A Sampling of Linear Models

Section 4.1 Linear Transformations in Computer Graphics

In this chapter we discuss some linear models in greater detail. Three of these models were introduced in Chapter 1, Markov chains, linear programming, and population growth. Using matrix algebra and solution techniques learned in the previous chapters, we shall be able to analyze and solve these linear models.

A general solution for most of the models in this chapter requires one to solve a system of linear equations; in matrix form, solve $Ax = b$. In solving some of these systems of linear equations, various theoretical difficulties will arise. Those problems will motivate the theory of solutions to systems of linear equations, which is discussed in Chapter 5.

A common use of linear models is to predict the values of a set of variables in the future as a linear function of the variables’ current values. A Markov chain is such a model, as was the rabbit–fox population model. Other models of this type are presented in this chapter. These models assume the matrix form

$$w' = Aw \quad (1)$$

or more generally (in future examples)
where \( w \) is the vector of current values and \( w' \) the vector of future values. In this section we give a geometric interpretation of (1) and (2).

A **linear transformation** \( T \) of the plane maps each point \( w = (x, y) \) into a point \( w' = T(w) \), where \( T(w) = Aw \), for some 2-by-2 matrix \( A \). A linear transformation in \( n \) dimensions is defined the same way (\( A \) is then an \( n \)-by-\( n \) matrix). When linear transformations are programmed on a computer, they can be used to move figures about and create the special visual effects we have come to associate with computer graphics. In this section the reader will learn how to build up complicated graphics effects out of simple transformations.

It will sometimes be convenient in this section to drop matrix notation and write \( w' = T(w) = Aw \) to represent the pair of linear equations

\[
\begin{align*}
x' &= ax + by \\
y' &= cx + dy
\end{align*}
\]  

A slightly more general transformation, corresponding to \( w' = Aw + b \), is an **affine linear transformation**.

\[
\begin{align*}
x' &= ax + by + e \\
y' &= cx + dy + f
\end{align*}
\]

Clearly, all linear transformations are affine linear transformations (with \( e = f = 0 \)). Shortly, we will see that affine linear transformations lack some very important properties that linear transformations have.

Figure 4.1b, c, and d show the effect of transformations \( T_1 \), \( T_2 \), and \( T_3 \), respectively, on the square in Figure 4.1a whose corners are \( A = (0, 0) \), \( B = (1, 0) \), \( C = (1, 1) \), \( D = (0, 1) \).

\[
\begin{align*}
T_1: & \quad x' = 2x + 4 \\
& \quad y' = 3y + 2 \\
T_2: & \quad x' = \cos 45^\circ x - \sin 45^\circ y = .707x - .707y \\
& \quad y' = \sin 45^\circ x + \cos 45^\circ y = .707x + .707y \\
T_3: & \quad x' = x + y \\
& \quad y' = y
\end{align*}
\]  

Transformation \( T_1 \) doubles the width and triples the height of the square and also moves it 4 units to the right and 2 units up. Transformation \( T_2 \) has the effect of revolving the square \( 45^\circ \) counterclockwise about the origin, but does not change the square's size. Transformation \( T_3 \) slants the y-axis and lines parallel to it by \( 45^\circ \). To help understand the effect of these transfor-
mations, readers should evaluate $T_1$, $T_2$, and $T_3$ at point $C = \{1, 1\}$ of the square in Figure 4.1a. Exercise 31 shows that revolving a point about the origin always has the form (6).

There is an important computational question to ask about transforming a square or any figure built out of line segments. Is it sufficient to compute just the new coordinates of the corners, and then connect these new corners with straight lines to obtain the full transformed figure? Fortunately, the answer is yes. This result is a simple consequence of two basic laws of matrix algebra. We state these laws in linear transformation form as a theorem.
**Theorem 1.** Let $T$ be a linear transformation with $w' = T(w)$ and $v' = T(v)$. Then for any scalar constants $r$ and $s$,

(i) $T(w + v) = T(w) + T(v) = w' + v'$

(ii) $T(rw) = rT(w) = rw'$

(iii) $T(rw + sv) = rT(w) + sT(v) = rw' + sv'$

When Theorem 1, parts (i) and (ii) are rewritten in matrix form with $T(w) = Aw$, they become the familiar matrix laws $A(w + v) = Aw + Av$ and $A(rw) = r(Aw)$. Theorem 1, part (iii) is just a combination of parts (i) and (ii). Theorem 1 is not true for affine linear transformations $T(w) = Aw + b$—parts (ii) and (iii) fail (see Exercise 33 for counterexamples). Theorem 1, part (iii) generalizes to linear combinations of three or more points.

If $w$ and $v$ are the two endpoints of a line segment $L$, any point $t$ on $L$ can be written as a linear combination of $w$ and $v$ of the form

$$t = rw + (1 - r)v, \quad \text{for some} \ r, \ 0 \leq r \leq 1 \quad (8)$$

The constant $r$ is the fraction of the distance $t$ is from $w$ to $v$. For example, if $t$ were halfway between $w$ and $v$, then $r = .5$. When $w$ and $v$ are mapped by some linear transformation $T$ to points $w'$ and $v'$, the line segment between them will all points of the form

$$t' = rw' + (1 - r)v', \quad \text{for some} \ r, \ 0 \leq r \leq 1 \quad (9)$$

By Theorem 1, part (iii), we see that if $t$ is as in (8), then $t' = T(t)$ is the expression in (9). So linear transformations map lines into lines. This result is also true for affine linear transformations (it is easily verified using matrix algebra (see Exercise 32)).

**Theorem 2.** An affine linear transformation $T$ maps line segments into line segments.

Theorem 2 allows us to compute transformations of straight-line figures simply by transforming corners of figures and then drawing lines between the transformed corners.

In computer graphics applications, Theorem 1 says that if the coordinates of corners, or other critical points, in a figure can be expressed as linear combinations of the coordinates of some “key” points, then to transform the figure we only need to apply the linear transformation to the coordinates of these key points; the coordinates of other points can quickly be obtained from the coordinates of the transformed key points.

Theorem 1 restated the basic fact of matrix algebra that vector addition and scalar multiplication are preserved by matrix-vector multiplication. Another almost-as-easy consequence from matrix algebra is the following result.
Theorem 3. If $T_1$ and $T_2$ are two affine linear transformations, the composite transformation $w'' = T_2(T_1(w))$, obtained by mapping $w$ to $w' = T_1(w)$ and then mapping $w'$ to $w'' = T_2(w')$, is also an affine linear transformation.

Proof. Let $T_1(w)$ be the mapping $w' = A_1w + b_1$ and $T_2(w)$ be the mapping $w' = A_2w + b_2$. Then using matrix algebra, $T_2(T_1(w))$ can be written

$$T_2(T_1(w)) = A_2(A_1w + b_1) + b_2$$

$$= A_2A_1w + A_2b_1 + b_2$$

(10)

Then $T_2(T_1(w))$ is the affine linear transformation $T_3(w) = A_3w + b_3$, where

$$A_3 = A_2A_1 \quad \text{and} \quad b_3 = A_2b_1 + b_2$$

(11)

This theorem was easy to prove using matrix algebra. Without it, we would have to substitute one system of equations for affine linear transformation $T_1$ into another system of equations for affine linear transformation $T_2$—a giant mess!

Theorem 3 lets us build up complicated transformations out of simpler transformations that revolve, expand distances along the $x$- or $y$-axis, move left (right) or up (down), slant axes, and other changes. Another use is in creating animated motion. If the rotation $T_2$ [in equation (6)] used an angle of $1^\circ$, we would obtain a linear transformation $T'_2$ that would revolve, say, a square around the origin by $1^\circ$. If we repeatedly applied $T'_2$ 360 times, we would obtain an ‘‘animated’’ sequence of figures that create one full revolution of the square around the origin.

Example 1. Transforming a Set of Squares

Draw a figure $F$ consisting of a set of eight unit squares whose corners are at points $(g, h)$, $g = 1, 2, 3$ and $h = -2, -1, 0, 1, 2$ (see Figure 4.2a). Observe that the coordinates of all corner points are linear combinations of the coordinates of the lower-left corner $(1, -2)$ and the “change in coordinates” points $(1, 0)$ and $(0, 1)$. For example, the point $(2, 2) = (1, -2) + 1(1, 0) + 4(0, 1)$. Now let us transform $F$ first by applying the linear transformation $T_2(T_3(x, y))$ ($T_3$ followed by $T_2$), and second by applying the linear transformation $T_3(T_2(x, y))$. The transformed figures are drawn in Figure 4.2b and c. For convenience, we restate $T_2$ and $T_3$ here:
We successively apply $T_2$ and $T_3$ in both orders to the key points $(1, -2), (1, 0)$, and $(0, 1)$.

We use Theorem 1, part (iii) to obtain the other corners as linear combinations of the transformed key points. For example, since the point $(2, 2) = (1, -2) + 1(1, 0) + 4(0, 1)$, then

$$T_2(T_3(2, 2)) = T_2T_3(1, -2) + 1T_2T_3(1, 0) + 4T_2T_3(0, 1))$$
$$= (.707, -2.121) + 1(.707, .707) + 4(0, 1.414)$$
$$= (1.414, 4.242)$$
Note that changing the order of the transformations changes the composite transformation. Composing linear transformations is not commutative! This is because matrix multiplication is not commutative.

There are many geometric properties of figures that we might hope affine linear transformations would preserve: an angle at a corner, the area of a square, and the distance of each point from the origin. Each of the properties is satisfied by some but not all of $T_1, T_2, T_3$, defined above. With a little thought, the reader should be able to guess conditions on affine linear transformations which make each of these properties true.

Another interesting property is reversibility—if $u' = T(u)$, does there exist another transformation $T^{-1}$ such that $u = T^{-1}(u')$? That is, does $T$ have an inverse? $T^{-1}$ should exist if the matrix used to define $T$ has an inverse; details are left as an exercise.

Next let us consider transformations to represent three-dimensional figures in two dimensions. Any time one draws a three-dimensional figure on a piece of paper or displays it on a computer screen, one is performing such a transformation. A transformation $(x', y') = T(x, y, z)$ that maps a three-dimensional figure into two dimensions is really operating in three dimensions, but the $z$-coordinate always becomes zero [i.e., $(x', y', 0) = T(x, y, z)$].

The simplest type of linear transformation from three to two dimensions is a projection onto the $x$-$y$ plane (or onto the $x$-$z$ or $y$-$z$ planes). This has the form $T(x, y, z) = (x, y)$—just delete the $z$-coordinate. A more general projection would project three-dimensional space onto some plane that is not parallel to any pair of coordinate axes. Figure 4.3 illustrates such a projection of two dimensions onto one dimension. Here we have projected the $x$-$y$ plane onto the (one-dimensional) line $x = y$ using the transformation

![Figure 4.3 Projecting x-y plane onto line y = x.](image-url)
The arrows in Figure 4.3 show how sample points are projected onto this line. When three-dimensional points are projected onto a plane, the situation is similar to Figure 4.3, but in three dimensions it is harder to illustrate with a figure. In Chapter 5 we will learn more about projection mappings. (We note as an aside that there is no way to reverse a projection, that is, there is no affine linear transformation that maps a line onto the whole plane, since by Theorem 2 lines are always mapped onto lines; similarly, a plane cannot be mapped linearly onto all of three-dimensional space.)

The following two examples illustrate a standard way to map three dimensions into two, as well as a way to map two dimensions into three.

### Example 2. Projection of a Cube into the Plane

Devise a linear transformation that projects the three-dimensional unit cube shown in Figure 4.4a into the (x, y)-plane so that the cube looks just the way it is drawn on the (two-dimensional) page of this book in Figure 4.4a. In Figure 4.4a the z-axis is represented as a line at a 30° angle to the x-axis. Moreover, distances along the z-axis in Figure 4.4a are drawn with half the length of distances along the x- or y-axis. So the projection we want acts on a point (x, y, z) as follows: The z-coordinate should alter the x, y coordinates in the direction of a 30° angle above the x-axis and the distance of the displacement should be half the value of the z-coordinate.
\[ T' : \begin{align*}
    x' &= x + \frac{1}{2} \cos 30^\circ z = x + 0.433z \\
y' &= y + \frac{1}{2} \sin 30^\circ z = y + 0.25z
\end{align*} \tag{13} \]

See Figure 4.4b.

**Example 3. Revolve a Letter Around the x-Axis**

Devise a set of linear transformations that take the letter L (for Linear) in Figure 4.5a and revolve it 5° around the x-axis, then 10°, then 15°, and so on, around the x-axis, to make an animated movie of the L revolving around the x-axis. The revolution around the x-axis takes place in three dimensions. Let us agree to represent the transformed L in two dimensions using the projection \( T' \) given by (13). That is, we treat the L as a figure in three dimensions, then successively revolve it 5° around the x-axis and display the results at each stage in two dimensions using \( T' \).

**Figure 4.5.** (a) Letter L. (b) Letter L revolved 50° about x-axis and projected back onto x-y plane. (c) Letter L revolved 50° about x-axis, then shrunk by \( T_{10} \) and projected onto x-y plane. (d) Letter L revolved 150° about x-axis, then shrunk by \( T_{30} \) and projected onto x-y plane.
The corners of $L$ are $(1, 3), (1, 0), (2, 0)$, which in three dimensions become $(1, 3, 0), (1, 0, 0), (2, 0, 0)$. Revolving $L$ around the $x$-axis is similar to revolving a figure in the $(x, y)$-plane around the origin; this is what $T_2$ did, at the beginning of this section. We keep the $x$-coordinate fixed and revolve the $y$- and $z$-coordinates the way $T_2$ did. Thus the required linear transformation $T$ in three dimensions is

\[
T: \begin{align*}
x' &= x \\
y' &= \cos 5^\circ y - \sin 5^\circ z \\
z' &= \sin 5^\circ y + \cos 5^\circ z 
\end{align*}
\] (14)

First we apply $T$ to the corners of $L$, $(1, 3, 0), (1, 0, 0), (2, 0, 0)$, and then apply $T$ again to the transformed corners and continue applying $T$. Each time we apply $T$, we display $L$ in two dimensions by using the projection $T'$. The original $L$ and the result after applying $T$ 10 times, a $50^\circ$ revolution (and then applying $T'$), are shown in Figure 4.5a and b. Note that $T$ leaves corners $(1, 0, 0)$ and $(2, 0, 0)$ unchanged.

The result $w^*$ of applying $T$ 10 times to an initial 3-vector $u$ could also be computed by multiplying $u$ by the tenth power of the matrix of coefficients in (14). Of course, the smart way to obtain a $50^\circ$ rotation is just to substitute $\cos 50^\circ$ and $\sin 50^\circ$ in (14).

There are several variations on the transformation in Example 3 that are used in computer graphics. In practice, we would probably want to transform a whole set of letters that spell out a word. We might shrink or magnify the letters by an amount $r$ as they revolve; we use the transformation $(x'', y'', z'') = (rx', ry', rz')$ composed with $T$. We might want the letters to recede back into the distance, away from the $(x, y)$-plane. This can be accomplished by increasing the $z$-coordinate a little more each time and shrinking the letters a little (in all coordinates). The following affine linear transformation could be used after $k$ applications of $T$.

\[
T_k: \quad (x'', y'', z'') = (.95^k x', .95^k y', .95^k z' + .1 k) 
\] (15)

Note that $T_k$ has the undesirable affect of shrinking the $x$-coordinate toward the left (toward the origin). Figure 4.5c shows the result of applying $T$ 10 times followed by $T_{10}$; Figure 4.5d shows the result of applying $T$ 30 times followed by $T_{30}$ (again $T'$ is used to get planar depictions).

Let us add a warning about roundoff errors. A small error in computing (14) (to revolve the letter $L$) may become a noticeable error after dozens of iterations of (14). One easy way to eliminate this type of error is on every tenth iteration to compute the new coordinates of the transformed $L$ directly from the original coordinates, $(1, 3), (1, 0), (2, 0)$, by performing a $50^\circ$ rotation [as noted above, do this by using $\cos 50^\circ$ and $\sin 50^\circ$ in (14)]. This method of eliminating an accumulation of errors by "updating" is used frequently in many different types of linear models.
Hopefully, Examples 1, 2, and 3 have given the reader a sense of how to generate a variety of transformations to move figures around in two and three dimensions. We have tried to provide the reader with the basic tools. We leave the creation of more interesting graphics transformations as projects for the readers (some simple graphics projects are suggested in the Exercises). Please note how extremely tedious animated graphics would be without computer programs to map sets of points repeatedly, as in Example 3, and to build a complex transformation from a sequence of simple transformations, as permitted by Theorem 3.

There is one very important problem in computer graphics that we have not discussed—the hidden surface problem. In Figure 4.4 we show all the corners and edges of the cube, but actually some are not visible because they lie at the back of the cube. The problem of determining which corners, edges, and surfaces of an object are visible is tricky and its solution relies heavily on linear algebra; even determinants are involved.

In earlier chapters we saw how eigenvectors can simplify the computation in iterating the system \( \mathbf{x}' = \mathbf{A}\mathbf{x} \). We now give a geometric interpretation of how eigenvectors simplify a linear transformation \( \mathbf{T}(\mathbf{w}) = \mathbf{A}\mathbf{w} \). Recall that an eigenvector \( \mathbf{u} \) of a matrix \( \mathbf{A} \) has the property that multiplying \( \mathbf{u} \) by \( \mathbf{A} \) has the effect of multiplying \( \mathbf{u} \) by a scalar. That is, there is some scalar \( \lambda \), called an eigenvalue of \( \mathbf{A} \), such that \( \mathbf{Au} = \lambda\mathbf{u} \). Since matrix \( \mathbf{A} \) and the linear transformation \( \mathbf{T}(\mathbf{w}) = \mathbf{A}\mathbf{w} \) are really "the same thing," we will speak interchangeably about a vector \( \mathbf{u} \) being an eigenvector of \( \mathbf{A} \) or of \( \mathbf{T} \).

Eigenvectors allow us to break a linear transformation into simple parts. (This reverses our previous goal of building up complicated transformations from simple ones.) The idea is to change to a coordinate system based on the eigenvectors of \( \mathbf{A} \), as was done in Example 8 of Section 2.5, and apply \( \mathbf{T} \) in terms of this new coordinate system (in Section 3.3 we showed that converting to eigenvector coordinates was equivalent to writing \( \mathbf{A} \) in the form \( \mathbf{UD}_u\mathbf{U}^{-1} \)).

---

**Example 4. Eigenvector Coordinates to Simplify a Linear Transformation**

Consider the linear transformation \( \mathbf{T} \),

\[
\begin{align*}
\mathbf{x}' &= \mathbf{x} + \mathbf{y} \\
\mathbf{y}' &= \frac{1}{2}\mathbf{x} + \frac{3}{2}\mathbf{y}
\end{align*}
\]

(16)

The standard \((x, y)\) coordinate system expresses a point as a linear combination of the point \((1, 0)\)—the distance along the \(x\)-axis—and the point \((0, 1)\)—the distance along the \(y\)-axis. That is, \((x, y) = x(1, 0) + y(0, 1)\). Now let us use a coordinate system in which points are expressed as a linear combination of two eigenvectors of \( \mathbf{T} \) (these are magically provided by the author). They are \( \mathbf{u}_1 = (1, 1) \) and \( \mathbf{u}_2 = (-2, 1) \). Applying \( \mathbf{T} \) to \( \mathbf{u}_1 \) and \( \mathbf{u}_2 \), we have
We want to give an arbitrary point \((x, y)\) new coordinates \((r, s)\) such that

\[
(x, y) = r(1, 1) + s(-2, 1) \tag{17}
\]

Since \(T\) multiplies vector \((1, 1)\) by 2, then by Theorem 1, \(T\) also multiplies vector \(r(1, 1)\) (for any \(r\)) by 2; similarly for multiples of \((-2, 1)\). In a coordinate system \((r, s)\) based on \((1, 1)\) and \((-2, 1)\), \(T\) becomes

\[
r' = 2r \\
s' = .5s \tag{18}
\]

It will require some work to convert a point in standard \((x, y)\) coordinates [based on points \((1, 0)\) and \((0, 1)\)] to this new coordinate system based on \((1, 1)\) and \((-2, 1)\). But if \(T\) is to be applied repeatedly, the simple form (18) of \(T\) in the new coordinates is worth the effort of conversion. Writing (17) as a system of equations for the coordinates, we have

\[
\begin{bmatrix} x \\ y \end{bmatrix} = r \begin{bmatrix} 1 \\ 1 \end{bmatrix} + s \begin{bmatrix} -2 \\ 1 \end{bmatrix} \tag{19}
\]

or

\[
w = Et, \quad \text{where} \quad E = \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix}, \quad t = \begin{bmatrix} r \\ s \end{bmatrix}
\]

Here \(E\) is the matrix whose columns are the eigenvectors.

