Linear systems of equations are associated with many problems in engineering and science, as well as with applications of mathematics to the social sciences and the quantitative study of business and economic problems.

In this chapter, direct techniques are considered to solve the linear system

$$\begin{aligned} E_1: &\quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ E_2: &\quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ &\qquad\qquad\qquad\qquad\qquad\quad \vdots \\ E_n: &\quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n, \end{aligned} \tag{6.1}$$

for $x_1, \ldots, x_n$, given the constants a_{ij}, for each $i, j = 1, 2, \ldots, n$, and b_i, for each $i = 1, 2, \ldots, n$. Direct techniques are methods that give an answer in a fixed number of steps, subject only to roundoff errors. In the presentation we shall also introduce some elementary notions from the subject of linear algebra.

Methods of approximating the solution to linear systems by iterative methods are discussed in Chapter 7.

6.1 Linear Systems of Equations

We use three operations to simplify the linear system given in (6.1):

1. Equation E_i can be multiplied by any nonzero constant λ with the resulting equation used in place of E_i. This operation is denoted $(\lambda E_i) \rightarrow (E_i)$.

2. Equation E_j can be multiplied by any constant λ and added to equation E_i with the resulting equation used in place of E_i. This operation is denoted $(E_i + \lambda E_j) \rightarrow (E_i)$.

3. Equations E_i and E_j can be transposed in order. This operation is denoted $(E_i) \leftrightarrow (E_j)$.

By a sequence of these operations, a linear system can be transformed to a more easily solved linear system that has the same solutions. The sequence of operations is illustrated in the next example.

EXAMPLE 1 The four equations

$$\begin{aligned} E_1: &\quad x_1 + x_2 + 3x_4 = 4, \\ E_2: &\quad 2x_1 + x_2 - x_3 + x_4 = 1, \\ E_3: &\quad 3x_1 - x_2 - x_3 + 2x_4 = -3, \\ E_4: &\quad -x_1 + 2x_2 + 3x_3 - x_4 = 4, \end{aligned} \tag{6.2}$$

will be solved for x_1, x_2, x_3, and x_4. We first use equation E_1 to eliminate the unknown x_1 from E_2, E_3, and E_4 by performing $(E_2 - 2E_1) \rightarrow (E_2)$, $(E_3 - 3E_1) \rightarrow (E_3)$, and $(E_4 + E_1) \rightarrow (E_4)$. The resulting system is

$$\begin{aligned} E_1: \quad & x_1 + x_2 + 3x_4 = 4, \\ E_2: \quad & -x_2 - x_3 - 5x_4 = -7, \\ E_3: \quad & -4x_2 - x_3 - 7x_4 = -15, \\ E_4: \quad & 3x_2 + 3x_3 + 2x_4 = 8, \end{aligned}$$

where, for simplicity, the new equations are again labeled E_1, E_2, E_3, and E_4.

In the new system, E_2 is used to eliminate x_2 from E_3 and E_4 by performing $(E_3 - 4E_2) \rightarrow (E_3)$ and $(E_4 + 3E_2) \rightarrow (E_4)$, resulting in

$$\begin{aligned} E_1: \quad & x_1 + x_2 + 3x_4 = 4, \\ E_2: \quad & -x_2 - x_3 - 5x_4 = -7, \\ E_3: \quad & 3x_3 + 13x_4 = 13, \\ E_4: \quad & -13x_4 = -13. \end{aligned} \tag{6.3}$$

The system of equations (6.3) is now in **triangular** (or **reduced**) **form** and can be solved for the unknowns by a **backward-substitution process**. Since E_4 implies $x_4 = 1$, we can solve E_3 for x_3 to give

$$x_3 = \frac{1}{3}(13 - 13x_4) = \frac{1}{3}(13 - 13) = 0.$$

Continuing, E_2 gives

$$x_2 = -(-7 + 5x_4 + x_3) = -(-7 + 5 + 0) = 2,$$

and E_1 gives

$$x_1 = 4 - 3x_4 - x_2 = 4 - 3 - 2 = -1.$$

The solution to (6.3), and consequently to (6.2), is therefore, $x_1 = -1$, $x_2 = 2$, $x_3 = 0$, and $x_4 = 1$. ■

When performing the calculations of Example 1, we did not need to write out the full equations at each step or to carry the variables x_1, x_2, x_3, and x_4 through the calculations, since they always remained in the same column. The only variation from system to system occurred in the coefficients of the unknowns and in the values on the right side of the equations. For this reason, a linear system is often replaced by a *matrix*, which contains all the information about the system that is necessary to determine its solution, but in a compact form.

Definition 6.1 An $n \times m$ (n by m) **matrix** is a rectangular array of elements with n rows and m columns in which not only is the value of an element important, but also its position in the array. ■

The notation for an $n \times m$ matrix will be a capital letter such as A for the matrix and lowercase letters with double subscripts, such as a_{ij}, to refer to the entry at the intersection of the ith row and jth column; that is,

$$A = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}.$$

EXAMPLE 2 The matrix

$$A = \begin{bmatrix} 2 & -1 & 7 \\ 3 & 1 & 0 \end{bmatrix}$$

is a 2×3 matrix with $a_{11} = 2$, $a_{12} = -1$, $a_{13} = 7$, $a_{21} = 3$, $a_{22} = 1$, and $a_{23} = 0$. ■

The $1 \times n$ matrix

$$A = [a_{11} \ a_{12} \ \cdots \ a_{1n}]$$

is called an ***n*-dimensional row vector**, and an $n \times 1$ matrix

$$A = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}$$

is called an ***n*-dimensional column vector**. Usually the unnecessary subscripts are omitted for vectors, and a boldface lowercase letter is used for notation. Thus,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

denotes a column vector, and

$$\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_n]$$

a row vector.

An $n \times (n + 1)$ matrix can be used to represent the linear system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \vdots \qquad\qquad & \qquad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n, \end{aligned}$$

by first constructing

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

and then combining these matrices to form the **augmented matrix**

$$[A, \mathbf{b}] = \left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right],$$

where the vertical dotted line is used to separate the coefficients of the unknowns from the values on the right-hand side of the equations.

Repeating the operations involved in Example 1 with the matrix notation results in first considering the augmented matrix:

$$\left[\begin{array}{cccc:c} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{array}\right].$$

Performing the operations as described in that example produces the matrices

$$\left[\begin{array}{cccc:c} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & -4 & -1 & -7 & -15 \\ 0 & 3 & 3 & 2 & 8 \end{array}\right] \quad \text{and} \quad \left[\begin{array}{cccc:c} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & -13 & -13 \end{array}\right].$$

The final matrix can now be transformed into its corresponding linear system, and solutions for x_1, x_2, x_3, and x_4, can be obtained. The procedure involved in this process is called **Gaussian elimination with backward substitution**.

The general Gaussian elimination procedure applied to the linear system

$$\begin{aligned} E_1: &\quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ E_2: &\quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ &\qquad\qquad \vdots \qquad\qquad\qquad\quad \vdots \\ E_n: &\quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n, \end{aligned} \tag{6.4}$$

is handled in a similar manner. First form the augmented matrix $\tilde{A}$:

$$\tilde{A} = [A, \mathbf{b}] = \left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right], \tag{6.5}$$

where A denotes the matrix formed by the coefficients. The entries in the $(n+1)$st column are the values of $\mathbf{b}$; that is, $a_{i,n+1} = b_i$ for each $i = 1, 2, \ldots, n$.

