^{(1)} +\ } + 0.25x_4^{(0)} + 25.00 = 75.00 \\
x_4^{(1)} &= \phantom{0.25x_1^{(1)} +\ } 0.25x_2^{(1)} + 0.25x_3^{(1)} \phantom{+\ 0.25x_4^{(1)}} + 25.00 = 68.75
\end{aligned} \tag{3}
$$

Use "new" values

These equations (3) are obtained from (2) by substituting on the right the ***most recent*** approximation for each unknown. In fact, corresponding values replace previous ones as soon as they have been computed, so that in the second and third equations we use $x_1^{(1)}$ (not $x_1^{(0)}$), and in the last equation of (3) we use $x_2^{(1)}$ and $x_3^{(1)}$ (not $x_2^{(0)}$ and $x_3^{(0)}$). Using the same principle, we obtain in the next step

$$
\begin{aligned}
x_1^{(2)} &= \phantom{0.25x_1^{(2)} +\ } 0.25x_2^{(1)} + 0.25x_3^{(1)} \phantom{+\ 0.25x_4^{(1)}} + 50.00 = 93.750 \\
x_2^{(2)} &= 0.25x_1^{(2)} \phantom{+\ 0.25x_2^{(2)} +\ } + 0.25x_4^{(1)} + 50.00 = 90.625 \\
x_3^{(2)} &= 0.25x_1^{(2)} \phantom{+\ 0.25x_2^{(2)} +\ } + 0.25x_4^{(1)} + 25.00 = 65.625 \\
x_4^{(2)} &= \phantom{0.25x_1^{(2)} +\ } 0.25x_2^{(2)} + 0.25x_3^{(2)} \phantom{+\ 0.25x_4^{(2)}} + 25.00 = 64.062
\end{aligned}
$$

Further steps give the values

x_1	x_2	x_3	x_4
89.062	88.281	63.281	62.891
87.891	87.695	62.695	62.598
87.598	87.549	62.549	62.524
87.524	87.512	62.512	62.506
87.506	87.503	62.503	62.502

Hence convergence to the exact solution $x_1 = x_2 = 87.5, x_3 = x_4 = 62.5$ (verify!) seems rather fast. ■

An algorithm for the Gauss–Seidel iteration is shown in Table 20.2. To obtain the algorithm, let us derive the general formulas for this iteration.

We assume that $a_{jj} = 1$ for $j = 1, \cdots, n$. (Note that this can be achieved if we can rearrange the equations so that no diagonal coefficient is zero; then we may divide each equation by the corresponding diagonal coefficient.) We now write

(4) $$\mathbf{A} = \mathbf{I} + \mathbf{L} + \mathbf{U} \qquad (a_{jj} = 1)$$

where $\mathbf{I}$ is the $n \times n$ unit matrix and $\mathbf{L}$ and $\mathbf{U}$ are, respectively, lower and upper triangular matrices with zero main diagonals. If we substitute (4) into $\mathbf{Ax} = \mathbf{b}$, we have

$$\mathbf{Ax} = (\mathbf{I} + \mathbf{L} + \mathbf{U})\mathbf{x} = \mathbf{b}.$$

Taking $\mathbf{Lx}$ and $\mathbf{Ux}$ to the right, we obtain, since $\mathbf{Ix} = \mathbf{x}$,

(5) $$\mathbf{x} = \mathbf{b} - \mathbf{Lx} - \mathbf{Ux}.$$

Remembering from (3) in Example 1 that below the main diagonal we took "new" approximations and above the main diagonal "old" ones, we obtain from (5) the desired iteration formulas

"New" "Old"

(6) $$\mathbf{x}^{(m+1)} = \mathbf{b} - \mathbf{L}\mathbf{x}^{(m+1)} - \mathbf{U}\mathbf{x}^{(m)} \qquad (a_{jj} = 1)$$

where $\mathbf{x}^{(m)} = [x_j^{(m)}]$ is the mth approximation and $\mathbf{x}^{(m+1)} = [x_j^{(m+1)}]$ is the $(m + 1)$st approximation. In components this gives the formula in line 1 in Table 20.2. The matrix $\mathbf{A}$ must satisfy $a_{jj} \neq 0$ for all j. In Table 20.2 our assumption $a_{jj} = 1$ is no longer required, but is automatically taken care of by the factor $1/a_{jj}$ in line 1.

Table 20.2 Gauss–Seidel Iteration

ALGORITHM GAUSS–SEIDEL ($\mathbf{A}$, $\mathbf{b}$, $\mathbf{x}^{(0)}$, ϵ, N)
This algorithm computes a solution $\mathbf{x}$ of the system $\mathbf{Ax} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$, where $\mathbf{A} = [a_{jk}]$ is an $n \times n$ matrix with $a_{jj} \neq 0$, $j = 1, \cdots, n$.

INPUT: **A, b,** initial approximation $\mathbf{x}^{(0)}$, tolerance $\epsilon > 0$, maximum number of iterations N

OUTPUT: Approximate solution $\mathbf{x}^{(m)} = [x_j^{(m)}]$ or failure message that $\mathbf{x}^{(N)}$ does not satisfy the tolerance condition

For $m = 0, \cdots, N - 1$, do:
  For $j = 1, \cdots, n$, do:
1 $$x_j^{(m+1)} = \frac{1}{a_{jj}}\left(b_j - \sum_{k=1}^{j-1} a_{jk}x_k^{(m+1)} - \sum_{k=j+1}^{n} a_{jk}x_k^{(m)}\right)$$
  End
2 If $\max_j |x_j^{(m+1)} - x_j^{(m)}| < \epsilon\, |x_j^{(m+1)}|$ then OUTPUT $\mathbf{x}^{(m+1)}$. Stop
    [*Procedure completed successfully*]
End
OUTPUT: "No solution satisfying the tolerance condition obtained after N iteration steps." Stop
[*Procedure completed unsuccessfully*]
End GAUSS–SEIDEL

Convergence and Matrix Norms

An iteration method for solving $\mathbf{Ax} = \mathbf{b}$ is said to **converge** for an initial $\mathbf{x}^{(0)}$ if the corresponding iterative sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots$ converges to a solution of the given system. Convergence depends on the relation between $\mathbf{x}^{(m)}$ and $\mathbf{x}^{(m+1)}$. To get this relation for the Gauss–Seidel method, we use (6). We first have

$$(\mathbf{I} + \mathbf{L})\,\mathbf{x}^{(m+1)} = \mathbf{b} - \mathbf{U}\mathbf{x}^{(m)}$$

and by multiplying by $(\mathbf{I} + \mathbf{L})^{-1}$ from the left,

$$\textbf{(7)} \qquad \mathbf{x}^{(m+1)} = \mathbf{C}\mathbf{x}^{(m)} + (\mathbf{I} + \mathbf{L})^{-1}\mathbf{b} \qquad \text{where} \qquad \mathbf{C} = -(\mathbf{I} + \mathbf{L})^{-1}\mathbf{U}.$$

The Gauss–Seidel iteration converges for every $\mathbf{x}^{(0)}$ if and only if all the eigenvalues (Sec. 8.1) of the "iteration matrix" $\mathbf{C} = [c_{jk}]$ have absolute value less than 1. (Proof in Ref. [E5], p. 191, listed in App. 1.)

CAUTION! If you want to get $\mathbf{C}$, first divide the rows of $\mathbf{A}$ by a_{jj} to have main diagonal $1, \cdots, 1$. If the **spectral radius** of $\mathbf{C}$ (= maximum of those absolute values) is small, then the convergence is rapid.

Sufficient Convergence Condition. A sufficient condition for convergence is

$$\textbf{(8)} \qquad \|\mathbf{C}\| < 1.$$

Here $\|\mathbf{C}\|$ is some **matrix norm**, such as

$$(9) \qquad \|\mathbf{C}\| = \sqrt{\sum_{j=1}^{n}\sum_{k=1}^{n} c_{jk}^2} \qquad \textbf{(Frobenius norm)}$$

or the greatest of the sums of the $|c_{jk}|$ in a *column* of $\mathbf{C}$

$$(10) \qquad \|\mathbf{C}\| = \max_k \sum_{j=1}^{n} |c_{jk}| \qquad \textbf{(Column "sum" norm)}$$

or the greatest of the sums of the $|c_{jk}|$ in a *row* of $\mathbf{C}$

$$(11) \qquad \|\mathbf{C}\| = \max_j \sum_{k=1}^{n} |c_{jk}| \qquad \textbf{(Row "sum" norm)}.$$

These are the most frequently used matrix norms in numerics.

In most cases the choice of one of these norms is a matter of computational convenience. However, the following example shows that sometimes one of these norms is preferable to the others.

EXAMPLE 2 Test of Convergence of the Gauss–Seidel Iteration

Test whether the Gauss–Seidel iteration converges for the system

$$\begin{aligned} 2x + \ y + \ z &= 4 \\ x + 2y + \ z &= 4 \\ x + \ y + 2z &= 4 \end{aligned} \qquad \text{written} \qquad \begin{aligned} x &= 2 - \tfrac{1}{2}y - \tfrac{1}{2}z \\ y &= 2 - \tfrac{1}{2}x - \tfrac{1}{2}z \\ z &= 2 - \tfrac{1}{2}x - \tfrac{1}{2}y. \end{aligned}$$

Solution. The decomposition (multiply the matrix by $\frac{1}{2}$ – why?) is

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} = \mathbf{I} + \mathbf{L} + \mathbf{U} = \mathbf{I} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix}.$$

It shows that

$$\mathbf{C} = -(\mathbf{I} + \mathbf{L})^{-1}\mathbf{U} = -\begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{4} & -\frac{1}{2} & 1 \end{bmatrix}\begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{4} & -\frac{1}{4} \\ 0 & \frac{1}{8} & \frac{3}{8} \end{bmatrix}.$$

We compute the Frobenius norm of $\mathbf{C}$

$$\|\mathbf{C}\| = \left(\tfrac{1}{4} + \tfrac{1}{4} + \tfrac{1}{16} + \tfrac{1}{16} + \tfrac{1}{64} + \tfrac{9}{64}\right)^{1/2} = \left(\tfrac{50}{64}\right)^{1/2} = 0.884 < 1$$

and conclude from (8) that this Gauss–Seidel iteration converges. It is interesting that the other two norms would permit no conclusion, as you should verify. Of course, this points to the fact that (8) is sufficient for convergence rather than necessary. ■

Residual. Given a system $\mathbf{Ax} = \mathbf{b}$, the **residual** $\mathbf{r}$ of $\mathbf{x}$ with respect to this system is defined by

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax}. \tag{12}$$

Clearly, $\mathbf{r} = \mathbf{0}$ if and only if $\mathbf{x}$ is a solution. Hence $\mathbf{r} \neq \mathbf{0}$ for an approximate solution. In the Gauss–Seidel iteration, at each stage we modify or *relax* a component of an approximate solution in order to reduce a component of $\mathbf{r}$ to zero. Hence the Gauss–Seidel iteration belongs to a class of methods often called **relaxation methods**. More about the residual follows in the next section.

Jacobi Iteration

The Gauss–Seidel iteration is a method of **successive corrections** because for each component we successively replace an approximation of a component by a corresponding new approximation as soon as the latter has been computed. An iteration method is called a method of **simultaneous corrections** if no component of an approximation $\mathbf{x}^{(m)}$ is used until ***all*** the components of $\mathbf{x}^{(m)}$ have been computed. A method of this type is the **Jacobi iteration**, which is similar to the Gauss–Seidel iteration but involves *not* using improved values until a step has been completed and then replacing $\mathbf{x}^{(m)}$ by $\mathbf{x}^{(m+1)}$ at once, directly before the beginning of the next step. Hence if we write $\mathbf{Ax} = \mathbf{b}$ (***with*** $a_{jj} = 1$ ***as before!***) in the form $\mathbf{x} = \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}$, the Jacobi iteration in matrix notation is

$$\mathbf{x}^{(m+1)} = \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}^{(m)} \qquad (a_{jj} = 1). \tag{13}$$

This method converges for every choice of $\mathbf{x}^{(0)}$ if and only if the spectral radius of $\mathbf{I} - \mathbf{A}$ is less than 1. It has recently gained greater practical interest since on parallel processors all n equations can be solved simultaneously at each iteration step.

For Jacobi, see Sec. 10.3. For exercises, see the problem set.

PROBLEM SET 20.3

1. Verify the solution in Example 1 of the text.

2. Show that for the system in Example 2 the Jacobi iteration diverges. *Hint.* Use eigenvalues.

3. Verify the claim at the end of Example 2.

4–10 GAUSS–SEIDEL ITERATION

Do 5 steps, starting from $\mathbf{x}_0 = [1 \quad 1 \quad 1]^T$ and using 6S in the computation. *Hint.* Make sure that you solve each equation for the variable that has the largest coefficient (why?). Show the details.

4.
$$\begin{aligned} 4x_1 - x_2 &= 21 \\ -x_1 + 4x_2 - x_3 &= -45 \\ -x_2 + 4x_3 &= 33 \end{aligned}$$

5.
$$\begin{aligned} 10x_1 + x_2 + x_3 &= 6 \\ x_1 + 10x_2 + x_3 &= 6 \\ x_1 + x_2 + 10x_3 &= 6 \end{aligned}$$

6.
$$\begin{aligned} x_2 + 7x_3 &= 25.5 \\ 5x_1 + x_2 &= 0 \\ x_1 + 6x_2 + x_3 &= -10.5 \end{aligned}$$

7.
$$\begin{aligned} 5x_1 - 2x_2 &= 18 \\ -2x_1 + 10x_2 - 2x_3 &= -60 \\ -2x_2 + 15x_3 &= 128 \end{aligned}$$

8.
$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 7 \\ x_1 + 3x_2 + 2x_3 &= 4 \\ 2x_1 + x_2 + 3x_3 &= 7 \end{aligned}$$

9.
$$\begin{aligned} 5x_1 + x_2 + 2x_3 &= 19 \\ x_1 + 4x_2 - 2x_3 &= -2 \\ 2x_1 + 3x_2 + 8x_3 &= 39 \end{aligned}$$

10.
$$\begin{aligned} 4x_1 + 5x_3 &= 12.5 \\ x_1 + 6x_2 + 2x_3 &= 18.5 \\ 8x_1 + 2x_2 + x_3 &= -11.5 \end{aligned}$$

11. Apply the Gauss–Seidel iteration (3 steps) to the system in Prob. 5, starting from **(a)** 0, 0, 0 **(b)** 10, 10, 10. Compare and comment.

12. In Prob. 5, compute $\mathbf{C}$ **(a)** if you solve the first equation for x_1, the second for x_2, the third for x_3, proving convergence; **(b)** if you nonsensically solve the third equation for x_1, the first for x_2, the second for x_3, proving divergence.

13. CAS Experiment. Gauss–Seidel Iteration. (a) Write a program for Gauss–Seidel iteration.

(b) Apply the program $\mathbf{A}(t)\mathbf{x} = \mathbf{b}$, to starting from $[0 \quad 0 \quad 0]^T$, where

$$\mathbf{A}(t) = \begin{bmatrix} 1 & t & t \\ t & 1 & t \\ t & t & 1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}.$$

For $t = 0.2, 0.5, 0.8, 0.9$ determine the number of steps to obtain the exact solution to 6S and the corresponding spectral radius of $\mathbf{C}$. Graph the number of steps and the spectral radius as functions of t and comment.

(c) Successive overrelaxation (SOR). Show that by adding and subtracting $\mathbf{x}^{(m)}$ on the right, formula (6) can be written

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \mathbf{b} - \mathbf{L}\mathbf{x}^{(m+1)} - (\mathbf{U} + \mathbf{I})\mathbf{x}^{(m)} \qquad (a_{jj} = 1).$$

Anticipation of further corrections motivates the introduction of an **overrelaxation factor** $\omega > 1$ to get the **SOR formula for Gauss–Seidel**

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \omega(\mathbf{b} - \mathbf{L}\mathbf{x}^{(m+1)} - (\mathbf{U} + \mathbf{I})\mathbf{x}^{(m)}) \qquad (a_{jj} = 1) \tag{14}$$

intended to give more rapid convergence. A recommended value is $\omega = 2/(1 + \sqrt{1 - \rho})$, where ρ is the spectral radius of $\mathbf{C}$ in (7). Apply SOR to the matrix in (b) for $t = 0.5$ and 0.8 and notice the improvement of convergence. (Spectacular gains are made with larger systems.)

14–17 JACOBI ITERATION

Do 5 steps, starting from $\mathbf{x}_0 = [1 \quad 1 \quad 1]$. Compare with the Gauss–Seidel iteration. Which of the two seems to converge faster? Show the details of your work.

14. The system in Prob. 4

15. The system in Prob. 9

16. The system in Prob. 10

17. Show convergence in Prob. 16 by verifying that $\mathbf{I} - \mathbf{A}$, where $\mathbf{A}$ is the matrix in Prob. 16 with the rows divided by the corresponding main diagonal entries, has the eigenvalues -0.519589 and $0.259795 \pm 0.246603i$.

