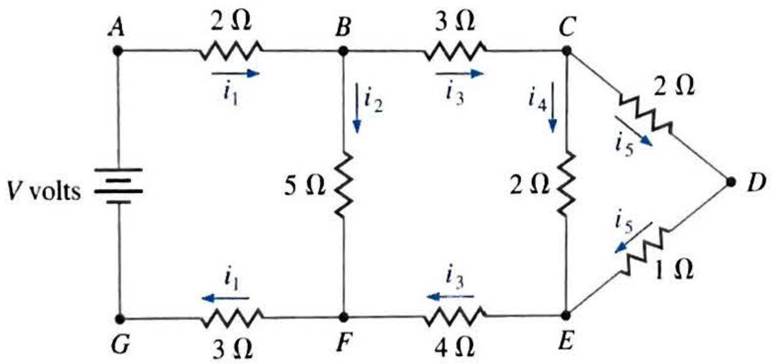
Direct Methods for Solving Linear Systems

Introduction

Kirchhoff's laws of electrical circuits state that both the net flow of current through each junction and the net voltage drop around each closed loop of a circuit are zero. Suppose that a potential of V volts is applied between the points A and G in the circuit and that i_1, i_2, i_3, i_4, and i_5 represent current flow as shown in the diagram. Using G as a reference point, Kirchhoff's laws imply that the currents satisfy the following system of linear equations:

$$\begin{aligned}
5i_1 + 5i_2 &= V,\\
i_3 - i_4 - i_5 &= 0,\\
2i_4 - 3i_5 &= 0,\\
i_1 - i_2 - i_3 &= 0,\\
5i_2 - 7i_3 - 2i_4 &= 0.
\end{aligned}$$



The solution of systems of this type will be considered in this chapter. This application is discussed in Exercise 23 of Section 6.6.

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, we consider *direct methods* for solving a linear system of n equations in n variables. Such a system has the form

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

In this system, we are given the constants a_{ij}, for each $i, j = 1, 2, \ldots, n$, and b_i, for each $i = 1, 2, \ldots, n$, and we need to determine the unknowns $x_1, \ldots, x_n$.

Direct techniques are methods that theoretically give the exact solution to the system in a finite number of steps. In practice, of course, the solution obtained will be contaminated by the round-off error that is involved with the arithmetic being used. Analyzing the effect of this round-off error and determining ways to keep it under control will be a major component of this chapter.

A course in linear algebra is not assumed to be prerequisite for this chapter, so we will include a number of the basic notions of the subject. These results will also be used in Chapter 7, where we consider methods of approximating the solution to linear systems using iterative methods.

6.1 Linear Systems of Equations

We use three operations to simplify the linear system given in Eq. (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 will be systematically transformed into to a new linear system that is more easily solved and has the same solutions (See Exercise 13). The sequence of operations is illustrated in the following.

Illustration The four equations

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

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 equations E_2, E_3, and E_4 by performing $(E_2 - 2E_1) \rightarrow (E_2)$, $(E_3 - 3E_1) \rightarrow (E_3)$, and $(E_4 + E_1) \rightarrow (E_4)$. For example, in the second equation,

$$(E_2 - 2E_1) \rightarrow (E_2)$$

produces

$$(2x_1 + x_2 - x_3 + x_4) - 2(x_1 + x_2 + 3x_4) = 1 - 2(4),$$

which simplifies to the result shown as E_2 in

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

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 the unknown x_2 from E_3 and E_4 by performing $(E_3 - 4E_2) \rightarrow (E_3)$ and $(E_4 + 3E_2) \rightarrow (E_4)$. This results 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 system (6.3) and, consequently, to system (6.2) is, therefore, $x_1 = -1$, $x_2 = 2$, $x_3 = 0$, and $x_4 = 1$. ■

Matrices and Vectors

When performing the calculations in the illustration, we would 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, if they always remained in the same column. The only variation from system to system occurs 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 and one that is easily represented in a computer.

Definition 6.1 An $\boldsymbol{n \times m}$ (**$\boldsymbol{n}$ by $\boldsymbol{m}$**) **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 1 Determine the size and respective entries of the matrix

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

Solution The matrix has two rows and three columns, so it is of size 2×3. Its entries are described by $a_{11} = 2$, $a_{12} = -1$, $a_{13} = 7$, $a_{21} = 3$, $a_{22} = 1$, 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. In addition, row vectors often have commas inserted between the entries to make the separation clearer. So, you might see $\mathbf{y}$ written as $\mathbf{y} = [y_1, \; y_2, \; \ldots, \; y_n]$.

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

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

by first constructing

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

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

Augmented refers to the fact that the right-hand side of the system has been included in the matrix.

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. The array $[A, \mathbf{b}]$ is called an **augmented matrix**.

Repeating the operations involved in the illustration on page 362 with the matrix notation results in first considering the augmented matrix:

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

Performing the operations as described in that example produces the augmented matrices

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

A technique similar to Gaussian elimination first appeared during the Han dynasty in China in the text Nine Chapters on the Mathematical Art, which was written about 200 B.C.E. Joseph Louis Lagrange (1736–1813) described a technique similar to this procedure in 1778 for the case when the value of each equation is 0. Gauss gave a more general description in *Theoria Motus corporum coelestium sectionibus solem ambientium*, which described the least squares technique he used in 1801 to determine the orbit of the minor planet Ceres.