Provided $a_{11} \neq 0$, the operations corresponding to $(E_j - (a_{j1}/a_{11})E_1) \rightarrow (E_j)$ are performed for each $j = 2, 3, \ldots, n$ to eliminate the coefficient of x_1 in each of these rows. Although the entries in rows $2, 3, \ldots, n$ are expected to change, for ease of notation we again denote the entry in the ith row and the jth column by a_{ij}. With this in mind, we follow a sequential procedure for $i = 2, 3, \ldots, n-1$ and perform the operation $(E_j - (a_{ji}/a_{ii})E_i) \rightarrow (E_j)$ for each $j = i+1, i+2, \ldots, n$, provided $a_{ii} \neq 0$. This eliminates (changes the coefficient to zero) x_i in each row below the ith for all values of $i = 1, 2, \ldots, n-1$. The resulting matrix has the form:

$$\tilde{\tilde{A}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & \vdots & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & \vdots & a_{2,n+1} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{nn} & \vdots & a_{n,n+1} \end{bmatrix},$$

where, except in the first row, the values of a_{ij} are not expected to agree with those in the original matrix $\tilde{A}$. The matrix $\tilde{\tilde{A}}$ represents a linear system with the same solution set as the original system (6.4). Since the new linear system is triangular,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= a_{1,n+1}, \\ a_{22}x_2 + \cdots + a_{2n}x_n &= a_{2,n+1}, \\ \ddots \quad \vdots \quad &\vdots \\ a_{nn}x_n &= a_{n,n+1}, \end{aligned}$$

backward substitution can be performed. Solving the nth equation for x_n gives

$$x_n = \frac{a_{n,n+1}}{a_{nn}}.$$

Solving the $(n-1)$st equation for x_{n-1} and using x_n yields

$$x_{n-1} = \frac{a_{n-1,n+1} - a_{n-1,n}x_n}{a_{n-1,n-1}}.$$

Continuing this process, we obtain

$$x_i = \frac{a_{i,n+1} - a_{i,n}x_n - a_{i,n-1}x_{n-1} - \cdots - a_{i,i+1}x_{i+1}}{a_{ii}} = \frac{a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}},$$

for each $i = n-1, n-2, \ldots, 2, 1$.

The Gaussian elimination procedure can be presented more precisely, although more intricately, by forming a sequence of augmented matrices $\tilde{A}^{(1)}, \tilde{A}^{(2)}, \ldots, \tilde{A}^{(n)}$, where $\tilde{A}^{(1)}$ is the matrix $\tilde{A}$ given in (6.5) and $\tilde{A}^{(k)}$, for each $k = 2, 3, \ldots, n$, has entries $a_{ij}^{(k)}$, where:

$$a_{ij}^{(k)} = \begin{cases} a_{ij}^{(k-1)}, & \text{when } i = 1, 2, \ldots, k-1 \text{ and } j = 1, 2, \ldots, n+1, \\ 0, & \text{when } i = k, k+1, \ldots, n \text{ and } j = 1, 2, \ldots, k-1, \\ a_{ij}^{(k-1)} - \dfrac{a_{i,k-1}^{(k-1)}}{a_{k-1,k-1}^{(k-1)}} a_{k-1,j}^{(k-1)}, & \text{when } i = k, k+1, \ldots, n \text{ and } j = k, k+1, \ldots, n+1. \end{cases}$$

Thus,

$$\tilde{A}^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1,k-1}^{(1)} & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2,k-1}^{(2)} & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} & a_{2,n+1}^{(2)} \\ \vdots & \ddots & & & \vdots & \vdots & & \vdots & \vdots \\ & & & & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & \cdots & a_{k-1,n}^{(k-1)} & a_{k-1,n+1}^{(k-1)} \\ & & & & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & a_{k,n+1}^{(k)} \\ & & & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & a_{n,n+1}^{(k)} \end{bmatrix} \tag{6.6}$$

represents the equivalent linear system for which the variable x_{k-1} has just been eliminated from equations $E_k, E_{k+1}, \ldots, E_n$.

The procedure will fail if one of the elements $a_{11}^{(1)}, a_{22}^{(2)}, a_{33}^{(3)}, \ldots, a_{n-1,n-1}^{(n-1)}, a_{nn}^{(n)}$ is zero because the step

$$\left(E_i - \frac{a_{i,k}^{(k)}}{a_{kk}^{(k)}} E_k\right) \rightarrow E_i$$

either cannot be performed (this occurs if one of $a_{11}^{(1)}, \ldots, a_{n-1,n-1}^{(n-1)}$ is zero), or the backward substitution cannot be accomplished (in the case $a_{nn}^{(n)} = 0$). The system may still have a solution, but the technique for finding the solution must be altered. An illustration is given in the following example.

EXAMPLE 3 Consider the linear system

$$\begin{aligned} E_1: &\quad x_1 - x_2 + 2x_3 - x_4 = -8, \\ E_2: &\quad 2x_1 - 2x_2 + 3x_3 - 3x_4 = -20, \\ E_3: &\quad x_1 + x_2 + x_3 = -2, \\ E_4: &\quad x_1 - x_2 + 4x_3 + 3x_4 = 4. \end{aligned}$$

The augmented matrix is

$$\tilde{A} = \tilde{A}^{(1)} = \left[\begin{array}{cccc:c} 1 & -1 & 2 & -1 & -8 \\ 2 & -2 & 3 & -3 & -20 \\ 1 & 1 & 1 & 0 & -2 \\ 1 & -1 & 4 & 3 & 4 \end{array}\right],$$

and performing the operations

$$(E_2 - 2E_1) \rightarrow (E_2), \quad (E_3 - E_1) \rightarrow (E_3), \quad \text{and} \quad (E_4 - E_1) \rightarrow (E_4),$$

gives

$$\tilde{A}^{(2)} = \left[\begin{array}{cccc:c} 1 & -1 & 2 & -1 & -8 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & 2 & 4 & 12 \end{array}\right].$$

Since $a_{22}^{(2)}$, called the **pivot element**, is zero, the procedure cannot continue in its present form. But the operation $(E_i) \leftrightarrow (E_j)$ is permitted, so a search is made of the elements $a_{32}^{(2)}$ and $a_{42}^{(2)}$ for the first nonzero element. Since $a_{32}^{(2)} \neq 0$, the operation $(E_2) \leftrightarrow (E_3)$ is performed to obtain a new matrix,

$$\tilde{A}^{(2)'} = \left[\begin{array}{cccc:c} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 0 & 2 & 4 & 12 \end{array}\right].$$

Since x_2 is already eliminated from E_3 and E_4, $\tilde{A}^{(3)}$ will be $\tilde{A}^{(2)'}$, and the computations continue with the operation $(E_4 + 2E_3) \rightarrow (E_4)$, giving

$$\tilde{A}^{(4)} = \left[\begin{array}{cccc:c} 1 & -1 & 2 & -1 & -8 \\ 0 & 2 & -1 & 1 & 6 \\ 0 & 0 & -1 & -1 & -4 \\ 0 & 0 & 0 & 2 & 4 \end{array}\right].$$

Finally, the backward substitution is applied:

$$\begin{aligned} x_4 &= \frac{4}{2} = 2, \\ x_3 &= \frac{[-4 - (-1)x_4]}{-1} = 2, \\ x_2 &= \frac{[6 - x_4 - (-1)x_3]}{2} = 3, \\ x_1 &= \frac{[-8 - (-1)x_4 - 2x_3 - (-1)x_2]}{1} = -7. \end{aligned}$$

■

Example 2 illustrates what is done if $a_{kk}^{(k)} = 0$ for some $k = 1, 2, \ldots, n-1$. The kth column of $\tilde{A}^{(k-1)}$ from the kth row to the nth row is searched for the first nonzero entry. If $a_{pk}^{(k)} \neq 0$ for some p,with $k+1 \leq p \leq n$, then the operation $(E_k) \leftrightarrow (E_p)$ is performed to obtain $\tilde{A}^{(k-1)'}$. The procedure can then be continued to form $\tilde{A}^{(k)}$, and so on. If $a_{pk}^{(k)} = 0$ for each p, it can be shown (see Theorem 6.16 in Section 6.4) that the linear system does not have a unique solution and the procedure stops. Finally, if $a_{nn}^{(n)} = 0$, the linear system does not have a unique solution, and again the procedure stops. Algorithm 6.1 summarizes Gaussian elimination with backward substitution. The algorithm incorporates pivoting when one of the pivots $a_{kk}^{(k)}$ is 0 by interchanging the kth row with the pth row, where p is the smallest integer greater than k for which a $a_{pk}^{(k)}$ is nonzero.