18–20 NORMS

Compute the norms (9), (10), (11) for the following (square) matrices. Comment on the reasons for greater or smaller differences among the three numbers.

18. The matrix in Prob. 10

19. The matrix in Prob. 5

20.
$$\begin{bmatrix} 2k & -k & -k \\ k & -2k & k \\ -k & -k & 2k \end{bmatrix}$$

20.4 Linear Systems: Ill-Conditioning, Norms

One does not need much experience to observe that some systems $\mathbf{Ax} = \mathbf{b}$ are good, giving accurate solutions even under roundoff or coefficient inaccuracies, whereas others are bad, so that these inaccuracies affect the solution strongly. We want to see what is going on and whether or not we can "trust" a linear system. Let us first formulate the two relevant concepts (ill- and well-conditioned) for general numeric work and then turn to linear systems and matrices.

A computational problem is called **ill-conditioned** (or *ill-posed*) if "small" changes in the data (the input) cause "large" changes in the solution (the output). On the other hand, a problem is called **well-conditioned** (or *well-posed*) if "small" changes in the data cause only "small" changes in the solution.

These concepts are qualitative. We would certainly regard a magnification of inaccuracies by a factor 100 as "large," but could debate where to draw the line between "large" and "small," depending on the kind of problem and on our viewpoint. Double precision may sometimes help, but if data are measured inaccurately, one should attempt ***changing the mathematical setting*** of the problem to a well-conditioned one.

Let us now turn to linear systems. Figure 445 explains that ill-conditioning occurs if and only if the two equations give two nearly parallel lines, so that their intersection point (the solution of the system) moves substantially if we raise or lower a line just a little. For larger systems the situation is similar in principle, although geometry no longer helps. We shall see that we may regard ill-conditioning as an approach to singularity of the matrix.

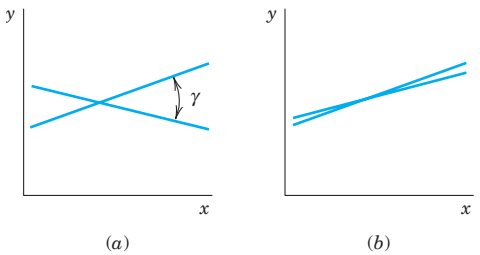

Fig. 445. (a) Well-conditioned and (b) ill-conditioned linear system of two equations in two unknowns

EXAMPLE 1 An Ill-Conditioned System

You may verify that the system

$$0.9999x - 1.0001y = 1$$
$$x - y = 1$$

has the solution $x = 0.5, y = -0.5$, whereas the system

$$0.9999x - 1.0001y = 1$$
$$x - y = 1 + \epsilon$$

has the solution $x = 0.5 + 5000.5\epsilon, y = -0.5 + 4999.5\epsilon$. This shows that the system is ill-conditioned because a change on the right of magnitude ϵ produces a change in the solution of magnitude 5000ϵ, approximately. We see that the lines given by the equations have nearly the same slope. ■

Well-conditioning can be asserted if the main diagonal entries of $\mathbf{A}$ have large absolute values compared to those of the other entries. Similarly if $\mathbf{A}^{-1}$ and $\mathbf{A}$ have maximum entries of about the same absolute value.

Ill-conditioning is indicated if $\mathbf{A}^{-1}$ has entries of large absolute value compared to those of the solution (about 5000 in Example 1) and if poor approximate solutions may still produce small residuals.

Residual. The *residual* $\mathbf{r}$ of an approximate solution $\tilde{\mathbf{x}}$ of $\mathbf{Ax} = \mathbf{b}$ is defined as

(1) $$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}.$$

Now $\mathbf{b} = \mathbf{Ax}$, so that

(2) $$\mathbf{r} = \mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}).$$

Hence $\mathbf{r}$ is small if $\tilde{\mathbf{x}}$ has high accuracy, but the converse may be false:

EXAMPLE 2 Inaccurate Approximate Solution with a Small Residual

The system

$$1.0001x_1 + x_2 = 2.0001$$
$$x_1 + 1.0001x_2 = 2.0001$$

has the exact solution $x_1 = 1, x_2 = 1$. Can you see this by inspection? The very inaccurate approximation $\tilde{x}_1 = 2.0000, \tilde{x}_2 = 0.0001$ has the very small residual (to 4D)

$$\mathbf{r} = \begin{bmatrix} 2.0001 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 1.0001 & 1.0000 \\ 1.0000 & 1.0001 \end{bmatrix} \begin{bmatrix} 2.0000 \\ 0.0001 \end{bmatrix} = \begin{bmatrix} 2.0001 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 2.0003 \\ 2.0001 \end{bmatrix} = \begin{bmatrix} -0.0002 \\ 0.0000 \end{bmatrix}.$$

From this, a naive person might draw the false conclusion that the approximation should be accurate to 3 or 4 decimals.

Our result is probably unexpected, but we shall see that it has to do with the fact that the system is ill-conditioned. ■

Our goal is to show that ill-conditioning of a linear system and of its coefficient matrix $\mathbf{A}$ can be measured by a number, the *condition number* $\kappa(\mathbf{A})$. Other measures for ill-conditioning

have also been proposed, but $\kappa(\mathbf{A})$ is probably the most widely used one. $\kappa(\mathbf{A})$ is defined in terms of norm, a concept of great general interest throughout numerics (and in modern mathematics in general!). We shall reach our goal in three steps, discussing

1. **Vector norms**
2. **Matrix norms**
3. **Condition number** κ of a square matrix

Vector Norms

A **vector norm** for column vectors $\mathbf{x} = [x_j]$ with n components (n fixed) is a generalized length or distance. It is denoted by $\|\mathbf{x}\|$ and is defined by four properties of the usual length of vectors in three-dimensional space, namely,

(3)
(a) $\|\mathbf{x}\|$ is a nonnegative real number.
(b) $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$.
(c) $\|k\mathbf{x}\| = |k|\,\|\mathbf{x}\|$ for all k.
(d) $\|\mathbf{x} + \mathbf{y}\| \leqq \|\mathbf{x}\| + \|\mathbf{y}\|$ (Triangle inequality).

If we use several norms, we label them by a subscript. Most important in connection with computations is the ***p-norm*** defined by

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p} \tag{4}$$

where p is a fixed number and $p \geqq 1$. In practice, one usually takes $p = 1$ or 2 and, as a third norm, $\|\mathbf{x}\|_\infty$ (the latter as defined below), that is,

$$\|\mathbf{x}\|_1 = |x_1| + \cdots + |x_n| \quad (\text{“}l_1\text{-norm”}) \tag{5}$$

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2} \quad (\text{“Euclidean” or “}l_2\text{-norm”}) \tag{6}$$

$$\|\mathbf{x}\|_\infty = \max_j |x_j| \quad (\text{“}l_\infty\text{-norm”}). \tag{7}$$

For $n = 3$ the l_2-norm is the usual length of a vector in three-dimensional space. The l_1-norm and l_∞-norm are generally more convenient in computation. But all three norms are in common use.

EXAMPLE 3 **Vector Norms**

If $\mathbf{x}^T = [2 \quad -3 \quad 0 \quad 1 \quad -4]$, then $\|\mathbf{x}\|_1 = 10$, $\|\mathbf{x}\|_2 = \sqrt{30}$, $\|\mathbf{x}\|_\infty = 4$. ■

In three-dimensional space, two points with position vectors $\mathbf{x}$ and $\tilde{\mathbf{x}}$ have distance $|\mathbf{x} - \tilde{\mathbf{x}}|$ from each other. For a linear system $\mathbf{Ax} = \mathbf{b}$, this suggests that we take $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ as a measure of inaccuracy and call it the **distance** between an exact and an approximate solution, or the **error** of $\tilde{\mathbf{x}}$.

Matrix Norm

If $\mathbf{A}$ is an $n \times n$ matrix and $\mathbf{x}$ any vector with n components, then $\mathbf{Ax}$ is a vector with n components. We now take a vector norm and consider $\|\mathbf{x}\|$ and $\|\mathbf{Ax}\|$. One can prove (see

Ref. [E17]. pp. 77, 92–93, listed in App. 1) that there is a number c (depending on $\mathbf{A}$) such that

$$\|\mathbf{Ax}\| \leqq c\|\mathbf{x}\| \qquad \text{for all } \mathbf{x}. \tag{8}$$

Let $\mathbf{x} \neq 0$. Then $\|\mathbf{x}\| > 0$ by (3b) and division gives $\|\mathbf{Ax}\|/\|\mathbf{x}\| \leqq c$. We obtain the smallest possible c valid for *all* $\mathbf{x}$ ($\neq \mathbf{0}$) by taking the maximum on the left. This smallest c is called the **matrix norm of A** *corresponding to the vector norm we picked* and is denoted by $\|\mathbf{A}\|$. Thus

$$\|\mathbf{A}\| = \max \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} \qquad (\mathbf{x} \neq \mathbf{0}), \tag{9}$$

the maximum being taken over all $\mathbf{x} \neq \mathbf{0}$. Alternatively [see (c) in Team Project 24],

$$\|\mathbf{A}\| = \max_{\|\mathbf{x}\|=1} \|\mathbf{Ax}\|. \tag{10}$$

The maximum in (10) and thus also in (9) exists. And the name "matrix ***norm***" is justified because $\|\mathbf{A}\|$ satisfies (3) with $\mathbf{x}$ and $\mathbf{y}$ replaced by $\mathbf{A}$ and $\mathbf{B}$. (Proofs in Ref. [E17] pp. 77, 92–93.)

Note carefully that $\|\mathbf{A}\|$ depends on the vector norm that we selected. In particular, one can show that

for the l_1-norm (5) one gets the column "sum" norm (10), Sec. 20.3,

for the l_∞-norm (7) one gets the row "sum" norm (11), Sec. 20.3.

By taking our best possible (our smallest) $c = \|\mathbf{A}\|$ we have from (8)

$$\|\mathbf{Ax}\| \leqq \|\mathbf{A}\|\,\|\mathbf{x}\|. \tag{11}$$

This is the formula we shall need. Formula (9) also implies for two $n \times n$ matrices (see Ref. [E17], p. 98)

$$\|\mathbf{AB}\| \leqq \|\mathbf{A}\|\,\|\mathbf{B}\|, \qquad \text{thus} \qquad \|\mathbf{A}^n\| \leqq \|\mathbf{A}\|^n. \tag{12}$$

See Refs. [E9] and [E17] for other useful formulas on norms.

Before we go on, let us do a simple illustrative computation.

EXAMPLE 4 Matrix Norms

Compute the matrix norms of the coefficient matrix $\mathbf{A}$ in Example 1 and of its inverse $\mathbf{A}^{-1}$, assuming that we use (a) the l_1-vector norm, (b) the l_∞-vector norm.

Solution. We use (4*), Sec. 7.8, for the inverse and then (10) and (11) in Sec. 20.3. Thus

$$\mathbf{A} = \begin{bmatrix} 0.9999 & -1.0001 \\ 1.0000 & -1.0000 \end{bmatrix}, \qquad \mathbf{A}^{-1} = \begin{bmatrix} -5000.0 & 5000.5 \\ -5000.0 & 4999.5 \end{bmatrix}.$$

(a) The l_1-vector norm gives the column "sum" norm (10), Sec. 20.3; from Column 2 we thus obtain $\|\mathbf{A}\| = |-1.0001| + |-1.0000| = 2.0001$. Similarly, $\|\mathbf{A}^{-1}\| = 10{,}000$.

(b) The l_∞-vector norm gives the row "sum" norm (11), Sec. 20.3; thus $\|\mathbf{A}\| = 2, \|\mathbf{A}^{-1}\| = 10000.5$ from Row 1. We notice that $\|\mathbf{A}^{-1}\|$ is surprisingly large, which makes the product $\|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|$ large (20,001). We shall see below that this is typical of an ill-conditioned system.

Condition Number of a Matrix

We are now ready to introduce the key concept in our discussion of ill-conditioning, the **condition number** $\kappa(\mathbf{A})$ of a (nonsingular) square matrix $\mathbf{A}$, defined by

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|. \tag{13}$$

The role of the condition number is seen from the following theorem.

THEOREM 1

Condition Number

A linear system of equations $\mathbf{Ax} = \mathbf{b}$ *and its matrix* $\mathbf{A}$ *whose condition number* (13) *is small are well-conditioned. A large condition number indicates ill-conditioning.*

PROOF $\mathbf{b} = \mathbf{Ax}$ and (11) give $\|\mathbf{b}\| \leqq \|\mathbf{A}\|\,\|\mathbf{x}\|$. Let $\mathbf{b} \neq \mathbf{0}$ and $\mathbf{x} \neq \mathbf{0}$. Then division by $\|\mathbf{b}\|\,\|\mathbf{x}\|$ gives

$$\frac{1}{\|\mathbf{x}\|} \leqq \frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}. \tag{14}$$

Multiplying (2) $\mathbf{r} = \mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}})$ by $\mathbf{A}^{-1}$ from the left and interchanging sides, we have $\mathbf{x} - \tilde{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{r}$. Now (11) with $\mathbf{A}^{-1}$ and $\mathbf{r}$ instead of $\mathbf{A}$ and $\mathbf{x}$ yields

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| = \|\mathbf{A}^{-1}\mathbf{r}\| \leqq \|\mathbf{A}^{-1}\|\,\|\mathbf{r}\|.$$

Division by $\|\mathbf{x}\|$ [note that $\|\mathbf{x}\| \neq 0$ by (3b)] and use of (14) finally gives

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leqq \frac{1}{\|\mathbf{x}\|}\,\|\mathbf{A}^{-1}\|\,\|\mathbf{r}\| \leqq \frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}\,\|\mathbf{A}^{-1}\|\,\|\mathbf{r}\| = \kappa(\mathbf{A})\,\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}. \tag{15}$$

Hence if $\kappa(\mathbf{A})$ is small, a small $\|\mathbf{r}\|/\|\mathbf{b}\|$ implies a small relative error $\|\mathbf{x} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\|$, so that the system is well-conditioned. However, this does not hold if $\kappa(\mathbf{A})$ is large; then a small $\|\mathbf{r}\|/\|\mathbf{b}\|$ does not necessarily imply a small relative error $\|\mathbf{x} - \tilde{\mathbf{x}}\|/\|\mathbf{x}\|$. ■

EXAMPLE 5 **Condition Numbers. Gauss–Seidel Iteration**

$$\mathbf{A} = \begin{bmatrix} 5 & 1 & 1 \\ 1 & 4 & 2 \\ 1 & 2 & 4 \end{bmatrix} \quad \text{has the inverse} \quad \mathbf{A}^{-1} = \frac{1}{56}\begin{bmatrix} 12 & -2 & -2 \\ -2 & 19 & -9 \\ -2 & -9 & 19 \end{bmatrix}.$$

Since $\mathbf{A}$ is symmetric, (10) and (11) in Sec. 20.3 give the same condition number

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\| = 7 \cdot \tfrac{1}{56} \cdot 30 = 3.75.$$

We see that a linear system $\mathbf{Ax} = \mathbf{b}$ with this $\mathbf{A}$ is well-conditioned.

For instance, if $\mathbf{b} = [14 \quad 0 \quad 28]^T$, the Gauss algorithm gives the solution $\mathbf{x} = [2 \quad -5 \quad 9]^T$, (confirm this). Since the main diagonal entries of $\mathbf{A}$ are relatively large, we can expect reasonably good convergence of the Gauss–Seidel iteration. Indeed, starting from, say, $\mathbf{x}_0 = [1 \quad 1 \quad 1]^T$, we obtain the first 8 steps (3D values)

x_1	x_2	x_3
1.000	1.000	1.000
2.400	−1.100	6.950
1.630	−3.882	8.534
1.870	−4.734	8.900
1.967	−4.942	8.979
1.993	−4.988	8.996
1.998	−4.997	8.999
2.000	−5.000	9.000
2.000	−5.000	9.000

■

EXAMPLE 6 Ill-Conditioned Linear System

Example 4 gives by (10) or (11), Sec. 20.3, for the matrix in Example 1 the very large condition number $\kappa(\mathbf{A}) = 2.0001 \cdot 10000 = 2 \cdot 10000.5 = 200001$. This confirms that the system is very ill-conditioned.

Similarly in Example 2, where by (4*), Sec. 7.8 and 6D-computation,

$$\mathbf{A}^{-1} = \frac{1}{0.0002}\begin{bmatrix} 1.0001 & -1.0000 \\ -1.0000 & 1.0001 \end{bmatrix} = \begin{bmatrix} 5000.5 & -5.000.0 \\ -5000.0 & 5000.5 \end{bmatrix}$$

so that (10), Sec. 20.3, gives a very large $\kappa(\mathbf{A})$, explaining the surprising result in Example 2,

$$\kappa(\mathbf{A}) = (1.0001 + 1.0000)(5000.5 + 5000.0) \approx 20{,}002.$$

■

In practice, $\mathbf{A}^{-1}$ will not be known, so that in computing the condition number $\kappa(\mathbf{A})$, one must estimate $\|\mathbf{A}^{-1}\|$. A method for this (proposed in 1979) is explained in Ref. [E9] listed in App. 1.