Solving (19) for \(r\) and \(s\), we obtain

\[
t = E^{-1}w: \quad r = \frac{1}{3}x + \frac{2}{3}y \tag{20} \\
s = -\frac{1}{3}x + \frac{2}{3}y
\]

Let us consider the effect of repeatedly applying \(T\) to the point \((1, 0)\) [given in \((x, y)\) coordinates]. Using (20), we convert \((1, 0)\) to \((\frac{1}{3}, -\frac{1}{3})\) in \((r, s)\) coordinates. Now we repeatedly apply \(T\) to \((\frac{1}{3}, -\frac{1}{3})\) using (18). We get the sequence of points [in \((r, s)\) coordinates]

\[
(\frac{1}{3}, -\frac{1}{3}), (\frac{2}{3}, -\frac{1}{3}), (\frac{1}{3}, -\frac{2}{3}), (\frac{2}{3}, -\frac{2}{3}), \ldots
\]

Each application of \(T\) doubles the first coordinate and halves the second coordinate (see Figure 4.6). So after \(k\) applications of \(T\) we get

\[
(r_k, s_k) = \frac{1}{3}(2^k, -(\frac{1}{2})^k)
\]
If we want to express the coordinates of this point $(r_k, s_k)$ back in the original $(x, y)$ coordinates, we simply convert back with (19) to obtain

$$
\begin{align*}
x_k &= r_k - 2s_k = \frac{1}{3}2^k + \frac{2}{3}\left(\frac{1}{2}\right)^k \\
y_k &= r_k + s_k = \frac{1}{3}2^k - \frac{1}{3}\left(\frac{1}{2}\right)^k
\end{align*}
$$

(21)

For large $k$, $2^k$ is much much larger than $\frac{1}{3}\left(\frac{1}{2}\right)^k$, so we can neglect the latter term (the effects of computer roundoff will eventually drop the smaller term for us).

In the long term, the largest eigenvalue always dominates the effects of all other eigenvalues (as was discussed in Section 2.5). In this case we have

$$
(x_k, y_k) = \frac{1}{3}(2^k, 2^k)
$$

(22)

This simple result would have been much harder to obtain without the change of coordinates. Clearly, using eigenvector-based coordinates can make calculations with linear transformations much easier.
Section 4.1 Exercises

Summary of Exercises
Exercises 1–24 call for the construction of various affine linear transformations and plotting their effect on certain figures. Exercises 25–27 have eigenvector-coordinate computations. Exercises 28–33 involve associated theory.

1. Construct affine linear transformations to do the following to the square in Figure 4.1a.
   (a) Rotate the square 180° counterclockwise around the origin (in the plane).
   (b) Move the square 7 units to the right, 3 units up, and double its width.
   (c) Make the vertical lines of the square slant at a 45° angle (height unchanged):
   
   (d) Reflect the square about the y-axis.

2. Write out the affine linear transformations in Exercise 1 in the form \( u' = Au + b \), giving \( A \) and \( b \).

3. Compute the new coordinates of the corners of the square in Figure 4.1a for each of the transformations in Exercise 1.

4. If \( T \) is the triangle with corners at \((-1, 1), (1, 1), (1, -1)\), draw \( T \) after it is transformed by each of the transformations in Exercise 1.

5. Apply each of the transformations of Exercise 1 to the grid in Figure 4.2a and plot your answer.

6. The following exercise verifies which types of affine transformations are commutative. All transformations act on the \( x-y \) plane. Let

   - \( T_a \) double the \( x \)-coordinate: \( x' = 2x, y' = y \).
   - \( T_b \) double the \( y \)-coordinate: \( x' = x, y' = 2y \).
   - \( T_c \) reflect about the \( y \)-axis: \( x' = -x, y' = y \).
   - \( T_d \) reflect about the \( x \)-axis: \( x' = x, y' = -y \).
   - \( T_e \) shift \( x \)-value 2 units: \( x' = x + 2, y' = y \).
   - \( T_f \) shift \( y \)-value 3 units: \( x' = x, y' = y + 3 \).
   - \( T_g \) rotate 45° around the origin [\( = T_2 \) in (6)].
(a) Compute $T_a T_c$ and $T_c T_a$. Do these two transformations commute?
(b) Compute $T_b T_e$ and $T_e T_b$. Do these two transformations commute?
(c) Compute $T_b T_f$ and $T_f T_b$. Do these two transformations commute?
(d) Compute $T_b T_g$ and $T_g T_b$. Do these two transformations commute?
(e) Compute $T_3 T_d$ and $T_d T_3$. Do these two transformations commute?
(f) Compute $T_T T_r$ and $T_r T_T$. Do these two transformations commute?
(g) Compute $T_c T_g$ and $T_g T_c$. Do these two transformations commute?

7. Construct affine linear transformations to do the following to the square in Figure 4.1a and plot the square after the transformation is performed.
(a) Double the width of the square (double $x$ coordinates) and rotate it $90^\circ$ around the origin.
(b) Reflect the square about the $y$-axis and then reflect about the $x$-axis.
(c) Move the square 7 units to the right, 3 units up, and then rotate the square $180^\circ$ counterclockwise around the origin.

8. Reflect the square in Figure 4.1a about the line $y = x$. Give your transformation.

9. Reflect the square in Figure 4.1a about the line $y = 2$. Give your transformation.

10. Rotate the square in Figure 4.1a $90^\circ$ about the point $(\frac{1}{2}, 0)$ by first moving the square so that its center is at the origin, then rotate it $90^\circ$, and finally reverse the initial move. Give your transformation.

11. Rotate the grid in Figure 4.2a $90^\circ$ counterclockwise about the point $(3, 1)$.
    Hint: See Exercise 10.
    Give your transformation.

12. Rotate the square in Figure 4.1a $45^\circ$ about its center.
    Hint: See Exercise 10.
    Give your transformation.

13. By squaring the associated matrix, determine the transformation of repeating twice the transformations in Exercise 7, parts (a) and (b). Plot the square in Figure 4.1a after applying each squared transformation. Finally, describe in words the effect of the squared transformations.

14. Cube the matrix associated with the linear transformation in Exercise 1, part (a) and verify that the resulting linear transformation is the same as the original one (rotating $180^\circ$ three times is the same as rotating $180^\circ$ once).

15. Square the matrix associated with the projection linear transformation $T'$ as (13) and verify that the square equals the original matrix. Explain this result in words.
16. Devise an affine linear transformation to project any \((x, y)\) point onto the following lines.
   (a) \(y = -x\)
   (b) \(y = 2x\)
   (c) \(y = x + 2\)
   (d) \(y = 3x - 8\)

17. (a) Verify that \(x' = 2x - y, y' = 2x - y\) projects any \(x-y\) point onto the line \(y' = x'\) and that the projection is not perpendicular like (12), but rather, the line segment from \((x, y)\) to \((2x - y, 2x - y)\) has a slope of 2.
   (b) Construct a projection \(T\) that maps any \(x-y\) point \(w\) onto the line \(y = 2x\) so that the line segment from \(w\) to \(T(w)\) has a slope of 1.

18. Give the \(x-y\) coordinates of the corners of the following \(x-y-z\) figures after they are projected onto the \(x-y\) plane using projection \(T'\) in (13) and draw the figures (in the \(x-y\) plane).
   (a) A triangle with corners \((1, 1, 0), (1, 0, 1), (0, 1, 1)\).
   (b) A pyramid with base \((0, 0, 0), (2, 0, 0), (0, 0, 2), (2, 0, 2)\) and top at \((1, 2, 1)\).

19. Construct a linear transformation to do the following to the letter \(L\) in Figure 4.5a.
   (a) Revolve it 30° around the \(y\)-axis.
   (b) Revolve it 10° around the \(z\)-axis.
   (c) Revolve it 30° around the \(y\)-axis and then 30° around the \(z\)-axis.

20. Give the composite linear transformation of performing the revolution \(T\) in Example 3 followed by the projection \(T'\) in (13).

21. Revolve the grid in Figure 4.2a 30° around the \(x\)-axis and project onto the \(x-y\) plane [using \(T'\) in (13)] (plot this).

22. Revolve the square in Figure 4.1a 60° around the \(x\)-axis (in three dimensions), then shrink all coordinates to half-size and project it onto the \(x-y\) plane using \(T'\) in (13).

23. (a) Construct a linear transformation to revolve an object 30° around the line of points \((x, 1, 0)\) (the line is parallel to the \(x\)-axis with \(y = 1, z = 0\)).
   \(Hint: \) See Exercise 8.
   (b) Apply the linear transformation in part (a) to the grid in Figure 4.2a and project the result onto the \(x-y\) plane with projection \(T'\) in (13).

24. Construct a linear transformation to make an animated movie in which in each successive frame the object
   (a) Revolves 30° around the \(y\)-axis and shrinks its \(x\)- and \(y\)-coordinates by 10%.
   (b) Revolves 10° around the \(y\)-axis, shrinks all coordinates by 10%, and then moves 2 units along the \(z\)-axis.
25. Repeat Example 4 to find the approximate point after \( k \) transformations using \( T \) in (16) if the initial point is
(a) \((3, 2)\)  (b) \((-1, 1)\)  (c) \((-2, -1)\)

26. Find the eigenvalues and associated eigenvectors of the following linear transformations (use the method in Section 3.1).
(a) \(x' = 3x + y, y' = 2x + 2y\)
(b) \(T'\) in (13)
(c) \(T_2\) in (6)

*Hint:* Eigenvalues are complex.
(d) \(T_3\) in (7)  (e) \(T_1\) in (5)

27. Use your results in Exercise 26, part (a) to find the result of applying that transformation 5 times to the point \([5, 2]\) and the approximate value of applying it 20 times.

28. This exercise gives a "picture" of how, when two columns of \( A \) are almost the same, the inverse of \( A \) almost does not exist. For the following matrices \( A \), solve the system \( Ax = x \).

29. A linear transformation \( w' = T(w) = Aw \) can be reversed if \( A \) is invertible. Then the reverse transformation is \( T^*(w) = A^{-1}w \). Find the reverse transformation, if possible, for
(a) \(T_3\) in (7)  (b) \(T_2\) in (6)  (c) \(T'\) in (13)

30. An affine linear transformation \( w' = T(w) = Aw + b \) can be reversed if \( A \) is invertible.
(a) In matrix notation, what is the inverse transformation \( T^*(w) \) for \( T(u) = Au + b \) (so that \( T^*T \) is the identity transformation)?
(b) Find the inverse transformation for \( T_1 \) in (5).

31. Consider the point \( u = (1, 0) \) and suppose that we want to rotate it \( \theta^\circ \) counterclockwise around the origin. Then its new position will be distance 1 from the origin and at an angle of \( \theta^\circ \) (with respect to the \( x \)-axis). Using a similar argument for the point \( v = (0, 1) \), show that for \( u' = Au \) and \( v' = Av \) to have the right values, \( A \) must be

\[
A = \begin{bmatrix}
\cos \theta \circ & -\sin \theta \circ \\
\sin \theta \circ & \cos \theta \circ
\end{bmatrix}
\]
32. Verify that affine linear transformations map lines into lines by showing if \( u' = T(u) = Au + b \), points of the form \( w = ru + (1 - r)v' \) are mapped into points of the form \( w' = ru' + (1 - r)v' \), where \( u' = T(u) \) and \( v' = T(v) \), and \( 0 \leq r \leq 1 \).

33. Make up counterexamples to show that Theorem 1, parts (i) and (ii), are false for affine linear transformations (virtually any affine example will do).

### Section 4.2 Linear Regression

One of the fundamental problems in building linear models is estimating coefficients and other constants in the linear equations. For example, in the refinery model introduced in Section 1.2 we stated that from 1 barrel of crude oil the first refinery would produce 20 gallons of heating oil, 10 gallons of diesel oil, and 5 gallons of gasoline. These production levels would vary from one batch of crude oil to another and might also vary depending on how much crude oil was being processed each day. The numbers given are estimates, not precise values. The first important work in linear algebra grew out of an estimation problem.

In 1818 the famous mathematician Karl Friedrich Gauss was commissioned to make a geodetic survey of the kingdoms of Denmark and Hanover (geodetic surveys create very accurate maps of a portion of the earth’s spherical surface). In making estimates for the positions of different locations on a map, Gauss developed the least-squares theory of regression that we present in this section. This theory yields a system of linear equations that must be solved. To solve them, Gauss invented the algorithm we now call Gaussian elimination, presented in Section 3.2. This method is still the best way known to solve systems of linear equations. It should be noted that not only did this survey project cause Gauss to start the theories of statistics and linear algebra, but to compensate for the slightly nonspherical change of the earth, Gauss was also led to develop the theory of differential geometry! The following equation summarizes this paragraph.

\[
\text{one good application} + \text{one genius} = \text{important new mathematics} \quad (*)
\]

Let us return humbly to the problem of estimating coefficients. Recall the linear model from Example 2 of Section 1.4 for predicting \( C \), the college grade average of a Scrooge High School graduate, in terms of the student’s Scrooge High average \( S \). The proposed model was

\[
C = 1.1 \times S - 0.9 \quad (1)
\]

The constants in (1) were chosen to “fit” as closely as possible data about eight graduates. The heart of these models is the choice of the constants.

Let us restate the problem of finding a linear model such as (1) in the following standardized form:
Linear Regression Model. Given a set of points \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\), find constants \(q\) and \(r\) such that the linear relation

\[
\hat{y}_i = qx_i + r
\]

gives the best possible fit for these points.

The point \((x_i, \hat{y}_i)\) is the estimate for \((x_i, y_i)\). The name "regression," which means movement back to a less developed state, comes from the idea that our model recaptures a simple relationship between the \(x_i\) and the \(y_i\), which randomness has obscured (the variables regress to a linear relationship). A model involving several input variables is called multiple linear regression. If we try to fit the data to a more complex function, such as \(\hat{y} = q_2 x^2 + q_1 x + r\) or \(\hat{y} = e^{qx}\), the model is called nonlinear regression. We shall concentrate first on simple linear regression. Once this is well understood, we can extend our analysis to the multiple regression problem (using matrix algebra). We shall also show how some problems in nonlinear regression can be transformed into linear regression problems.

Example 1. Using the Model \(\hat{y} = qx\)

Let us consider a very simple regression problem. Suppose that we want to fit the three points \((0, 1), (2, 1),\) and \((4, 4)\) to a line of the form

\[
\hat{y} = qx
\]

(see Figure 4.7). The \(x\)-value might represent the number of semesters of college mathematics a student has taken and the \(y\)-value the student's score on some test. There are thousands of other settings that might give rise to these values. The estimate (3) would help us predict the...
y-values for other x-values, for example, predict how other students might do on the test based on the amount of mathematics they have taken.

The three points in this problem are readily seen not to lie on a common line, much less a line through the origin [any line of the form (3) passes through the origin]. So we have to find a choice of q that gives the best possible fit, that is, a line $\hat{y} = qx$ passing as close to these three points as possible.

What do “best possible fit” and “as close as possible” mean? The most common approach used in such problems is to minimize the sum of the squares of the errors. The error at a point $(x_i, \hat{y}_i)$ will be $|\hat{y}_i - y_i| = |qx_i - y_i|$, the absolute difference between the value $qx_i$ predicted by (3) and the true value $y_i$ (the absolute value is needed so that a “negative” error and a “positive” error cannot offset each other). However, absolute values are not easy to use in mathematical equations. Taking the squares of differences yields positive numbers without using absolute values. There is also a geometric reason we shall give for using squares.

For the points (0, 1), (2, 1), (4, 4), the expression $\sum (\hat{y}_i - y_i)^2$ for the sum of squares of the errors (SSE) is

$$
SSE = (0q - 1)^2 + (2q - 1)^2 + (4q - 4)^2 \\
= 1 + (4q^2 - 4q + 1) + (16q^2 - 32q + 16) \\
= 20q^2 - 36q + 18
$$

The geometric justification for using a sum of squares is based on the following interpretation of our estimation problem. Let $x$ be the vector of our x-values and $y$ be the vector of our corresponding y-values. In this case, $x = [0, 2, 4]$ and $y = [1, 1, 4]$. Further, let $\hat{y}$ be the vector of estimates for $y$. Equation (3) can now be rewritten

$$
\hat{y} = qx
$$

That is, the estimates $\hat{y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3]$ from (3) will be $q$ times the x-values [0, 2, 4].

Think of $x, y,$ and $\hat{y}$ as points in three-dimensional space, where $\hat{y}$ is a multiple of $x$. Then the obvious strategy is to pick the value of $q$ that makes $qx (= \hat{y})$ as close as possible to $y$ (see Figure 4.8). That is, we want to minimize the distance $|qx - y|_e$ (in the euclidean norm) in three-dimensional space between $qx$ and $y$. This distance between $qx = [0q, 2q, 4q]$ and $y = [1, 1, 4]$ is simply

$$
|qx - y|_e = \sqrt{(0q - 1)^2 + (2q - 1)^2 + (4q - 4)^2}
$$

Comparing (4) and (6), we see that $|qx - y|_e$ is the square root of SSE. So minimizing SSE will also minimize the distance $|qx - y|_e$. Recall that in vector notation $|a|_e^2$ equals $a \cdot a$. So

$$
SSE = |qx - y|_e^2 = (qx - y) \cdot (qx - y)
$$
The value of \( q \) that minimizes \( 20q^2 - 36q + 18 \) is found by differentiating this expression with respect to \( q \) and setting this derivative equal to 0:

\[
\frac{d\text{SSE}}{dq} = 40q - 36 = 0 \rightarrow q = 0.9
\]

(8)

The desired regression equation is thus \( \hat{y} = 0.9x \) (see Figure 4.7). Using our regression equation \( \hat{y} = 0.9x \), our estimate for (0, 1) is (0, 0), for (2, 1) is (2, 1.8), and for (4, 4) is (4, 3.6) [the bad estimate for (0, 1) arose from the fact that any line \( \hat{y} = qx \) must go through the origin]. So \( \text{SSE} = 1^2 + 0.8^2 + 0.4^2 = 1.80 \), and the distance in 3-space between our estimate vector \( \hat{y} \) and the true \( y \) is \( |\hat{y} - y| = \sqrt{\text{SSE}} = \sqrt{1.80} = 1.34 \).