ALGORITHM 6.1

Gaussian Elimination with Backward Substitution

To solve the $n \times n$ linear system

$$\begin{aligned} E_1 &: \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_{1,n+1} \\ E_2 &: \quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = a_{2,n+1} \\ &\vdots \qquad \vdots \qquad \vdots \qquad\qquad \vdots \qquad \vdots \\ E_n &: \quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = a_{n,n+1} \end{aligned}$$

INPUT number of unknowns and equations n; augmented matrix $A = (a_{ij})$, where $1 \le i \le n$ and $1 \le j \le n+1$.

OUTPUT solution $x_1, x_2, \ldots, x_n$ or message that the linear system has no unique solution.

Step 1 For $i = 1, \ldots, n-1$ do Steps 2–4. (*Elimination process.*)

Step 2 Let p be the smallest integer with $i \le p \le n$ and $a_{pi} \neq 0$.
If no integer p can be found
then OUTPUT ('no unique solution exists');
STOP.

Step 3 If $p \neq i$ then perform $(E_p) \leftrightarrow (E_i)$.

Step 4 For $j = i+1, \ldots, n$ do Steps 5 and 6.

Step 5 Set $m_{ji} = a_{ji}/a_{ii}$.

Step 6 Perform $(E_j - m_{ji}E_i) \rightarrow (E_j)$;

Step 7 If $a_{nn} = 0$ then OUTPUT ('no unique solution exists');
STOP.

Step 8 Set $x_n = a_{n,n+1}/a_{nn}$. (*Start backward substitution.*)

Step 9 For $i = n-1, \ldots, 1$ set $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right] \Big/ a_{ii}$.

Step 10 OUTPUT $(x_1, \ldots, x_n)$; (*Procedure completed successfully.*)
STOP. ■

All CAS have matrix routines. To define matrices and perform Gaussian elimination using Maple, you must first access the linear algebra library using the command

```
>with(linalg);
```

To define the matrix $\tilde{A}^{(1)}$ of Example 2, which we will call AA, use the command

```
>AA:=matrix(4,5,[1,-1,2,-1,-8,2,-2,3,-3,-20,1,1,1,0,-2,1,-1,4,3,4]);
```

The first two parameters, 4 and 5, give the number of rows and columns, respectively, and the last parameter is a list of the entries of $\tilde{A}^{(1)} \equiv AA$. The function `addrow(AA,i,j,m)`

performs the operation $(E_j + mE_i) \rightarrow (E_j)$ and the function `swaprow(AA,i,j)` performs the operation $(E_i) \leftrightarrow (E_j)$. So, the sequence of operations

```
>AA:=addrow(AA,1,2,-2);
>AA:=addrow(AA,1,3,-1);
>AA:=addrow(AA,1,4,-1);
>AA:=swaprow(AA,2,3);
>AA:=addrow(AA,3,4,2);
```

gives the reduction to $\tilde{A}^{(4)}$, which is again called AA. Alternatively, the single command `AA:=gausselim(AA);` returns the reduced matrix. In either case, the final operation

```
>x:=backsub(AA);
```

produces the solution $x := [-7, 3, 2, 2]$.

EXAMPLE 4 The purpose of this example is to show what can happen if Algorithm 6.1 fails. The computations will be done simultaneously on two linear systems:

$$\begin{aligned} x_1 + x_2 + x_3 &= 4, \\ 2x_1 + 2x_2 + x_3 &= 6, \\ x_1 + x_2 + 2x_3 &= 6, \end{aligned} \quad \text{and} \quad \begin{aligned} x_1 + x_2 + x_3 &= 4, \\ 2x_1 + 2x_2 + x_3 &= 4, \\ x_1 + x_2 + 2x_3 &= 6. \end{aligned}$$

These systems produce matrices

$$\tilde{A} = \left[\begin{array}{ccc:c} 1 & 1 & 1 & 4 \\ 2 & 2 & 1 & 6 \\ 1 & 1 & 2 & 6 \end{array}\right] \quad \text{and} \quad \tilde{A} = \left[\begin{array}{ccc:c} 1 & 1 & 1 & 4 \\ 2 & 2 & 1 & 4 \\ 1 & 1 & 2 & 6 \end{array}\right].$$

Since $a_{11} = 1$, we perform $(E_2 - 2E_1) \rightarrow (E_2)$ and $(E_3 - E_1) \rightarrow (E_3)$ to produce

$$\tilde{A} = \left[\begin{array}{ccc:c} 1 & 1 & 1 & 4 \\ 0 & 0 & -1 & -2 \\ 0 & 0 & 1 & 2 \end{array}\right] \quad \text{and} \quad \tilde{A} = \left[\begin{array}{ccc:c} 1 & 1 & 1 & 4 \\ 0 & 0 & -1 & -4 \\ 0 & 0 & 1 & 2 \end{array}\right].$$

At this point, $a_{22} = a_{32} = 0$. The algorithm requires that the procedure be halted, and no solution to either system is obtained. Writing the equations for each system gives

$$\begin{aligned} x_1 + x_2 + x_3 &= 4, \\ -x_3 &= -2, \\ x_3 &= 2, \end{aligned} \quad \text{and} \quad \begin{aligned} x_1 + x_2 + x_3 &= 4, \\ -x_3 &= -4, \\ x_3 &= 2. \end{aligned}$$

The first linear system has an infinite number of solutions; $x_3 = 2$, $x_2 = 2 - x_1$, and x_1 arbitrary. The second system leads to the contradiction $x_3 = 2$ and $x_3 = 4$, so no solution exists. In each case, however, there is no *unique* solution, as we conclude from Algorithm 6.1. ■

Although Algorithm 6.1 can be viewed as the construction of the augmented matrices $\tilde{A}^{(1)}, \ldots, \tilde{A}^{(n)}$, the computations can be performed in a computer using only one $n \times (n+1)$ array for storage. At each step we simply replace the previous value of a_{ij} by the new one.

In addition, we can store the multipliers m_{ji} in the locations of a_{ji} since a_{ji} has the value 0 for each $i = 1, 2, \ldots, n-1$ and $j = i+1, i+2, \ldots, n$. Thus, A can be overwritten by the multipliers below the main diagonal and by the nonzero entries of $\tilde{A}^{(n)}$ on and above the main diagonal. These values can be used to solve other linear systems involving the original matrix A, as we will see in Section 6.5.

Both the amount of time required to complete the calculations and the subsequent roundoff error depend on the number of floating-point arithmetic operations needed to solve a routine problem. In general, the amount of time required to perform a multiplication or division on a computer is approximately the same and is considerably greater than that required to perform an addition or subtraction. The actual differences in execution time, however, depend on the particular computing system. To demonstrate the counting operations for a given method, we will count the operations required to solve a typical linear system of n equations in n unknowns using Algorithm 6.1. We will keep the count of the additions/subtractions separate from the count of the multiplications/divisions because of the time differential.