Inaccurate Matrix Entries. $\kappa(\mathbf{A})$ can be used for estimating the effect $\delta\mathbf{x}$ of an inaccuracy $\delta\mathbf{A}$ of $\mathbf{A}$ (errors of measurements of the a_{jk}, for instance). Instead of $\mathbf{Ax} = \mathbf{b}$ we then have

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b}.$$

Multiplying out and subtracting $\mathbf{Ax} = \mathbf{b}$ on both sides, we obtain

$$\mathbf{A}\delta\mathbf{x} + \delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{0}.$$

Multiplication by $\mathbf{A}^{-1}$ from the left and taking the second term to the right gives

$$\delta\mathbf{x} = -\mathbf{A}^{-1}\delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x}).$$

Applying (11) with $\mathbf{A}^{-1}$ and vector $\delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x})$ instead of $\mathbf{A}$ and $\mathbf{x}$, we get

$$\|\delta\mathbf{x}\| = \|\mathbf{A}^{-1}\delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x})\| \leqq \|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}(\mathbf{x} + \delta\mathbf{x})\|.$$

Applying (11) on the right, with $\delta\mathbf{A}$ and $\mathbf{x} - \delta\mathbf{x}$ instead of $\mathbf{A}$ and $\mathbf{x}$, we obtain

$$\|\delta\mathbf{x}\| \leqq \|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\|\,\|\mathbf{x} + \delta\mathbf{x}\|.$$

Now $\|\mathbf{A}^{-1}\| = \kappa(\mathbf{A})/\|\mathbf{A}\|$ by the definition of $\kappa(\mathbf{A})$, so that division by $\|\mathbf{x} + \delta\mathbf{x}\|$ shows that the relative inaccuracy of $\mathbf{x}$ is related to that of $\mathbf{A}$ via the condition number by the inequality

$$\textbf{(16)} \qquad \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \approx \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x} + \delta\mathbf{x}\|} \leqq \|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\| = \kappa(\mathbf{A})\,\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}.$$

Conclusion. If the system is well-conditioned, small inaccuracies $\|\delta\mathbf{A}\|/\|\mathbf{A}\|$ can have only a small effect on the solution. However, in the case of ill-conditioning, if $\|\delta\mathbf{A}\|/\|\mathbf{A}\|$ is small, $\|\delta\mathbf{x}\|/\|\mathbf{x}\|$ ***may*** be large.

Inaccurate Right Side. You may show that, similarly, when $\mathbf{A}$ is accurate, an inaccuracy $\delta\mathbf{b}$ of $\mathbf{b}$ causes an inaccuracy $\delta\mathbf{x}$ satisfying

$$\textbf{(17)} \qquad \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leqq \kappa(\mathbf{A})\,\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

Hence $\|\delta\mathbf{x}\|/\|\mathbf{x}\|$ must remain relatively small whenever $\kappa(\mathbf{A})$ is small.

EXAMPLE 7 **Inaccuracies. Bounds (16) and (17)**

If each of the nine entries of $\mathbf{A}$ in Example 5 is measured with an inaccuracy of 0.1, then $\|\delta\mathbf{A}\| = 9 \cdot 0.1$ and (16) gives

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leqq 7.5 \cdot \frac{3 \cdot 0.1}{7} = 0.321 \qquad \text{thus} \qquad \|\delta\mathbf{x}\| \leqq 0.321\,\|\mathbf{x}\| = 0.321 \cdot 16 = 5.14.$$

By experimentation you will find that the actual inaccuracy $\|\delta\mathbf{x}\|$ is only about 30% of the bound 5.14. This is typical.

Similarly, if $\delta\mathbf{b} = [0.1 \quad 0.1 \quad 0.1]^T$, then $\|\delta\mathbf{b}\| = 0.3$ and $\|\mathbf{b}\| = 42$ in Example 5, so that (17) gives

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leqq 7.5 \cdot \frac{0.3}{42} = 0.0536, \qquad \text{hence} \qquad \|\delta\mathbf{x}\| \leqq 0.0536 \cdot 16 = 0.857$$

but this bound is again much greater than the actual inaccuracy, which is about 0.15. ■

Further Comments on Condition Numbers. The following additional explanations may be helpful.

1. There is no sharp dividing line between "well-conditioned" and "ill-conditioned," but generally the situation will get worse as we go from systems with small $\kappa(\mathbf{A})$ to systems with larger $\kappa(\mathbf{A})$. Now always $\kappa(\mathbf{A}) \geqq 1$, so that values of 10 or 20 or so give no reason for concern, whereas $\kappa(\mathbf{A}) = 100$, say, calls for caution, and systems such as those in Examples 1 and 2 are extremely ill-conditioned.

2. If $\kappa(\mathbf{A})$ is large (or small) in one norm, it will be large (or small, respectively) in any other norm. See Example 5.

3. The literature on ill-conditioning is extensive. For an introduction to it, see [E9].

This is the end of our discussion of numerics for solving linear systems. In the next section we consider curve fitting, an important area in which solutions are obtained from linear systems.

PROBLEM SET 20.4

1–6 VECTOR NORMS

Compute the norms (5), (6), (7). Compute a corresponding **unit vector** (vector of norm 1) with respect to the l_∞-norm.

1. $[1 \quad -3 \quad 8 \quad 0 \quad -6 \quad 0]$

2. $[4 \quad -1 \quad 8]$

3. $[0.2 \quad 0.6 \quad -2.1 \quad 3.0]$

4. $[k^2, \quad 4k, \quad k^3], \quad k > 4$

5. $[1 \quad 1 \quad 1 \quad 1 \quad 1]$

6. $[0 \quad 0 \quad 0 \quad 1 \quad 0]$

7. For what $\mathbf{x} = [a \quad b \quad c]$ will $\|\mathbf{x}\|_1 = \|\mathbf{x}\|_2$?

8. Show that $\|\mathbf{x}\|_\infty \leqq \|\mathbf{x}\|_2 \leqq \|\mathbf{x}\|_1$.

9–16 MATRIX NORMS, CONDITION NUMBERS

Compute the matrix norm and the condition number corresponding to the l_1-vector norm.

9. $\begin{bmatrix} 2 & 1 \\ 0 & 4 \end{bmatrix}$

10. $\begin{bmatrix} 2.1 & 4.5 \\ 0.5 & 1.8 \end{bmatrix}$

11. $\begin{bmatrix} \sqrt{5} & 5 \\ 0 & -\sqrt{5} \end{bmatrix}$

12. $\begin{bmatrix} 7 & 6 \\ 6 & 5 \end{bmatrix}$

13. $\begin{bmatrix} -2 & 4 & -1 \\ -2 & 3 & 0 \\ 7 & -12 & 2 \end{bmatrix}$

14. $\begin{bmatrix} 1 & 0.01 & 0 \\ 0.01 & 1 & 0.01 \\ 0 & 0.01 & 1 \end{bmatrix}$

15. $\begin{bmatrix} -20 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 20 \end{bmatrix}$

16. $\begin{bmatrix} 21 & 10.5 & 7 & 5.25 \\ 10.5 & 7 & 5.25 & 4.2 \\ 7 & 5.25 & 4.2 & 3.5 \\ 5.25 & 4.2 & 3.5 & 3 \end{bmatrix}$

17. Verify (11) for $\mathbf{x} = [3 \quad 15 \quad -4]^T$ taken with the l_∞-norm and the matrix in Prob. 13.

18. Verify (12) for the matrices in Probs. 9 and 10.

19–20 ILL-CONDITIONED SYSTEMS

Solve $\mathbf{Ax} = \mathbf{b}_1$, $\mathbf{Ax} = \mathbf{b}_2$. Compare the solutions and comment. Compute the condition number of $\mathbf{A}$.

19. $\mathbf{A} = \begin{bmatrix} 4.50 & 3.55 \\ 3.55 & 2.80 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 5.2 \\ 4.1 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 5.2 \\ 4.0 \end{bmatrix}$

20. $\mathbf{A} = \begin{bmatrix} 3.0 & 1.7 \\ 1.7 & 1.0 \end{bmatrix}, \quad \mathbf{b}_1 = \begin{bmatrix} 4.7 \\ 2.7 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 4.7 \\ 2.71 \end{bmatrix}$

21. Residual. For $\mathbf{Ax} = \mathbf{b}_1$ in Prob. 19 guess what the residual of $\tilde{\mathbf{x}} = [-10.0 \quad 14.1]^T$, very poorly approximating $[-2 \quad 4]^T$, might be. Then calculate and comment.

22. Show that $\kappa(\mathbf{A}) \geqq 1$ for the matrix norms (10), (11), Sec. 20.3, and $\kappa(\mathbf{A}) \geqq \sqrt{n}$ for the Frobenius norm (9), Sec. 20.3.

23. CAS EXPERIMENT. Hilbert Matrices. The 3×3 Hilbert matrix is

$$\mathbf{H}_3 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}.$$

The $n \times n$ Hilbert matrix is $\mathbf{H}_n = [h_{jk}]$, where $h_{jk} = 1/(j + k - 1)$. (Similar matrices occur in curve fitting by least squares.) Compute the condition number $\kappa(\mathbf{H}_n)$ for the matrix norm corresponding to the l_∞- (or l_1-) vector norm, for $n = 2, 3, \cdots, 6$ (or further if you wish). Try to find a formula that gives reasonable approximate values of these rapidly growing numbers.

Solve a few linear systems of your choice, involving an $\mathbf{H}_n$.

24. TEAM PROJECT. Norms. (a) Vector norms in our text are **equivalent**, that is, they are related by double inequalities; for instance,

$$(18) \qquad \begin{aligned} &\text{(a)} \quad \|\mathbf{x}\|_\infty \leqq \|\mathbf{x}\|_1 \leqq n\|\mathbf{x}\|_\infty \\ &\text{(b)} \quad \frac{1}{n}\|\mathbf{x}\|_1 \leqq \|\mathbf{x}\|_\infty \leqq \|\mathbf{x}\|_1. \end{aligned}$$

Hence if for some $\mathbf{x}$, one norm is large (or small), the other norm must also be large (or small). Thus in many investigations the particular choice of a norm is not essential. Prove (18).

(b) The Cauchy–Schwarz inequality is

$$|\mathbf{x}^T\mathbf{y}| \leqq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2.$$

It is very important. (Proof in Ref. [GenRef7] listed in App. 1.) Use it to prove

(19a) $$\|\mathbf{x}\|_2 \leqq \|\mathbf{x}\|_1 \leqq \sqrt{n}\,\|\mathbf{x}\|_2$$

(19b) $$\frac{1}{\sqrt{n}}\|\mathbf{x}\|_1 \leqq \|\mathbf{x}\|_2 \leqq \|\mathbf{x}\|_1.$$

(c) Formula (10) is often more practical than (9). Derive (10) from (9).

(d) Matrix norms. Illustrate (11) with examples. Give examples of (12) with equality as well as with strict inequality. Prove that the matrix norms (10), (11) in Sec. 20.3 satisfy the *axioms of a norm*

$$\|\mathbf{A}\| \geqq \mathbf{0}.$$
$$\|\mathbf{A}\| = \mathbf{0} \text{ if and only if } \mathbf{A} = \mathbf{0},$$
$$\|k\mathbf{A}\| = |k|\,\|\mathbf{A}\|,$$
$$\|\mathbf{A} + \mathbf{B}\| \leqq \|\mathbf{A}\| + \|\mathbf{B}\|.$$

25. WRITING PROJECT. Norms and Their Use in This Section. Make a list of the most important of the many ideas covered in this section and write a two-page report on them.

20.5 Least Squares Method

Having discussed numerics for linear systems, we now turn to an important application, curve fitting, in which the solutions are obtained from linear systems.

In **curve fitting** we are given n points (pairs of numbers) $(x_1, y_1), \cdots, (x_n, y_n)$ and we want to determine a function $f(x)$ such that

$$f(x_1) \approx y_1, \cdots, f(x_n) \approx y_n,$$

approximately. The type of function (for example, polynomials, exponential functions, sine and cosine functions) may be suggested by the nature of the problem (the underlying physical law, for instance), and in many cases a polynomial of a certain degree will be appropriate.

Let us begin with a motivation.

If we require strict equality $f(x_1) = y_1, \cdots, f(x_n) = y_n$ and use polynomials of sufficiently high degree, we may apply one of the methods discussed in Sec. 19.3 in connection with interpolation. However, in certain situations this would not be the appropriate solution of the actual problem. For instance, to the four points

(1) $$(-1.3, 0.103), \quad (-0.1, 1.099), \quad (0.2, 0.808), \quad (1.3, 1.897)$$

there corresponds the interpolation polynomial $f(x) = x^3 - x + 1$ (Fig. 446), but if we graph the points, we see that they lie nearly on a straight line. Hence if these values are obtained in an experiment and thus involve an experimental error, and if the nature of the experiment suggests a linear relation, we better fit a straight line through the points (Fig. 446). Such a line may be useful for predicting values to be expected for other values of x. A widely used principle for fitting straight lines is the **method**

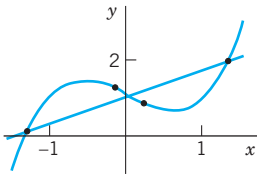

Fig. 446. Approximate fitting of a straight line

of least squares by Gauss and Legendre. In the present situation it may be formulated as follows.

Method of Least Squares. *The straight line*

$$y = a + bx \tag{2}$$

should be fitted through the given points $(x_1, y_1), \cdots, (x_n, y_n)$ *so that the sum of the squares of the distances of those points from the straight line is minimum, where the distance is measured in the vertical direction* (*the y-direction*).

The point on the line with abscissa x_j has the ordinate $a + bx_j$. Hence its distance from (x_j, y_j) is $|y_j - a - bx_j|$ (Fig. 447) and that sum of squares is

$$q = \sum_{j=1}^{n} (y_j - a - bx_j)^2.$$

q depends on a and b. A necessary condition for q to be minimum is

$$\frac{\partial q}{\partial a} = -2\sum (y_j - a - bx_j) = 0$$
$$\frac{\partial q}{\partial b} = -2\sum x_j (y_j - a - bx_j) = 0 \tag{3}$$

(where we sum over j from 1 to n). Dividing by 2, writing each sum as three sums, and taking one of them to the right, we obtain the result

$$\begin{aligned} an \quad\quad + b\sum x_j &= \sum y_j \\ a\sum x_j + b\sum x_j^2 &= \sum x_j y_j. \end{aligned} \tag{4}$$

These equations are called the **normal equations** of our problem.

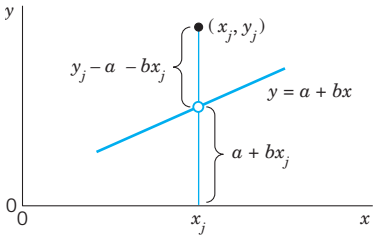

Fig. 447. Vetrical distance of a point (x_j, y_j) from a straight line $y = a + bx$

EXAMPLE 1 **Straight Line**

Using the method of least squares, fit a straight line to the four points given in formula (1).

Solution. We obtain

$$n = 4, \quad \sum x_j = 0.1, \quad \sum x_j^2 = 3.43, \quad \sum y_j = 3.907, \quad \sum x_j y_j = 2.3839.$$

Hence the normal equations are

$$4a + 0.10b = 3.9070$$

$$0.1a + 3.43b = 2.3839.$$

The solution (rounded to 4D) is $a = 0.9601$, $b = 0.6670$, and we obtain the straight line (Fig. 446)

$$y = 0.9601 + 0.6670x.$$

Curve Fitting by Polynomials of Degree *m*

Our method of curve fitting can be generalized from a polynomial $y = a + bx$ to a polynomial of degree m

$$p(x) = b_0 + b_1x + \cdots + b_mx^m \tag{5}$$

where $m \leqq n - 1$. Then q takes the form

$$q = \sum_{j=1}^{n} (y_j - p(x_j))^2$$

and depends on $m + 1$ parameters $b_0, \cdots, b_m$. Instead of (3) we then have $m + 1$ conditions

$$\frac{\partial q}{\partial b_0} = 0, \qquad \cdots, \qquad \frac{\partial q}{\partial b_m} = 0 \tag{6}$$

which give a system of $m + 1$ normal equations.

In the case of a quadratic polynomial

$$p(x) = b_0 + b_1x + b_2x^2 \tag{7}$$

the normal equations are (summation from 1 to n)

$$\begin{aligned} b_0 n \quad &+ b_1 \sum x_j + b_2 \sum x_j^2 = \sum y_j \\ b_0 \sum x_j \; &+ b_1 \sum x_j^2 + b_2 \sum x_j^3 = \sum x_j y_j \\ b_0 \sum x_j^2 \; &+ b_1 \sum x_j^3 + b_2 \sum x_j^4 = \sum x_j^2 y_j. \end{aligned} \tag{8}$$

The derivation of (8) is left to the reader.