Readers should pause a moment to get their geometric bearings. Example 1 started as a problem of estimating a relationship between some \( x \)- and \( y \)-values that we plotted in Figure 4.7 in \( x \)-\( y \) space. But then we considered a new geometric picture with three-dimensional vectors, formed by the \( x \)-values, the \( y \)-values, and the \( \hat{y} \) estimates. Let us present the data in a matrix:

<table>
<thead>
<tr>
<th>x-value</th>
<th>y-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>First reading</td>
<td>0</td>
</tr>
<tr>
<td>Second reading</td>
<td>2</td>
</tr>
<tr>
<td>Third reading</td>
<td>4</td>
</tr>
</tbody>
</table>
In this new setting, our objective is to find a multiple \( q \) of the first column \( x \) that would estimate the second column \( y \) as closely as possible. Geometrically, we want to find a point \( p \) on the line formed by multiplies \( x = [0, 2, 4] \)—\( p = qx \), for some \( q \)—such that \( p \) is as close to \( y = [1, 1, 4] \) as possible. Point \( p = \hat{y} \) is the projection of \( y \) onto the line from the origin through \( x \).

**Example 2. Using the Model \( \hat{y} = qx + r \)**

Let us fit the points in Example 1, \((0, 1), (2, 1), (4, 4)\), using the full linear regression model (2): \( \hat{y} = qx + r \). We can also write our regression model as

\[
\hat{y} = qx + r1: \quad \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix} = q \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix} + r \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (9)
\]

We want to find an estimate vector \( \hat{y} \) as close to the vector \( y \) of \( y \)-values as possible in 3-space. Now \( \hat{y} \) is formed from a linear combination of the vectors \( x \) and \( 1 \). (The set of all possible linear combinations of \( x \) and \( 1 \) will be a plane.)

Again we pick \( q \) and \( r \) by minimizing the sum of squares of errors,

\[
\text{SSE} = |\hat{y} - y|^2 = \Sigma (\hat{y}_i - y_i)^2 = \Sigma (qx_i + r - y_i)^2 \\
= (0q + r - 1)^2 + (2q + r - 1)^2 + (4q + r - 4)^2 \\
= (r^2 - 2r + 1) + (4q^2 + r^2 + 4qr - 4q - 2r + 1) \\
+ (16q^2 + r^2 + 8qr - 32q - 8r + 16) \\
= 20q^2 + 3r^2 + 12qr - 36q - 12r + 18 \quad (10)
\]

To minimize (10) with respect to \( q \) and \( r \), we differentiate with respect to \( q \) and \( r \) and set the partial derivatives equal to 0.

It is left as an exercise for the reader to verify that the partial derivatives of SSE in (10) with respect to \( q \) and \( r \) are

\[
\frac{\partial \text{SSE}}{\partial q} = 40q + 12r - 36 = 0 \quad \text{or} \quad 40q + 12r = 36 \quad (11)
\]

\[
\frac{\partial \text{SSE}}{\partial r} = 6r + 12q - 12 = 0 \quad \text{or} \quad 12q + 6r = 12
\]

Solving the pair of equations in (11) for \( q \) and \( r \), we obtain

\( q = .75, \quad r = .5 \quad (12) \)

So our regression equation is

\( \hat{y} = .75x + .5 \quad (13) \)
Sec. 4.2 Linear Regression

(see Figure 4.7). This time our estimate for \((0, 1)\) is \((0, .5)\), for \((2, 1)\) is \((2, 2)\), and for \((4, 4)\) is \((4, 3.5)\), and \(\text{SSE} = .5^2 + 1^2 + .5^2 = 1.50\). The distance \(|\hat{y} - y|\) between our estimate vector \(\hat{y}\) and the true \(y\)-value vector \(y\) is \(\sqrt{\text{SSE}} = \sqrt{1.50} = 1.22\). In Example 1 the distance was 1.34. Thus, for these data, our fuller model provided little improvement over the simple model \(\hat{y} = qx\).

If we were applying linear regression models (2) or (3) to \(n\) points, the \(x\)- and \(y\)-values would form \(n\)-vectors and the distance \(|\hat{y} - y|\) would be calculated in \(n\) dimensions (the reasoning is the same). These calculations would be quite tedious. However, nowadays we have computers to handle the tedium. It is as easy to program a computer to do regression on \(n\) points as it is on three points with the following observation.

**Proposition.** The derivative of a sum of functions is the sum of the derivatives of each function.

This proposition greatly simplifies taking derivatives to find the minimum of \(\text{SSE}\). In the model \(\hat{y} = qx\), \(\text{SSE}\) has the following form for the set of points \((x_i, y_i), i = 1, 2, \ldots, n:\)

\[
\text{SSE} = \sum (qx_i - y_i)^2 = \sum (q^2 x_i^2 - 2qx_i y_i + y_i^2) \tag{14}
\]

By the proposition, the derivative of \(\text{SSE}\) is the sum of the derivatives of the individual terms in (14):

\[
\frac{d\text{SSE}}{dq} = \sum (2x_i^2 q - 2x_i y_i) = 2(\sum x_i^2)q - 2 \sum x_i y_i \tag{15}
\]

Setting (15) equal to 0 (to find the minimizing value of \(q\)), we obtain

**Formula for Regression Model \(\hat{y} = qx\)**

\[
q = \frac{\sum x_i y_i}{\sum x_i^2} = \frac{x \cdot y}{x \cdot x} \tag{16}
\]

Let us informally rederive (16) using matrix algebra. Recall that \(\text{SSE} = (qx - y) \cdot (qx - y)\). In vector calculus, we treat \((qx - y) \cdot (qx - y)\) like \((qx - y)^2\), and \(\text{SSE}\)'s derivative is

\[
2x \cdot (qx - y) = 2qx \cdot x - 2x \cdot y
\]

If we set this expression equal to zero, we get

\[
2qx \cdot x - 2x \cdot y = 0 \quad \text{or} \quad qx \cdot x = x \cdot y
\]

Hence we have directly \(q = x \cdot y / x \cdot x\).
The simple form of the formula (16) for $q$ was not apparent in the calculations we did in Example 1. If we were working with 10 or more points, it would clearly be much easier to do the general calculations just performed to obtain (16) and then plug in the given $x$- and $y$-values, rather than to multiply out all the squared factors in SSE and collect terms, as was done in equation (4) in Example 1. General symbolic computations can make life a lot easier! This is what mathematics is all about. The details of an individual problem often hide a nice general structure for solutions. In programming terms, it is often easier to write a computer program to solve a general class of problems and use it to solve one specific problem, rather than write a specialized program for the single problem.

The calculations for the regression model $\hat{y}_i = qx_i + r$ are obtained similarly by generalizing the equations in Example 2. The partial derivatives of SSE can be shown to have the form (see Exercise 10)

\[
\begin{align*}
\frac{\partial \text{SSE}}{\partial q} &= 2(\sum x_i^2)q + 2(\sum x_i)r - 2 \sum x_iy_i \\
\frac{\partial \text{SSE}}{\partial r} &= 2(\sum x_i)q + 2nr - 2 \sum y_i
\end{align*}
\]

(17)

We set the derivatives equal to 0 and solve this pair of equations for $q$ and $r$. That is, we solve the equations

\[
\begin{align*}
aq + br &= e \\
cq + dr &= f
\end{align*}
\]

(18)

where

\[
\begin{align*}
a &= 2 \sum x_i^2, \quad b = 2 \sum x_i, \quad e = 2 \sum x_iy_i, \\
c &= 2 \sum x_i, \quad d = 2n, \quad f = 2 \sum y_i
\end{align*}
\]

The solution is

\[
\boxed{	ext{Formula for Regression Model } \hat{y} = qx + r}
\]

\[
\begin{align*}
q &= \frac{n \sum x_iy_i - (\sum x_i)(\sum y_i)}{n \sum x_i^2 - (\sum x_i)^2} \\
r &= \frac{(\sum y_i)(\sum x_i^2) - (\sum x_i)(\sum x_iy_i)}{n \sum x_i^2 - (\sum x_i)^2}
\end{align*}
\]

(19)

Again we note the advantage of generality. Solving a pair of linear equations in terms of constants $a$, $b$, and so on, and then substituting complex expressions for the constants of (18), is much easier than directly solving the specific system of equations arising from (17).
Although the formulas in (19) are certainly complicated, they are still quite simple to program. In fact, because linear regression is so widely used, many (nonprogrammable) hand-held calculators have built-in routines to calculate $q$ and $r$. One simply enters successive $(x_i, y_i)$ pairs. When a pair is entered, the calculator updates sums that it is computing for $\Sigma x_i$, $\Sigma x_i^2$, $\Sigma y_i$, and $\Sigma x_i y_i$. After the last pair is entered, the user presses a Regression key and the calculator inserts these sums into the formulas in (19). The following BASIC program shows how the calculator works.

```
1 N=0: SX=0: SX2=0: SY=0: SXY=0
10 INPUT X,Y
20 N=N+1
30 SX = SX + X
40 SX2 = SX2 + X*X
50 SY = SY + Y
60 SXY = SXY + X*Y
70 IF Regression key not pushed THEN GOTO 10
100 D = N*SX2 - SX*SX
120 PRINT "'R = '"; (N*SXY-SX*SY)/D
130 PRINT "'Q = '"; (SY*SX2-SX*SXY)/D
140 END
```

A word of warning about the formulas in (19). Roundoff errors in computing the terms in the denominators of these formulas can sometimes seriously affect the accuracy of the results. In line 100 of the BASIC program, if N*SX2 and SX*SX were large numbers that were nearly equal, then their difference D may be very inaccurate. Also, one data point quite different from the rest (caused by an unusual event or a recording error) can significantly affect the values of $q$ and $r$; such points are called outliers. Exercise 5 illustrates the effect of outliers.

We now show a convenient shortcut for the linear regression model $\hat{y} = qx + r$. With a computer or calculator programmed to do regression, this shortcut does not save time, but it does eliminate roundoff-error difficulties.

We shall perform an elementary transformation of $x$-coordinates.

$$x' = x - \bar{x}, \quad \text{where} \quad \bar{x} = \frac{1}{n} \Sigma x_i$$

(20)

The term $\bar{x}$ we use to shift the $x$-coordinate is just the average of the $x_i$. Note that if the $x$-coordinates are integers (as is common) and are equally spaced along the $x$-axis, their average will be the middle integer, if $n$ is odd (or midway between the middle two integers, if $n$ is even). In the case in Example 2, the average of $x_i$ is $(0 + 2 + 4)/3 = 2$. Since the $y$-values are unchanged, we are simply renumbering the $x$-axis (see Figure 4.9).
Figure 4.9 Figure 4.7 transformed so that $x' = 0$ is the mean of $x'$-values. The old regression line $\hat{y} = 0.75x + 0.5$ becomes $\hat{y} = 0.75x' + 2$, where the constant 2 is the mean of the $y$-values.

In the new coordinate system, the average of the $x_i'$ will be 0—that was the whole idea of the shift, to center the $x$-values around the origin. If the average of the $x_i'$ is 0, so is the sum of the $x_i'$ (since the average is the sum divided by $n$). Now in (19) all products involving $x_i$ are 0. The formulas for $r$ and $q$ in (19) simplify considerably, to become

$$q' = \frac{\sum x_i'y_i}{\sum x_i'^2}$$

$$r' = \frac{1}{n} \sum y_i$$

(21)

Here $r'$ is just the average of the $y_i$, and $q'$ is the same formula that we obtained in (16) for the regression model $\hat{y} = qx$. Roundoff error can no longer distort the denominator in (21) as was possible in (19).

Shifting the $x$-coordinate will not change the slope of a line, so $q'$ equals $q$ in the original model $\hat{y} = qx + r$. The reader can verify with a geometric argument that $r = r' - q \frac{\sum x_i'}{n}$.

If we had also transformed the $y$-values by their average, then $r' = 0$. However, the regression formula for $q'$ does not simplify further if we transform the $y$-values, so a $y$-transformation serves no purpose. Also, transforming just the $y$-values instead of the $x$-values will not simplify the denominator in (19) as happened in (21).

**Example 3. Predicting Printing Costs**

A copy center bases its fees on the number of (duplicate) units that have been ordered (a unit is 100 pages). Table 4.1 gives some sample fees. Based on these sample fees, what would be a reasonable charge for 15 units?

Let us fit a line $\hat{y} = qx + r$ to these five data points, $(1, 6)$, $(3, 5.5)$, $(5, 5)$, $(10, 3.5)$, $(12, 3)$, and then determine $\hat{y}$ when $x =$
Sec. 4.2 Linear Regression

Table 4.1

<table>
<thead>
<tr>
<th>Number of Units</th>
<th>Cost per Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$6</td>
</tr>
<tr>
<td>3</td>
<td>5.5</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>3.5</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>

15. Using the shortcut described above, we transform the coordinates by subtracting 6.2 ($= \sum x_i/n$) from each $x_i$. Then (21) gives

$$
q' = \frac{\sum x_i y_i}{\sum x_i^2} = \frac{(-5.2) \times 6 + (-3.2) \times 5.5 + (-1.2) \times 5 + 3.8 \times 3.5 + 5.8 \times 3}{(-5.2)^2 + (-3.2)^2 + (-1.2)^2 + (3.8)^2 + (5.8)^2} = -0.3
$$

and $r' = \sum y_i/5 = 4.6$. Then

$$
q = q' = -0.3 \quad \text{and} \quad r = r' - \frac{q \sum x_i}{n} = 4.6 - (-0.3)(6.2) = 6.5
$$

Thus our regression model is $\hat{y} = -0.3x + 6.5$. And when $x = 15$, we obtain $\hat{y} = -0.3 \times 15 + 6.5 = 2$.

Next we give an example of a nonlinear regression problem and show how it can be converted into simple linear regression.

**Example 4. Nonlinear Regression**

Consider the following pairs of $x$- and $y$-values; the points are shown in Figure 4.1a. The $x$-values could be the age of a wine and the $y$-values ratings by expert wine tasters.

$$
\begin{array}{r|cccccccc}
  x & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  y & .4 & .6 & 1.1 & 1.6 & 2.9 & 5 & 8 \\
\end{array}
$$

Suppose that by inspection and experience we believe that the relationship between $x$- and $y$-values is best explained by an exponential model

$$
\hat{y} = re^{ax}
$$

(23)
Then we perform the following (nonlinear) transformation on the $y$-values.

$$y' = \log y$$  \hspace{1cm} (24)

The new data values are

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y'$</td>
<td>-.9</td>
<td>-.5</td>
<td>.1</td>
<td>.5</td>
<td>1.1</td>
<td>1.6</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Clearly, we have a fairly good linear fit here (see Figure 4.10b). (One can also plot the original data points on log paper.) We let $\hat{y}'_i$ be the estimate for the transformed problem:

$$\hat{y}'_i = \log \hat{y}_i = \log re^{qx_i} = \log r + qx_i$$

Letting $r' = \log r$, we have the standard simple linear regression model

$$\hat{y}' = qx + r'$$  \hspace{1cm} (25)

That is, the logarithm function (24) transforms exponential curves into straight lines. We can apply the formulas in (19) to determine $q$ and $r'$ from the transformed data and insert these into (23) to get a model in the original coordinate system. We obtain

$$q = \frac{1}{2}, \quad r' = -1.4 \quad (r = e^{-1.4} = \frac{1}{4})$$

The curve $\hat{y} = \frac{1}{4}e^{x/2}$ is plotted in dashed lines in Figure 4.10a. \qed
This technique can be used any time that there is a mapping that transforms the proposed regression model, such as (23), into a simple linear regression model. A statistician will test out several different regression models and settle on the one that gives the best fit (e.g., that minimizes the sum of the squares of the errors).

Another way to perform nonlinear regression is to fit the y values to a polynomial in x, such as $y_i = ax_i^2 + bx_i^2 + cx_i + d$, where we treat $x_i^2$, $x_i^3$, and $x_i$ like the three distinct variables, $v_i$, $w_i$, and $x_i$. We cannot solve a problem with several variables on the right yet, but we will come back to least-squares polynomial fitting in Section 5.3.

**Optional**

We conclude this section with a vector calculus derivation of the general multivariable regression model in which we allow $y$ to be a linear function of several input values. The same results will be obtained in Section 5.3 more simply using vector space techniques.

To be concrete, we consider the following model for $\hat{y}_i$:

$$\hat{y}_i = q_1 v_i + q_2 w_i + q_3 x_i + r$$

In matrix notation, we write

$$\hat{y} = q_1 v + q_2 w + q_3 x + r 1$$

$$= Xq$$

where $q = [q_1, q_2, q_3, r]$ and $X = [v \ w \ x \ 1]$.

Now let us compute SSE and its derivative in terms of (27a). Later we do it in terms of (27b).

$$SSE = (q_1 v + q_2 w + q_3 x + r 1 - y) \cdot (q_1 v + q_2 w + q_3 x + r 1 - y)$$

By the informal vector calculus used previously,

$$\frac{dSSE}{dq_1} = 2v \cdot (q_1 v + q_2 w + q_3 x + r 1 - y)$$

The derivatives with respect to $q_2$ and $q_3$ and $r$ will be similar. Multiplying $v$ by the vectors in the parentheses in (29) and setting the result equal to 0, we obtain

$$q_1 v \cdot v + q_2 v \cdot w + q_3 v \cdot x + rv \cdot 1 = v \cdot y$$

Three other similar equations will be obtained from the other three derivatives. This gives us four equations in the four unknowns $q_1$, $q_2$, $q_3$, and $r$ that can be solved by Gaussian elimination. (Recall that it was this system
of equations of estimating regression unknowns that forced Gauss to develop Gaussian elimination.)

Let us indicate how (30) and its sister equations can be obtained as a matrix equation. We write the SSE using (27b):

\[ \text{SSE} = (\hat{y} - y) \cdot (\hat{y} - y) = (Xq - y) \cdot (Xq - y) = Xq \cdot Xq - 2Xq \cdot y + y \cdot y \]  

(31)

By matrix algebra \( Xq \cdot y = X^T y \cdot q \) (see Exercise 14). Then by vector calculus, the derivative of \( X^T y \cdot q \) with respect to \( q \) is \( X^T y \). By more advanced vector calculus, the derivative of \( Xq \cdot Xq \) is \( 2(X^T X)q \). Thus

\[ \frac{d\text{SSE}}{dq} = 2(X^T X)q - 2X^T y \]  

(32)

Setting (32) equal to 0, we obtain the famous normal equations of regression.

\[ X^T X q = X^T y \]  

(33)

whose solution, using inverses, is

\[ q = (X^T X)^{-1} X^T y \]  

(34)

The matrix expression \((X^T X)^{-1} X^T\) is called the pseudoinverse of \( X \), since it allows us to solve (approximately) the system \( Xq = y \). Pseudoinverses are discussed in Section 5.3.

Section 4.2 Exercises

Summary of Exercises

Exercises 1–8 involve regression models, Exercises 6–8 being nonlinear. Exercises 9–14 are theoretical.

1. Seven students earned the following scores on a test after studying the subject matter for different numbers of weeks:

<table>
<thead>
<tr>
<th>Student</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of Study</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Test Score</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>10</td>
</tr>
</tbody>
</table>

(a) Fit these data with a regression model of the form \( \hat{y} = qx \), where \( x \) is number of weeks studied and \( y \) is the test score. Plot the observed scores and the predicted scores. What is the sum of squares of errors?
(b) Fit these data with a regression model of the form \( \hat{y} = qx + r \). Plot the observed scores and the predicted scores. What is the sum of squares of errors?

(c) Repeat the calculations in part (b) by first shifting the \( x \)-values to make the average \( x \)-value be 0 [see equations (21)].

2. The following data indicate the numbers of accidents that bus drivers had in one year as a function of the numbers of years on the job.

<table>
<thead>
<tr>
<th>Years on Job</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accidents</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

(a) Fit these data with a regression model of the form \( \hat{y} = qx \), where \( x \) is number of years experience and \( y \) is number of bus accidents. Plot the observed numbers of accidents and the predicted numbers. What is the sum of squares of errors?

(b) Fit these data with a regression model of the form \( \hat{y} = qx + r \). Plot the observed numbers of accidents and the predicted numbers. What is the sum of squares of errors? Is this model significantly better than the model in part (a)?