No arithmetic operations are performed until Steps 5 and 6 in the algorithm. Step 5 requires that $(n - i)$ divisions be performed. The replacement of the equation E_j by $(E_j - m_{ji}E_i)$ in Step 6 requires that m_{ji} be multiplied by each term in E_i, resulting in a total of $(n-i)(n-i+1)$ multiplications. After this is completed, each term of the resulting equation is subtracted from the corresponding term in E_j. This requires $(n - i)(n - i + 1)$ subtractions. For each $i = 1, 2, \ldots, n-1$, the operations required in Steps 5 and 6 are as follows.

Multiplications/divisions

$$(n - i) + (n - i)(n - i + 1) = (n - i)(n - i + 2).$$

Additions/subtractions

$$(n - i)(n - i + 1).$$

The total number of operations required by these steps is obtained by summing the operation counts for each i. Recalling from calculus that

$$\sum_{j=1}^{m} 1 = m, \quad \sum_{j=1}^{m} j = \frac{m(m+1)}{2}, \quad \text{and} \quad \sum_{j=1}^{m} j^2 = \frac{m(m+1)(2m+1)}{6},$$

we have the following operation counts.

Multiplications/divisions

$$\sum_{i=1}^{n-1}(n - i)(n - i + 2) = \sum_{i=1}^{n-1}(n^2 - 2ni + i^2 + 2n - 2i)$$

$$= (n^2 + 2n)\sum_{i=1}^{n-1} 1 - 2(n+1)\sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} i^2 = \frac{2n^3 + 3n^2 - 5n}{6}.$$

Additions/subtractions

$$\sum_{i=1}^{n-1}(n-i)(n-i+1) = \sum_{i=1}^{n-1}(n^2 - 2ni + i^2 + n - i)$$

$$= (n^2+n)\sum_{i=1}^{n-1} 1 - (2n+1)\sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} i^2 = \frac{n^3 - n}{3}.$$

The only other steps in Algorithm 6.1 that involve arithmetic operations are those required for backward substitution, Steps 8 and 9. Step 8 requires one division. Step 9 requires $(n - i)$ multiplications and $(n - i - 1)$ additions for each summation term and then one subtraction and one division. The total number of operations in Steps 8 and 9 is as follows.

Multiplications/divisions

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

Additions/subtractions

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

The total number of arithmetic operations in Algorithm 6.1 is, therefore:

Multiplications/divisions

$$\frac{2n^3+3n^2-5n}{6} + \frac{n^2+n}{2} = \frac{n^3}{3} + n^2 - \frac{n}{3}.$$

Additions/subtractions

$$\frac{n^3-n}{3} + \frac{n^2-n}{2} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}.$$

For large n, the total number of multiplications and divisions is approximately $n^3/3$, as is the total number of additions and subtractions. Thus, the amount of computation and the time required increases with n in proportion to n^3, as shown in Table 6.1.

Table 6.1

n	Multiplications/Divisions	Additions/Subtractions
3	17	11
10	430	375
50	44,150	42,875
100	343,300	338,250

EXERCISE SET 6.1

1. For each of the following linear systems, obtain a solution by graphical methods, if possible. Explain the results from a geometrical standpoint.

a. $x_1 + 2x_2 = 3,$
$x_1 - x_2 = 0.$

b. $x_1 + 2x_2 = 0,$
$x_1 - x_2 = 0.$

c. $x_1 + 2x_2 = 3,$
$2x_1 + 4x_2 = 6.$

d. $x_1 + 2x_2 = 3,$
$-2x_1 - 4x_2 = 6.$

e. $x_1 + 2x_2 = 0,$
$2x_1 + 4x_2 = 0.$

f. $2x_1 + x_2 = -1,$
$x_1 + x_2 = 2,$
$x_1 - 3x_2 = 5.$

g. $2x_1 + x_2 = -1,$
$4x_1 + 2x_2 = -2,$
$x_1 - 3x_2 = 5.$

h. $2x_1 + x_2 + x_3 = 1,$
$2x_1 + 4x_2 - x_3 = -1.$

2. Use Gaussian elimination with backward substitution and two-digit rounding arithmetic to solve the following linear systems. Do not reorder the equations. (The exact solution to each system is $x_1 = 1, x_2 = -1, x_3 = 3.$)

a. $4x_1 - x_2 + x_3 = 8,$
$2x_1 + 5x_2 + 2x_3 = 3,$
$x_1 + 2x_2 + 4x_3 = 11.$

b. $4x_1 + x_2 + 2x_3 = 9,$
$2x_1 + 4x_2 - x_3 = -5,$
$x_1 + x_2 - 3x_3 = -9.$

3. Use the Gaussian Elimination Algorithm to solve the following linear systems, if possible, and determine whether row interchanges are necessary:

a. $x_1 - x_2 + 3x_3 = 2,$
$3x_1 - 3x_2 + x_3 = -1,$
$x_1 + x_2 = 3.$

b. $2x_1 - 1.5x_2 + 3x_3 = 1,$
$-x_1 + 2x_3 = 3,$
$4x_1 - 4.5x_2 + 5x_3 = 1.$

c. $2x_1 = 3,$
$x_1 + 1.5x_2 = 4.5,$
$-3x_2 + 0.5x_3 = -6.6,$
$2x_1 - 2x_2 + x_3 + x_4 = 0.8.$

d. $x_1 - \frac{1}{2}x_2 + x_3 = 4,$
$2x_1 - x_2 - x_3 + x_4 = 5,$
$x_1 + x_2 = 2,$
$x_1 - \frac{1}{2}x_2 + x_3 + x_4 = 5.$

e. $x_1 + x_2 + x_4 = 2,$
$2x_1 + x_2 - x_3 + x_4 = 1,$
$4x_1 - x_2 - 2x_3 + 2x_4 = 0,$
$3x_1 - x_2 - x_3 + 2x_4 = -3.$

f. $x_1 + x_2 + x_4 = 2,$
$2x_1 + x_2 - x_3 + x_4 = 1,$
$-x_1 + 2x_2 + 3x_3 - x_4 = 4,$
$3x_1 - x_2 - x_3 + 2x_4 = -3.$

4. Use the Gaussian Elimination Algorithm and single-precision arithmetic on a computer to solve the following linear systems.

a. $\begin{aligned} \tfrac{1}{4}x_1 + \tfrac{1}{5}x_2 + \tfrac{1}{6}x_3 &= 9, \\ \tfrac{1}{3}x_1 + \tfrac{1}{4}x_2 + \tfrac{1}{5}x_3 &= 8, \\ \tfrac{1}{2}x_1 + x_2 + 2x_3 &= 8. \end{aligned}$

b. $\begin{aligned} 3.333x_1 + 15920x_2 - 10.333x_3 &= 15913, \\ 2.222x_1 + 16.71x_2 + 9.612x_3 &= 28.544, \\ 1.5611x_1 + 5.1791x_2 + 1.6852x_3 &= 8.4254. \end{aligned}$

c. $\begin{aligned} x_1 + \tfrac{1}{2}x_2 + \tfrac{1}{3}x_3 + \tfrac{1}{4}x_4 &= \tfrac{1}{6}, \\ \tfrac{1}{2}x_1 + \tfrac{1}{3}x_2 + \tfrac{1}{4}x_3 + \tfrac{1}{5}x_4 &= \tfrac{1}{7}, \\ \tfrac{1}{3}x_1 + \tfrac{1}{4}x_2 + \tfrac{1}{5}x_3 + \tfrac{1}{6}x_4 &= \tfrac{1}{8}, \\ \tfrac{1}{4}x_1 + \tfrac{1}{5}x_2 + \tfrac{1}{6}x_3 + \tfrac{1}{7}x_4 &= \tfrac{1}{9}. \end{aligned}$

d. $\begin{aligned} 2x_1 + x_2 - x_3 + x_4 - 3x_5 &= 7, \\ x_1 + 2x_3 - x_4 + x_5 &= 2, \\ -2x_2 - x_3 + x_4 - x_5 &= -5, \\ 3x_1 + x_2 - 4x_3 + 5x_5 &= 6, \\ x_1 - x_2 - x_3 - x_4 + x_5 &= 3. \end{aligned}$

5. Given the linear system

$$\begin{aligned} 2x_1 - 6\alpha x_2 &= 3, \\ 3\alpha x_1 - x_2 &= \tfrac{3}{2}. \end{aligned}$$

a. Find value(s) of α for which the system has no solutions.

b. Find value(s) of α for which the system has an infinite number of solutions.

c. Assuming a unique solution exists for a given α, find the solution.