EXAMPLE 2 Quadratic Parabola by Least Squares

Fit a parabola through the data (0, 5), (2, 4), (4, 1), (6, 6), (8, 7).

Solution. For the normal equations we need $n = 5$, $\Sigma x_j = 20$, $\Sigma x_j^2 = 120$, $\Sigma x_j^3 = 800$, $\Sigma x_j^4 = 5664$, $\Sigma y_j = 23$, $\Sigma x_j y_j = 104$, $\Sigma x_j^2 y_j = 696$. Hence these equations are

$$5b_0 + 20b_1 + 120b_2 = 23$$

$$20b_0 + 120b_1 + 800b_2 = 104$$

$$120b_0 + 800b_1 + 5664b_2 = 696.$$

Solving them we obtain the quadratic least squares parabola (Fig. 448)

$$y = 5.11429 - 1.41429x + 0.21429x^2.$$

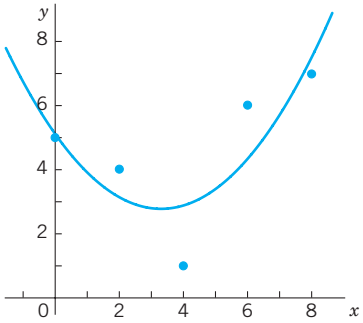

Fig. 448. Least squares parabola in Example 2

For a general polynomial (5) the normal equations form a linear system of equations in the unknowns $b_0, \cdots, b_m$. When its matrix $\mathbf{M}$ is nonsingular, we can solve the system by Cholesky's method (Sec. 20.2) because then $\mathbf{M}$ is positive definite (and symmetric). When the equations are nearly linearly dependent, the normal equations may become ill-conditioned and should be replaced by other methods; see [E5], Sec. 5.7, listed in App. 1.

The least squares method also plays a role in statistics (see Sec. 25.9).

PROBLEM SET 20.5

1–6 FITTING A STRAIGHT LINE

Fit a straight line to the given points (x, y) by least squares. Show the details. Check your result by sketching the points and the line. Judge the goodness of fit.

1. (0, 2), (2, 0), (3, −2), (5, −3)

2. How does the line in Prob. 1 change if you add a point far above it, say, (1, 3)? Guess first.

3. (0, 1.8), (1, 1.6), (2, 1.1), (3, 1.5), (4, 2.3)

4. Hooke's law $F = ks$. Estimate the spring modulus k from the force F [lb] and the elongation s [cm], where $(F, s) = (1, 0.3)$, $(2, 0.7)$, $(4, 1.3)$, $(6, 1.9)$, $(10, 3.2)$, $(20, 6.3)$.

5. Average speed. Estimate the average speed v_{av} of a car traveling according to $s = v \cdot t$ [km] (s = distance traveled, t [hr] = time) from $(t, s) = (9, 140)$, $(10, 220)$, $(11, 310)$, $(12, 410)$.

6. Ohm's law $U = Ri$. Estimate R from $(i, U) = (2, 104)$, $(4, 206)$, $(6, 314)$, $(10, 530)$.

7. Derive the normal equations (8).

8–11 FITTING A QUADRATIC PARABOLA

Fit a parabola (7) to the points (x, y). Check by sketching.

8. (−1, 5), (1, 3), (2, 4), (3, 8)

9. (2, −3), (3, 0), (5, 1), (6, 0) (7, −2)

10. t [hr] = Worker's time on duty, y [sec] = His/her reaction time, $(t, y) = (1, 2.0)$, $(2, 1.78)$, $(3, 1.90)$, $(4, 2.35)$, $(5, 2.70)$

11. The data in Prob. 3. Plot the points, the line, and the parabola jointly. Compare and comment.

12. Cubic parabola. Derive the formula for the normal equations of a cubic least squares parabola.

13. Fit curves (2) and (7) and a cubic parabola by least squares to $(x, y) = (-2, -30)$, $(-1, -4)$, $(0, 4)$, $(1, 4)$, $(2, 22)$, $(3, 68)$. Graph these curves and the points on common axes. Comment on the goodness of fit.

14. TEAM PROJECT. The least squares approximation of a function $f(x)$ on an interval $a \leqq x \leqq b$ by a function

$$F_m(x) = a_0y_0(x) + a_1y_1(x) + \cdots + a_my_m(x)$$

where $y_0(x), \cdots, y_m(x)$ are given functions, requires the determination of the coefficients $a_0, \cdots, a_m$ such that

$$(9) \qquad \int_a^b [f(x) - F_m(x)]^2 \, dx$$

becomes minimum. This integral is denoted by $\|f - F_m\|^2$, and $\|f - F_m\|$ is called the $\boldsymbol{L_2}$**-norm** of $f - F_m$ (L suggesting Lebesgue[5]). A necessary condition for that minimum is given by $\partial \|f - F_m\|^2 / \partial a_j = 0$, $j = 0, \cdots, m$ [the analog of (6)]. **(a)** Show that this leads to $m + 1$ normal equations ($j = 0, \cdots, m$)

$$\sum_{k=0}^{m} h_{jk} a_k = b_j \qquad \text{where}$$

$$(10) \qquad h_{jk} = \int_a^b y_j(x) y_k(x) \, dx,$$

$$b_j = \int_a^b f(x) y_j(x) \, dx.$$

(b) Polynomial. What form does (10) take if $F_m(x) = a_0 + a_1 x + \cdots + a_m x^m$? What is the coefficient matrix of (10) in this case when the interval is $0 \leqq x \leqq 1$?

(c) Orthogonal functions. What are the solutions of (10) if $y_0(x), \cdots, y_m(x)$ are orthogonal on the interval $a \leqq x \leqq b$? (For the definition, see Sec. 11.5. See also Sec. 11.6.)

15. CAS EXPERIMENT. Least Squares versus Interpolation. For the given data and for data of your choice find the interpolation polynomial and the least squares approximations (linear, quadratic, etc.). Compare and comment.

(a) $(-2, 0)$, $(-1, 0)$, $(0, 1)$, $(1, 0)$, $(2, 0)$

(b) $(-4, 0)$, $(-3, 0)$, $(-2, 0)$, $(-1, 0)$, $(0, 1)$, $(1, 0)$, $(2, 0)$, $(3, 0)$, $(4, 0)$

(c) Choose five points on a straight line, e.g., $(0, 0)$, $(1, 1), \cdots, (4, 4)$. Move one point 1 unit upward and find the quadratic least squares polynomial. Do this for each point. Graph the five polynomials on common axes. Which of the five motions has the greatest effect?

20.6 Matrix Eigenvalue Problems: Introduction

We now come to the second part of our chapter on numeric linear algebra. In the *first part of this chapter* we discussed methods of solving systems of linear equations, which included Gauss elimination with backward substitution. This method is known as a direct method since it gives solutions after a prescribed amount of computation. The Gauss method was modified by Doolittle's method, Crout's method, and Cholesky's method, each requiring fewer arithmetic operations than Gauss. Finally we presented indirect methods of solving systems of linear equations, that is, the Gauss–Seidel method and the Jacobi iteration. The indirect methods require an undetermined number of iterations. That number depends on how far we start from the true solution and what degree of accuracy we require. Moreover, depending on the problem, convergence may be fast or slow or our computation cycle might not even converge. This led to the concepts of ill-conditioned problems and condition numbers that help us gain some control over difficulties inherent in numerics.

The second part of this chapter deals with some of the most important ideas and numeric methods for matrix eigenvalue problems. This very extensive part of numeric linear algebra is of great practical importance, with much research going on, and hundreds, if not thousands, of papers published in various mathematical journals (see the references in [E8], [E9], [E11], [E29]). We begin with the concepts and general results we shall need in explaining and applying numeric methods for eigenvalue problems. (For typical models of eigenvalue problems see Chap. 8.)

[5]HENRI LEBESGUE (1875–1941), great French mathematician, creator of a modern theory of measure and integration in his famous doctoral thesis of 1902.

An **eigenvalue** or **characteristic value** (or *latent root*) of a given $n \times n$ matrix $\mathbf{A} = [a_{jk}]$ is a real or complex number λ such that the vector equation

$$\mathbf{Ax} = \lambda\mathbf{x} \tag{1}$$

has a nontrivial solution, that is, a solution $\mathbf{x} \neq \mathbf{0}$, which is then called an **eigenvector** or **characteristic vector** of $\mathbf{A}$ corresponding to that eigenvalue λ. The set of all eigenvalues of $\mathbf{A}$ is called the **spectrum** of $\mathbf{A}$. Equation (1) can be written

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{2}$$

where $\mathbf{I}$ is the $n \times n$ unit matrix. This homogeneous system has a nontrivial solution if and only if the **characteristic determinant** $\det(\mathbf{A} - \lambda\mathbf{I})$ is 0 (see Theorem 2 in Sec. 7.5). This gives (see Sec. 8.1)

THEOREM 1

Eigenvalues

The eigenvalues of $\mathbf{A}$ *are the solutions* λ *of the* **characteristic equation**

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{vmatrix} = 0. \tag{3}$$

Developing the characteristic determinant, we obtain the **characteristic polynomial** of $\mathbf{A}$, which is of degree n in λ. Hence $\mathbf{A}$ has at least one and at most n numerically different eigenvalues. If $\mathbf{A}$ is real, so are the coefficients of the characteristic polynomial. By familiar algebra it follows that then the roots (the eigenvalues of $\mathbf{A}$) are ***real or complex conjugates*** in pairs.

To give you some orientation of the underlying approaches of numerics for eigenvalue problems, note the following. For large or very large matrices it may be very difficult to determine the eigenvalues, since, in general, it is difficult to find the roots of characteristic polynomials of higher degrees. We will discuss different numeric methods for finding eigenvalues that achieve different results. Some methods, such as in Sec. 20.7, will give us only regions in which complex eigenvalues lie (Geschgorin's method) or the intervals in which the largest and smallest real eigenvalue lie (Collatz method). Other methods compute all eigenvalues, such as the Householder tridiagonalization method and the QR-method in Sec. 20.9.

To continue our discussion, we shall usually denote the eigenvalues of $\mathbf{A}$ by

$$\lambda_1, \lambda_2, \cdots, \lambda_n$$

with the understanding that some (or all) of them may be equal.

The sum of these n eigenvalues equals the sum of the entries on the main diagonal of $\mathbf{A}$, called the trace of $\mathbf{A}$; thus

$$\text{trace } \mathbf{A} = \sum_{j=1}^{n} a_{jj} = \sum_{k=1}^{n} \lambda_k. \tag{4}$$

Also, the product of the eigenvalues equals the determinant of $\mathbf{A}$,

$$\det \mathbf{A} = \lambda_1 \lambda_2 \cdots \lambda_n. \tag{5}$$

Both formulas follow from the product representation of the characteristic polynomial, which we denote by $f(\lambda)$,

$$f(\lambda) = (-1)^n(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n).$$

If we take equal factors together and denote the *numerically distinct* eigenvalues of $\mathbf{A}$ by $\lambda_1, \cdots, \lambda_r$ $(r \leqq n)$, then the product becomes

$$f(\lambda) = (-1)^n(\lambda - \lambda_1)^{m_1}(\lambda - \lambda_2)^{m_2}\cdots(\lambda - \lambda_r)^{m_r}. \tag{6}$$

The exponent m_j is called the **algebraic multiplicity** of λ_j. The maximum number of linearly independent eigenvectors corresponding to λ_j is called the **geometric multiplicity** of λ_j. It is equal to or smaller than m_j.

A subspace S of R^n or C^n (if $\mathbf{A}$ is complex) is called an **invariant subspace** of $\mathbf{A}$ if for every $\mathbf{v}$ in S the vector $\mathbf{Av}$ is also in S. **Eigenspaces** of $\mathbf{A}$ (spaces of eigenvectors; Sec. 8.1) are important invariant subspaces of $\mathbf{A}$.

An $n \times n$ matrix $\mathbf{B}$ is called **similar** to $\mathbf{A}$ if there is a nonsingular $n \times n$ matrix $\mathbf{T}$ such that

$$\mathbf{B} = \mathbf{T}^{-1}\mathbf{AT}. \tag{7}$$

Similarity is important for the following reason.

THEOREM 2

Similar Matrices

Similar matrices have the same eigenvalues. If $\mathbf{x}$ *is an eigenvector of* $\mathbf{A}$*, then* $\mathbf{y} = \mathbf{T}^{-1}\mathbf{x}$ *is an eigenvector of* $\mathbf{B}$ *in* (7) *corresponding to the same eigenvalue.* (Proof in Sec. 8.4.)

Another theorem that has various applications in numerics is as follows.

THEOREM 3

Spectral Shift

If $\mathbf{A}$ *has the eigenvalues* $\lambda_1, \cdots, \lambda_n$*, then* $\mathbf{A} - k\mathbf{I}$ *with arbitrary* k *has the eigenvalues* $\lambda_1 - k, \cdots, \lambda_n - k$.

This theorem is a special case of the following **spectral mapping theorem**.

THEOREM 4

Polynomial Matrices

If λ *is an eigenvalue of* $\mathbf{A}$*, then*

$$q(\lambda) = \alpha_s\lambda^s + \alpha_{s-1}\lambda^{s-1} + \cdots + \alpha_1\lambda + \alpha_0$$

is an eigenvalue of the **polynomial matrix**

$$q(\mathbf{A}) = \alpha_s\mathbf{A}^s + \alpha_{s-1}\mathbf{A}^{s-1} + \cdots + \alpha_1\mathbf{A} + a_0\mathbf{I}.$$

PROOF $\mathbf{Ax} = \lambda\mathbf{x}$ implies $\mathbf{A}^2\mathbf{x} = \mathbf{A}\lambda\mathbf{x} = \lambda\mathbf{Ax} = \lambda^2\mathbf{x}$, $\mathbf{A}^3\mathbf{x} = \lambda^3\mathbf{x}$, etc. Thus

$$\begin{aligned} q(\mathbf{A})\mathbf{x} &= (\alpha_s\mathbf{A}^s + \alpha_{s-1}\mathbf{A}^{s-1} + \cdots)\,\mathbf{x} \\ &= \alpha_s\mathbf{A}^s\mathbf{x} + \alpha_{s-1}A^{s-1}\mathbf{x} + \cdots \\ &= \alpha_s\lambda^s\mathbf{x} + \alpha_{s-1}\lambda^{s-1}\mathbf{x} + \cdots = q(\lambda)\,\mathbf{x}. \end{aligned}$$

■

The eigenvalues of important special matrices can be characterized as follows.

THEOREM 5

Special Matrices

The eigenvalues of Hermitian matrices (i.e., $\overline{\mathbf{A}}^{\mathsf{T}} = \mathbf{A}$), *hence of real symmetric matrices* (i.e., $\mathbf{A}^{\mathsf{T}} = \mathbf{A}$), *are real. The eigenvalues of skew-Hermitian matrices* (i.e., $\overline{\mathbf{A}}^{\mathsf{T}} = -\mathbf{A}$), *hence of real skew-symmetric matrices* (i.e., $\mathbf{A}^{\mathsf{T}} = -\mathbf{A}$), *are pure imaginary or* 0. *The eigenvalues of unitary matrices* (i.e., $\overline{\mathbf{A}}^{\mathsf{T}} = \mathbf{A}^{-1}$), *hence of orthogonal matrices* (i.e., $\mathbf{A}^{\mathsf{T}} = \mathbf{A}^{-1}$), *have absolute value* 1. (Proofs in Secs. 8.3 and 8.5.)

The **choice of a numeric method** for matrix eigenvalue problems depends essentially on two circumstances, on the kind of matrix (real symmetric, real general, complex, sparse, or full) and on the kind of information to be obtained, that is, whether one wants to know all eigenvalues or merely specific ones, for instance, the largest eigenvalue, whether eigenvalues *and* eigenvectors are wanted, and so on. It is clear that we cannot enter into a systematic discussion of all these and further possibilities that arise in practice, but we shall concentrate on some basic aspects and methods that will give us a general understanding of this fascinating field.

20.7 Inclusion of Matrix Eigenvalues

The whole of numerics for matrix eigenvalues is motivated by the fact that, except for a few trivial cases, we cannot determine eigenvalues *exactly* by a finite process because these values are the roots of a polynomial of nth degree. Hence we must mainly use iteration.

In this section we state a few general theorems that give approximations and error bounds for eigenvalues. Our matrices will continue to be real (except in formula (5) below), but since (nonsymmetric) matrices may have complex eigenvalues, complex numbers will play a (very modest) role in this section.

The important theorem by Gerschgorin gives a region consisting of closed circular disks in the complex plane and including all the eigenvalues of a given matrix. Indeed, for each $j = 1, \cdots, n$ the inequality (1) in the theorem determines a closed circular disk in the complex λ-plane with center a_{jj} and radius given by the right side of (1); and Theorem 1 states that each of the eigenvalues of $\mathbf{A}$ lies in one of these n disks.