(c) Repeat the calculations in part (b) by first shifting the \( x \)-values to make the average \( x \)-value be 0 [see equations (21)].

3. (a) Reverse the roles of \( y \) and \( x \) in Exercise 2—now \( y \) is number of years of experience—and fit the regression model \( \hat{y} = qx + r \) to these data. Plot the observed years experience and the predicted numbers. What is the sum of squares of errors?

(b) Compare your results with those in Exercise 2, part (b) or (c)—why are the numbers not the same?

4. (a) The following data show the GPA and the job salary (5 years after graduation) of six mathematics majors from Podunk U.

<table>
<thead>
<tr>
<th>GPA</th>
<th>2.3</th>
<th>3.1</th>
<th>2.7</th>
<th>3.4</th>
<th>3.7</th>
<th>2.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salary</td>
<td>25,000</td>
<td>38,000</td>
<td>28,000</td>
<td>35,000</td>
<td>30,000</td>
<td>32,000</td>
</tr>
</tbody>
</table>

Fit these data with a regression model of the form \( \hat{y} = qx + r \). Plot the observed salaries and the predicted salaries. Is the regression fit reasonably good?

(b) Repeat the calculations in part (a) by first shifting the \( x \)-values to make the average \( x \)-value be 0 [see equations (21)].

5. Consider the following relationship between the height of a student's mother and the number of F's the student gets at Podunk U.
(a) Determine $q$ and $r$ in the regression model $\hat{y} = qx + r$ (where $x$ is mother's height and $y$ is number of F's).

(b) Delete student $E$ from your study and repeat part (a). Does deleting $E$ make much of a difference?

(c) Repeat the calculations in part (b) by first shifting the $x$-values to make the average $x$-value be 0 [see equations (21)].

6. Consider the following set of data, which are believed to obey (approximately) a relation of the form $y = qx^2$:

<table>
<thead>
<tr>
<th>$x$-value</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$-value</td>
<td>.5</td>
<td>1</td>
<td>2</td>
<td>3.5</td>
<td>7</td>
<td>9</td>
<td>12</td>
</tr>
</tbody>
</table>

Perform a transformation $y' = f(y)$ on $y$ so that the regression model $\hat{y}' = q'x$ is fairly accurate. Then determine $q'$, reverse the transformation to determine $q$, and plot the curve $\hat{y} = qx^2$.

7. Consider the following set of data, which are believed to obey an inverse relation of the form $y = q(1/x)$.

<table>
<thead>
<tr>
<th>Experience ($x$)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Accidents ($y$)</td>
<td>10</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Perform a transformation $y' = f(y)$ on $y$ so that the regression model $\hat{y}' = q'x$ is fairly accurate. Then determine $q'$, reverse the transformation to determine $q$, and plot the curve $\hat{y} = q(1/x)$.

8. Consider the following set of data, which are believed to obey a square root law $y = q\sqrt{x}$. 
Perform a transformation \( y' = f(y) \) on \( y \) so that the regression model \( y' = q'x \) is fairly accurate. Then determine \( q' \), reverse the transformation to determine \( q \), and plot the curve \( \hat{y} = q\sqrt{x} \).

9. Verify the calculation of the partial derivatives in (11).

10. Verify (17).

11. Verify the expression for \( r = r' - q \sum x_i/n \), where \( r' \) and \( q \) (= \( q' \)) are the regression coefficients in the transformed problem [see (21)].

12. Show that the formula for \( q \) and \( r \) makes the regression line \( y = qx + r \) go through the point \((\bar{x}, \bar{y})\), where \( \bar{x} \) is the average \( x \)-value and \( \bar{y} \) is the average \( y \)-value.

   \textit{Hint:} First shift the \( x \)-values so that \( \bar{x} = 0 \).

13. In vector notation, the sum of squares to be minimized in the model \( y = qx + r \) is \( \text{SSE} = (qx + r1 - y) \cdot (qx + r1 - y) \). Compute the vector derivative of this expression with respect to \( q \) and with respect to \( r \) [see (29)]. Show that the two derivatives are the same as the expressions in (17).

14. Verify that \( Xq \cdot y = X'y \cdot q \).
For this reason, none of the models examined in depth in this book will come from the physical sciences. The linear models, such as Markov chains and growth models, that we shall repeatedly use to illustrate concepts are "neutral" models that are easily comprehended by all students. However, for completeness in this section a few models based on basic physical laws will be presented. (Social scientists should consider this section as the book’s "College of Arts and Letters’ distribution requirement in science.")

The same reasoning applies to differential equations. Students who will use differential equations in courses in their major should have a full course in differential equations. We will only sample the subject to see some basic ways that linear systems of equations arise in differential equations.

Example 1. Balancing Chemical Equations

In a chemical reaction, a collection of molecules are brought together in the proper setting (e.g., in boiling water) and they rearrange themselves into new molecules. In this process the number of atoms of each element is conserved. If the molecules put into the reaction have a total of 12 hydrogen (H) atoms, the resulting set of molecules must also contain 12 H’s. Consider the reaction in which permanganate (MnO₄) and hydrogen (H) ions combine to form manganese (Mn) and water (H₂O):

\[ \text{MnO}_4^- + \text{H} \rightarrow \text{Mn} + \text{H}_2\text{O} \]  

where O represents oxygen. Let \( x_1 \) be the number of permanganate ions, \( x_2 \) the number of hydrogen ions, \( x_3 \) the number of manganese atoms, and \( x_4 \) the number of water molecules. To have the same number of atoms in the molecules on each side of the reaction, we obtain the system of equations.

\[
\begin{align*}
\text{H:} & \quad x_2 = 2x_4 \\
\text{Mn:} & \quad x_1 = x_3 \\
\text{O:} & \quad 4x_1 = x_4
\end{align*}
\]

or

\[
\begin{align*}
x_2 - 2x_4 &= 0 \\
x_1 - x_3 &= 0 \\
4x_1 - x_4 &= 0
\end{align*}
\]

Notice that we have four unknowns but only three equations.

Let us solve this system using elimination by pivoting. We pivot on entry (2, 1) to obtain

\[
\begin{align*}
x_2 - 2x_4 &= 0 \\
x_1 - x_3 &= 0 \\
4x_3 - x_4 &= 0
\end{align*}
\]
We do not need to pivot in column 2. Next we pivot on entry (3, 3)

\[
\begin{align*}
x_2 - 2x_4 &= 0 & x_2 &= 2x_4 \\
x_1 - \frac{1}{4}x_4 &= 0 \quad \text{or} \quad x_1 = \frac{1}{4}x_4 \\
x_3 - \frac{1}{4}x_4 &= 0 & x_3 &= \frac{1}{4}x_4
\end{align*}
\]

As vectors, the solutions in (3) have the form

\[
\begin{bmatrix} \frac{1}{4}x_4, 2x_4, \frac{1}{4}x_4, x_4 \end{bmatrix} \quad \text{or} \quad x_4[\frac{1}{4}, 2, \frac{1}{4}, 1] \quad (3)
\]

For example, if \(x_4 = 4\), then \(x_2 = 8\), \(x_1 = x_3 = 1\), and the reaction equation becomes

\[
\text{MnO}_4^- + 8\text{H} \rightarrow \text{Mn} + 4\text{H}_2\text{O}
\]

The solution we obtain makes the amounts of the first three types of molecules fixed ratios of the amount of the fourth type, which are free to give any value (i.e., \(x_4\) is a free variable). This makes sense in physical terms, since doubling our chemical "recipe" of inputs should just double the output.

In another series of pivots, say at entries (1, 2), (2, 3), and (3, 4), we get \(x_1\) as the free variable.

\[
\begin{align*}
-8x_1 + x_2 &= 0 & x_2 &= 8x_1 \\
x_1 + x_3 &= 0 \quad \text{or} \quad x_3 = x_1 \\
-4x_1 + x_4 &= 0 & x_4 &= 4x_1
\end{align*}
\]

yielding solution vectors

\[
\begin{bmatrix} x_1, 8x_1, x_1, 4x_1 \end{bmatrix} \quad \text{or} \quad x_1[1, 8, 1, 4] \quad (4)
\]

In this solution the values of the last three variables are fixed ratios of the first variable.

---

**Example 2. Currents in an Electrical Network**

In this example we compute the current in different parts of the electrical network in Figure 4.11a. There are three basic laws that are used to analyze simple electrical networks. The following review of elementary physics summarizes the concepts behind these laws. A battery or other source of electrical power "forces" electricity through electrical devices, such as a light or a doorbell. The force applied to a device depends on two factors: (i) the resistance of the device—a measure of how hard it is to push electricity through the device; and (ii) the current, the rate at which electricity flows through the device. The fundamental law of electricity, due to Ohm, says

**Ohm's Law:** force = resistance \(\times\) current.
Force is measured in volts, current in amperes, and resistance in ohms. In terms of these units, Ohm’s law is

\[ \text{volts} = \text{ohms} \times \text{amperes} \]

A battery supplies a fixed force into a network. Batteries send their electricity out from a positive terminal and receive it back at a negative terminal. All the voltage (i.e., force) provided by a battery is used up by the time the electricity returns to the terminal. This property of voltage is called

**Kirchhoff’s Voltage Law.** In any cycle (closed path) in a network, the sum of the voltages used by resistive devices equals the voltage from the battery(ies).

The final law says that current is conserved at any branch node.

**Kirchhoff’s Current Law.** The sum of the currents flowing into any node is equal to the sum of the current flowing out of the node.

Let us use these three laws to derive a set of linear equations modeling the behavior of currents in the network in Figure 4.11a. Later we shall express each of Kirchhoff’s laws in the form of a matrix equation.

Assume that the battery delivers 14 volts. Further let \( c_1 \) be the current flowing through the section of the network with the battery and the light, whose resistance is 1 ohm; let \( c_2 \) be the current through the section with the doorbell, whose resistance is 2 ohms; and let \( c_3 \) be
the current through the section with the motor, whose resistance is 4 ohms. Currents flow in the direction of the arrows along these edges. By the current law, at node $s$ we have the following equation:

$$c_1 = c_2 + c_3$$

(5)

Note that at node $t$ we get the same equation. Next we use the voltage law. Following the cycle, battery to light to doorbell to battery, we have

voltage at light + voltage at doorbell = battery voltage  (6a)

We use Ohm’s law (voltage = resistance $\times$ current) to determine the voltages at the devices. The battery voltage is 14. So (6a) becomes

$$1c_1 + 2c_2 = 14$$

(6b)

Following a second cycle, battery to light to motor to battery, we have

$$1c_1 + 4c_3 = 14$$

(7)

There is still a third cycle that we could use, node $s$ to doorbell to node $t$ to motor (going against the current flow) to node $s$. If we go against the current flow, the voltage is treated as negative. So the voltage law for this cycle is

$$2c_2 - 4c_3 = 0$$

(8)

Note that equation (8) is simply what is obtained when we subtract (7) from (6b). Intuitively, the third cycle is the net result of going forward on the first cycle and then backward on the second cycle.

We need to solve the three equations (5), (6), and (7) in the three unknown currents, which we write as

$$c_1 - c_2 - c_3 = 0$$

$$c_1 + 2c_2 = 14$$

$$c_1 + 4c_3 = 14$$

(9)

Solving by Gaussian elimination, we find

$$c_1 = 6 \text{ amperes, } \quad c_2 = 4 \text{ amperes, } \quad c_3 = 2 \text{ amperes}$$

(10)

A general analysis of currents in networks involves a combination of physics and mathematics. The critical mathematical problem is proving that there will always be enough different equations to determine uniquely all the currents.
Before leaving this problem, let us show how this problem can be cast in matrix notation. Kirchhoff's current law can be restated in matrix form using the incidence matrix $M(G)$ of the underlying graph (when batteries and resistive devices are ignored). Figure 4.11b shows the graph $G$ associated with the network in Figure 4.11a. $G$ contains two nodes $s, t$ and three edges $e_1, e_2, e_3$ joining $s$ and $t$. In this graph each edge has a direction; the directions for $G$ are shown in Figure 4.11b. The current $c_i$ in edge $e_i$ is positive if it flows in the direction of $e_i$ and negative if it flows in the opposite direction.

Recall from Section 2.3 that $M(G)$ has a row for each node of $G$ and a column for each edge. In the case of directed edges, entry $m_{ij} = +1$ if the $j$th edge is directed into $i$th node, $= -1$ if the $j$th edge is directed out from the $i$th node, and $= 0$ if the $j$th edge does not touch the $i$th node. For the graph $G$ in Figure 4.11b, $M(G)$ is

$$M(G) = \begin{bmatrix} e_1 & e_2 & e_3 \\ s & +1 & -1 & -1 \\ t & -1 & +1 & +1 \end{bmatrix}$$

Kirchhoff's current law says that the flow into a node equals the flow out of the node, or in other words, the net current flow is zero. At node $s$, this means that

$$+c_1 - c_2 - c_3 = 0$$

Observe that currents going into $s$ are associated with edges that have $+1$, in row $s$ of $M(G)$, and currents going out are associated with edges that have $-1$ in row $s$ of $M(G)$. If $m_s$ denotes row $s$ of $M(G)$ and $c$ is the vector of currents ($c_1, c_2, c_3$), then (12) can be rewritten as

$$m_s \cdot c = 0$$

This equation is true for all rows of $M(G)$. Thus (13) generalizes to

Kirchhoff's current law:

$$M(G)c = 0$$

This result is true for any associated graph $G$.

We can also define a special cycle matrix $K(G)$ for $G$ with a row for each cycle (closed path) of $G$ and a column for each edge. Let $r_j$ be the resistance in the $j$th edge. Then define entry $k_{ij}$ of $K(G) = +r_j$ if the $i$th cycle uses the $j$th edge traversing this edge in the direction of its arrow, $-r_j$ if the $i$th circuit uses the $j$th edge in the opposite direction of its arrow, and $= 0$ otherwise. For example, $K(G)$ in Example 2 would be, with cycles listed in the order they were discussed,

$$K(G) = \begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & 4 \\ 0 & 2 & -4 \end{bmatrix}$$
As occurred in Example 1, some circuits in a graph are always redundant and there is a simple rule for picking a minimal set of cycles for $K(G)$. If $k_i$ is the $i$th row of $K(G)$, then $k_i \cdot c$ will be the voltages used on the $i$th cycle. If $f_i$ is the voltage force of batteries on the $i$th cycle, then Kirchhoff’s voltage law becomes

$$k_i \cdot c = f_i$$

and if $f$ is the vector of $f_i$’s, we have

Kirchhoff’s voltage law: \[ K(G)c = f \] (16)

Our current problem can now be stated: Solve the system of equations (14) and (16) for $c$.

The next four examples involve differential equations. A common mathematical model for many dynamic systems, such as falling objects, vibrating strings, or economic growth, is a differential equation of the form

$$y''(t) = a_1 y'(t) + a_0 y(t)$$

where $y(t)$ is a function that measures the “position” of the quantity, $y'(t)$ denotes the first derivative with respect to $t$ (representing time), $y''(t)$ denotes the second derivative, and $a_1$ and $a_0$ are constants. The differential equation is called linear because the right side is a linear combination of the function and its derivative. Solutions of (17) are functions of the form

$$y(t) = Ae^{kt}$$

where $e$ is Euler’s constant, and $A$ and $k$ are constants that depend on the particular problem. This form of solution also works if higher derivatives are involved in the linear differential equation.

**Example 3. Differential Equation for Instantaneous Interest**

The simple differential equation

$$y'(t) = .10y(t)$$

(19)

describes the amount of money $y(t)$ in a savings account after $t$ years when the account earns 10% interest compounded instantaneously. Recall that $y'(t)$ is the instantaneous rate of change, or graphically, the slope, of $y(t)$. Thus (19) says that the instantaneous growth rate of the savings account is 10% of the account’s current value.

Recall that the derivative of $e^{kt}$ is $ke^{kt}$. Let us try setting $y(t) = Ae^{kt}$ [as given in (18)]. Now (19) becomes

$$kAe^{kt} = .10Ae^{kt}$$

(20)
Dividing both sides of (20) by $Ae^{kt}$, we obtain $k = .10$. So the solution to (19) is

$$y(t) = Ae^{.10t}$$  \hspace{1cm} (21)

The constant $A$ is still to be determined because to know how much money we shall have after $t$ years, we must know how much we started with. Suppose that we started with 1000 dollars. The starting time is $t = 0$. Thus we have (using the fact $e^0 = 1$)

$$1000 = y(0) = Ae^0 = A$$  \hspace{1cm} (22)

Then the solution of (19) with $y(0) = 1000$ is

$$y(t) = 1000e^{.10t}$$  \hspace{1cm} (23)

**Example 4. Solving Second-Order Linear Differential Equations**

Consider the following differential equation that might describe the height of a falling particle in a special force field.

$$y''(t) = 6y'(t) - 8y(t)$$  \hspace{1cm} (24)

This differential equation is called a second-order equation because it involves the second derivative. To solve this equation, we also need to know the starting conditions, what are the initial height $y(0)$ and the initial speed $y'(0)$. Here the derivative $y'(t)$ measures speed, that is, the rate of change of the height. Suppose in this problem that

$$f(0) = 100 \quad \text{and} \quad f'(0) = -20$$  \hspace{1cm} (25)

We solve this problem in two stages. First we substitute (18) for $y(t)$ in (24).

$$\frac{d^2}{dt^2} Ae^{kt} = 6 \frac{d}{dt} Ae^{kt} - 8Ae^{kt}$$  \hspace{1cm} (26)

Recall that the second derivative of $e^{kt}$ is $k^2e^{kt}$. So (26) becomes

$$k^2Ae^{kt} = 6kAe^{kt} - 8Ae^{kt}$$  \hspace{1cm} (27)

If we divide by $Ae^{kt}$, (27) becomes

$$k^2 = 6k - 8 \quad \text{or} \quad k^2 - 6k + 8 = 0$$  \hspace{1cm} (28)
Equation (28) is called the characteristic equation of the linear differential equation (24). The roots of (28) are easily verified to be 2 and 4. So we have two possible types of solutions to (24).

\[ y(t) = Ae^{4t} \quad \text{and} \quad y(t) = A'e^{2t} \]

The constants \( A \) and \( A' \) can have any value and these solutions will still satisfy (24). In fact, it can readily be checked [see Exercise 16, part (b)] that any linear combination of the basic solutions \( e^{2t} \) and \( e^{4t} \) is a solution. Thus

\[ y(t) = Ae^{4t} + A'e^{2t} \quad (29) \]

is the general form of a solution to (24). The constants \( A \) and \( A' \) depend on the starting values. From (25) we have

\[
\begin{align*}
100 &= y(0) = Ae^0 + A'e^0 \\
-20 &= y'(0) = 4Ae^0 + 2A'e^0
\end{align*} \quad (30a)
\]

which simplifies to

\[
\begin{align*}
A + A' &= 100 \\
4A + 2A' &= -20
\end{align*} \quad (30b)
\]

In (30a), we obtain \( y'(0) \) by differentiating (29) and setting \( t = 0 \). Now we have our old "friend," a system of two equations in two unknowns. We solve (30b) and obtain

\[
A = -110 \quad \text{and} \quad A' = 210 \quad (31)
\]

Substituting these values in (29), we obtain the required solution

\[ y(t) = -110e^{4t} + 210e^{2t} \quad (32) \]

The calculations for any other second-order differential equation would proceed in a similar fashion: First substitute (18) in the differential equation to obtain the characteristic equation [as in (28)] and solve for its roots; then determine \( A \) and \( A' \) from the pair of equations for starting values [as in (30)]. This method generalizes to \( k \)-th order differential equations; then the characteristic equation has \( k \) roots, we need \( k \) initial values, and we have to solve \( k \) equations in \( k \) unknowns.