6. Given the linear system

$$\begin{aligned} x_1 - x_2 + \alpha x_3 &= -2, \\ -x_1 + 2x_2 - \alpha x_3 &= 3, \\ \alpha x_1 + x_2 + x_3 &= 2. \end{aligned}$$

a. Find value(s) of α for which the system has no solutions.

b. Find value(s) of α for which the system has an infinite number of solutions.

c. Assuming a unique solution exists for a given α, find the solution.

7. Show that the operations

a. $(\lambda E_i) \to (E_i)$ **b.** $(E_i + \lambda E_j) \to (E_i)$ **c.** $(E_i) \leftrightarrow (E_j)$

do not change the solution set of a linear system.

8. **Gauss-Jordan Method:** This method is described as follows. Use the ith equation to eliminate not only x_i from the equations $E_{i+1}, E_{i+2}, \ldots, E_n$, as was done in the Gaussian elimination method, but also from $E_1, E_2, \ldots, E_{i-1}$. Upon reducing $[A, \mathbf{b}]$ to:

$$\begin{bmatrix} a_{11}^{(1)} & 0 & \cdots & 0 & \vdots & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & \ddots & \vdots & \vdots & a_{2,n+1}^{(2)} \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ 0 & \cdots & 0 & a_{nn}^{(n)} & \vdots & a_{n,n+1}^{(n)} \end{bmatrix},$$

the solution is obtained by setting

$$x_i = \frac{a_{i,n+1}^{(i)}}{a_{ii}^{(i)}},$$

for each $i = 1, 2, \ldots, n$. This procedure circumvents the backward substitution in the Gaussian elimination. Construct an algorithm for the Gauss-Jordan procedure patterned after that of Algorithm 6.1.

9. Use the Gauss-Jordan method and two-digit rounding arithmetic to solve the systems in Exercise 2.

10. Repeat Exercise 4 using the Gauss-Jordan method.

11. **a.** Show that the Gauss-Jordan method requires

$$\frac{n^3}{2} + n^2 - \frac{n}{2} \quad \text{multiplications/divisions}$$

and

$$\frac{n^3}{2} - \frac{n}{2} \quad \text{additions/subtractions.}$$

b. Make a table comparing the required operations for the Gauss-Jordan and Gaussian elimination methods for $n = 3, 10, 50, 100$. Which method requires less computation?

12. Consider the following Gaussian-elimination-Gauss-Jordan hybrid method for solving the system (6.4). First, apply the Gaussian-elimination technique to reduce the system to triangular form. Then use the nth equation to eliminate the coefficients of x_n in each of the first $n - 1$ rows. After this is completed use the $(n - 1)$st equation to eliminate the coefficients of x_{n-1} in the first $n-2$ rows, etc. The system will eventually appear as the reduced system in Exercise 8.

a. Show that this method requires

$$\frac{n^3}{3} + \frac{3}{2}n^2 - \frac{5}{6}n \quad \text{multiplications/divisions}$$

and

$$\frac{n^3}{3} + \frac{n^2}{2} - \frac{5}{6}n \quad \text{additions/subtractions.}$$

b. Make a table comparing the required operations for the Gaussian elimination, Gauss-Jordan, and hybrid methods, for $n = 3, 10, 50, 100$.

13. Use the hybrid method described in Exercise 12 and two-digit rounding arithmetic to solve the systems in Exercise 2.

14. Repeat Exercise 4 using the method described in Exercise 12.

15. Suppose that in a biological system there are n species of animals and m sources of food. Let x_j represent the population of the jth species, for each $j = 1, \ldots, n$; b_i represent the available daily supply of the ith food; and a_{ij} represent the amount of the ith food consumed on the average by a member of the jth species. The linear system

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1,\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2,\\
\vdots \qquad\quad \vdots \qquad\qquad \vdots \quad &\quad \vdots\\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m
\end{aligned}$$

represents an equilibrium where there is a daily supply of food to precisely meet the average daily consumption of each species.

a. Let

$$A = (a_{ij}) = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

$\mathbf{x} = (x_j) = [1000, 500, 350, 400]$, and $\mathbf{b} = (b_i) = [3500, 2700, 900]$. Is there sufficient food to satisfy the average daily consumption?

b. What is the maximum number of animals of each species that could be individually added to the system with the supply of food still meeting the consumption?

c. If species 1 became extinct, how much of an individual increase of each of the remaining species could be supported?

d. If species 2 became extinct, how much of an individual increase of each of the remaining species could be supported?

16. A Fredholm integral equation of the second kind is an equation of the form

$$u(x) = f(x) + \int_a^b K(x, t)u(t)\, dt,$$

where a and b and the functions f and K are given. To approximate the function u on the interval $[a, b]$, a partition $x_0 = a < x_1 < \cdots < x_{m-1} < x_m = b$ is selected and the equations

$$u(x_i) = f(x_i) + \int_a^b K(x_i, t)u(t)\, dt, \quad \text{for each } i = 0, \ldots, m,$$

are solved for $u(x_0), u(x_1), \ldots, u(x_m)$. The integrals are approximated using quadrature formulas based on the nodes $x_0, \ldots, x_m$. In our problem, $a = 0$, $b = 1$, $f(x) = x^2$, and $K(x, t) = e^{|x-t|}$.

a. Show that the linear system

$$u(0) = f(0) + \frac{1}{2}[K(0, 0)u(0) + K(0, 1)u(1)],$$

$$u(1) = f(1) + \frac{1}{2}[K(1, 0)u(0) + K(1, 1)u(1)]$$

must be solved when the Trapezoidal rule is used.

b. Set up and solve the linear system that results when the Composite Trapezoidal rule is used with $n = 4$.

c. Repeat part (b) using the Composite Simpson's rule.

6.2 Pivoting Strategies

In deriving Algorithm 6.1, we found that a row interchange is needed when one of the pivot elements $a_{kk}^{(k)}$ is 0. This row interchange has the form $(E_k) \leftrightarrow (E_p)$, where p is the smallest integer greater than k with $a_{pk}^{(k)} \neq 0$. To reduce roundoff error, it is often necessary to perform row interchanges even when the pivot elements are not zero.

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{jk}^{(k)}$, the magnitude of the multiplier

$$m_{jk} = \frac{a_{jk}^{(k)}}{a_{kk}^{(k)}}$$

will be much larger than 1. Roundoff error introduced in the computation of one of the terms $a_{kl}^{(k)}$ is multiplied by m_{jk} when computing $a_{jl}^{(k+1)}$, which compounds the original error. Also, when performing the backward substitution for

$$x_k = \frac{a_{k,n+1}^{(k)} - \sum_{j=k+1}^{n} a_{kj}^{(k)}}{a_{kk}^{(k)}},$$

with a small value of $a_{kk}^{(k)}$, any error in the numerator can be dramatically increased because of the division by $a_{kk}^{(k)}$. An illustration of this difficulty is given in the following example.

EXAMPLE 1 The linear system

$$\begin{aligned} E_1: \quad & 0.003000x_1 + 59.14x_2 = 59.17 \\ E_2: \quad & 5.291x_1 - 6.130x_2 = 46.78, \end{aligned}$$

has the exact solution $x_1 = 10.00$ and $x_2 = 1.000$. Suppose Gaussian elimination is performed on this system using four-digit arithmetic with rounding.