THEOREM 1

Gerschgorin's Theorem[6]

Let λ be an eigenvalue of an arbitrary $n \times n$ matrix $\mathbf{A} = [a_{jk}]$. Then for some integer j $(1 \leqq j \leqq n)$ we have

$$(1) \qquad |a_{jj} - \lambda| \leqq |a_{j1}| + |a_{j2}| + \cdots + |a_{j,\,j-1}| + |a_{j,\,j+1}| + \cdots + |a_{jn}|.$$

[6]SEMYON ARANOVICH GERSCHGORIN (1901–1933), Russian mathematician.

PROOF Let $\mathbf{x}$ be an eigenvector corresponding to an eigenvalue λ of $\mathbf{A}$. Then

$$\text{(2)} \qquad \mathbf{Ax} = \lambda\mathbf{x} \qquad \text{or} \qquad (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}.$$

Let x_j be a component of $\mathbf{x}$ that is largest in absolute value. Then we have $|x_m/x_j| \leqq 1$ for $m = 1, \cdots, n$. The vector equation (2) is equivalent to a system of n equations for the n components of the vectors on both sides. The jth of these n equations with j as just indicated is

$$a_{j1}x_1 + \cdots + a_{j,j-1}x_{j-1} + (a_{jj} - \lambda)x_j + a_{j,j+1}x_{j+1} + \cdots + a_{jn}x_n = 0.$$

Division by x_j (which cannot be zero; why?) and reshuffling terms gives

$$a_{jj} - \lambda = -a_{j1}\frac{x_1}{x_j} - \cdots - a_{j,j-1}\frac{x_{j-1}}{x_j} - a_{j,j+1}\frac{x_{j+1}}{x_j} - \cdots - a_{jn}\frac{x_n}{x_j}.$$

By taking absolute values on both sides of this equation, applying the triangle inequality $|a + b| \leqq |a| + |b|$ (where a and b are any complex numbers), and observing that because of the choice of j (which is crucial!), $|x_1/x_j| \leqq 1, \cdots, |x_n/x_j| \leqq 1$, we obtain (1), and the theorem is proved. ■

EXAMPLE 1 Gerschgorin's Theorem

For the eigenvalues of the matrix

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 5 & 1 \\ \frac{1}{2} & 1 & 1 \end{bmatrix}$$

we get the Gerschgorin disks (Fig. 449)

$$D_1\text{: Center 0, radius 1,} \qquad D_2\text{: Center 5, radius 1.5,} \qquad D_3\text{: Center 1, radius 1.5.}$$

The centers are the main diagonal entries of $\mathbf{A}$. These would be the eigenvalues of $\mathbf{A}$ if $\mathbf{A}$ were diagonal. We can take these values as crude approximations of the unknown eigenvalues (3D-values) $\lambda_1 = -0.209$, $\lambda_2 = 5.305$, $\lambda_3 = 0.904$ (verify this); then the radii of the disks are corresponding error bounds.

Since $\mathbf{A}$ is symmetric, it follows from Theorem 5, Sec. 20.6, that the spectrum of $\mathbf{A}$ must actually lie in the intervals $[-1, 2.5]$ and $[3.5, 6.5]$.

It is interesting that here the Gerschgorin disks form two disjoint sets, namely, $D_1 \cup D_3$, which contains two eigenvalues, and D_2, which contains one eigenvalue. This is typical, as the following theorem shows. ■

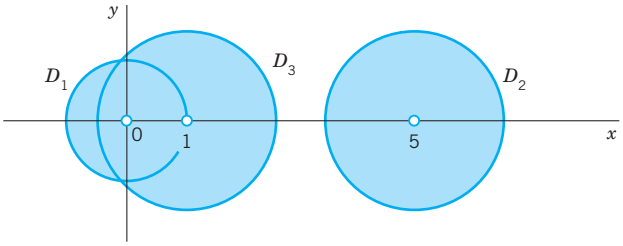

Fig. 449. Gerschgorin disks in Example 1

THEOREM 2

Extension of Gerschgorin's Theorem

If p Gerschgorin disks form a set S that is disjoint from the n − p other disks of a given matrix **A**, *then S contains precisely p eigenvalues of* **A** (*each counted with its algebraic multiplicity, as defined in Sec.* 20.6).

Idea of Proof. Set $\mathbf{A} = \mathbf{B} + \mathbf{C}$, where $\mathbf{B}$ is the diagonal matrix with entries a_{jj}, and apply Theorem 1 to $\mathbf{A}_t = \mathbf{B} + t\mathbf{C}$ with real t growing from 0 to 1. ■

EXAMPLE 2 **Another Application of Gerschgorin's Theorem. Similarity**

Suppose that we have diagonalized a matrix by some numeric method that left us with some off-diagonal entries of size 10^{-5}, say,

$$\mathbf{A} = \begin{bmatrix} 2 & 10^{-5} & 10^{-5} \\ 10^{-5} & 2 & 10^{-5} \\ 10^{-5} & 10^{-5} & 4 \end{bmatrix}.$$

What can we conclude about deviations of the eigenvalues from the main diagonal entries?

Solution. By Theorem 2, one eigenvalue must lie in the disk of radius $2 \cdot 10^{-5}$ centered at 4 and two eigenvalues (or an eigenvalue of algebraic multiplicity 2) in the disk of radius $2 \cdot 10^{-5}$ centered at 2. Actually, since the matrix is symmetric, these eigenvalues must lie in the intersections of these disks and the real axis, by Theorem 5 in Sec. 20.6.

We show how an isolated disk can always be reduced in size by a similarity transformation. The matrix

$$\mathbf{B} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^{-5} \end{bmatrix} \begin{bmatrix} 2 & 10^{-5} & 10^{-5} \\ 10^{-5} & 2 & 10^{-5} \\ 10^{-5} & 10^{-5} & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10^{5} \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 10^{-5} & 1 \\ 10^{-5} & 2 & 1 \\ 10^{-10} & 10^{-10} & 4 \end{bmatrix}$$

is similar to **A**. Hence by Theorem 2, Sec. 20.6, it has the same eigenvalues as **A**. From Row 3 we get the smaller disk of radius $2 \cdot 10^{-10}$. Note that the other disks got bigger, approximately by a factor of 10^5. And in choosing **T** we have to watch that the new disks do not overlap with the disk whose size we want to decrease.

For further interesting facts, see the book [E28]. ■

By definition, a **diagonally dominant** matrix $\mathbf{A} = [a_{jk}]$ is an $n \times n$ matrix such that

$$|a_{jj}| \geqq \sum_{k \neq j} |a_{jk}| \qquad j = 1, \cdots, n \tag{3}$$

where we sum over all off-diagonal entries in Row j. The matrix is said to be **strictly diagonally dominant** if $>$ in (3) for all j. Use Theorem 1 to prove the following basic property.

THEOREM 3

Strict Diagonal Dominance

Strictly diagonally dominant matrices are nonsingular.

Further Inclusion Theorems

An **inclusion theorem** is a theorem that specifies a set which contains at least one eigenvalue of a given matrix. Thus, Theorems 1 and 2 are inclusion theorems; they even include the whole spectrum. We now discuss some famous theorems that yield further inclusions of eigenvalues. We state the first two of them without proofs (which would exceed the level of this book).

THEOREM 4

Schur's Theorem[7]

Let $\mathbf{A} = [a_{jk}]$ *be a* $n \times n$ *matrix. Then for each of its eigenvalues* $\lambda_1, \cdots, \lambda_n$,

$$|\lambda_m|^2 \leqq \sum_{i=1}^{n} |\lambda_i|^2 \leqq \sum_{j=1}^{n} \sum_{k=1}^{n} |a_{jk}|^2 \quad \textbf{(Schur's inequality)}. \tag{4}$$

In (4) *the second equality sign holds if and only if* $\mathbf{A}$ *is such that*

$$\bar{\mathbf{A}}^{\mathsf{T}}\mathbf{A} = \mathbf{A}\bar{\mathbf{A}}^{\mathsf{T}}. \tag{5}$$

Matrices that satisfy (5) are called **normal matrices**. It is not difficult to see that Hermitian, skew-Hermitian, and unitary matrices are normal, and so are real symmetric, skew-symmetric, and orthogonal matrices.

EXAMPLE 3 **Bounds for Eigenvalues Obtained from Schur's Inequality**

For the matrix

$$\mathbf{A} = \begin{bmatrix} 26 & -2 & 2 \\ 2 & 21 & 4 \\ 4 & 2 & 28 \end{bmatrix}$$

we obtain from Schur's inequality $|\lambda| \leqq \sqrt{1949} = 44.1475$. You may verify that the eigenvalues are 30, 25, and 20. Thus $30^2 + 25^2 + 20^2 = 1925 < 1949$; in fact, $\mathbf{A}$ is not normal. ■

The preceding theorems are valid for every real or complex square matrix. Other theorems hold for special classes of matrices only. Famous is the following one, which has various applications, for instance, in economics.

THEOREM 5

Perron's Theorem[8]

Let $\mathbf{A}$ *be a real* $n \times n$ *matrix whose entries are all positive. Then* $\mathbf{A}$ *has a positive real eigenvalue* $\lambda = \rho$ *of multiplicity* 1. *The corresponding eigenvector can be chosen with all components positive. (The other eigenvalues are less than* ρ *in absolute value.)*

---

[7]ISSAI SCHUR (1875–1941), German mathematician, also known by his important work in group theory.

[8]OSKAR PERRON (1880–1975) and GEORG FROBENIUS (1849–1917), German mathematicians, known for their work in potential theory, ODEs (Sec. 5.4), and group theory.

For a proof see Ref. [B3], vol. II, pp. 53–62. The theorem also holds for matrices with *nonnegative* real entries ("**Perron–Frobenius Theorem**"[8]) provided **A** is **irreducible**, that is, it cannot be brought to the following form by interchanging rows and columns; here **B** and **F** are square and **0** is a zero matrix.

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{F} \end{bmatrix}$$

Perron's theorem has various applications, for instance, in economics. It is interesting that one can obtain from it a theorem that gives a numeric algorithm:

THEOREM 6

Collatz Inclusion Theorem[9]

Let $\mathbf{A} = [a_{jk}]$ *be a real* $n \times n$ *matrix whose elements are all positive. Let* $\mathbf{x}$ *be any real vector whose components* $x_1, \cdots, x_n$ *are positive, and let* $y_1, \cdots, y_n$ *be the components of the vector* $\mathbf{y} = \mathbf{Ax}$. *Then the closed interval on the real axis bounded by the smallest and the largest of the* n *quotients* $q_j = y_j/x_j$ *contains at least one eigenvalue of* $\mathbf{A}$.

PROOF We have $\mathbf{Ax} = \mathbf{y}$ or

$$\mathbf{y} - \mathbf{Ax} = \mathbf{0}. \tag{6}$$

The transpose $\mathbf{A}^T$ satisfies the conditions of Theorem 5. Hence $\mathbf{A}^T$ has a positive eigenvalue λ and, corresponding to this eigenvalue, an eigenvector $\mathbf{u}$ whose components u_j are all positive. Thus $\mathbf{A}^T\mathbf{u} = \lambda\mathbf{u}$ and by taking the transpose we obtain $\mathbf{u}^T\mathbf{A} = \lambda\mathbf{u}^T$. From this and (6) we have

$$\mathbf{u}^T(\mathbf{y} - \mathbf{Ax}) = \mathbf{u}^T\mathbf{y} - \mathbf{u}^T\mathbf{Ax} = \mathbf{u}^T\mathbf{y} - \lambda\mathbf{u}^T\mathbf{x} = \mathbf{u}^T(\mathbf{y} - \lambda\mathbf{x}) = 0$$

or written out

$$\sum_{j=1}^{n} u_j(y_j - \lambda x_j) = 0.$$

Since all the components u_j are positive, it follows that

$$\text{(7)} \qquad \begin{aligned} y_j - \lambda x_j \geqq 0, \quad &\text{that is,} \quad q_j \geqq \lambda \quad \text{for at least one } j, \\ y_j - \lambda x_j \leqq 0, \quad &\text{that is,} \quad q_j \leqq \lambda \quad \text{for at least one } j. \end{aligned} \qquad \text{and}$$

Since $\mathbf{A}$ and $\mathbf{A}^T$ have the same eigenvalues, λ is an eigenvalue of $\mathbf{A}$, and from (7) the statement of the theorem follows. ■

[9]LOTHAR COLLATZ (1910–1990), German mathematician known for his work in numerics.

EXAMPLE 4 Bounds for Eigenvalues from Collatz's Theorem. Iteration

For a given matrix $\mathbf{A}$ with positive entries we choose an $\mathbf{x} = \mathbf{x}_0$ and **iterate**, that is, we compute $\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0$, $\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1, \cdots, \mathbf{x}_{20} = \mathbf{A}\mathbf{x}_{19}$. In each step, taking $\mathbf{x} = \mathbf{x}_j$ and $\mathbf{y} = \mathbf{A}\mathbf{x}_j = \mathbf{x}_{j+1}$ we compute an inclusion interval by Collatz's theorem. This gives (6S)

$$\mathbf{A} = \begin{bmatrix} 0.49 & 0.02 & 0.22 \\ 0.02 & 0.28 & 0.20 \\ 0.22 & 0.20 & 0.40 \end{bmatrix}, \mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{x}_1 = \begin{bmatrix} 0.73 \\ 0.50 \\ 0.82 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 0.5481 \\ 0.3186 \\ 0.5886 \end{bmatrix},$$

$$\cdots, \mathbf{x}_{19} = \begin{bmatrix} 0.00216309 \\ 0.00108155 \\ 0.00216309 \end{bmatrix}, \mathbf{x}_{20} = \begin{bmatrix} 0.00155743 \\ 0.000778713 \\ 0.00155743 \end{bmatrix}$$

and the intervals $0.5 \leqq \lambda \leqq 0.82$, $0.3186/0.50 = 0.6372 \leqq \lambda \leqq 0.5481/0.73 = 0.750822$, etc. These intervals have length

j	1	2	3	10	15	20
Length	0.32	0.113622	0.0539835	0.0004217	0.0000132	0.0000004

Using the characteristic polynomial, you may verify that the eigenvalues of $\mathbf{A}$ are 0.72, 0.36, 0.09, so that those intervals include the largest eigenvalue, 0.72. Their lengths decreased with j, so that the iteration was worthwhile. The reason will appear in the next section, where we discuss an iteration method for eigenvalues. ■

PROBLEM SET 20.7

1–6 GERSCHGORIN DISKS

Find and sketch disks or intervals that contain the eigenvalues. If you have a CAS, find the spectrum and compare.

1. $\begin{bmatrix} 5 & 2 & 4 \\ -2 & 0 & 2 \\ 2 & 4 & 7 \end{bmatrix}$ **2.** $\begin{bmatrix} 5 & 10^{-2} & 10^{-2} \\ 10^{-2} & 8 & 10^{-2} \\ 10^{-2} & 10^{-2} & 9 \end{bmatrix}$

3. $\begin{bmatrix} 0 & 0.4 & -0.1 \\ -0.4 & 0 & 0.3 \\ 0.1 & -0.3 & 0 \end{bmatrix}$ **4.** $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 4 & 3 \\ 1 & 3 & 12 \end{bmatrix}$

5. $\begin{bmatrix} 2 & i & 1+i \\ -i & 3 & 0 \\ 1-i & 0 & 8 \end{bmatrix}$ **6.** $\begin{bmatrix} 10 & 0.1 & -0.2 \\ 0.1 & 6 & 0 \\ -0.2 & 0 & 3 \end{bmatrix}$

7. Similarity. In Prob. 2, find $\mathbf{T}^{-\mathsf{T}}\mathbf{A}\mathbf{T}$ such that the radius of the Gerschgorin circle with center 5 is reduced by a factor 1/100.

8. By what integer factor can you at most reduce the Gerschgorin circle with center 3 in Prob. 6?

9. If a symmetric $n \times n$ matrix $\mathbf{A} = [a_{jk}]$ has been diagonalized except for small off-diagonal entries of size 10^{-5}, what can you say about the eigenvalues?

10. Optimality of Gerschgorin disks. Illustrate with a 2×2 matrix that an eigenvalue may very well lie on a Gerschgorin circle, so that Gerschgorin disks can generally not be replaced with smaller disks without losing the inclusion property.

11. Spectral radius $\rho(\mathbf{A})$. Using Theorem 1, show that $\rho(\mathbf{A})$ cannot be greater than the row sum norm of $\mathbf{A}$.

12–16 SPECTRAL RADIUS

Use (4) to obtain an upper bound for the spectral radius:

12. In Prob. 4 **13.** In Prob. 1

14. In Prob. 6 **15.** In Prob. 3

16. In Prob. 5

17. Verify that the matrix in Prob. 5 is normal.

18. Normal matrices. Show that Hermitian, skew-Hermitian, and unitary matrices (hence real symmetric, skew-symmetric, and orthogonal matrices) are normal. Why is this of practical interest?

19. Prove Theorem 3 by using Theorem 1.

20. Extended Gerschgorin theorem. Prove Theorem 2. *Hint.* Let $\mathbf{A} = \mathbf{B} + \mathbf{C}$, $\mathbf{B} = \text{diag}\,(a_{jj})$, $\mathbf{A}_t = \mathbf{B} + t\mathbf{C}$, and let t increase continuously from 0 to 1.