For completeness, we note that if the two roots of the characteristic equation (28) were the same, such as 2 and 2, then the starting value equations cannot be solved, since the two equations of (30a) will be the same. In the case of identical roots of the characteristic equation, \( y(t) \) instead has the form

\[ y(t) = Ae^{rt} + A'te^{rt} \quad (33) \]
where \( r \) is the double root. It is an exercise to check that in the case of a multiple root (and only then), \( te^r \) is also a solution to the differential equation.

**Example 5. A System of Differential Equations**

Let us consider a pair of first-order differential equations which describe motion of an object in \( x-y \) space with one equation governing the \( x \)-coordinate and one the \( y \)-coordinate.

\[
\begin{align*}
x'(t) &= 2x(t) - y(t) \\
y'(t) &= -x(t) + 2y(t)
\end{align*}
\]  

(34)

The starting values are \( x(0) = y(0) = 1 \). Let \( \mathbf{u}(t) \) be the vector function \( \mathbf{u}(t) = (x(t), y(t)) \). Then (34) can be written in matrix notation as

\[
\mathbf{u}'(t) = \mathbf{B}\mathbf{u}(t), \quad \text{where } \mathbf{B} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}
\]  

(35)

and \( \mathbf{u}(0) = [1, 1] = 1 \). In Example 3 we saw that

\[
y'(t) = b\mathbf{y}(t), \quad y(0) = A \rightarrow y(t) = Ae^{bt}
\]  

(36)

Substituting \( \mathbf{B} \) for \( b \) and \( \mathbf{1} \) for \( A \) in (36), we obtain the solution to (35):

\[
\mathbf{u}(t) = e^{\mathbf{B}t}\mathbf{1}
\]  

(37)

A matrix in the exponent looks strange. But one definition of \( e^x \) is in terms of the power series.

\[
e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^k}{k!} + \cdots
\]  

(38)

Similarly, \( e^\mathbf{B} \) is defined

\[
e^\mathbf{B} = \mathbf{I} + \mathbf{B} + \frac{\mathbf{B}^2}{2!} + \frac{\mathbf{B}^3}{3!} + \cdots + \frac{\mathbf{B}^k}{k!} + \cdots
\]  

(39)

The power series (39) is well defined for all matrices. Although this power series of matrices may look forbidding, it is easy to use if we work in eigenvector-based coordinates so that \( \mathbf{B} \) and its powers act like scalar multipliers (as in \( \mathbf{B}\mathbf{u} = \lambda\mathbf{u} \)). Recall Theorem 5 of Section 3.3, which said that if \( \mathbf{U} \) is a matrix whose columns were different eigenvectors of \( \mathbf{B} \) and if \( \mathbf{D}_\lambda \) is a diagonal matrix of associated eigenvalues, then

\[
\mathbf{B} = \mathbf{UD}_\lambda\mathbf{U}^{-1}
\]  

(40)
Substituting with (40) for $B$ in (39), we obtain

$$e^B = I + UD_0 U^{-1} + \frac{UD_1^2 U^{-1}}{2!} + \frac{UD_2^3 U^{-1}}{3!} + \cdots$$

$$+ \frac{UD_k^k U^{-1}}{k!} + \cdots$$

$$= U\left(I + D_0 + \frac{D_1^2}{2!} + \frac{D_2^3}{3!} + \cdots + \frac{D_k^k}{k!} + \cdots\right)U^{-1}$$

$$= U e^{D_0} U^{-1} = U \begin{bmatrix}
e^{\lambda_1} & 0 & 0 & \cdots \\
0 & e^{\lambda_2} & 0 & \cdots \\
0 & 0 & e^{\lambda_3} & \cdots \\
\vdots & \vdots & \vdots & \ddots
\end{bmatrix} U^{-1}$$

The reason that $e^{D_0}$ turns out to be simply a diagonal matrix with diagonal entries $e^{\lambda_1}, e^{\lambda_2}, \ldots, e^{\lambda_n}$ is that in (41b), the matrices $D_k^k/k!$ are diagonal with entries $\lambda_1/k!, \lambda_2/k!, \ldots$ and summing these matrices we get a matrix whose entry $(1, 1)$ is $1 + \lambda_1 + \lambda_1^2/2! + \lambda_1^3/3! + \cdots$, which equals $e^{\lambda_1}$; and similarly for the other diagonal entries.

---

**Example 6. Converting a Second-Order Differential Equation into a Pair of First-Order Differential Equations**

Let us consider again the second-order equation from Example 4:

$$y''(t) = 6y'(t) - 8y(t)$$

(42)

We convert (42) into a pair of first-order equations by introducing a second function $x(t)$ defined

$$x(t) = y'(t) \quad \text{and thus} \quad x'(t) = y''(t)$$

(43)

Now (42) can be written as the pair of the first-order equations

$$x'(t) = 6x(t) - 8y(t)$$

(44)

$$y'(t) = x(t)$$

Defining the vector function $u(t) = [x(t), y(t)]$, we have

$$u'(t) = Bu(t), \quad \text{where} \quad B = \begin{bmatrix} 6 & -8 \\ 1 & 0 \end{bmatrix}$$

(45)

with initial conditions from Example 4 of $u(0) = [-20, 100]$. 
As in Example 5, the solution to (45) should be

\[ u(t) = e^{Br}u(0) \]  

(46)

where \( e^{Br} \) is defined by the power series

\[ e^{Br} = I + tB + \frac{t^2B^2}{2!} + \frac{t^3B^3}{3!} + \cdots + \frac{t^kB^k}{k!} + \cdots \]  

(47)

Remember that we already know from Example 4 the solution of (42), so \( u(t) \) in (46) must equal

\[ u(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} -440e^{4t} + 420e^{2t} \\ -110e^{4t} + 210e^{2t} \end{bmatrix} \]  

(48)

where \( x(t) = y'(t) = -440e^{4t} + 420e^{2t} \) is obtained by differentiating the solution for \( y(t) \).

We now use the eigenvector-coordinates approach from Sections 2.5 and 3.3 to show how the intimidating formula for \( u(t) \) in (46) is the same as (48). Since by (47) \( e^{Br} \) involves powers of \( B \), the computation of multiplying \( e^{Br} \) times \( u(0) \) will be simplified if we express \( u(0) \) in terms of \( B \)'s eigenvectors.

In Section 3.1 we learned how to find the eigenvalues of a matrix \( B \)—they are the roots of the characteristic polynomial \( \det (B - \lambda I) \)—and from them, the associated eigenvectors. The characteristic polynomial for \( B \) is \( \lambda^2 - 6\lambda + 8 \) and its roots are 4 and 2. Eigenvectors \( u_1, u_2 \) of \( B \) associated with the eigenvalues 4 and 2 are (Exercise 13):

\[ u_1 = [4, 1] \quad \text{for } \lambda_1 = 4 \quad u_2 = [2, 1] \quad \text{for } \lambda_2 = 2 \]

Writing \( u(0) = [-20, 100] \) as a linear combination of \( u_1 \) and \( u_2 \) (we must solve the system \( u(0) = au_1 + bu_2 \) for \( a \) and \( b \) (see Section 2.5 for details), we obtain

\[ u(0) = [-20, 100] = -110u_1 + 210u_2 \]

(49)

We now can compute \( e^{Br}u(0) \), which we rewrite using (47) as

\[ u(t) = e^{Br}u(0) \]

\[ = Bu(0) + tBu(0) + \frac{t^2B^2u(0)}{2!} + \cdots \]

\[ + \frac{t^kB^ku(0)}{k!} + \cdots \]

(50)

Substituting \( u(0) = -110u_1 + 210u_2 \) in (50), we have
\[ u(t) = I(-110u_1 + 210u_2) + tB(-110u_1 + 210u_2) + \cdots \]
\[ + \frac{t^2}{2!} B^2(-110u_1 + 210u_2) + \cdots \]
\[ + \frac{t^k}{k!} B^k(-110u_1 + 210u_2) + \cdots \]
\[ = -110 \left\{ t u_1 + tBu_1 + \frac{t^2}{2!} B^2 u_1 + \cdots + \frac{t^k}{k!} B^k u_1 + \cdots \right\} \]
\[ + 210 \left\{ t u_2 + tBu_2 + \frac{t^2}{2!} B^2 u_2 + \cdots + \frac{t^k}{k!} B^k u_2 + \cdots \right\} \]
\[ = -110 \left\{ 1 + t4 + \frac{t^24^2}{2!} + \cdots + \frac{t^k4^k}{k!} + \cdots \right\} u_1 \]
\[ + 210 \left\{ 1 + t2 + \frac{t^22^2}{2!} + \cdots + \frac{t^k2^k}{k!} + \cdots \right\} u_2 \]
\[ = -110e^{4t}u_1 + 210e^{2t}u_2 \]
\[ = -110e^{4t} \begin{bmatrix} 4 \\ 1 \end{bmatrix} + 210e^{2t} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \]
\[ = \begin{bmatrix} -440e^{4t} + 420e^{2t} \\ -110e^{4t} + 210e^{2t} \end{bmatrix} \quad (52) \]

Observe that for a different starting vector \( u^*(0) \), we would get
\[ u^*(0) = a'u_1 + b'u_2, \]
for some \( a' \), \( b' \) and then the result in (52) would be
\[ u(t) = a'e^{4t}u_1 + b'e^{2t}u_2. \]

We now give a shorter derivation of this result using the matrix formula in (41) to handle the exponential series. For \( e^{Bt} \), (41) becomes
\[ e^{Bt} = Ue^{D't}U^{-1} \quad (53) \]

where \( e^{D't} \) is a diagonal matrix with diagonal entries \( e^{\lambda't} \). Recall that \( U \) has eigenvectors \( u_1 \) and \( u_2 \) as its columns. Thus
\[ U = \begin{bmatrix} 4 & 2 \\ 1 & 1 \end{bmatrix} \]
and we compute
\[ U^{-1} = \begin{bmatrix} 1/2 & -1 \\ -1/2 & 2 \end{bmatrix} \quad (54) \]
Using (53) to substitute for $e^{Bu}$ in $u(t) = e^{Bu(0)}$, we obtain

$$u(t) = e^{Bu(0)} = Ue^{Dv}U^{-1}u(0)$$

$$= \begin{bmatrix} 4 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} e^{4t} \\ e^{2t} \end{bmatrix} \begin{bmatrix} 0.5 & -1 \\ -0.5 & 2 \end{bmatrix} \begin{bmatrix} -20 \\ 100 \end{bmatrix}$$

$$= \begin{bmatrix} 4e^{4t} & 2e^{2t} \\ e^{4t} & e^{2t} \end{bmatrix} \begin{bmatrix} -110 \\ 210 \end{bmatrix}$$

$$= \begin{bmatrix} -440e^{4t} + 420e^{2t} \\ -110e^{4t} + 210e^{2t} \end{bmatrix} \tag{55}$$

The conversion of (42) to a system of first-order differential equations can be applied to any linear higher-order differential equation.

**Example 7. Converting a Third-Order Differential Equation into a System of Three First-Order Differential Equations**

Consider the third-order linear differential equation

$$y'''(t) = y''(t) + 2y'(t) + 3y(t) \tag{56}$$

We introduce the two new functions $w(t)$ and $z(t)$:

$$w(t) = y'(t) \quad \text{and} \quad z(t) = w'(t) \quad \left[ = y''(t) \right] \tag{57}$$

Then (56) can be written

$$z'(t) = z(t) + 2w(t) + 3y(t) \tag{58}$$

Defining the vector function $u(t) = [y(t), w(t), z(t)]$, we can rewrite (57) and (58) as the matrix equation

$$u'(t) = Bu(t): \begin{bmatrix} z'(t) \\ w'(t) \\ y'(t) \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} z(t) \\ w(t) \\ y(t) \end{bmatrix} \tag{59}$$

The solution to (59) is $u(t) = e^{Bu(0)}$, which we would evaluate using eigenvectors, as discussed in Example 6. \[\blacksquare\]
Section 4.3 Exercises

Summary of Exercises
Exercises 1–4 are chemical reaction balancing problems. Exercises 5–7 are electrical circuit problems. Exercises 8–18 involve differential equations, with Exercises 16–18 being theory questions.

1. Write out a system of equations required to balance the following chemical reactions and solve. Here C represents carbon, N represents nitrogen, H represents hydrogen, and O represents oxygen.
   (a) \( \text{N}_2\text{H}_4 + \text{N}_2\text{O}_4 \rightarrow \text{N}_2 + \text{H}_2\text{O} \)
   (b) \( \text{C}_6\text{H}_6 + \text{O}_2 \rightarrow \text{CO}_2 + \text{H}_2\text{O} \)

2. Write out a system of equations required to balance the following chemical reaction and solve.
   \( \text{SO}_2 + \text{NO}_3 + \text{H}_2\text{O} \rightarrow \text{H} + \text{SO}_4 + \text{NO} \)
   where S represents sulfur, N represents nitrogen, H represents hydrogen, and O represents oxygen.

3. Write out a system of equations required to balance the following chemical reaction and solve.
   \( \text{PbN}_6 + \text{CrMn}_2\text{O}_8 \rightarrow \text{Cr}_2\text{O}_3 + \text{MnO}_2 + \text{Pb}_3\text{O}_4 + \text{NO} \)
   where Pb represents lead, N represents nitrogen, Cr represents chromium, Mn represents manganese, and O represents oxygen.

4. Write out a system of equations required to balance the following chemical reaction and solve.
   \( \text{H}_2\text{SO}_4 + \text{MnS} + \text{As}_2\text{Cr}_{10}\text{O}_{35} \rightarrow \text{HMnO}_4 + \text{AsH}_3 + \text{CrS}_2\text{O}_{12} + \text{H}_2\text{O} \)
   where H represents hydrogen, S represents sulfur, O represents oxygen, Mn represents manganese, As represents arsenic, and Cr represents chromium.

5. Determine the currents in each branch of the following circuit.
6. Determine the currents in each branch of the following circuits, in which the incoming amperage (on the left) is given.

(a)  
\[ i_1 = 5 \]
\[ \begin{array}{c}
  i_2 \\
  i_3 \\
  i_4 
\end{array} \]

(b)  
\[ i_1 = 10 \]
\[ \begin{array}{c}
  i_2 \\
  i_3 \\
  i_4 
\end{array} \]

7. Determine the currents in each branch of the following circuit. The voltage in the battery is 19.

\[ \begin{array}{c}
  i_5 \\
  19 \\
  1 \\
  2 \\
  3 \\
  i_4 \\
  i_1 \\
  i_2 \\
  i_3 
\end{array} \]

8. Solve the following first-order differential equations, with given initial values.
   (a) \( y'(t) = 0.5y(t), \quad y(0) = 100 \)
   (b) \( y'(t) - 4y(t) = 0, \quad y(0) = 10 \)

9. Suppose that a population of bacteria is continuously doubling its size every unit of time. Write a differential equation for \( y(t) \), the size of the population.

10. Solve the following second-order differential equations, with given initial values. Use the method based on the characteristic equation (see Example 4).
    (a) \( y''(t) = 5y'(t) - 4y(t), \quad y(0) = 20, \quad y'(0) = 5 \)
    (b) \( y''(t) = -5y'(t) + 6y(t), \quad y(0) = 1, \quad y'(0) = 15 \)
    (c) \( y''(t) = 2y'(t) + 8y(t), \quad y(0) = 2, \quad y'(0) = 0 \)

11. Convert the following differential equations into systems of simultaneous first-order differential equations. Do not solve.
    (a) \( y''(t) = 5y'(t) - 4y(t) \)
    (b) \( y''(t) = -5y'(t) - 6y(t) \)
    (c) \( y'''(t) = 4y''(t) + 3y'(t) - 2y(t) \)
    (d) \( y'''(t) = 2y'(t) + y(t) \)

12. Check that 3 and 1 are the eigenvalues for \( B = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \) in Example 5 and that associated eigenvectors are \([1, -1]\) and \([1, 1]\). Solve the system of differential equations in Example 5 using the method in Example 6.
13. Check that 4 and 2 are the eigenvalues for $B = \begin{bmatrix} 6 & -8 \\ 1 & 0 \end{bmatrix}$ in Example 6 and that associated eigenvectors are [4, 1] and [2, 1]. Also verify (49).

14. Re-solve the second-order differential equations in Exercise 10 by converting them to a pair of first-order differential equations and solving by the method in Example 6 (you must find the eigenvalues and eigenvectors).

15. Solve the following pairs of first-order differential equations by using the solution technique in Example 6 (you must find the eigenvalues and eigenvectors). The initial condition is $x(0) = y(0) = 10$.
   (a) $x'(t) = 4x(t), \quad y'(t) = 2x(t) + 2y(t)$
   (b) $x'(t) = 2x(t) + y(t), \quad y'(t) = 2x(t) + 3y(t)$
   (c) $x'(t) = x(t) + 4y(t), \quad y'(t) = 2x(t) + 3y(t)$

16. (a) Show that any multiple $ry^*(t)$ of a solution $y^*(t)$ to a second-order differential equation $y''(t) = ay'(t) + by(t)$ is again a solution.
   (b) Show that any linear combination $ry^*(t) + sy_0(t)$ of solutions $y^*(t), y_0(t)$ to $y''(t) = ay'(t) + by(t)$ is again a solution.

17. Suppose that $y_0(t)$ is some solution to $y''(t) - ay'(t) - by(t) = f(t)$ and $y^*(t)$ is a solution to $y''(t) - ay'(t) - by(t) = 0$. Then show that for any $r$, $y_0(t) + ry^*(t)$ is also a solution to $y''(t) - ay'(t) - by(t) = f(t)$.

18. Verify that $y(t) = te^{\lambda t}$ is a solution to the differential equation $y''(t) = cy'(t) + dy(t)$ whose characteristic equation $\lambda^2 - \lambda c - d = 0$ has $\lambda$ as its double root.

Note: If $\lambda_1^2 - \lambda_1 c - d = 0$ has $\lambda_1$ as a double root, the characteristic equation can be factored as $(\lambda - \lambda_1)^2 = 0$. This means that $c = 2\lambda_1$ and $d = -\lambda_1^2$. Use these values for $c$ and $d$ in verifying that $te^{\lambda_1 t}$ is a solution.

### Section 4.4 Markov Chains

Markov chains were introduced in Section 1.3. They are probability models for simulating the behavior of a system that randomly moves among different "states" over successive periods of time. If a Markov chain is currently in state $S_i$, there is a transition probability $a_{ij}$ that 1 unit of time later it will be in state $S_j$. The matrix $A$ of transition probabilities completely describes the Markov chain. If $p = [p_1, p_2, \ldots, p_n]$ is the vector giving the probabilities $p_i$ that $S_i$ is the current state of the chain and $p' = [p'_1, p'_2, \ldots, p'_n]$ is the...
vector of probabilities $p'_i$ that $S_i$ is the next state of the chain, then we have

$$p' = Ap$$

(1a)

For a particular $p'_i$, this is

$$p'_i = a_{i1}p_1 + a_{i2}p_2 + \cdots + a_{in}p_n$$

(1b)

**Example 1. Frog in Highway Revisited**

In Section 1.3 we considered the Markov chain for a frog wandering across a highway that was divided into six states. The transition matrix $A$ was

<table>
<thead>
<tr>
<th>Current State</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.50</td>
<td>.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>.50</td>
<td>.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>.25</td>
<td>.50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>.25</td>
<td>.50</td>
<td>.25</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.25</td>
<td>.50</td>
<td>.50</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.25</td>
<td>.50</td>
</tr>
</tbody>
</table>

We started with probability vector $p = [1, 0, 0, 0, 0, 0]$, that is, the frog started in state 1. We computed a table of the probability distributions after varying numbers of periods. In matrix notation, we computed

$$p^{(k)} = A^k p$$

for increasing values of $k$. We found that as $k$ got large, $p^{(k)}$ converged to the probability vector

$$p^* = [.1, .2, .2, .2, .2, .1]$$

(3)

which satisfied the equation

$$p^* = Ap^*$$

(4)

Thus $p^*$ is a stable (unchanging) probability distribution for this Markov chain. In matrix terminology, $p^*$ is an eigenvector of $A$ with associated eigenvalue 1.