The first pivot element, $a_{11}^{(1)} = 0.003000$, is small, and its associated multiplier,

$$m_{21} = \frac{5.291}{0.003000} = 1763.6\overline{6},$$

rounds to the large number 1764. Performing $(E_2 - m_{21}E_1) \rightarrow (E_2)$ and the appropriate rounding gives

$$\begin{aligned} 0.003000x_1 + 59.14x_2 &\approx 59.17 \\ -104300x_2 &\approx -104400, \end{aligned}$$

instead of the precise values,

$$\begin{aligned} 0.003000x_1 + 59.14x_2 &= 59.17 \\ -104309.37\overline{6}x_2 &= -104309.37\overline{6}. \end{aligned}$$

The disparity in the magnitudes of $m_{21}a_{13}$ and a_{23} has introduced roundoff error, but the roundoff error has not yet been propagated. Backward substitution yields

$$x_2 \approx 1.001,$$

which is a close approximation to the actual value, $x_2 = 1.000$. However, because of the small pivot $a_{11} = 0.003000$,

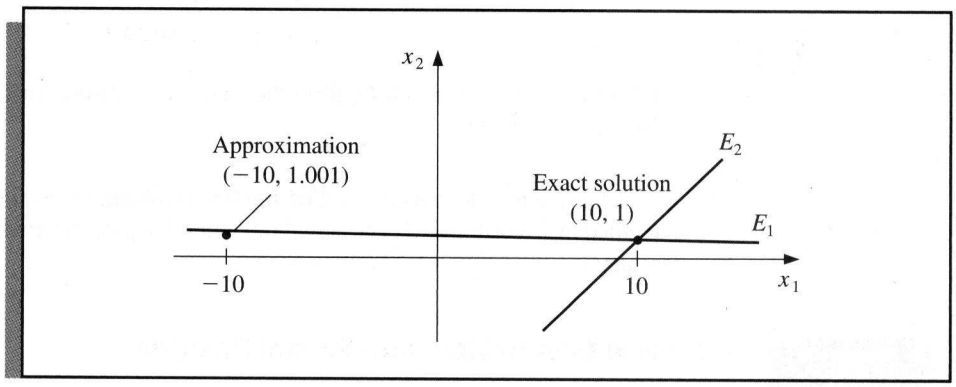
$$x_1 \approx \frac{59.17 - (59.14)(1.001)}{0.003000} = -10.00$$

contains the small error of 0.001 multiplied by

$$\frac{59.14}{0.003000} \approx 20000.$$

This ruins the approximation to the actual value $x_1 = 10.00$. (See Figure 6.1.) ■

Figure 6.1



Example 1 shows how difficulties arise when the pivot element $a_{kk}^{(k)}$ is small relative to the entries $a_{ij}^{(k)}$, for $k \leq i \leq n$ and $k \leq j \leq n$. To avoid this problem, pivoting is performed by selecting a larger element $a_{pq}^{(k)}$ for the pivot and interchanging the kth and pth rows, followed by the interchange of the kth and qth columns, if necessary. The simplest strategy is to select an element in the same column that is below the diagonal and has the largest absolute value; specifically, we determine the smallest $p \geq k$ such that

$$|a_{pk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$$

and perform $(E_k) \leftrightarrow (E_p)$. In this case no interchange of columns is used.

EXAMPLE 2 Reconsider the system

$$\begin{aligned} E_1: &\quad 0.003000x_1 + 59.14x_2 = 59.17, \\ E_2: &\quad 5.291x_1 - 6.130x_2 = 46.78. \end{aligned}$$

The pivoting procedure just described results in first finding

$$\max\left\{|a_{11}^{(1)}|, |a_{21}^{(1)}|\right\} = \max\{|0.003000|, |5.291|\} = |5.291| = |a_{21}^{(1)}|.$$

The operation $(E_2) \leftrightarrow (E_1)$ is then performed to give the system

$$\begin{aligned} E_1: &\quad 5.291x_1 - 6.130x_2 = 46.78, \\ E_2: &\quad 0.003000x_1 + 59.14x_2 = 59.17. \end{aligned}$$

The multiplier for this system is

$$m_{21} = \frac{a_{21}^{(1)}}{a_{11}^{(1)}} = 0.0005670,$$

and the operation $(E_2 - m_{21}E_1) \rightarrow (E_2)$ reduces the system to

$$5.291x_1 - 6.130x_2 \approx 46.78,$$
$$59.14x_2 \approx 59.14.$$

The four-digit answers resulting from the backward substitution are the correct values $x_1 = 10.00$ and $x_2 = 1.000$. ■

The technique just described is called **partial pivoting**, or *maximal column pivoting*, and is detailed in Algorithm 6.2. The actual row interchanging is simulated in the algorithm by interchanging the values of *NROW* in Step 5.

ALGORITHM **6.2**

Gaussian Elimination with Partial Pivoting

To solve the $n \times n$ linear system

$$\begin{aligned} E_1: &\quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_{1,n+1} \\ E_2: &\quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = a_{2,n+1} \\ &\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ E_n: &\quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = a_{n,n+1} \end{aligned}$$

INPUT number of unknowns and equations n; augmented matrix $A = (a_{ij})$ where $1 \leq i \leq n$ and $1 \leq j \leq n + 1$.

OUTPUT solution $x_1, \ldots, x_n$ or message that the linear system has no unique solution.

Step 1 For $i = 1, \ldots, n$ set $NROW(i) = i$. (*Initialize row pointer.*)

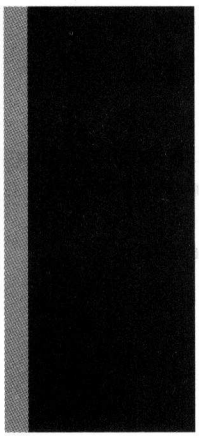
Step 2 For $i = 1, \ldots, n - 1$ do Steps 3–6. (*Elimination process.*)

Step 3 Let p be the smallest integer with $i \leq p \leq n$ and
$|a(NROW(p), i)| = \max_{i \leq j \leq n} |a(NROW(j), i)|$.
(*Notation:* $a(NROW(i), j) \equiv a_{NROW_i, j}$.)

Step 4 If $a(NROW(p), i) = 0$ then OUTPUT ('no unique solution exists');
STOP.

Step 5 If $NROW(i) \neq NROW(p)$ then set $NCOPY = NROW(i)$;
$NROW(i) = NROW(p)$;
$NROW(p) = NCOPY$.
(*Simulated row interchange.*)

Step 6 For $j = i + 1, \ldots, n$ do Steps 7 and 8.



Step 7 Set $m(NROW(j), i) = a(NROW(j), i)/a(NROW(i), i)$.

Step 8 Perform $(E_{NROW(j)} - m(NROW(j), i) \cdot E_{NROW(i)}) \rightarrow (E_{NROW(j)})$.

Step 9 If $a(NROW(n), n) = 0$ then OUTPUT ('no unique solution exists');
STOP.

Step 10 Set $x_n = a(NROW(n), n + 1)/a(NROW(n), n)$.
(*Start backward substitution.*)

Step 11 For $i = n - 1, \ldots, 1$

$$\text{set } x_i = \frac{a(NROW(i), n + 1) - \sum_{j=i+1}^{n} a(NROW(i), j) \cdot x_j}{a(NROW(i), i)}.$$

Step 12 OUTPUT $(x_1, \ldots, x_n)$; (*Procedure completed successfully.*)
STOP. ■

Each multiplier m_{ji} in the partial pivoting algorithm has magnitude less than or equal to 1. Although this strategy is sufficient for most linear systems, situations do arise when it is inadequate.