20.8 Power Method for Eigenvalues

A simple standard procedure for computing approximate values of the eigenvalues of an $n \times n$ matrix $\mathbf{A} = [a_{jk}]$ is the **power method**. In this method we start from any vector $\mathbf{x}_0 \ (\neq \mathbf{0})$ with n components and compute successively

$$\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0, \qquad \mathbf{x}_2 = \mathbf{A}\mathbf{x}_1, \qquad \cdots, \qquad \mathbf{x}_s = \mathbf{A}\mathbf{x}_{s-1}.$$

For simplifying notation, we denote $\mathbf{x}_{s-1}$ by $\mathbf{x}$ and $\mathbf{x}_s$ by $\mathbf{y}$, so that $\mathbf{y} = \mathbf{A}\mathbf{x}$.

The method applies to any $n \times n$ matrix $\mathbf{A}$ that has a **dominant eigenvalue** (a λ such that $|\lambda|$ is greater than the absolute values of the other eigenvalues). If $\mathbf{A}$ is *symmetric*, it also gives the error bound (2), in addition to the approximation (1).

THEOREM 1

Power Method, Error Bounds

Let $\mathbf{A}$ be an $n \times n$ real symmetric matrix. Let $\mathbf{x} \ (\neq \mathbf{0})$ be any real vector with n components. Furthermore, let

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \qquad m_0 = \mathbf{x}^\mathsf{T}\mathbf{x}, \qquad m_1 = \mathbf{x}^\mathsf{T}\mathbf{y}, \qquad m_2 = \mathbf{y}^\mathsf{T}\mathbf{y}.$$

Then the quotient

(1) $$q = \frac{m_1}{m_0}$$ **(Rayleigh[10] quotient)**

is an approximation for an eigenvalue λ of $\mathbf{A}$ (usually that which is greatest in absolute value, but no general statements are possible).

Furthermore, if we set $q = \lambda - \epsilon$, so that ϵ is the error of q, then

(2) $$|\epsilon| \leqq \delta = \sqrt{\frac{m_2}{m_0} - q^2}.$$

PROOF δ^2 denotes the radicand in (2). Since $m_1 = qm_0$ by (1), we have

(3) $$(\mathbf{y} - q\mathbf{x})^\mathsf{T}(\mathbf{y} - q\mathbf{x}) = m_2 - 2qm_1 + q^2 m_0 = m_2 - q^2 m_0 = \delta^2 m_0.$$

Since $\mathbf{A}$ is real symmetric, it has an orthogonal set of n real unit eigenvectors $\mathbf{z}_1, \cdots, \mathbf{z}_n$ corresponding to the eigenvalues $\lambda_1, \cdots, \lambda_n$, respectively (some of which may be equal). (Proof in Ref. [B3], vol. 1, pp. 270–272, listed in App. 1.) Then $\mathbf{x}$ has a representation of the form

$$\mathbf{x} = a_1\mathbf{z}_1 + \cdots + a_n\mathbf{z}_n.$$

[10] LORD RAYLEIGH (JOHN WILLIAM STRUTT) (1842–1919), great English physicist and mathematician, professor at Cambridge and London, known for his important contributions to various branches of applied mathematics and theoretical physics, in particular, the theory of waves, elasticity, and hydrodynamics. In 1904 he received a Nobel Prize in physics.

Now $\mathbf{A}\mathbf{z}_1 = \lambda_1\mathbf{z}_1$, etc., and we obtain

$$\mathbf{y} = \mathbf{A}\mathbf{x} = a_1\lambda_1\mathbf{z}_1 + \cdots + a_n\lambda_n\mathbf{z}_n$$

and, since the $\mathbf{z}_j$ are orthogonal unit vectors,

$$m_0 = \mathbf{x}^\mathsf{T}\mathbf{x} = a_1^2 + \cdots + a_n^2. \tag{4}$$

It follows that in (3),

$$\mathbf{y} - q\mathbf{x} = a_1(\lambda_1 - q)\mathbf{z}_1 + \cdots + a_n(\lambda_n - q)\mathbf{z}_n.$$

Since the $\mathbf{z}_j$ are orthogonal unit vectors, we thus obtain from (3)

$$\delta^2 m_0 = (y - q\mathbf{x})^\mathsf{T}(\mathbf{y} - q\mathbf{x}) = a_1^2(\lambda_1 - q)^2 + \cdots + a_n^2(\lambda_n - q)^2. \tag{5}$$

Now let λ_c be an eigenvalue of $\mathbf{A}$ to which q is closest, where c suggests "closest." Then $(\lambda_c - q)^2 \leqq (\lambda_j - q)^2$ for $j = 1, \cdots, n$. From this and (5) we obtain the inequality

$$\delta^2 m_0 \geqq (\lambda_c - q)^2(a_1^2 + \cdots + a_n^2) = (\lambda_c - q)^2 m_0.$$

Dividing by m_0, taking square roots, and recalling the meaning of δ^2 gives

$$\delta = \sqrt{\frac{m_2}{m_0} - q^2} \geqq |\lambda_c - q|.$$

This shows that δ is a bound for the error ϵ of the approximation q of an eigenvalue of $\mathbf{A}$ and completes the proof. ■

The main advantage of the method is its simplicity. And it can handle *sparse matrices* too large to store as a full square array. Its disadvantage is its possibly slow convergence. From the proof of Theorem 1 we see that the speed of convergence depends on the ratio of the dominant eigenvalue to the next in absolute value (2:1 in Example 1, below).

If we want a convergent sequence of **eigenvectors,** then at the beginning of each step we **scale** the vector, say, by dividing its components by an absolutely largest one, as in Example 1, as follows.

EXAMPLE 1 Application of Theorem 1. Scaling

For the symmetric matrix $\mathbf{A}$ in Example 4, Sec. 20.7, and $\mathbf{x}_0 = [1 \quad 1 \quad 1]^\mathsf{T}$ we obtain from (1) and (2) and the indicated scaling

$$\mathbf{A} = \begin{bmatrix} 0.49 & 0.02 & 0.22 \\ 0.02 & 0.28 & 0.20 \\ 0.22 & 0.20 & 0.40 \end{bmatrix}, \quad \mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}_1 = \begin{bmatrix} 0.890244 \\ 0.609756 \\ 1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0.931193 \\ 0.541284 \\ 1 \end{bmatrix}$$

$$\mathbf{x}_5 = \begin{bmatrix} 0.990663 \\ 0.504682 \\ 1 \end{bmatrix}, \quad \mathbf{x}_{10} = \begin{bmatrix} 0.999707 \\ 0.500146 \\ 1 \end{bmatrix}, \quad \mathbf{x}_{15} = \begin{bmatrix} 0.999991 \\ 0.500005 \\ 1 \end{bmatrix}.$$

Here $\mathbf{Ax}_0 = [0.73 \quad 0.5 \quad 0.82]^T$, scaled to $\mathbf{x}_1 = [0.73/0.82 \quad 0.5/0.82 \quad 1]^T$, etc. The dominant eigenvalue is 0.72, an eigenvector $[1 \quad 0.5 \quad 1]^T$. The corresponding q and δ are computed each time before the next scaling. Thus in the first step,

$$q = \frac{m_1}{m_0} = \frac{\mathbf{x}_0^T\mathbf{A}\mathbf{x}_0}{\mathbf{x}_0^T\mathbf{x}_0} = \frac{2.05}{3} = 0.683333$$

$$\delta = \left(\frac{m_2}{m_0} - q^2\right)^{1/2} = \left(\frac{(\mathbf{Ax}_0)^T\mathbf{Ax}_0}{\mathbf{x}_0^T\mathbf{x}_0} - q^2\right)^{1/2} = \left(\frac{1.4553}{3} - q^2\right)^{1/2} = 0.134743.$$

This gives the following values of q, δ, and the error $\epsilon = 0.72 - q$ (calculations with 10D, rounded to 6D):

j	1	2	5	10
q	0.683333	0.716048	0.719944	0.720000
δ	0.134743	0.038887	0.004499	0.000141
ϵ	0.036667	0.003952	0.000056	$5 \cdot 10^{-8}$

The error bounds are much larger than the actual errors. This is typical, although the bounds cannot be improved; that is, for special symmetric matrices they agree with the errors.

Our present results are somewhat better than those of Collatz's method in Example 4 of Sec. 20.7, at the expense of more operations. ■

Spectral shift, the transition from $\mathbf{A}$ to $\mathbf{A} - k\mathbf{I}$, shifts every eigenvalue by $-k$. Although finding a good k can hardly be made automatic, it may be helped by some other method or small preliminary computational experiments. In Example 1, Gerschgorin's theorem gives $-0.02 \leqq \lambda \leqq 0.82$ for the whole spectrum (verify!). Shifting by -0.4 might be too much (then $-0.42 \leqq \lambda \leqq 0.42$), so let us try -0.2.

EXAMPLE 2 **Power Method with Spectral Shift**

For $\mathbf{A} - 0.2\mathbf{I}$ with $\mathbf{A}$ as in Example 1 we obtain the following substantial improvements (where the index 1 refers to Example 1 and the index 2 to the present example).

j	1	2	5	10
δ_1	0.134743	0.038887	0.004499	0.000141
δ_2	0.134743	0.034474	0.000693	$1.8 \cdot 10^{-6}$
ϵ_1	0.036667	0.003952	0.000056	$5 \cdot 10^{-8}$
ϵ_2	0.036667	0.002477	$1.3 \cdot 10^{-6}$	$9 \cdot 10^{-12}$

■

PROBLEM SET 20.8

1–4 POWER METHOD WITHOUT SCALING

Apply the power method without scaling (3 steps), using $\mathbf{x}_0 = [1, \quad 1]^T$ or $[1 \quad 1 \quad 1]^T$. Give Rayleigh quotients and error bounds. Show the details of your work.

1. $\begin{bmatrix} 9 & 4 \\ 4 & 3 \end{bmatrix}$

2. $\begin{bmatrix} 7 & -3 \\ -3 & -1 \end{bmatrix}$

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

4. $\begin{bmatrix} 3.6 & -1.8 & 1.8 \\ -1.8 & 2.8 & -2.6 \\ 1.8 & -2.6 & 2.8 \end{bmatrix}$

5–8 POWER METHOD WITH SCALING

Apply the power method (3 steps) with scaling, using $\mathbf{x}_0 = [1 \quad 1 \quad 1]^T$ or $[1 \quad 1 \quad 1 \quad 1]^T$, as applicable. Give

Rayleigh quotients and error bounds. Show the details of your work.

5. The matrix in Prob. 3

6. $\begin{bmatrix} 4 & 2 & 3 \\ 2 & 7 & 6 \\ 3 & 6 & 4 \end{bmatrix}$

7. $\begin{bmatrix} 5 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 5 \end{bmatrix}$

8. $\begin{bmatrix} 2 & 4 & 0 & 1 \\ 4 & 1 & 2 & 8 \\ 0 & 2 & 5 & 2 \\ 1 & 8 & 2 & 0 \end{bmatrix}$

9. Prove that if $\mathbf{x}$ is an eigenvector, then $\delta = 0$ in (2). Give two examples.

10. Rayleigh quotient. Why does q generally approximate the eigenvalue of greatest absolute value? When will q be a good approximation?

11. Spectral shift, smallest eigenvalue. In Prob. 3 set $\mathbf{B} = \mathbf{A} - 3\mathbf{I}$ (as perhaps suggested by the diagonal entries) and see whether you may get a sequence of q's converging to an eigenvalue of $\mathbf{A}$ that *is smallest* (not largest) in absolute value. Use $\mathbf{x}_0 = [1 \quad 1 \quad 1]^T$. Do 8 steps. Verify that $\mathbf{A}$ has the spectrum $\{0, 3, 5\}$.

12. CAS EXPERIMENT. Power Method with Scaling. Shifting. (a) Write a program for $n \times n$ matrices that prints every step. Apply it to the (nonsymmetric!) matrix (20 steps), starting from $[1 \quad 1 \quad 1]^T$.

$$\mathbf{A} = \begin{bmatrix} 15 & 12 & 3 \\ 18 & 44 & 18 \\ -19 & -36 & -7 \end{bmatrix}.$$

(b) Experiment in (a) with shifting. Which shift do you find optimal?

(c) Write a program as in (a) but for symmetric matrices that prints vectors, scaled vectors, q, and δ. Apply it to the matrix in Prob. 8.

(d). Optimality of δ. Consider $\mathbf{A} = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}$ and take $\mathbf{x}_0 = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$. Show that $q = 0$, $\delta = 1$ for all steps and the eigenvalues are ± 1, so that the interval $[q - \delta, q + \delta]$ cannot be shortened (by omitting ± 1) without losing the inclusion property. Experiment with other $\mathbf{x}_0$'s.

(e) Find a (nonsymmetric) matrix for which δ in (2) is no longer an error bound.

(f) Experiment systematically with speed of convergence by choosing matrices with the second greatest eigenvalue (i) almost equal to the greatest, (ii) somewhat different, (iii) much different.

20.9 Tridiagonalization and QR-Factorization

We consider the problem of computing *all* the eigenvalues of a ***real symmetric*** matrix $\mathbf{A} = [a_{jk}]$, discussing a method widely used in practice. In the ***first stage*** we reduce the given matrix stepwise to a **tridiagonal matrix**, that is, a matrix having all its nonzero entries on the main diagonal and in the positions immediately adjacent to the main diagonal (such as $\mathbf{A}_3$ in Fig. 450, Third Step). This reduction was invented by A. S. Householder[11] (*J. Assn. Comput. Machinery* **5** (1958), 335–342). See also Ref. [E29] in App. 1.

This Householder tridiagonalization will simplify the matrix without changing its eigenvalues. The latter will then be determined (approximately) by factoring the tridiagonalized matrix, as discussed later in this section.

[11]ALSTON SCOTT HOUSEHOLDER (1904–1993), American mathematician, known for his work in numerical analysis and mathematical biology. He was head of the mathematics division at Oakridge National Laboratory and later professor at the University of Tennessee. He was both president of ACM (Association for Computing Machinery) 1954–1956 and SIAM (Society for Industrial and Applied Mathematics) 1963–1964.

Householder's Tridiagonalization Method[11]

An $n \times n$ real symmetric matrix $\mathbf{A} = [a_{jk}]$ being given, we reduce it by $n - 2$ successive similarity transformations (see Sec. 20.6) involving matrices $\mathbf{P}_1, \cdots, \mathbf{P}_{n-2}$ to tridiagonal form. These matrices are orthogonal and symmetric. Thus $\mathbf{P}_1^{-1} = \mathbf{P}_1^{\mathsf{T}} = \mathbf{P}_1$ and similarly for the others. These transformations produce, from the given $\mathbf{A}_0 = \mathbf{A} = [a_{jk}]$, the matrices $\mathbf{A}_1 = [a_{jk}^{(1)}]$, $\mathbf{A}_2 = [a_{jk}^{(2)}], \cdots, \mathbf{A}_{n-2} = [a_{jk}^{(n-2)}]$ in the form

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{P}_1\mathbf{A}_0\mathbf{P}_1 \\ \mathbf{A}_2 &= \mathbf{P}_2\mathbf{A}_1\mathbf{P}_2 \\ &\cdots\cdots\cdots\cdots \\ \mathbf{B} &= \mathbf{A}_{n-2} = \mathbf{P}_{n-2}\mathbf{A}_{n-3}\mathbf{P}_{n-2}. \end{aligned} \tag{1}$$

The transformations (1) create the necessary zeros, in the first step in Row 1 and Column 1, in the second step in Row 2 and Column 2, etc., as Fig. 450 illustrates for a 5×5 matrix. $\mathbf{B}$ is tridiagonal.

Fig. 450. Householder's method for a 5×5 matrix. Positions left blank are zeros created by the method.

How do we determine $\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_{n-2}$? Now, all these $\mathbf{P}_r$ are of the form

$$\mathbf{P}_r = \mathbf{I} - 2\mathbf{v}_r\mathbf{v}_r^{\mathsf{T}} \qquad (r = 1, \cdots, n-2) \tag{2}$$

where $\mathbf{I}$ is the $n \times n$ unit matrix and $\mathbf{v}_r = [v_{jr}]$ is a unit vector with its first r components 0; thus

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ * \\ * \\ \vdots \\ * \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ * \\ \vdots \\ * \end{bmatrix}, \quad \cdots, \quad \mathbf{v}_{n-2} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ * \\ * \end{bmatrix} \tag{3}$$

where the asterisks denote the other components (which will be nonzero in general).