In Section 3.5 we solved the eigenvector equations (4)—actually, we solved $(A - I)p = 0$—and obtained a general solution of the form

$$[q, 2q, 2q, 2q, 2q, q]$$

(5)

Making the components in (5) sum to 1 (to be a probability distribu-
tion), we obtained $p^* = (.1, .2, .2, .2, .2, .1)$. So this $p^*$ is the unique stable distribution of the frog Markov chain. The fact was brought out in the Exercises of Section 1.3 that if the starting probability vector $p$ had been different, we still would have found that $p^{(k)}$ converged to this $p^*$.

Suppose that the starting probability vector were the $j$th unit vector $e_j$, with a 1 in the $j$th position and 0’s elsewhere (the original $p$ was $e_j$). It was noted in Section 2.4 that for any matrix $B$,

$$Be_j = b_j^C \quad (b_j^C = \text{the } j\text{th column of } B)$$

Then

$$p^{(k)} = A^k e_j = (A^k)^C_j \quad (\text{the } j\text{th column of } A^k) \quad (6)$$

Since $p^{(k)}$ converges to $p^*$, we conclude that the $j$th column of $A^k$ approaches $p^*$, for $k$ large,

$$A^k \rightarrow \begin{bmatrix} .1 & .1 & .1 & .1 & .1 \\ .2 & .2 & .2 & .2 & .2 \\ .2 & .2 & .2 & .2 & .2 \\ .2 & .2 & .2 & .2 & .2 \\ .1 & .1 & .1 & .1 & .1 \end{bmatrix} \quad (7)$$

Does this property of any starting probability vector converging to a stable probability distribution hold true for all Markov chains? The answer is no.

**Example 2. Markov Chain Not Converging to Stable Distribution**

Consider the simple two-state Markov chain with transition matrix and starting vector

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad p = [1, 0] \quad (8)$$

It is easy to check that $p^{(k)} = [0, 1]$ for $k$ odd, and $p^{(k)} = [1, 0]$ for $k$ even. More generally, for a starting vector of $p = [r, 1 - r]$ for any $r$, $0 \leq r \leq 1$, we have $p^{(k)} = [1 - r, r]$ for $k$ odd, and $p^{(k)} = [r, 1 - r]$ for $k$ even. Note that $p^0 = [.5, .5]$ is a stable vector (an eigenvector with eigenvalue 1), but this Markov chain will not converge to $p^0$ if we do not start at $p^0$, we never get to $p^0$.

The powers of $A$ have a similar odd–even cyclic pattern, with $A^k = A$ for $k$ odd, and $A^k = I$ for $k$ even.
So now the question is: Under what conditions does a Markov transition matrix \( A \) have a stable probability vector to which any starting vector will converge, or equivalently, when will the columns in powers of \( A \) all converge to a given probability vector as in (7)? The question is not tied to the existence of an eigenvector with eigenvalue 1, since in Example 2, \([.5, .5]\) was such an eigenvector, but there was no convergence. Instead, the answer depends on the absence of any cyclic or other nonrandom pattern, as seen in Example 2.

**Definition.** A Markov chain with transition matrix \( A \) is **regular** if for some positive integer \( h \), the matrix \( A^h \) has all positive entries.

The matrix in Example 2 was not regular. A regular Markov chain mixes, or randomizes, patterns so as to eliminate any cyclic behavior. If a Markov chain is regular, then every column of \( A^h \) has all positive entries, meaning that starting from state \( j \) it is possible after \( h \) periods to be in any of the states. The following theorem requires a lengthy, but not advanced, proof that may be found in any of the texts on Markov chains listed in the References.

**Theorem 1.** Every regular Markov chain with transition matrix \( A \) has a stable probability vector \( p^* \) to which \( p^{(k)} = A^k p \) converges, for any probability vector \( p \). All the columns of \( A^k \) also converge to \( p^* \).

One way to find the stable distribution of a regular Markov chain is, as done in Section 3.5, by solving \((A - I)p = 0\) and then picking the constant in the solution to make the components sum to 1 [see equation (5)]. Another approach is to add the additional constraint \( 1 \cdot p \) (\( = \sum p_i \)) = 1.

\[
(A - I)p = 0 \\
1 \cdot p = 1
\]

This is a set of \( n + 1 \) equations in \( n \) unknowns.

**Example 3. Solving for Stable Distribution**

Consider a simpler Markov chain which involves just two states that represent two islands, isle 1 and isle 2, in an isolated country. We are interested in the flow of money between these two islands. Assume that no money enters or leaves the country. Then a Markov chain should provide a reasonable model for currency flow. We shall perform a general analysis of this model rather than use specific values for the transition probabilities \( a_{ij} \). Since columns must sum to 1, the transition matrix \( A \) can be written in terms of the off-diagonal entries thus:

\[
A = \begin{bmatrix}
1 - b & a \\
b & 1 - a
\end{bmatrix}
\]

Current State

\[
\begin{array}{c|cc}
& 1 & 2 \\
\hline
1 & 1 - b & a \\
2 & b & 1 - a
\end{array}
\]
Sec. 4.4 Markov Chains

We require $0 < a, b < 1$ so that the Markov chain will be regular. Let us solve (9) for this $A$.

\[(A - I)p = 0: \quad -bp_1 + ap_2 = 0 \quad bp_1 - ap_2 = 0\]

\[1 \cdot p = 1: \quad p_1 + p_2 = 1\] (10)

The second equation in (10) is just the first equation multiplied by $-1$. So the second equation is redundant, and we are back to the standard situation of two equations in two unknowns. When solved, they yield the stable distribution

\[p_1 = \frac{a}{a + b}, \quad p_2 = \frac{b}{a + b}\] (11)

Note that (11) will always be well defined unless $a$ and $b$ are both 0, in which case we get the trivial Markov chain: $p_1' = p_1, p_2' = p_2$.

For any Markov transition matrix, the system $(A - I)p = 0$ always has redundancy because the sum of the right-hand sides of all the equations is 0 (A has column sums of 1, but the $-I$ term makes the column sums of $A - I$ equal to 0) or, equivalently, the last row is minus the sum of all the preceding rows. When such redundancy exists, the last row will be zeroed out in Gaussian elimination (the reasons for this are discussed in Section 5.2). Thus the last row can be replaced by the constraint $1 \cdot p = 1$, as implicitly happened in Example 3.

However, Gaussian elimination is so simple in tridiagonal systems, like the frog Markov chain, that adding this new row creates as much trouble as it saves. We illustrate the advantage of a tridiagonal matrix with the following large-scale example.

### Example 4. An $n$-State Frog Markov Chain

Let us generalize the frog Markov chain to a chain with $n$ states, where $n$ is an arbitrary number. The system of equations $(A - I)p = 0$ is

\[\begin{align*}
-0.50p_1 + 0.25p_2 &= 0 \\
0.50p_1 - 0.50p_2 + 0.25p_3 &= 0 \\
0.25p_2 - 0.50p_3 + 0.25p_4 &= 0 \\
\vdots & \quad \vdots \\
0.25p_{n-3} - 0.50p_{n-2} + 0.25p_{n-1} &= 0 \\
0.25p_{n-2} - 0.50p_{n-1} + 0.50p_n &= 0 \\
0.25p_{n-1} - 0.50p_n &= 0
\end{align*}\] (12)
Performing elimination in the first column requires us simply to add the first equation to the second.

\[-.50p_1 + .25p_2 \quad = 0\]
\[-.50p_2 + .25p_3 \quad = 0\]
\[-.25p_2 - .50p_3 + .25p_4 \quad = 0\]

\[.25p_2 - .50p_3 + .25p_4 \quad = 0\]

\[.25p_{n-3} - .50p_{n-2} + .25p_{n-1} \quad = 0\]
\[.25p_{n-2} - .50p_{n-1} + .50p_n = 0\]
\[.25p_{n-1} - .50p_n = 0\]

Similarly, for elimination in the second column we add the second equation to the third.

\[-.50p_1 + .25p_2 \quad = 0\]
\[-.25p_2 + .25p_3 \quad = 0\]
\[-.25p_3 + .25p_4 \quad = 0\]

\[-.25p_3 - .50p_4 + .25p_5 \quad = 0\]

\[.25p_{n-3} - .50p_{n-2} + .25p_{n-1} \quad = 0\]
\[.25p_{n-2} - .50p_{n-1} + .50p_n = 0\]
\[.25p_{n-1} - .50p_n = 0\]

The situation when we come to perform elimination in the third column is the same as in the second column and again involves adding the third equation to the fourth. This situation will stay the same for every column from the second through the \((n - 1)\)st. After elimination in the first \(n - 1\) columns, we have

\[-.50p_1 + .25p_2 \quad = 0\]
\[-.50p_2 + .25p_3 \quad = 0\]
\[-.25p_2 + .25p_3 \quad = 0\]

\[-.25p_3 + .25p_4 \quad = 0\]

\[.25p_{n-3} - .50p_{n-2} + .25p_{n-1} \quad = 0\]
\[.25p_{n-2} - .50p_{n-1} + .50p_n = 0\]
\[.25p_{n-1} - .50p_n = 0\]

\[-.25p_{n-2} + .25p_n = 0\]
\[-.25p_{n-1} + .50p_n = 0\]
\[0 \quad = 0\]

\[(13)\]

Note that the \((n - 1)\)st equation in \((13)\) is the negative of the original last equation, so the last equation is zeroed out when we perform elimination in the \((n - 1)\)st column. For concreteness, the reader may
want to refer back to Example 1 of Section 3.5, where we solved this system for the original frog Markov chain, where \( n = 6 \).

Letting \( p_n = q \), we perform back substitution and find from the \((n - 1)\)st equation in (13) that \( .25p_{n-1} = .50p_n (= .50q) \), so \( p_{n-1} = 2q \). Equations 2 through \( n - 2 \) in (13) say that successive \( p_i, p_{i+1} \) pairs from \( p_2 \) through \( p_{n-1} \) are equal. Since \( p_{n-1} = 2q \), all these \( p_i \)'s equal \( 2q \). Finally, we see that \( p_1 = q \). So our solution has the form

\[
[q, 2q, 2q, \ldots, 2q, 2q, q]
\]

The sum of the entries in this general solution is \((2n - 2)q\). For this sum to equal 1, we require that \( q = 1/(2n - 2) \). So our stable distribution is

\[
p^* = \begin{bmatrix}
1 & 2 & 2 & \cdots & 2 & 2 & 1 \\
2n - 2 & 2n - 2 & 2n - 2 & \cdots & 2n - 2 & 2n - 2 & 2n - 2
\end{bmatrix}
\]

The effect of replacing the last equation in (12) by \( 1 \cdot p = 1 \) is discussed in the Exercises.

Next we consider an important type of nonregular Markov chain, called an absorbing Markov chain. A state \( S_i \) in a Markov chain is called an \textbf{absorbing state} if \( a_{ii} = 1 \), that is, once you enter state \( S_i \), you never leave it. A Markov chain with one or more absorbing states is called an \textbf{absorbing Markov chain}.

**Example 5. A Gambling Model with Absorbing States**

Absorbing states complicate the behavior of a Markov chain and lead to a variety of different stable probabilities. Consider the following Markov chain for gambling, with states representing the gambler's winnings. Each round, the gambler has a probability .3 of winning $1, .33 of losing $1, and .37 of staying the same. The gambler stops if he or she loses all of the money, and also stops if the winnings reach $6. So 0 and 6 will be absorbing states in this Markov chain.
After playing a very long time, a person is certain to have stopped, either having gone broke or having won. Thus the stable probabilities should only involve the absorbing states 0 and 6, that is, the stable vector \( \mathbf{p}^* \) has the form

\[
P_0^* = p, \quad P_6^* = 1 - p, \quad \text{and} \quad P_1^* = P_2^* = P_3^* = P_4^* = P_5^* = 0
\]  

(15)

By looking at the transition probabilities for states 0 and 6 alone,

\[
\begin{array}{c|cc}
\text{Current} & \text{Next} & \text{State} \\
0 & 1 & 0 \\
6 & 0 & 1 \\
\end{array}
\]

it is easy to see that \textit{any} \( \mathbf{p}^* \) of the form in (15), with \( 0 \leq p \leq 1 \), is a stable vector.

Before doing any mathematical analysis of absorbing Markov chains, let us explore the behavior of (14) by letting a computer pro-

\[\textbf{Table 4.2}
\begin{array}{c|cccccc}
\text{Rounds} & 0 & 1 & 2 & 3 & 4 & 5 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & .33 & .37 & .30 & 0 \\
2 & 0 & .109 & .244 & .335 & .222 & .09 \\
3 & .036 & .121 & .234 & .357 & .212 & .100 \\
4 & .076 & .128 & .212 & .240 & .193 & .101 \\
5 & .116 & .115 & .194 & .216 & .177 & .095 \\
6 & .154 & .107 & .178 & .196 & .161 & .088 \\
8 & .222 & .090 & .149 & .164 & .135 & .074 \\
10 & .278 & .075 & .125 & .137 & .113 & .062 \\
15 & .383 & .048 & .080 & .088 & .072 & .040 \\
20 & .451 & .031 & .051 & .056 & .047 & .026 \\
25 & .494 & .020 & .033 & .036 & .030 & .016 \\
50 & .563 & .002 & .003 & .004 & .003 & .002 \\
75 & .570 & ~0 & ~0 & ~0 & ~0 & ~0 \\
100 & .571 & ~0 & ~0 & ~0 & ~0 & ~0 \\
\end{array}
\]
gram produce a table of probability distributions over many rounds when we start with $3. Since $3 is halfway between losing and winning, we could expect the chances of losing to be close to .5, but a little above .5, since there is always .33-versus-.30 bias toward losing a dollar in any state.

So if we start with $3, the probability of eventually losing (before we reach $6) is $p = .571$. Instead of repeating this computer simulation for other starting values, we shall now develop a theory that lets us calculate directly the probability of losing or winning, when we start with different amounts of money.

The first step in our development is to divide the states of an absorbing Markov chain into two groups, the absorbing states and the nonabsorbing states. Assume that there are $r$ absorbing states and $s$ nonabsorbing states. If the absorbing states are listed first, the transition matrix $A$ can be partitioned into the form

$$A = \begin{bmatrix} Ab & NAb \\ Ab & NAb \end{bmatrix}$$

where $I$ is an $r$-by-$r$ identity matrix, $O$ is an $s$-by-$r$ matrix of 0’s, $R$ is an $r$-by-$s$ matrix with entry $r_{ij}$ giving the probability of going from nonabsorbing state $j$ to absorbing state $i$, and $Q$ is the $s$-by-$s$ transition matrix among the nonabsorbing states. The transition matrix (14) in Example 5 becomes

$$A = \begin{bmatrix} 0 & 6 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & .33 & 0 & 0 & 0 & 0 \\ 6 & 0 & 1 & 0 & 0 & 0 & .30 \\ 1 & 0 & 0 & .37 & .33 & 0 & 0 \\ 2 & 0 & 0 & .30 & .37 & .33 & 0 \\ 3 & 0 & 0 & 0 & .30 & .37 & .33 \\ 4 & 0 & 0 & 0 & 0 & .30 & .37 \\ 5 & 0 & 0 & 0 & 0 & 0 & .30 & .37 \end{bmatrix}$$

Using the rule for matrix multiplication of a partitioned matrix from Section 2.6 (just treat the submatrices like individual entries), we have

$$A^2 = \begin{bmatrix} I & R \\ O & Q \end{bmatrix} \begin{bmatrix} I & R \\ O & Q \end{bmatrix} = \begin{bmatrix} I \phantom{+} + \phantom{0} RO & IR + \phantom{0} RQ \\ IO + \phantom{0} OQ & OR + \phantom{0} QQ \end{bmatrix}$$

and multiplying $A$ times $A^2$, we find that
It is left as an exercise to check that for higher powers of \( A \), the partitioned form is

\[
A^k = \begin{bmatrix}
I & R^* \\
O & Q^k
\end{bmatrix}.
\]

where \( R^*_k = R + RQ + RQ^2 + \cdots + RQ^{k-1} \)

Note that \( Q^k \) is the standard \( k \)th power of the nonabsorbing transition matrix \( Q \), for play among the nonabsorbing states.

As \( k \) gets large, the entries in \( Q^k \) will approach 0, since over time the probability of not getting absorbed approaches 0. The important submatrix in (19) is \( R^*_k \). Entry \((i, j)\) of \( R^*_k \) is the probability of being in absorbing state \( i \) after \( k \) rounds if we start in nonabsorbing state \( j \). Let us explain what this probability is in detail. To go from nonabsorbing state \( j \) to absorbing state \( i \) after \( k \) rounds, we can either go immediately on the first round from \( j \) to \( i \)—with probability \( r_{ij} \)—(and remain in absorbing state \( i \)), or we can wander among the nonabsorbing states for several rounds, ending up after \( w \) rounds in nonabsorbing state \( h \)—with probability given by entry \((j, h)\) in \( Q^w \)—and then go from state \( h \) to absorbing state \( i \)—with probability \( r_{ih} \) (and thereafter remaining in state \( i \)). The total probability of starting in a nonabsorbing state \( j \), wandering among nonabsorbing states for \( w \) rounds, and then going from some nonabsorbing state to absorbing state \( i \) is given by entry \((i, j)\) in \( RQ^w \). Since the number \( w \) can range up to \( k - 1 \), we obtain the sum for \( R^*_k \) given in (19).

The limiting matrix \( R^* \) for \( R^*_k \) as \( k \) approaches infinity will give the probabilities \( r^*_{ij} \) that starting in nonabsorbing state \( j \) we eventually end up in absorbing state \( i \).

\[
R^* = R + RQ + RQ^2 + \cdots = R(I + Q + Q^2 + \cdots) \quad (20)
\]

\( R^* \) is the matrix that would tell us in the gambling model the probability of eventually losing or winning, when we start with different amounts.

There are two ways to compute \( R^* \). The first way is to compute \( Q^k \) for all \( k \) up to some large number, say 50, and sum these matrices and multiply by \( R \) to obtain \( R^* \) as in (20). The other way rewrites \( R^* \) as

\[
R^* = R(I + Q + Q^2 + \cdots) = R(I - Q)^{-1} \quad (21)
\]

using the geometric series identity introduced in equation (7) of Section 3.4.
We call \((I - Q)^{-1}\) the fundamental matrix of an absorbing Markov chain and use the matrix \(N\) to denote it.

\[
N = I + Q + Q^2 + \cdots = (I - Q)^{-1}
\] and \(R^* = RN\) (22)

We can calculate this inverse using elimination by pivoting. \(N\) is given in (24).

The geometric series identity used in (21) required that \(\|Q\| < 1\) for some matrix norm. The sum norm \(\|Q\|_s\) (largest column sum) of \(Q\) in (17) is 1, and the max norm is \(> 1\). However, \(\|Q\| < 1\) in the euclidean norm.

The matrix \(N\) contains some very useful information by itself. It tells us the expected number of times we will visit (nonabsorbing) state \(i\) if we start in (nonabsorbing) state \(j\). The reasoning is as follows. The average number of times we visit state \(i\) starting from state \(j\) after exactly one round is simply \(0(1 - q_{ij}) + 1q_{ij} = q_{ij}\)—the weighted average of visiting state \(i\) zero times and of visiting state \(i\) one time. The average number of times we visit state \(i\) starting from state \(j\) after exactly two rounds is \(0(1 - q_{ij}^2) + 1q_{ij}^2 = q_{ij}^2\), where \(q_{ij}^2\) denotes entry \((i, j)\) in \(Q^2\). The average number of visits after exactly \(k\) rounds is entry \((i, j)\) in \(Q^k\). Probability theory states that the average number of visits from state \(j\) to state \(i\) totaled over all rounds is simply the sum of the average number of visits on each specific round. So the expected number of times we visit nonabsorbing state \(i\) starting from nonabsorbing state \(j\) is the sum of the \((i, j)\) entries in \(Q^k\) for all \(Q^k\), that is, entry \((i, j)\) in \(N\).