EXAMPLE 3 The linear system

$$\begin{aligned} E_1: \quad & 30.00x_1 + 591400x_2 = 591700, \\ E_2: \quad & 5.291x_1 - \quad 6.130x_2 = 46.78, \end{aligned}$$

is the same as that in Examples 1 and 2 except that all the entries in the first equation have been multiplied by 10^4. The procedure described in Algorithm 6.2 with four-digit arithmetic leads to the same results as obtained in Example 1. The maximal value in the first column is 30.00, and the multiplier

$$m_{21} = \frac{5.291}{30.00} = 0.1764$$

leads to the system

$$\begin{aligned} 30.00x_1 + 591400x_2 &\approx 591700, \\ -104300x_2 &\approx 104400, \end{aligned}$$

which has the same inaccurate solutions as in Example 1: $x_2 \approx 1.001$ and $x_1 \approx -10.00$. ■

Scaled partial pivoting, also called *scaled-column pivoting*, is appropriate for the system in Example 3. It places the element in the pivot position that is largest relative to the entries in its row. The first step in this procedure is to define a scale factor s_i for each row as

$$s_i = \max_{1 \le j \le n} |a_{ij}|.$$

If, for some i, we have $s_i = 0$, then the system has no unique solution since all entries in the ith row are 0. Assuming that this is not the case, the appropriate row interchange to

place zeros in the first column is determined by choosing the least integer p with

$$\frac{|a_{p1}|}{s_p} = \max_{1 \le k \le n} \frac{|a_{k1}|}{s_k}$$

and performing $(E_1) \leftrightarrow (E_p)$. The effect of scaling is to ensure that the largest element in each row has a *relative* magnitude of 1 before the comparison for row interchange is performed.

In a similar manner, before eliminating the variable x_i using the operations

$$E_k - m_{ki}E_i, \quad \text{for } k = i+1, \ldots, n,$$

we select the smallest integer $p \ge i$ with

$$\frac{|a_{pi}|}{s_p} = \max_{i \le k \le n} \frac{|a_{ki}|}{s_k}$$

and perform the row interchange $E_i \leftrightarrow E_p$ if $i \ne p$. The scale factors $s_1, \ldots, s_n$ are computed only once, at the start of the procedure, and must also be interchanged when row interchanges are performed.

Applying scaled partial pivoting to Example 3 gives

$$s_1 = \max\{|30.00|, |591400|\} = 591400$$

and

$$s_2 = \max\{|5.291|, |-6.130|\} = 6.130.$$

Consequently,

$$\frac{|a_{11}|}{s_1} = \frac{30.00}{591400} = 0.5073 \times 10^{-4}, \qquad \frac{|a_{21}|}{s_2} = \frac{5.291}{6.130} = 0.8631,$$

and the interchange $(E_1) \leftrightarrow (E_2)$ is made.

Applying Gaussian elimination to the new system

$$5.291x_1 - 6.130x_2 = 46.78$$

$$30.00x_1 + 591400x_2 = 591700$$

produces the correct results: $x_1 = 10.00$ and $x_2 = 1.000$.

Algorithm 6.3 implements scaled partial pivoting.

ALGORITHM 6.3

Gaussian Elimination with Scaled Partial Pivoting

The only steps in this algorithm that differ from those of Algorithm 6.2 are:

Step 1 For $i = 1, \ldots, n$ set $s_i = \max_{1 \le j \le n} |a_{ij}|$;
if $s_i = 0$ then OUTPUT ('no unique solution exists');
STOP.
set $NROW(i) = i$.

Step 2 For $i = 1, \ldots, n-1$ do Steps 3–6. (*Elimination process.*)

Step 3 Let p be the smallest integer with $i \le p \le n$ and

$$\frac{|a(NROW(p), i)|}{s(NROW(p))} = \max_{i \le j \le n} \frac{|a(NROW(j), i)|}{s(NROW(j))}.$$

■

The next example illustrates scaled partial pivoting using Maple with finite-digit rounding arithmetic.

EXAMPLE 4 Solve the linear system using three-digit rounding arithmetic.

$$\begin{aligned} 2.11x_1 - \ 4.21x_2 + 0.921x_3 &= \ \ 2.01, \\ 4.01x_1 + \ 10.2x_2 - \ 1.12x_3 &= -3.09, \\ 1.09x_1 + 0.987x_2 + 0.832x_3 &= \ \ 4.21. \end{aligned}$$

To obtain three-digit rounding arithmetic, enter

```
>Digits:=3;
```

We have $s_1 = 4.21$, $s_2 = 10.2$, and $s_3 = 1.09$.

So

$$\frac{|a_{11}|}{s_1} = \frac{2.11}{4.21} = 0.501, \quad \frac{|a_{21}|}{s_1} = \frac{4.01}{10.2} = 0.393, \quad \text{and} \quad \frac{|a_{31}|}{s_3} = \frac{1.09}{1.09} = 1.$$

The augmented matrix AA is defined by

```
>AA:=matrix(3,4,[2.11,-4.21,0.921,2.01,4.01,10.2,-1.12,-3.09,1.09,
                 0.987,0.832,4.21]);
```

which gives

$$AA := \begin{bmatrix} 2.11 & -4.21 & .921 & 2.01 \\ 4.01 & 10.2 & -1.12 & -3.09 \\ 1.09 & .987 & .832 & 4.21 \end{bmatrix}.$$

Since $|a_{31}|/s_3$ is largest, we perform $(E_1) \leftrightarrow (E_3)$ using

```
>AA:=swaprow(AA,1,3);
```

to obtain

$$AA := \begin{bmatrix} 1.09 & .987 & .832 & 4.21 \\ 4.01 & 10.2 & -1.12 & -3.09 \\ 2.11 & -4.21 & .921 & 2.01 \end{bmatrix}.$$

Computing multipliers gives

```
>m21:=4.01/1.09;
```

$$m21 := 3.68$$

```
>m31:=2.11/1.09;
```

$$m31 := 1.94$$

We perform the first two eliminations using

```
>AA:=addrow(AA,1,2,-m21);
```

and

```
>AA:=addrow(AA,1,3,-m31);
```

to obtain

$$AA := \begin{bmatrix} 1.09 & .987 & .832 & 4.21 \\ 0 & 6.57 & -4.18 & -18.6 \\ 0 & -6.12 & -.689 & -6.16 \end{bmatrix}.$$

Since

$$\frac{|a_{22}|}{s_2} = \frac{6.57}{10.2} = 0.644 < \frac{|a_{32}|}{s_3} = \frac{6.12}{4.21} = 1.45,$$

we perform

```
>AA:=swaprow(AA,2,3);
```

giving

$$AA := \begin{bmatrix} 1.09 & .987 & .832 & 4.21 \\ 0 & -6.12 & -.689 & -6.16 \\ 0 & 6.57 & -4.18 & -18.6 \end{bmatrix}.$$

The multiplier m_{32} is computed by

```
>m32:=6.57/(-6.12);
```

$$m_{32} := -1.07.$$

The elimination step

```
>AA:=addrow(AA,2,3,-m32);
```

gives

$$AA := \begin{bmatrix} 1.09 & .987 & .832 & 4.21 \\ 0 & -6.12 & -.689 & -6.16 \\ 0 & .02 & -4.92 & -25.2 \end{bmatrix}.$$