Step 1. $\mathbf{v}_1$ has the components

(4)

$$v_{11} = 0$$

(a) $$v_{21} = \sqrt{\frac{1}{2}\left(1 + \frac{|a_{21}|}{S_1}\right)}$$

(b) $$v_{j1} = \frac{a_{j1} \operatorname{sgn} a_{21}}{2v_{21}S_1} \qquad j = 3, 4, \cdots, n$$

where

(c) $$S_1 = \sqrt{a_{21}^2 + a_{31}^2 + \cdots + a_{n1}^2}$$

where $S_1 > 0$, and $\operatorname{sgn} a_{21} = +1$ if $a_{21} \geqq 0$ and $\operatorname{sgn} a_{21} = -1$ if $a_{21} < 0$. With this we compute $\mathbf{P}_1$ by (2) and then $\mathbf{A}_1$ by (1). This was the first step.

Step 2. We compute $\mathbf{v}_2$ by (4) with all subscripts increased by 1 and the a_{jk} replaced by $a_{jk}^{(1)}$, the entries of $\mathbf{A}_1$ just computed. Thus [see also (3)]

(4*)

$$v_{12} = v_{22} = 0$$

$$v_{32} = \sqrt{\frac{1}{2}\left(1 + \frac{|a_{32}^{(1)}|}{S_2}\right)}$$

$$v_{j2} = \frac{a_{j2}^{(1)} \operatorname{sgn} a_{32}^{(1)}}{2v_{32}S_2} \qquad j = 4, 5, \cdots, n$$

where

$$S_2 = \sqrt{a_{32}^{(1)^2} + a_{42}^{(1)^2} + \cdots + a_{n2}^{(1)^2}}.$$

With this we compute $\mathbf{P}_2$ by (2) and then $\mathbf{A}_2$ by (1).

Step 3. We compute $\mathbf{v}_3$ by (4*) with all subscripts increased by 1 and the $a_{jk}^{(1)}$ replaced by the entries $a_{jk}^{(2)}$ of $\mathbf{A}_2$, and so on.

EXAMPLE 1 Householder Tridiagonalization

Tridiagonalize the real symmetric matrix

$$\mathbf{A} = \mathbf{A}_0 = \begin{bmatrix} 6 & 4 & 1 & 1 \\ 4 & 6 & 1 & 1 \\ 1 & 1 & 5 & 2 \\ 1 & 1 & 2 & 5 \end{bmatrix}.$$

Solution. ***Step 1.*** We compute $S_1^2 = 4^2 + 1^2 + 1^2 = 18$ from (4c). Since $a_{21} = 4 > 0$, we have $\operatorname{sgn} a_{21} = +1$ in (4b) and get from (4) by straightforward computation

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ v_{21} \\ v_{31} \\ v_{41} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.98559856 \\ 0.11957316 \\ 0.11957316 \end{bmatrix}.$$

From this and (2),

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.94280904 & -0.23570227 & -0.23570227 \\ 0 & -0.23570227 & 0.97140452 & -0.02859548 \\ 0 & -0.23570227 & -0.02859548 & 0.97140452 \end{bmatrix}.$$

From the first line in (1) we now get

$$\mathbf{A}_1 = \mathbf{P}_1\mathbf{A}_0\mathbf{P}_1 = \begin{bmatrix} 6 & -\sqrt{18} & 0 & 0 \\ -\sqrt{18} & 7 & -1 & -1 \\ 0 & -1 & \frac{9}{2} & \frac{3}{2} \\ 0 & -1 & \frac{3}{2} & \frac{9}{2} \end{bmatrix}.$$

Step 2. From (4*) we compute $S_2^2 = 2$ and

$$\mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ v_{32} \\ v_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.92387953 \\ 0.38268343 \end{bmatrix}.$$

From this and (2),

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 0 & -1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}.$$

The second line in (1) now gives

$$\mathbf{B}_2 = \mathbf{A}_2 = \mathbf{P}_2\mathbf{A}_1\mathbf{P}_2 = \begin{bmatrix} 6 & -\sqrt{18} & 0 & 0 \\ -\sqrt{18} & 7 & \sqrt{2} & 0 \\ 0 & \sqrt{2} & 6 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}.$$

This matrix $\mathbf{B}$ is tridiagonal. Since our given matrix has order $n = 4$, we needed $n - 2 = 2$ steps to accomplish this reduction, as claimed. (Do you see that we got more zeros than we can expect in general?)

$\mathbf{B}$ is similar to $\mathbf{A}$, as we now show in general. This is essential because $\mathbf{B}$ thus has the same spectrum as $\mathbf{A}$, by Theorem 2 in Sec. 20.6. ■

B *Similar to* A. We assert that $\mathbf{B}$ in (1) is similar to $\mathbf{A} = \mathbf{A}_0$. The matrix $\mathbf{P}_r$ is symmetric; indeed,

$$\mathbf{P}_r{}^{\mathsf{T}} = (\mathbf{I} - 2\mathbf{v}_r\mathbf{v}_r{}^{\mathsf{T}})^{\mathsf{T}} = \mathbf{I}^{\mathsf{T}} - 2(\mathbf{v}_r\mathbf{v}_r{}^{\mathsf{T}})^{\mathsf{T}} = \mathbf{I} - 2\mathbf{v}_r\mathbf{v}_r{}^{\mathsf{T}} = \mathbf{P}_r$$

Also, $\mathbf{P}_r$ is orthogonal because $\mathbf{v}_r$ is a unit vector, so that $\mathbf{v}_r^{\mathsf{T}}\mathbf{v}_r = 1$ and thus

$$\mathbf{P}_r\mathbf{P}_r^{\mathsf{T}} = \mathbf{P}_r^2 = (\mathbf{I} - 2\mathbf{v}_r\mathbf{v}_r^{\mathsf{T}})^2 = \mathbf{I} - 4\mathbf{v}_r\mathbf{v}_r^{\mathsf{T}} + 4\mathbf{v}_r\mathbf{v}_r^{\mathsf{T}}\mathbf{v}_r\mathbf{v}_r^{\mathsf{T}}$$
$$= \mathbf{I} - 4\mathbf{v}_r\mathbf{v}_r^{\mathsf{T}} + 4\mathbf{v}_r(\mathbf{v}_r^{\mathsf{T}}\mathbf{v}_r)\mathbf{v}_r^{\mathsf{T}} = \mathbf{I}.$$

Hence $\mathbf{P}_r^{-1} = \mathbf{P}_r^{\mathsf{T}} = \mathbf{P}_r$ and from (1) we now obtain

$$\begin{aligned}\mathbf{B} &= \mathbf{P}_{n-2}\mathbf{A}_{n-3}\mathbf{P}_{n-2} = \cdots \\ \cdots &= \mathbf{P}_{n-2}\mathbf{P}_{n-3}\cdots\mathbf{P}_1\mathbf{A}\mathbf{P}_1\cdots\mathbf{P}_{n-3}\mathbf{P}_{n-2} \\ &= \mathbf{P}_{n-2}^{-1}\mathbf{P}_{n-3}^{-1}\cdots\mathbf{P}_1^{-1}\mathbf{A}\mathbf{P}_1\cdots\mathbf{P}_{n-3}\mathbf{P}_{n-2} \\ &= \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\end{aligned}$$

where $\mathbf{P} = \mathbf{P}_1\mathbf{P}_2\cdots\mathbf{P}_{n-2}$. This proves our assertion. ■

QR-Factorization Method

In 1958 H. Rutishauser[12] of Switzerland proposed the idea of using the LU-factorization (Sec. 20.2; he called it LR-factorization) in solving eigenvalue problems. An improved version of Rutishauser's method (avoiding breakdown if certain submatrices become singular, etc.; see Ref. [E29]) is the QR-method, independently proposed by the American J. G. F. Francis (*Computer J.* **4** (1961–62), 265–271, 332–345) and the Russian V. N. Kublanovskaya (*Zhurnal Vych. Mat. i Mat. Fiz.* **1** (1961), 555–570). The QR-method uses the factorization **QR** with orthogonal **Q** and upper triangular **R.** We discuss the **QR**-method for a real ***symmetric*** matrix. (For extensions to general matrices see Ref. [E29] in App. 1.)

In this method we first transform a given real symmetric $n \times n$ matrix $\mathbf{A}$ into a tridiagonal matrix $\mathbf{B}_0 = \mathbf{B}$ by Householder's method. This creates many zeros and thus reduces the amount of further work. Then we compute $\mathbf{B}_1, \mathbf{B}_2, \cdots$ stepwise according to the following iteration method.

Step 1. Factor $\mathbf{B}_0 = \mathbf{Q}_0\mathbf{R}_0$ with orthogonal $\mathbf{Q}_0$ and upper triangular $\mathbf{R}_0$. Then compute $\mathbf{B}_1 = \mathbf{R}_0\mathbf{Q}_0$.

Step 2. Factor $\mathbf{B}_1 = \mathbf{Q}_1\mathbf{R}_1$. Then compute $\mathbf{B}_2 = \mathbf{R}_1\mathbf{Q}_1$.

General Step s + 1.

(5)

(a) Factor $\mathbf{B}_s = \mathbf{Q}_s\mathbf{R}_s$.

(b) Compute $\mathbf{B}_{s+1} = \mathbf{R}_s\mathbf{Q}_s$.

Here $\mathbf{Q}_s$ is orthogonal and $\mathbf{R}_s$ upper triangular. The factorization (5a) will be explained below.

$\mathbf{B}_{s+1}$ Similar to B. Convergence to a Diagonal Matrix. From (5a) we have $\mathbf{R}_s = \mathbf{Q}_s^{-1}\mathbf{B}_s$. Substitution into (5b) gives

$$\mathbf{B}_{s+1} = \mathbf{R}_s\mathbf{Q}_s = \mathbf{Q}_s^{-1}\mathbf{B}_s\mathbf{Q}_s. \tag{6}$$

[12]HEINZ RUTISHAUSER (1918–1970). Swiss mathematician, professor at ETH Zurich. Known for his pioneering work in numerics and computer science.

Thus $\mathbf{B}_{s+1}$ is similar to $\mathbf{B}_s$. Hence $\mathbf{B}_{s+1}$ is similar to $\mathbf{B}_0 = \mathbf{B}$ for all s. By Theorem 2, Sec. 20.6, this implies that $\mathbf{B}_{s+1}$ has the same eigenvalues as $\mathbf{B}$.

Also, $\mathbf{B}_{s+1}$ is symmetric. This follows by induction. Indeed, $\mathbf{B}_0 = \mathbf{B}$ is symmetric. Assuming $\mathbf{B}_s$ to be symmetric, that is, $\mathbf{B}_s{}^{\mathsf{T}} = \mathbf{B}_s$, and using $\mathbf{Q}_s^{-1} = \mathbf{Q}_s{}^{\mathsf{T}}$ (since $\mathbf{Q}_s$ is orthogonal), we get from (6) the symmetry,

$$\mathbf{B}_{s+1}{}^{\mathsf{T}} = (\mathbf{Q}_s{}^{\mathsf{T}}\mathbf{B}_s\mathbf{Q}_s)^{\mathsf{T}} = Q_s{}^{\mathsf{T}}\mathbf{B}_s{}^{\mathsf{T}}\mathbf{Q}_s = \mathbf{Q}_s{}^{\mathsf{T}}\mathbf{B}_s\mathbf{Q}_s = \mathbf{B}_{s+1}.$$

If the eigenvalues of $\mathbf{B}$ are different in absolute value, say, $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$, then

$$\lim_{s\to\infty} \mathbf{B}_s = \mathbf{D}$$

where $\mathbf{D}$ is diagonal, with main diagonal entries $\lambda_1, \lambda_2, \cdots, \lambda_n$. (Proof in Ref. [E29] listed in App. 1.)

How to Get the QR-Factorization, say, $\mathbf{B} = \mathbf{B}_0 = [b_{jk}] = \mathbf{Q}_0\mathbf{R}_0$. The tridiagonal matrix $\mathbf{B}$ has $n - 1$ generally nonzero entries below the main diagonal. These are $b_{21}, b_{32}, \cdots, b_{n,n-1}$. We multiply $\mathbf{B}$ from the left by a matrix $\mathbf{C}_2$ such that $\mathbf{C}_2\mathbf{B} = [b_{jk}^{(2)}]$ has $b_{21}^{(2)} = 0$. We multiply this by a matrix $\mathbf{C}_3$ such that $\mathbf{C}_3\mathbf{C}_2\mathbf{B} = [b_{jk}^{(3)}]$ has $b_{32}^{(3)} = 0$, etc. After $n - 1$ such multiplications we are left with an upper triangular matrix $\mathbf{R}_0$, namely,

$$\mathbf{C}_n\mathbf{C}_{n-1}\cdots\mathbf{C}_3\mathbf{C}_2B_0 = \mathbf{R}_0. \tag{7}$$

These $n \times n$ matrices $\mathbf{C}_j$ are very simple. $\mathbf{C}_j$ has the 2×2 submatrix

$$\begin{bmatrix} \cos\theta_j & \sin\theta_j \\ -\sin\theta_j & \cos\theta_j \end{bmatrix} \qquad (\theta_j \text{ suitable})$$

in Rows $j - 1$ and j and Columns $j - 1$ and j; everywhere else on the main diagonal the matrix $\mathbf{C}_j$ has entries 1; and all its other entries are 0. (This submatrix is the matrix of a plane rotation through the angle θ_j; see Team Project 30, Sec. 7.2.) For instance, if $n = 4$, writing $c_j = \cos\theta_j$, $s_j = \sin\theta_j$, we have

$$\mathbf{C}_2 = \begin{bmatrix} c_2 & s_2 & 0 & 0 \\ -s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_3 & s_3 & 0 \\ 0 & -s_3 & c_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_4 & s_4 \\ 0 & 0 & -s_4 & c_4 \end{bmatrix}.$$

These $\mathbf{C}_j$ are orthogonal. Hence their product in (7) is orthogonal, and so is the inverse of this product. We call this inverse $\mathbf{Q}_0$. Then from (7),

$$\mathbf{B}_0 = \mathbf{Q}_0\mathbf{R}_0 \tag{8}$$

where, with $\mathbf{C}_j^{-1} = \mathbf{C}_j{}^{\mathsf{T}}$,

$$\mathbf{Q}_0 = (\mathbf{C}_n\mathbf{C}_{n-1}\cdots\mathbf{C}_3\mathbf{C}_2)^{-1} = \mathbf{C}_2{}^{\mathsf{T}}\mathbf{C}_3{}^{\mathsf{T}}\cdots\mathbf{C}_{n-1}{}^{\mathsf{T}}\mathbf{C}_n{}^{\mathsf{T}}. \tag{9}$$

This is our QR-factorization of $\mathbf{B}_0$. From it we have by (5b) with $s = 0$

$$\mathbf{B}_1 = \mathbf{R}_0\mathbf{Q}_0 = \mathbf{R}_0\mathbf{C}_2{}^{\mathsf{T}}\mathbf{C}_3{}^{\mathsf{T}}\cdots\mathbf{C}_{n-1}{}^{\mathsf{T}}\mathbf{C}_n{}^{\mathsf{T}}. \tag{10}$$

We do not need $\mathbf{Q}_0$ explicitly, but to get $\mathbf{B}_1$ from (10), we first compute $\mathbf{R}_0\mathbf{C}_2{}^{\mathsf{T}}$, then $(\mathbf{R}_0\mathbf{C}_2^{\mathsf{T}})\mathbf{C}_3^{\mathsf{T}}$, etc. Similarly in the further steps that produce $\mathbf{B}_2, \mathbf{B}_3, \cdots$.

Determination of $\cos\theta_j$ and $\sin\theta_j$. We finally show how to find the angles of rotation. $\cos\theta_2$ and $\sin\theta_2$ in $\mathbf{C}_2$ must be such that $b_{21}^{(2)} = 0$ in the product

$$\mathbf{C}_2\mathbf{B} = \begin{bmatrix} c_2 & s_2 & 0 & \cdots \\ -s_2 & c_2 & 0 & \cdots \\ \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdots \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots \\ b_{21} & b_{22} & b_{23} & \cdots \\ \cdot & \cdot & \cdot & \cdots \\ \cdot & \cdot & \cdot & \cdots \end{bmatrix}.$$

Now $b_{21}^{(2)}$ is obtained by multiplying the second row of $\mathbf{C}_2$ by the first column of $\mathbf{B}$,

$$b_{21}^{(2)} = -s_2b_{11} + c_2b_{21} = -(\sin\theta_2)b_{11} + (\cos\theta_2)b_{21} = 0.$$

Hence $\tan\theta_2 = s_2/c_2 = b_{21}/b_{11}$, and

$$\cos\theta_2 = \frac{1}{\sqrt{1 + \tan^2\theta_2}} = \frac{1}{\sqrt{1 + (b_{21}/b_{11})^2}}$$

$$\sin\theta_2 = \frac{\tan\theta_2}{\sqrt{1 + \tan^2\theta_2}} = \frac{b_{21}/b_{11}}{\sqrt{1 + (b_{21}/b_{11})^2}}. \tag{11}$$

Similarly for $\theta_3, \theta_4, \cdots$. The next example illustrates all this.