Furthermore, if we sum the entries of the \(j\)th column of \(N\)—the expected number of times, starting from state \(j\), that we visit state \(1\) plus the expected number of times we visit state \(2\), and so on—we obtain the expected number of rounds until we are absorbed. The vector-matrix product \(1N\) computes the sum of each column of \(N\).

We summarize this wealth of information about absorbing Markov chains we can get from \(N\) with the following theorem. The term absorption is used in this theorem to mean going to an absorbing state.

**Theorem 2.** Let \(N\) be the fundamental matrix of an absorbing Markov chain [\(N\) is defined in (22)]. Then the following are true.

(i) Entry \(n_{ij}\) of \(N\) is the expected number of times we visit the nonabsorbing state \(i\) (before absorption) when we start in nonabsorbing state \(j\).

(ii) The \(j\)th entry in the vector \(1N\) gives the expected number of rounds before absorption when we start in nonabsorbing state \(j\).

(iii) Entry \((i, j)\) in \(RN\) is the probability of eventually ending up in absorbing state \(i\) when we start in nonabsorbing state \(j\).

**Example 5 (continued). A Gambling Model.**

With Theorem 2 we can answer a variety of interesting questions about this model. We must compute the matrix \(N\) by finding the inverse of \(I - Q\), where \(Q\) is
Ch. 4 A Sampling of Linear Models

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & .37 & .33 & 0 & 0 \\
2 & .30 & .37 & .33 & 0 \\
\end{bmatrix}
\]

\[Q = \begin{bmatrix}
3 & 0 & .30 & .37 & .33 & 0 \\
4 & 0 & 0 & .30 & .37 & .33 \\
5 & 0 & 0 & 0 & .30 & .37 \\
\end{bmatrix}
\]  \hspace{1cm} (23)

Using elimination by pivoting (described in Section 3.3), we obtain

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & 2.64 & 2.21 & 1.73 & 1.21 & .63 \\
2 & 2.01 & 4.21 & 3.30 & 2.31 & 1.21 \\
\end{bmatrix}
\]

\[N = (I - Q)^{-1} = \begin{bmatrix}
3 & 1.43 & 3.00 & 4.73 & 3.30 & 1.73 \\
4 & .91 & 1.91 & 3.00 & 4.21 & 2.21 \\
5 & .43 & .91 & 1.43 & 2.01 & 2.64 \\
\end{bmatrix}
\]  \hspace{1cm} (24)

From (24) and Theorem 2, we see that if we started with $3, there would be 3.3 rounds during an average gambling session when we would be in state 2 (when we would have $2).

Next we sum the columns of $N$:

\[1N = [7.42, 12.24, 14.19, 13.04, 8.42] \hspace{1cm} (25)
\]

The third entry in (25) tells us that if we start with $3, we get to play about 14 rounds, on average, before the game ends.

Finally, we compute $R^* = RN$, where we see from (17) that $R$ is

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 \\
0 & .33 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[R^* = RN = \begin{bmatrix}
0 & .87 & .73 & .57 & .40 & .21 \\
6 & .13 & .27 & .43 & .60 & .79 \\
\end{bmatrix}
\]  \hspace{1cm} (26)

Entry (0, 3) of $R^*$ confirms our earlier simulation result that the probability of going broke when we start with $3 is .57.

We close this section by noting that some of these results about absorbing Markov chains can be applied to regular Markov chains with the following trick. Let $A$ be the transition matrix of a regular Markov chain. We convert one state, say state $p$, into an absorbing state by replacing the $p$th column $a_p^c$ of $A$ by the $p$th unit vector $e_p$. Now whenever we come to state $p$, we stay there. Our theory of absorbing Markov chains can be applied
to this modified transition matrix to determine the expected number of rounds it takes to get to state $p$ if we start from any other state $j$ (and also the expected number of visits in any third state on the journey from $j$ to $p$).

If we convert two states $p$ and $r$ into absorbing states with unit vector columns, then we compute the relative probability of reaching $p$ or $r$ first when we start from some other state $j$. These calculations are requested for Example 1 in the Exercises.

**Example 6. Ice Cream Selection**

We surveyed a group of students eating blueberry, mint, and strawberry ice cream about which flavor they would choose next time. Suppose that $\frac{3}{4}$ of those eating blueberry would choose blueberry the next time, while the remaining quarter would choose strawberry. Responses from others yielded the following transition matrix:

<table>
<thead>
<tr>
<th>Current Flavor</th>
<th>Blueberry</th>
<th>Mint</th>
<th>Strawberry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blueberry</td>
<td>$\frac{3}{4}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>Mint</td>
<td>0</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{3}$</td>
</tr>
<tr>
<td>Strawberry</td>
<td>$\frac{1}{4}$</td>
<td>0</td>
<td>$\frac{2}{3}$</td>
</tr>
</tbody>
</table>

We treat the selection of flavors as a Markov process and pose the question: How many rounds does it take on average for a person to switch from strawberry to blueberry?

To answer this question, we change the transition matrix in (27) by making blueberry an absorbing state. The modified transition matrix is

$$A = \begin{bmatrix} b & m & s \\ b & 1 & \frac{1}{2} & 0 \\ m & 0 & \frac{1}{2} & \frac{1}{3} \\ s & 0 & 0 & \frac{2}{3} \end{bmatrix} = \begin{bmatrix} I & R \\ 0 & Q \end{bmatrix}$$

with $Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{2}{3} \end{bmatrix}$ (28)

Then we find that

$$N = (I - Q)^{-1} = \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{3} \end{bmatrix}^{-1} = \begin{bmatrix} 2 & 2 \\ 0 & 3 \end{bmatrix}$$

and

$$1N = [2, 5]$$

By Theorem 2, part (ii), the second entry, 5, in $1N$ is the average number of rounds until absorption (blueberry) when starting from strawberry. Moreover, by Theorem 2, part (i), the second column of
N tells us that on average a person starting with strawberry would choose strawberry three times (including the initial time) and choose mint twice before choosing blueberry.

With modest effort (see any textbook on Markov chains in the References), one can also prove the following interesting result.

**Theorem 3.** Let \( p^* = [p_1^*, p_2^*, \ldots, p_k^*] \) be the stable probability vector of a regular Markov chain. Then, if we start in state \( i \), the expected number of rounds before we return to \( i \) again is \( 1/p_i^* \).

### Section 4.4 Exercises

**Summary of Exercises**


1. Describe the behavior of the Markov chain

\[
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
\]

with starting vector \([1, 0, 0]\). Are there any stable vectors?

2. Which of the following transition matrices belong to regular Markov chains? Find a stable distribution for each chain.

(a) \( \begin{bmatrix} 0 & \frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix} \)  
(b) \( \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{bmatrix} \)  
(c) \( \begin{bmatrix} \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \)

3. Compute the stable distribution for the weather Markov chain introduced in Section 1.3 with transition matrix

\[
\begin{array}{c|cc}
& Sunny & Cloudy \\
\hline
Sunny & \frac{3}{4} & \frac{1}{2} \\
Cloudy & \frac{1}{4} & \frac{1}{2}
\end{array}
\]

4. The printing press in a newspaper has the following pattern of breakdowns. If it is working today, tomorrow it has 90% chance of working (and 10% chance of breaking down). If the press is broken today, it has a 60% chance of working tomorrow (and 40% chance by being broken again). Compute the stable distribution for this Markov chain.
5. If the local professional basketball team, the Sneakers, wins today's game, they have a $\frac{3}{2}$ chance of winning their next game. If they lose this game, they have a $\frac{1}{2}$ chance of winning their next game. Compute the stable distribution for this Markov chain and give the approximate values of entries in $A^{100}$, where $A$ is this Markov chain's transition matrix.

6. If the stock market went up today, historical data show that it has a 60% chance of going up tomorrow, a 20% chance of staying the same, and a 20% chance of going down. If the market was unchanged today, it has a 20% chance of being unchanged tomorrow, a 40% chance of going up, and a 40% chance of going down. If the market goes down today, it has a 20% of going up tomorrow, a 20% chance of being unchanged, and a 60% chance of going down. Compute the stable distribution for the stock market.

7. Write down a Markov chain to model the following situation: Assume that there are three types of voters in Texas: Republicans, Democrats, and Independent. From one (national) election to the next, 60% of Republicans remain Republican and similarly for the two other groups; among the 40% who change parties, 30% become Independent and 10% go to the other major party, except that the Independents who change all become Republicans. Determine the stable distribution among the three parties and from it give the approximate values of entries in $A^{1000}$.

8. (a) Make a Markov chain model for a rat wandering through the following maze if, at the end of each period, the rat is equally likely to leave its current room through any of the doorways. (It never stays where it is.)

(b) What is the stable distribution?

9. Repeat the questions in Exercise 8 for the following maze.
10. Find the stable distribution for Markov chains with the following transition matrices.

\[
\begin{bmatrix}
\frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\
\frac{2}{3} & \frac{1}{3} & 0 & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
0 & 0 & \frac{1}{3} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & \frac{1}{3} \\
\end{bmatrix}
\quad \text{(a)}
\]

\[
\begin{bmatrix}
\frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\
0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & \frac{1}{3} \\
\end{bmatrix}
\quad \text{(b)}
\]

\[
\begin{bmatrix}
\frac{2}{3} & \frac{2}{3} & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\
0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\
0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\
0 & 0 & 0 & 0 & \frac{1}{3} \\
\end{bmatrix}
\quad \text{(c)}
\]

\[
\begin{bmatrix}
0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\
1 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\
0 & 0 & 0 & \frac{1}{2} & 0 & 1 \\
0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\
\end{bmatrix}
\quad \text{(d)}
\]

11. Repeat Exercise 10, parts (a) and (d) with the number of states expanded from six to \( n \), as done in Example 4.

12. Determine the two eigenvalues and associated eigenvectors for the Markov chain in the following exercises. Give the distribution after six periods for the given starting distribution \( \mathbf{p} \) by representing \( \mathbf{p} \) as a linear combination of the eigenvectors as was done in the end of Section 3.1.

(a) Exercise 3, starting \( \mathbf{p} \): Sunny 0, Cloudy 1.

(b) Exercise 4, starting \( \mathbf{p} \): Working 1, Broken 0.

(c) Exercise 5, starting \( \mathbf{p} \): Winning 1/2, Losing 1/2.

13. Show that if \( A \) is a tridiagonal Markov transition matrix, then in solving \( (A - I)\mathbf{p} = 0 \) by Gaussian elimination, the \( L \) in \( LU \) decomposition of \( A \) is a matrix with 1's on the main diagonal and -1 just below the diagonal entries. That is, in Gaussian elimination one always adds the current row (times 1) to the next row.

14. Re-solve the stable distribution problems in Exercise 10 with the last row of the matrix equation \( (A - I)\mathbf{p} = 0 \) replaced by the constraint \( \mathbf{1} \cdot \mathbf{p} = 1 \) (the last row always drops out—becomes 0—and the additional constraint that the probabilities sum to 1 can be put in its place).

15. The following questions refer to the gambling Markov chain in Example 5. If you started with $4, what is the expected number of rounds that you have $3, and what is the expected number of rounds until the game ends?

16. The following model for learning a concept over a set of lessons identifies four states of learning: \( I = \) ignorance, \( E = \) exploratory thinking, \( S = \) superficial understanding, and \( M = \) mastery. If now in state \( I \),
after one lesson you have $\frac{1}{2}$ probability of still being in $I$ and $\frac{1}{2}$ probability of being in $E$. If now in state $E$, you have $\frac{1}{4}$ probability of being in $I$, $\frac{1}{3}$ in $E$, and $\frac{1}{6}$ in $S$. If now in state $S$, you have $\frac{1}{4}$ probability of being in $E$, $\frac{1}{3}$ in $S$, and $\frac{1}{6}$ in $M$. If in $M$, you always stay in $M$ (with probability 1).

(a) Write out the transition matrix for this Markov chain with the absorbing state as the first state.
(b) Compute the fundamental matrix $N$.
(c) What is the expected number of rounds until mastery is attained if currently in the state of ignorance?

17. In the following maze suppose that a rat has a 20% chance of going into the middle room, which is an absorbing state, and a 40% chance each of going to the room on the left or on the right.

(a) What is the expected number of times a rat starting in room 1 enters room 2?
(b) What is the expected number of rounds until the rat goes to the middle room?

18. (a) Make a Markov chain model for a rat wandering through the following maze if, at the end of each period, the rat is equally likely to leave its current room through any of the doorways. The center room is an absorbing state. (It never stays in the same room.)

(b) If the rat starts in room 4, what is the expected number of times it will be in room 2?
(c) If the rat starts in room 4, what is the expected rounds until absorption?
19. Consider the game of Ping-Pong with the following states:

- A: Player 1 is hitting the ball.
- B: Player 2 is hitting the ball.
- C: Play is dead because 1 hit the ball out or in the net.
- D: Play is dead because 2 hit the ball out or in the net.

The transition matrix is

\[
\begin{bmatrix}
1 & 2 & 3 & 4 \\
A \text{ hitting ball} & 1 & .9 & 0 & 0 \\
B \text{ hitting ball} & 2 & .4 & 0 & 0 \\
A \text{ hit ball out} & 3 & 0 & 1 & 0 \\
B \text{ hit ball out} & 4 & 0 & .1 & 1
\end{bmatrix}
\]

If we start play with player A hitting the ball (in state 1)

(a) What is the expected number of times player A hits the ball (before the point is over)?
(b) What is the expected number of hits by A and B (before the point is over)?
(c) What is the probability that player A hits the ball out (i.e., that player B wins the point)?

20. Repeat Exercise 18 for the following maze, in which rooms 1 and 5 are absorbing. Start in room 2.

21. Repeat the Markov chain model of a poker game given in Example 5 but now with probability \( \frac{1}{3} \) that a player wins 1 dollar in a period, with probability \( \frac{1}{3} \) a player loses 1 dollar, and with probability \( \frac{1}{3} \) a player stays the same. The game ends if the player loses all his or her money or if the player has 6 dollars. Compute \( N \), \( lN \), and \( RN \) for this problem.

22. Three tanks A, B, and C are engaged in a battle. Tank A, when it fires, hits its target with hit probability \( \frac{1}{3} \). B hits its target with hit probability \( \frac{1}{3} \), and C with hit probability \( \frac{1}{3} \). Initially (in the first period), B and C fire at A and A fires at B. Once one tank is hit, the remaining tanks aim at each other. The battle ends when there is one or no tank left. The transition matrix for this game is (the states are the subsets of tanks surviving)
Sec. 4.5 Growth Models

In this section we examine three models for growing populations. We have already seen a simple linear model, introduced in Section 1.3, for the growth of two competing species, rabbits and foxes. Here we will study models for the growth of one species that is subdivided into different age groups. For simplicity we again let rabbits be the object of study in the models. However, our models apply to any renewable natural resource, from animals to forests, and to many human enterprises, be they economic or social. The first model has been applied to human populations to predict population cycles and to set insurance rates.

Example 1. Age-Specific Population Model

We want a model that breaks down a population into different age groups. Human population models commonly have about 20 age groups, with each age group spanning 5 years, plus a special group for the first year of life (since mortality rates for newborns are different from other young children) and a last group consisting of everyone past some advanced age, say 90 years. Each age group is really two
groups, one for men and one for women. The study of the sizes of human populations is called demography.

To make matters simple, we will content ourselves here with a three-age-group model of rabbits.

\[
y = \text{young rabbits, up to 2 years old} \\
m = \text{midlife rabbits, between 2 and 4 years old} \\
o = \text{old rabbits, 4 to 6 years old}
\]

Let one period of time equal 2 years. Our model for the next period’s population vector \( a' = [y', m', o'] \) in terms of the current population \( a = [y, m, o] \) is

\[
\begin{align*}
y' &= 4m + o \\
m' &= .4y \\
o' &= .6m
\end{align*}
\]

The first equation in (1) says that each midlife rabbit gives birth to 4 young each period and that each old rabbit gives birth to 1 young each period (of course, only females have babies, but in this initial model we are not differentiating between sexes). The second equation says that 40% of all young rabbits survive through their first 2 years (one period). The third equation says that 60% of midlife rabbits live through a period to become old rabbits. Finally, assume that all old rabbits die within 2 years. If \( L \) is the matrix of coefficients in (1),

\[
L = \begin{bmatrix}
0 & 4 & 1 \\
.4 & 0 & 0 \\
0 & .6 & 0
\end{bmatrix}
\]

then (1) has the matrix form

\[
a' = La
\]

This population model is called a **Leslie model**. If there were more age groups, the matrix \( L \) of coefficients would have the form

\[
L = \begin{bmatrix}
0 & b_2 & b_3 & b_4 & \cdots & b_n \\
p_1 & 0 & 0 & 0 & \cdots & 0 \\
0 & p_2 & 0 & 0 & \cdots & 0 \\
0 & 0 & p_3 & 0 & \cdots & 0 \\
0 & 0 & 0 & p_4 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & p_{n-1} \\
& & & & & 0
\end{bmatrix}
\]
where $b_i$ is the number of offspring per individual in group $i$ and $p_i$ is the probability that an individual in group $i$ survives one period to become a member of group $i + 1$.

The model is somewhat like a Markov chain, except that the numbers $b_i$ are not probabilities. Rather than summing to 1, the three variables $y$, $m$, and $o$ will grow larger or smaller over time. We want to know the behavior of this model over many periods. Will the total number of rabbits increase or decrease? Will there be any cyclic patterns in the population, such as a surge in the young one year followed a period later by a surge in midlifes, then the next period a surge in the young, continuing back and forth? Or after several periods, will there be a steady distribution of the population; for example, will the fractions of rabbits that are young, are midlife, and are old remain the same from period to period?

The answers to these questions depend on the eigenvalues and eigenvectors of $L$. As we saw in Sections 2.5 and 3.4, the *long-term population distribution* $L^k a$, for large $k$, will be a multiple of the dominant eigenvector of $L$ (the eigenvector associated with the largest eigenvalue), and the long-term growth rate will be the dominant (largest) eigenvalue. Whether the model converges quickly to the dominant eigenvalue depends on how much the largest eigenvalue dominates the second largest eigenvalue.