We cannot use `backsub` because of the entry .02 in the (3, 2) position. This entry is nonzero due to rounding, but we can remedy this minor problem using the command

```
>AA[3,2]:=0;
```

which replaces the entry .02 with a 0. To see this enter

```
>evalm(AA);
```

which displays the matrix AA. Finally,

```
>x:=backsub(AA);
```

gives the solution

$$x := [-.431 \quad .430 \quad 5.12]. \qquad \blacksquare$$

The first additional computations required for scaled partial pivoting result from the determination of the scale factors; there are $(n-1)$ comparisons for each of the n rows, for a total of

$$n(n-1) \text{ comparisons.}$$

To determine the correct first interchange, n divisions are performed, followed by $n-1$ comparisons. So the first interchange determination adds

$$n \text{ divisions and } (n-1) \text{ comparisons.}$$

Since the scaling factors are computed only once, the second step requires

$$(n-1) \text{ divisions and } (n-2) \text{ comparisons.}$$

We proceed in a similar manner until there are zeros below the main diagonal in all but the nth row. The final step requires that we perform

$$2 \text{ divisions and } 1 \text{ comparison.}$$

As a consequence, scaled partial pivoting adds a total of

$$n(n-1) + \sum_{k=1}^{n-1} k = n(n-1) + \frac{(n-1)n}{2} = \frac{3}{2}n(n-1) \quad \text{comparisons} \tag{6.7}$$

and

$$\sum_{k=2}^{n} k = \sum_{k=1}^{n} k - 1 = \frac{n(n+1)}{2} - 1 \quad \text{divisions}$$

to the Gaussian elimination procedure. The time required to perform a comparison is about the same as an addition/subtraction. Since the total time to perform the basic Gaussian elimination procedure is $O(n^3/3)$ multiplications/divisions and $O(n^3/3)$ additions/subtractions, scaled partial pivoting does not add significantly to the computational time required to solve a system for large values of n.

To emphasize the importance of choosing the scale factors only once, consider the amount of additional computation that would be required if the procedure were modified

so that new scale factors were determined each time a row interchange decision was to be made. In this case, the term $n(n-1)$ in Eq. (6.7) would be replaced by

$$\sum_{k=2}^{n} k(k-1) = \frac{1}{3}n(n^2-1).$$

As a consequence, this pivoting technique would add $O(n^3/3)$ comparisons, in addition to the $[n(n+1)/2]-1$ divisions. If a system warrants this type of pivoting, **complete** (or *maximal*) **pivoting** should instead be used. Complete pivoting at the kth step searches all the entries a_{ij}, for $i = k, k+1, \ldots, n$ and $j = k, k+1, \ldots, n$, to find the entry with the largest magnitude. Both row and column interchanges are performed to bring this entry to the pivot position. The first step of total pivoting requires that n^2-1 comparisons be performed, the second step requires $(n-1)^2-1$ comparisons, and so on. Hence the total additional time required to incorporate complete pivoting into Gaussian elimination is

$$\sum_{k=2}^{n}(k^2-1) = \frac{n(n-1)(2n+5)}{6}$$

comparisons. This figure is comparable to the number required for the modified scaled-column pivoting technique, but no divisions are required. Complete pivoting is, consequently, the strategy recommended for systems where accuracy is essential and the amount of execution time needed for this method can be justified.

EXERCISE SET 6.2

1. Find the row interchanges that are required to solve the following linear systems using Algorithm 6.1.

a.
$$\begin{aligned} x_1 - 5x_2 + x_3 &= 7,\\ 10x_1 + 20x_3 &= 6,\\ 5x_1 - x_3 &= 4. \end{aligned}$$

b.
$$\begin{aligned} x_1 + x_2 - x_3 &= 1,\\ x_1 + x_2 + 4x_3 &= 2,\\ 2x_1 - x_2 + 2x_3 &= 3. \end{aligned}$$

c.
$$\begin{aligned} 2x_1 - 3x_2 + 2x_3 &= 5,\\ -4x_1 + 2x_2 - 6x_3 &= 14,\\ 2x_1 + 2x_2 + 4x_3 &= 8. \end{aligned}$$

d.
$$\begin{aligned} x_2 + x_3 &= 6,\\ x_1 - 2x_2 - x_3 &= 4,\\ x_1 - x_2 + x_3 &= 5. \end{aligned}$$

2. Repeat Exercise 1 using Algorithm 6.2.

3. Repeat Exercise 1 using Algorithm 6.3.

4. Repeat Exercise 1 using complete pivoting.

5. Use Gaussian eliminiation and three-digit chopping arithmetic to solve the following linear systems, and compare the approximations to the actual solution.

a.
$$\begin{aligned} 0.03x_1 + 58.9x_2 &= 59.2,\\ 5.31x_1 - 6.10x_2 &= 47.0. \end{aligned}$$
Actual solution $(10, 1)^t$.

b.
$$\begin{aligned} 58.9x_1 + 0.03x_2 &= 59.2,\\ -6.10x_1 + 5.31x_2 &= 47.0. \end{aligned}$$
Actual solution $(1, 10)^t$.

c. $3.03x_1 - 12.1x_2 + 14x_3 = -119,$
$-3.03x_1 + 12.1x_2 - 7x_3 = 120,$
$6.11x_1 - 14.2x_2 + 21x_3 = -139.$
Actual solution $(0, 10, \frac{1}{7})^t$.

d. $3.3330x_1 + 15920x_2 + 10.333x_3 = 7953,$
$2.2220x_1 + 16.710x_2 + 9.6120x_3 = 0.965,$
$-1.5611x_1 + 5.1792x_2 - 1.6855x_3 = 2.714.$
Actual solution $(1, 0.5, -1)^t$.

e. $1.19x_1 + 2.11x_2 - 100x_3 + x_4 = 1.12,$
$14.2x_1 - 0.122x_2 + 12.2x_3 - x_4 = 3.44,$
$100x_2 - 99.9x_3 + x_4 = 2.15,$
$15.3x_1 + 0.110x_2 - 13.1x_3 - x_4 = 4.16.$
Actual solution $(0.17682530, 0.01269269, -0.02065405, -1.18260870)^t$.

f. $\pi x_1 - ex_2 + \sqrt{2}x_3 - \sqrt{3}x_4 = \sqrt{11},$
$\pi^2 x_1 + ex_2 - e^2 x_3 + \frac{3}{7}x_4 = 0,$
$\sqrt{5}x_1 - \sqrt{6}x_2 + x_3 - \sqrt{2}x_4 = \pi,$
$\pi^3 x_1 + e^2 x_2 - \sqrt{7}x_3 + \frac{1}{9}x_4 = \sqrt{2}.$
Actual solution $(0.78839378, -3.12541367, 0.16759660, 4.55700252)^t$.

6. Repeat Exercise 5 using three-digit rounding arithmetic.

7. Repeat Exercise 5 using Gaussian elimination with partial pivoting.

8. Repeat Exercise 6 using Gaussian elimination with partial pivoting.

9. Repeat Exercise 5 using Gaussian elimination with scaled partial pivoting.

10. Repeat Exercise 6 using Gaussian elimination with scaled partial pivoting.

11. Repeat Exercise 5 using Algorithm 6.1 with single-precision computer arithmetic.

12. Repeat Exercise 5 using Algorithm 6.2 with single-precision computer arithmetic.

13. Repeat Exercise 5 using Algorithm 6.3 with single-precision computer arithmetic.

14. Construct an algorithm for the complete pivoting procedure discussed in the text.

15. Use the complete pivoting algorithm developed in Exercise 14 to obtain solutions to

a. Exercise 5 **b.** Exercise 6 **c.** Exercise 11

16. Suppose that

$$2x_1 + x_2 + 3x_3 = 1,$$
$$4x_1 + 6x_2 + 8x_3 = 5,$$
$$6x_1 + \alpha x_2 + 10x_3 = 5,$$

with $|\alpha| < 10$. For which of the following values of α will there be no row interchange required when solving this system using scaled partial pivoting?

a. $\alpha = 6$ **b.** $\alpha = 9$ **c.** $\alpha = -3$