EXAMPLE 2 QR-Factorization Method

Compute all the eigenvalues of the matrix

$$\mathbf{A} = \begin{bmatrix} 6 & 4 & 1 & 1 \\ 4 & 6 & 1 & 1 \\ 1 & 1 & 5 & 2 \\ 1 & 1 & 2 & 5 \end{bmatrix}.$$

Solution. We first reduce $\mathbf{A}$ to tridiagonal form. Applying Householder's method, we obtain (see Example 1)

$$\mathbf{A}_2 = \begin{bmatrix} 6 & -\sqrt{18} & 0 & 0 \\ -\sqrt{18} & 7 & \sqrt{2} & 0 \\ 0 & \sqrt{2} & 6 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}.$$

From the characteristic determinant we see that $\mathbf{A}_2$, hence $\mathbf{A}$, has the eigenvalue 3. (Can you see this directly from $\mathbf{A}_2$?) Hence it suffices to apply the QR-method to the tridiagonal 3×3 matrix

$$\mathbf{B}_0 = \mathbf{B} = \begin{bmatrix} 6 & -\sqrt{18} & 0 \\ -\sqrt{18} & 7 & \sqrt{2} \\ 0 & \sqrt{2} & 6 \end{bmatrix}.$$

Step 1. We multiply $\mathbf{B}$ from the left by

$$\mathbf{C}_2 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 \\ -\sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and then } \mathbf{C}_2\mathbf{B} \text{ by} \quad \mathbf{C}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix}.$$

Here $(-\sin\theta_2) \cdot 6 + (\cos\theta_2)(-\sqrt{18}) = 0$ gives (11) $\cos\theta_2 = 0.81649658$ and $\sin\theta_2 = -0.57735027$. With these values we compute

$$\mathbf{C}_2\mathbf{B} = \begin{bmatrix} 7.34846923 & -7.50555350 & -0.81649658 \\ 0 & 3.26598632 & 1.15470054 \\ 0 & 1.41421356 & 6.00000000 \end{bmatrix}.$$

In $\mathbf{C}_3$ we get from $(-\sin\theta_3) \cdot 3.26598632 + (\cos\theta_3) \cdot 1.41421356 = 0$ the values $\cos\theta_3 = 0.91766294$ and $\sin\theta_3 = 0.39735971$. This gives

$$\mathbf{R}_0 = \mathbf{C}_3\mathbf{C}_2\mathbf{B} = \begin{bmatrix} 7.34846923 & -7.50555350 & -0.81649658 \\ 0 & 3.55902608 & 3.44378413 \\ 0 & 0 & 5.04714615 \end{bmatrix}.$$

From this we compute

$$\mathbf{B}_1 = \mathbf{R}_0\mathbf{C}_2{}^{\mathsf{T}}\mathbf{C}_3{}^{\mathsf{T}} = \begin{bmatrix} 10.33333333 & -2.05480467 & 0 \\ -2.05480467 & 4.03508772 & 2.00553251 \\ 0 & 2.00553251 & 4.63157895 \end{bmatrix}$$

which is symmetric and tridiagonal. The off-diagonal entries in $\mathbf{B}_1$ are still large in absolute value. Hence we have to go on.

Step 2. We do the same computations as in the first step, with $\mathbf{B}_0 = \mathbf{B}$ replaced by $\mathbf{B}_1$ and $\mathbf{C}_2$ and $\mathbf{C}_3$ changed accordingly, the new angles being $\theta_2 = -0.196291533$ and $\theta_3 = 0.513415589$. We obtain

$$\mathbf{R}_1 = \begin{bmatrix} 10.53565375 & -2.80232241 & -0.39114588 \\ 0 & 4.08329584 & 3.98824028 \\ 0 & 0 & 3.06832668 \end{bmatrix}$$

and from this

$$\mathbf{B}_2 = \begin{bmatrix} 10.87987988 & -0.79637918 & 0 \\ -0.79637918 & 5.44738664 & 1.50702500 \\ 0 & 1.50702500 & 2.67273348 \end{bmatrix}.$$

We see that the off-diagonal entries are somewhat smaller in absolute value than those of $\mathbf{B}_1$, but still much too large for the diagonal entries to be good approximations of the eigenvalues of $\mathbf{B}$.

Further Steps. We list the main diagonal entries and the absolutely largest off-diagonal entry, which is $|b_{12}^{(j)}| = |b_{21}^{(j)}|$ in all steps. You may show that the given matrix **A** has the spectrum 11, 6, 3, 2.

Step j	$b_{11}^{(j)}$	$b_{22}^{(j)}$	$b_{33}^{(j)}$	$\max_{j \neq k} \lvert b_{jk}^{(J)} \rvert$
3	10.9668929	5.94589856	2.08720851	0.58523582
5	10.9970872	6.00181541	2.00109738	0.12065334
7	10.9997421	6.00024439	2.00001355	0.03591107
9	10.9999772	6.00002267	2.00000017	0.01068477

Looking back at our discussion, we recognize that the purpose of applying Householder's tridiagonalization before the QR-factorization method is a substantial reduction of cost in each QR-factorization, in particular if **A** is large.

Convergence acceleration and thus further reduction of cost can be achieved by a **spectral shift,** that is, by taking $\mathbf{B}_s - k_s\mathbf{I}$ instead of $\mathbf{B}_s$ with a suitable k_s. Possible choices of k_s are discussed in Ref. [E29], p. 510.

PROBLEM SET 20.9

1–5 HOUSEHOLDER TRIDIAGONALIZATION

Tridiagonalize. Show the details.

1. $\begin{bmatrix} 0.98 & 0.04 & 0.44 \\ 0.04 & 0.56 & 0.40 \\ 0.44 & 0.40 & 0.80 \end{bmatrix}$

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

3. $\begin{bmatrix} 7 & 2 & 3 \\ 2 & 10 & 6 \\ 3 & 6 & 7 \end{bmatrix}$

4. $\begin{bmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{bmatrix}$

5. $\begin{bmatrix} 3 & 52 & 10 & 42 \\ 52 & 59 & 44 & 80 \\ 10 & 44 & 39 & 42 \\ 42 & 80 & 42 & 35 \end{bmatrix}$

6–9 QR-FACTORIZATION

Do three QR-steps to find approximations of the eigenvalues of:

6. The matrix in the answer to Prob. 1

7. The matrix in the answer to Prob. 3

8. $\begin{bmatrix} 14.2 & -0.1 & 0 \\ -0.1 & -6.3 & 0.2 \\ 0 & 0.2 & 2.1 \end{bmatrix}$

9. $\begin{bmatrix} 140 & 10 & 0 \\ 10 & 70 & 2 \\ 0 & 2 & -30 \end{bmatrix}$

10. CAS EXPERIMENT. QR-Method. Try to find out experimentally on what properties of a matrix the speed of decrease of off-diagonal entries in the QR-method depends. For this purpose write a program that first tridiagonalizes and then does QR-steps. Try the program out on the matrices in Probs. 1, 3, and 4. Summarize your findings in a short report.

CHAPTER 20 REVIEW QUESTIONS AND PROBLEMS

1. What are the main problem areas in numeric linear algebra?
2. When would you apply Gauss elimination and when Gauss–Seidel iteration?
3. What is pivoting? Why and how is it done?
4. What happens if you apply Gauss elimination to a system that has no solutions?
5. What is Cholesky's method? When would you apply it?

6. What do you know about the convergence of the Gauss–Seidel iteration?
7. What is ill-conditioning? What is the condition number and its significance?
8. Explain the idea of least squares approximation.
9. What are eigenvalues of a matrix? Why are they important? Give typical examples.
10. How did we use similarity transformations of matrices in designing numeric methods?
11. What is the power method for eigenvalues? What are its advantages and disadvantages?
12. State Gerschgorin's theorem from memory. Give typical applications.
13. What is tridiagonalization and QR? When would you apply it?

14–17 GAUSS ELIMINATION

Solve

14.
$$\begin{aligned} 3x_2 - 6x_3 &= 0 \\ 4x_1 - x_2 + 2x_3 &= 16 \\ -5x_1 + 2x_2 - 4x_3 &= -20 \end{aligned}$$

15.
$$\begin{aligned} 8x_2 - 6x_3 &= 23.6 \\ 10x_1 + 6x_2 + 2x_3 &= 68.4 \\ 12x_1 - 14x_2 + 4x_3 &= -6.2 \end{aligned}$$

16.
$$\begin{aligned} 5x_1 + x_2 - 3x_3 &= 17 \\ -5x_2 + 15x_3 &= -10 \\ 2x_1 - 3x_2 + 9x_3 &= 0 \end{aligned}$$

17.
$$\begin{aligned} 42x_1 + 74x_2 + 36x_3 &= 96 \\ -46x_1 - 12x_2 - 2x_3 &= 82 \\ 3x_1 + 25x_2 + 5x_3 &= 19 \end{aligned}$$

18–20 INVERSE MATRIX

Compute the inverse of:

18.
$$\begin{bmatrix} 2.0 & 0.1 & 3.3 \\ 1.6 & 4.4 & 0.5 \\ 0.3 & -4.3 & 2.8 \end{bmatrix}$$

19.
$$\begin{bmatrix} 15 & 20 & 10 \\ 20 & 35 & 15 \\ 10 & 15 & 90 \end{bmatrix}$$

20.
$$\begin{bmatrix} 5 & 1 & 1 \\ 1 & 6 & 0 \\ 1 & 0 & 8 \end{bmatrix}$$

21–23 GAUSS–SEIDEL ITERATION

Do 3 steps without scaling, starting from $[1 \quad 1 \quad 1]^T$.

21.
$$\begin{aligned} 4x_1 - x_2 &= 22.0 \\ 4x_2 - x_3 &= 13.4 \\ -x_1 + 4x_3 &= -2.4 \end{aligned}$$

22.
$$\begin{aligned} 0.2x_1 + 4.0x_2 - 0.4x_3 &= 32.0 \\ 0.5x_1 - 0.2x_2 + 2.5x_3 &= -5.1 \\ 7.5x_1 + 0.1x_2 - 1.5x_3 &= -12.7 \end{aligned}$$

23.
$$\begin{aligned} 10x_1 + x_2 - x_3 &= 17 \\ 2x_1 + 20x_2 + x_3 &= 28 \\ 3x_1 - x_2 + 25x_3 &= 105 \end{aligned}$$

24–26 VECTOR NORMS

Compute the ℓ_1-, ℓ_2-, and ℓ_∞-norms of the vectors.

24. $[0.2 \quad -8.1 \quad 0.4 \quad 0 \quad 0 \quad -1.3 \quad 2]^T$

25. $[8 \quad -21 \quad 13 \quad 0]^T$

26. $[0 \quad 0 \quad 0 \quad -1 \quad 0]^T$

27–30 MATRIX NORM

Compute the matrix norm corresponding to the ℓ_∞-vector norm for the coefficient matrix:

27. In Prob. 15

28. In Prob. 17

29. In Prob. 21

30. In Prob. 22

31–33 CONDITION NUMBER

Compute the condition number (corresponding to the ℓ_∞-vector norm) of the coefficient matrix:

31. In Prob. 19

32. In Prob. 18

33. In Prob. 21

34–35 FITTING BY LEAST SQUARES

Fit and graph:

34. A straight line to $(-1, 0)$, $(0, 2)$, $(1, 2)$, $(2, 3)$, $(3, 3)$

35. A quadratic parabola to the data in Prob. 34.

36–39 EIGENVALUES

Find and graph three circular disks that must contain all the eigenvalues of the matrix:

36. In Prob. 18

37. In Prob. 19

38. In Prob. 20

39. Of the coefficients in Prob. 14

40. Power method. Do 4 steps with scaling for the matrix in Prob. 19, starting for [1 1 1] and computing the Rayliegh quotients and error bounds.

SUMMARY OF CHAPTER 20
Numeric Linear Algebra

Main tasks are the numeric solution of linear systems (Secs. 20.1–20.4), curve fitting (Sec. 20.5), and eigenvalue problems (Secs. 20.6–20.9).

Linear systems $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} = [a_{jk}]$, written out

$$\begin{aligned} \mathrm{E}_1\text{:} \quad & a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \mathrm{E}_2\text{:} \quad & a_{21}x_1 + \cdots + a_{2n}x_n = b_2 \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \mathrm{E}_n\text{:} \quad & a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{aligned} \tag{1}$$

can be solved by a **direct method** (one in which the number of numeric operations can be specified in advance, e.g., Gauss's elimination) or by an **indirect** or **iterative method** (in which an initial approximation is improved stepwise).

The **Gauss elimination** (Sec. 20.1) is direct, namely, a systematic elimination process that reduces (1) stepwise to triangular form. In Step 1 we eliminate x_1 from equations E_2 to E_n by subtracting $(a_{21}/a_{11})\,\mathrm{E}_1$ from E_2, then $(a_{31}/a_{11})\,\mathrm{E}_1$ from E_3, etc. Equation E_1 is called the **pivot equation** in this step and a_{11} the **pivot.** In Step 2 we take the new second equation as pivot equation and eliminate x_2, etc. If the triangular form is reached, we get x_n from the last equation, then x_{n-1} from the second last, etc. **Partial pivoting** (= interchange of equations) is *necessary* if candidates for pivots are zero, and *advisable* if they are small in absolute value.

Doolittle's, Crout's, and **Cholesky's methods** in Sec. 20.2 are variants of the Gauss elimination. They factor $\mathbf{A} = \mathbf{LU}$ ($\mathbf{L}$ lower triangular, $\mathbf{U}$ upper triangular) and solve $\mathbf{Ax} = \mathbf{LUx} = \mathbf{b}$ by solving $\mathbf{Ly} = \mathbf{b}$ for $\mathbf{y}$ and then $\mathbf{Ux} = \mathbf{y}$ for $\mathbf{x}$.

In the **Gauss–Seidel iteration** (Sec. 20.3) we make $a_{11} = a_{22} = \cdots = a_{nn} = 1$ (by division) and write $\mathbf{Ax} = (\mathbf{I} + \mathbf{L} + \mathbf{U})\mathbf{x} = \mathbf{b}$; thus $\mathbf{x} = \mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}$, which suggests the iteration formula

$$\mathbf{x}^{(m+1)} = \mathbf{b} - \mathbf{L}\mathbf{x}^{(m+1)} - \mathbf{U}\mathbf{x}^{(m)} \tag{2}$$

in which we always take the most recent approximate x_j's on the right. If $\|\mathbf{C}\| < 1$, where $\mathbf{C} = -(\mathbf{I} + \mathbf{L})^{-1}\mathbf{U}$, then this process converges. Here, $\|\mathbf{C}\|$ denotes any matrix norm (Sec. 20.3).

If the **condition number** $k(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|$ of $\mathbf{A}$ is large, then the system $\mathbf{Ax} = \mathbf{b}$ is **ill-conditioned** (Sec. 20.4), and a small **residual** $\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}$ does ***not*** imply that $\tilde{\mathbf{x}}$ is close to the exact solution.

The fitting of a polynomial $p(x) = b_0 + b_1x + \cdots + b_mx^m$ through given data (points in the xy-plane) $(x_1, y_1), \cdots, (x_n, y_n)$ by the method of **least squares** is discussed in Sec. 20.5 (and in statistics in Sec. 25.9). If $m = n$, the least squares polynomial will be the same as an interpolating polynomial (uniqueness).

Eigenvalues λ (values λ for which $\mathbf{Ax} = \lambda\mathbf{x}$ has a solution $\mathbf{x} \neq \mathbf{0}$, called an **eigenvector**) can be characterized by inequalities (Sec. 20.7), e.g. in **Gerschgorin's theorem,** which gives n circular disks which contain the whole spectrum (all eigenvalues) of $\mathbf{A}$, of centers a_{jj} and radii $\Sigma|a_{jk}|$ (sum over k from 1 to n, $k \neq j$).

Approximations of eigenvalues can be obtained by iteration, starting from an $\mathbf{x}_0 \neq \mathbf{0}$ and computing $\mathbf{x}_1 = \mathbf{Ax}_0$, $\mathbf{x}_2 = \mathbf{Ax}_1, \cdots, \mathbf{x}_n = \mathbf{Ax}_{n-1}$. In this **power method** (Sec. 20.8) the **Rayleigh quotient**

$$q = \frac{(\mathbf{Ax})^{\mathsf{T}})\mathbf{x}}{\mathbf{x}^{\mathsf{T}}\mathbf{x}} \qquad (\mathbf{x} = \mathbf{x}_n) \tag{3}$$

gives an approximation of an eigenvalue (usually that of the greatest absolute value) and, if $\mathbf{A}$ is symmetric, an error bound is

$$|\epsilon| \leqq \sqrt{\frac{(\mathbf{Ax})^{\mathsf{T}}\mathbf{Ax}}{\mathbf{x}^{\mathsf{T}}\mathbf{x}} - q^2}. \tag{4}$$

Convergence may be slow but can be improved by a *spectral shift.*

For determining all the eigenvalues of a symmetric matrix $\mathbf{A}$ it is best to first tridiagonalize $\mathbf{A}$ and then to apply the QR-method (Sec. 20.9), which is based on a factorization $\mathbf{A} = \mathbf{QR}$ with orthogonal $\mathbf{Q}$ and upper triangular $\mathbf{R}$ and uses similarity transformations.