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 is called **Gaussian elimination with backward substitution.**

The general Gaussian elimination procedure applied to the linear system

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

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 $a_{11} \neq 0$, we perform the operations corresponding to

$$(E_j - (a_{j1}/a_{11})E_1) \rightarrow (E_j) \quad \text{for each } j = 2, 3, \ldots, n$$

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

$$(E_j - (a_{ji}/a_{ii})E_i) \rightarrow (E_j) \quad \text{for each } j = i+1, i+2, \ldots, n,$$

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

$$\tilde{\tilde{A}} = \left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & a_{nn} & a_{n,n+1} \end{array}\right],$$

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.

The new linear system is triangular,

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

so *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 the known value for 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

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

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

Gaussian elimination procedure is described 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 Eq. (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,

$$\tilde{A}^{(k)} = \left[\begin{array}{cccccccc:c} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1,k-1}^{(1)} & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2,k-1}^{(2)} & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} & a_{2,n+1}^{(2)} \\ \vdots & \ddots & & & \vdots & \vdots & & \vdots & \vdots \\ & & & & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & \cdots & a_{k-1,n}^{(k-1)} & a_{k-1,n+1}^{(k-1)} \\ \vdots & & & & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} & a_{k,n+1}^{(k)} \\ \vdots & & & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} & a_{n,n+1}^{(k)} \end{array}\right] \quad (6.6)$$

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 either the step

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

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 2 Represent the linear system

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

as an augmented matrix and use Gaussian elimination to find its solution.

Solution The augmented matrix is

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

Performing the operations

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

gives

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

The pivot element for a specific column is the entry that is used to place zeros in the other entries in that column.

The diagonal entry $a_{22}^{(2)}$, called the **pivot element**, is 0, so the procedure cannot continue in its present form. But operations $(E_i) \leftrightarrow (E_j)$ are 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 $(E_4 + 2E_3) \to (E_4)$, 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 matrix is converted back into a linear system that has a solution equivalent to the solution of the original system, and backward substitution is applied:

$$x_4 = \frac{4}{2} = 2,$$

$$x_3 = \frac{[-4 - (-1)x_4]}{-1} = 2,$$

$$x_2 = \frac{[6 - [(-1)x_3 + x_4]]}{2} = 3,$$

$$x_1 = \frac{[-8 - [(-1)x_2 + 2x_3 + (-1)x_4]]}{1} = -7.$$

■

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 6.17 on page 402) 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 $a_{pk}^{(k)} \neq 0$.

ALGORITHM 6.1

Gaussian Elimination with Backward Substitution

To solve the $n \times n$ linear system

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

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 $x_1, x_2, \ldots, x_n$ 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 $i \leq p \leq n$ 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 $(E_j - m_{ji}E_i) \to (E_j)$;

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.

Illustration The purpose of this illustration 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 the augmented 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 $(E_2 - 2E_1) \to (E_2)$ and $(E_3 - E_1) \to (E_3)$ 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, which can be described by $x_3 = 2$, $x_2 = 2 - x_1$, 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} because 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 in the entries that are below the main diagonal (that is, the entries of the form a_{ji}, with $j > i$) and by the newly computed entries of $\tilde{A}^{(n)}$ on and above the main diagonal (the entries of the form a_{ij}, with $j \leq i$). These values can be used to solve other linear systems involving the original matrix A, as we will see in Section 6.5.

Operation Counts

Both the amount of time required to complete the calculations and the subsequent round-off 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 Steps 5 and 6 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:

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

Additions/subtractions:

$$\begin{aligned}\sum_{i=1}^{n-1}(n-i)(n-i+1) &= \sum_{i=1}^{n-1}(n^2-2ni+i^2+n-i)\\ &= \sum_{i=1}^{n-1}(n-i)^2 + \sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-1} i\\ &= \frac{(n-1)n(2n-1)}{6} + \frac{(n-1)n}{2} = \frac{n^3-n}{3}.\end{aligned}$$

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:

$$\begin{aligned}1 + \sum_{i=1}^{n-1}((n-i)+1) &= 1 + \left(\sum_{i=1}^{n-1}(n-i)\right) + n - 1\\ &= n + \sum_{i=1}^{n-1}(n-i) = n + \sum_{i=1}^{n-1} i = \frac{n^2+n}{2}.\end{aligned}$$

Additions/subtractions:

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

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. $x_1 + 2x_2 = 3,$ $\quad x_1 - x_2 = 0.$

   b. $x_1 + 2x_2 = 3,$ $\quad 2x_1 + 4x_2 = 6.$

   c. $x_1 + 2x_2 = 0,$ $\quad 2x_1 + 4x_2 = 0.$

   d. $2x_1 + x_2 = -1,$ $\quad 4x_1 + 2x_2 = -2,$ $\quad x_1 - 3x_2 = 5.$

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

   a. $x_1 + 2x_2 = 0,$ $\quad x_1 - x_2 = 0.$

   b. $x_1 + 2x_2 = 3,$ $\quad -2x_1 - 4x_2 = 6.$

   c. $2x_1 + x_2 = -1,$ $\quad x_1 + x_2 = 2,$ $\quad x_1 - 3x_2 = 5.$

   d. $2x_1 + x_2 + x_3 = 1,$ $\quad 2x_1 + 4x_2 - x_3 = -1.$