**Table 4.3**

<table>
<thead>
<tr>
<th>Period</th>
<th>Young</th>
<th>Midlife</th>
<th>Old</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>50</td>
<td>30</td>
<td>180</td>
</tr>
<tr>
<td>1</td>
<td>230</td>
<td>40</td>
<td>30</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>190</td>
<td>92</td>
<td>24</td>
<td>306</td>
</tr>
<tr>
<td>3</td>
<td>392</td>
<td>76</td>
<td>55</td>
<td>523</td>
</tr>
<tr>
<td>4</td>
<td>359</td>
<td>157</td>
<td>47</td>
<td>563</td>
</tr>
<tr>
<td>5</td>
<td>673</td>
<td>144</td>
<td>94</td>
<td>916</td>
</tr>
<tr>
<td>6</td>
<td>669</td>
<td>269</td>
<td>86</td>
<td>1024</td>
</tr>
<tr>
<td>7</td>
<td>1162</td>
<td>266</td>
<td>161</td>
<td>1589</td>
</tr>
<tr>
<td>8</td>
<td>1232</td>
<td>465</td>
<td>160</td>
<td>1857</td>
</tr>
<tr>
<td>9</td>
<td>2021</td>
<td>493</td>
<td>279</td>
<td>2793</td>
</tr>
<tr>
<td>10</td>
<td>2250</td>
<td>808</td>
<td>295</td>
<td>3353</td>
</tr>
<tr>
<td>11</td>
<td>3529</td>
<td>900</td>
<td>485</td>
<td>4914</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>14</td>
<td>7382</td>
<td>2474</td>
<td>980</td>
<td>11836</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>19</td>
<td>33873</td>
<td>9553</td>
<td>4602</td>
<td>48028</td>
</tr>
<tr>
<td>20</td>
<td>42815</td>
<td>13549</td>
<td>5732</td>
<td>62096</td>
</tr>
</tbody>
</table>
To start, let us examine the behavior of our model with a computer simulation. Starting with an initial population of 100 young, 50 midlife, and 30 old, we compute a table of populations in successive periods using (1); we have rounded values to whole numbers in Table 4.3. Initially, we see a very pronounced cycling behavior between young and midlife rabbits, and this in turn leads to an uneven growth in the total population—there is little growth between periods 1 and 2, between 3 and 4, or between 5 and 6. The cycling is much smaller after 20 periods but still present. If we run the model a little longer, we see that the population stabilizes with a distribution and growth multiplier

Long-term distribution: 70% young, 21% midlifes, and 9% old  
Growth multiplier: 1.334 (33.4% growth rate) (5)

That is, the population vector in the next period is about 1.334 times this period's population vector.

Using more advanced techniques introduced in the Appendix to Section 5.5 (or using the appropriate mathematical software), we find that the eigenvalues of \( L \) in decreasing absolute size are

\[
\lambda_1 = 1.334, \quad \lambda_2 = -1.118, \quad \lambda_3 = -0.152
\]

and the dominant eigenvector (associated with \( \lambda_1 \)) is

\[
\mathbf{u}_1 = [0.697, 0.209, 0.094]
\]

The fact that \( \lambda_2 \) is close to \( \lambda_1 \) in absolute size is why the simulation took a long time to stabilize at \( \mathbf{u}_1 \).

In mathematics, the best way to understand a property of interest is often to study cases where the property fails to be true. We shall now take this approach and look at a Leslie model whose group percentages do not converge to the dominant eigenvector.

**Example 2. A Cyclic Leslie Model**

Consider the following Leslie model:

\[
\begin{align*}
y' &= 4o \\
m' &= 0.5y \\
o' &= 0.5m
\end{align*}
\]

with the Leslie matrix

\[
L = \begin{bmatrix}
0 & 0 & 4 \\
0.5 & 0 & 0 \\
0 & 0.5 & 0
\end{bmatrix}
\]
Let the initial population be 100 young and no midlife or old rabbits. Then the next period’s population is easily seen to have just 50 midlifes, the next following period just 25 old rabbits, and the third following period just 100 young—returning to the initial population. This cycle will repeat over and over again, and there will never be a stable distribution of age groups—we have pure cycling. No matter what the starting population is, it will repeat every three periods. Such cycling is very unusual and results from properties of the eigenvalues of this problem.

Let us compute the eigenvalues of $L$ in (7). Recall from Section 3.1 that the eigenvalues $\lambda$ are the zeros of the determinant of the matrix $(L - \lambda I)$. In this example, $\det(L - \lambda I)$ has a simple form that is easy to work with.

$$L - \lambda I = \begin{bmatrix} -\lambda & 0 & 4 \\ .5 & -\lambda & 0 \\ 0 & .5 & -\lambda \end{bmatrix}$$

so

$$\det(L - \lambda I) = (-\lambda)(-\lambda)(-\lambda) + (4)(.5)(.5)$$

$$= -\lambda^3 + 1$$

Although the determinant of a 3-by-3 matrix involves taking six diagonal products (see Section 3.1), the three 0’s in (8) eliminate all but two of these products.

Setting the determinant $-\lambda^3 + 1$ equal to 0, we get

$$\lambda^3 = 1$$

One obvious root of (10) is $\lambda = 1$. But all cubic equations must have three roots. The other two roots, called roots of unity, involve complex numbers. They are $-\frac{1}{2} \pm (\sqrt{3}/2)i$, where $i = \sqrt{-1}$. These complex numbers have absolute value 1 also. So there are three dominant eigenvalues of size 1, instead of a single one as is usually the case. This is why there is not a single dominant long-term effect as occurred in Example 1.

Perpetual cycling occurs if there are several largest eigenvalues (in absolute size). If the largest eigenvalue is complex, we must get cyclic behavior—since complex zeros of a polynomial always come in conjugate pairs ($c + id$ and $c - id$) of the same absolute value. The cyclic Markov chain in Example 2 of Section 4.4 had two largest eigenvalues. Recall that its transition matrix was

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
for which \( \det(A - \lambda I) = \lambda^2 - 1 \). So its eigenvalues were 1 and -1.

The following theorem, whose proof is beyond the scope of this book, gives a condition for assuring noncyclic behavior in a Leslie model.

**Theorem.** A Leslie growth model with matrix \( L \) given in (4) will have a unique largest eigenvalue that is a real number, and hence a stable long-term population distribution, if some consecutive pair \( b_i, b_{i+1} \) of entries in the first row of \( L \) are both positive, that is, if two consecutive age groups give birth to offspring.

The condition in this theorem is satisfied in Example 1 but not in Example 2.

**Example 3. Harvesting a Renewable Resource**

This time we shall grow rabbits for profit, to sell some of the rabbit population every year. The goal will be to determine the proper population size and distribution among age groups so that we can "harvest" a given number of rabbits each year without depleting the population. That is, we want a minimal-size collection of rabbits that will sustain a given harvest forever. This time we shall use a model that differentiates between females and males. We shall again use three different age categories, but now the age spans in the groups will vary. The groups are

- \( bm \) = baby male rabbits (less than 1 year old)
- \( bf \) = baby female rabbits
- \( ym \) = yearling male rabbits (between 1 and 2 years old)
- \( yf \) = yearling female rabbits
- \( am \) = adult male rabbits (2 or more years old)
- \( af \) = adult female rabbits

The time period is 1 year. In this model adults do not all die at the end of one time period but rather survive to the next year with probability .75. We shall only harvest adults, \( hm \) male rabbits harvested, and \( hf \) female rabbits harvested. Let us try using the following equations.

\[
\begin{align*}
    bm' &= 2af \\
    bf' &= 2af \\
    ym' &= .6bm \\
    yf' &= .6bf \\
    am' &= .6ym + .75am - hm \\
    af' &= .6yf + .75af - hf
\end{align*}
\]

If \( A \) is the matrix of coefficients on the right-hand side of (13), ex-
Sec. 4.5 Growth Models

cluding terms \(-hm\) and \(-hf\), \(x\) and \(x'\) are the 6-entry vectors of current and next-period age-group sizes, and \(h\) is the harvesting vector \((0, 0, 0, 0, hm, hf)\), we have

\[
x' = Ax - h
\]  

(14)

For stable harvesting, we seek values for the variables in (12) such that specified amounts can be harvested from each group in a year [in (13) only am and af are harvested] and in the next year all the groups will be the same sizes, ready to be harvested again. Mathematically, this means that in (14), \(x' = x\). So (14) becomes

\[
x = Ax - h
\]

With matrix algebra, we have

\[
(A - I)x = h
\]  

(15)

or

\[
\begin{align*}
-bm & \quad 2af = 0 \\
-bf & \quad 2af = 0 \\
.6bm & - ym \quad = 0 \\
.6bf & - yf \quad = 0 \\
.6ym & - .25am \quad = hm \\
.6yf & - .25af = hf
\end{align*}
\]  

(16)

Let us also compare, in a very general way, the difference between solving this model and the Leslie growth model in Example 1. The Leslie model is a dynamic growing model whose solution involves an eigenvalue problem, while here we have a static model (one period is like the next period) whose solution involves solving a standard system of \(n\) equations in \(n\) unknowns. A Leslie model like (4) in Example 1 will converge to a constant distribution of ages in all but the most exceptional cases, whereas system (15) can easily have no solution of constant population with harvesting. For example, if without harvesting the population naturally decreases (i.e., \(\|A\| < 1\)), then with harvesting it will decrease even faster and eventually become extinct.

We cannot find a solution to (15) by iterating (13) and hoping for the variables to converge after many periods to the stable harvest distribution. The reverse happens: If you do not start with the right population, the populations over successive periods will move farther away from the desired answer. Such divergence means that one has to be very careful about roundoff errors in solving (16).

Let us choose the values \(hm = hf = 100\), and solve (16) by Gaussian elimination. We obtain (with answers rounded to whole numbers)
Observe that while only females produce offspring, it appears that this model really groups males and females together, in that there are the same numbers of males and females in each age group. So far we have seen only that this equality occurs when $h_m = h_f = 100$. To see that it is true for $h_m = h_f = k$, for any $k > 0$, write (16) in matrix form as

$$(A - I)x = kh^*, \quad \text{where } h^* = [0, 0, 0, 0, 100, 100]$$

In (17), we have the solution $x^* = [426, 426, 255, 255, 213, 213]$ to (18) when $k = 1$: that is, $(A - I)x^* = h^*$. Then by linearity,

$$A(kx^*) = k(Ax^*) = kh^*$$

So when $h_m = h_f = 100k$, the stable population in our harvesting model will be $kx^* = [426k, 426k, 255k, 255k, 213k, 213k]$.

We see from (17) that the sex differentiation can be dropped from the original model in (13) when $h_m = h_f$, yielding the simpler model

$$b' = 2a$$

$$y' = .6b$$

$$a' = .6 + .75a + h$$

(20)

where $b = \text{baby rabbits}$, $y = \text{yearlings}$, $a = \text{adults}$, $h = \text{harvest}$. It was not at all obvious in advance that these two models would be equivalent.

Suppose that $h_m \neq h_f$. If we harvest twice as many of one sex as of the other, we obtain the results shown in Table 4.4 by re-solving (16) for these new values of $h_m$ and $h_f$.

<table>
<thead>
<tr>
<th>Harvest</th>
<th>Stable Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_m$</td>
<td>$h_f$</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
</tr>
</tbody>
</table>

Surprise! Our model gives us a negative answer when we try to harvest twice as many adult males as females. Why did this happen? What is the smallest ratio of harvested males to females that can occur?

To get a fuller understanding of our harvesting model, let us compute the inverse of the coefficient matrix in (16).
Sec. 4.5 Growth Models

\[
(A - I)^{-1} = \begin{bmatrix}
-1 & 1.53 & 0 & 2.55 & 0 & 4.25 \\
0 & .53 & 0 & 2.55 & 0 & 4.25 \\
-.6 & .91 & -1 & 1.53 & 0 & 2.55 \\
0 & .32 & 0 & .53 & 0 & 2.55 \\
-1.44 & 2.21 & -2.4 & 3.68 & -4 & 6.13 \\
0 & .77 & 0 & 1.28 & 0 & 2.13 
\end{bmatrix}
\]

(21)

Notice how the sparsity of the original coefficient matrix is lost in the inverse. This inverse gives an explicit formula for \( x \) in terms of \( h \) that can be invaluable. For example, if only adult males and females are harvested, then \( h = [0, 0, 0, 0, hm, hf] \) and multiplying \((A - I)^{-1}\) times this \( h \) yields a formula for the stable population vector \( x \).

\[
x = (A - I)^{-1}h
\]

\[= [4.25hf, 4.25hf, 2.55hf, 2.55hf, -4hm + 6.13hf, 2.13hf]
\]

(22)

The form of the solution vector in (22) answers all questions about the behavior of this particular model. In particular, for the fifth component to be \( \geq 0 \), \( hf \) should be at least \( \frac{2}{3} \) of \( hm \).

We now consider a simplified model for rabbit growth that results in a single equation. However, this equation will involve the population in the current period and the previous period.

**Example 4. Recurrence Model for Rabbit Growth**

First we consider an extremely simplified model, in which the population doubles in each successive period. If \( r_n \) is the rabbit population in the \( n \)th period, we have

\[
r_n = 2r_{n-1}
\]

(23)

If we started with \( r_0 = A \) rabbits in period 0, then in period 1 we would have \( r_1 = 2A \) rabbits, and in the next period \( r_2 = 4A \) rabbits. It is not hard to see that the formula for \( r_n \) is

\[
r_n = 2^n A
\]

For the general problem,

\[
r_n = c r_{n-1} \quad \text{has solution} \quad r_n = c^n r_0
\]

(24)

Now let us consider a model with adults and young rabbits. We suppose that once a pair of rabbits are 1 year old, they have one pair of offspring every year for the rest of their lives. Assume that all pairs
consist of one female and one male. We make a two-group mathemat-ical model of the rabbit population.

Let \( m_n \) denote the number of mature pairs (at least 1 year old) during period \( n \), \( y_n \) denote the number of young pairs (under 1 year old) during period \( n \), and \( r_n \) denote the total number of rabbit pairs during period \( n \). If we assume that no rabbits die (the model will not be valid for long periods), then from one period of time to the next, these quantities obey the equations

\[
\begin{align*}
  y_n &= m_{n-1} \\
  m_n &= m_{n-1} + y_{n-1}
\end{align*}
\]  

(25)

and

\[
  r_n = m_n + y_n
\]  

(26)

Observe that \( r_n \) can also be expressed as \( r_{n-1} \) plus the number of new pairs born in the \( n \)th year (i.e., \( y_n \)).

\[
  r_n = r_{n-1} + y_n
\]  

(27)

Comparing the right-hand sides of (26) and (27), we conclude that

\[
  m_n = r_{n-1}
\]  

(28)

In words, we explain (28) by the fact that any rabbit, young or mature, alive one period ago will be an adult this period. And restating (28) for the previous year, we have \( m_{n-1} = r_{n-2} \). When this identity is combined with \( y_n = m_{n-1} \) [equation (25)], we have

\[
  y_n = m_{n-1} = r_{n-2}
\]  

(29)

Substituting (29) in the equation for \( y_n \) in (27), we obtain the following simple relation for \( r_n \):

\[
  r_n = r_{n-1} + r_{n-2}
\]  

(30)

Equations such as (30) that tell how to compute the next number in a sequence \( r_0, r_1, r_2, \ldots \) are called recurrence relations. Equation (30) is called a second-order relation because the right-hand side goes back 2 years. Recurrence relations are the discrete counterpart to differential equations, that is, when time is measured in discrete units rather than continuously.

Equation (30) is called the Fibonacci relation (named after the thirteenth-century Italian mathematician Fibonacci, who first studied this growth model). For example, if we started with one young pair of rabbits (i.e., \( r_0 = 1 \)), after one period we would have one adult pair (\( r_1 = 1 \)) and thereafter we could use (30) to get the following sequence of population sizes:

\[
  r_0, r_1, r_2, \ldots
\]
Sec. 4.5 Growth Models

331

\[ r_2 = 2, \quad r_3 = 3, \quad r_4 = 5, \quad r_5 = 8, \]
\[ r_6 = 13, \quad r_7 = 21, \quad r_8 = 34 \]

and so on. The numbers in this sequence are called the Fibonacci numbers. Fibonacci numbers arise in many settings in nature and mathematics. Perhaps the most famous example is the spiral pattern of leaves around a blossom: On apple or oak stems, there are 5 (\( = r_4 \)) leaves for every two (\( = r_2 \)) spiral turns; on pear stems, 8 (\( = r_5 \)) leaves for every three (\( = r_3 \)) spiral turns; on willow stems, 13 (\( = r_6 \)) leaves for every five (\( = r_4 \)) spiral turns. There are biological reasons involving the Fibonacci relation for these numbers.

The Leslie model in Example 1 was a system of recurrence relations [as was our original model (25)]. That is, (1) could have been written

\[
\begin{align*}
Y_n &= 4m_{n-1} + o_{n-1} \\
m_n &= .4y_{n-1} \\
o_n &= .6m_{n-1}
\end{align*}
\]

Harvesting equations (13) in Example 3 are recurrence relations (but there we are not interested in growth of the model over many periods; instead, we seek a starting distribution which will remain constant with annual harvesting). The transition equations of a Markov chain are also recurrence relations.

[Note that the matrix generalization of the solution for a first-order recurrence relation, given in (24), tells us that the solution of the system \( a_n = L a_{n-1} \) in (31) is

\[ a_n = L^n a_0 \]

Unfortunately, this is not new information.]

Let us return to the second-order recurrence relation for \( r_n \) in (30). The theory of recurrence relations says that any linear recurrence relation for \( r_n \) of the form (where \( k \) and the \( c_i \) are constants)

\[ r_n = c_1 r_{n-1} + c_2 r_{n-2} + \cdots + c_k r_{n-k} \] (32)

has solutions of the form

\[ r_n = b \alpha^n \] (33)

where \( b \) and \( \alpha \) are values to be determined. Note the similarity with the form of solution for linear differential equations given in Section 4.3. We can determine \( \alpha \) by substituting (33) into equation (32). For the relation \( r_n = r_{n-1} + r_{n-2} \), (33) yields

\[ b \alpha^n = b \alpha^{n-1} + b \alpha^{n-2} \]
We can simplify this equation by dividing both sides by \( b_0 n^{-2} \) to obtain

\[
\alpha^2 = \alpha + 1 \quad \text{or} \quad \alpha^2 - \alpha - 1 = 0 \tag{34}
\]

Paralleling the terminology for differential equations, equation (34) is called the characteristic equation of the recurrence relation. The left-side polynomial in (34) is called its characteristic polynomial.

The two roots of the characteristic equation in (34) are found by the quadratic formula

\[
ax^2 + bx + c = 0 \quad \text{has roots} \quad \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{35}
\]

In our case, the solutions are \((1 \pm \sqrt{5})/2\), or approximately \(1.681\) and \(-.618\). It is quite surprising that the simple sequence formed by the Fibonacci numbers should turn out to be a function of an irrational number such as \(\sqrt{5}\).

As with differential equations, the general solution to (24) is a linear combination of the two solutions we have found:

\[
r_n = b_1 \left( \frac{1 + \sqrt{5}}{2} \right)^n + b_2 \left( \frac{1 - \sqrt{5}}{2} \right)^n \tag{36}
\]

For simplicity, let \(\alpha_1 = (1 + \sqrt{5})/2\) \((= 1.618)\) and \(\alpha_2 = (1 - \sqrt{5})/2\) \((= -0.618)\). So (36) becomes

\[
r_n = b_1 \alpha_1^n + b_2 \alpha_2^n \tag{37}
\]

As with differential equations, we solve for \(b_1\) and \(b_2\) by inserting the starting conditions in (37). Suppose that \(r_0 = r_1 = 1\).

\[
1 = r_0 = b_1 \alpha_1^0 + b_2 \alpha_2^0 = b_1 + b_2' \\
1 = r_1 = b_1 \alpha_1 + b_2 \alpha_2 = \alpha_1 b_1 + \alpha_2 b_2 \tag{38}
\]

Solving these two equations in two unknowns, we obtain

\[
b_1 = \frac{\alpha_2 - 1}{\alpha_2 - \alpha_1} = \frac{1 + \sqrt{5}}{2\sqrt{5}} = \frac{\alpha_1}{\sqrt{5}} \\
b_2 = \frac{1 - \alpha_1}{\alpha_2 - \alpha_1} = \frac{1 - \sqrt{5}}{2\sqrt{5}} = \frac{-\alpha_2}{\sqrt{5}} \tag{39}
\]

Remember that \(\alpha_2 = -0.618\), so \(|\alpha_2^2|\) will always be < \(\frac{1}{2}\). Thus the formula for the Fibonacci numbers is

\[
r_n = \text{closest integer to} \ \frac{1}{\sqrt{5}} \left[ \frac{1 + \sqrt{5}}{2} \right]^{n+1} \tag{40}
\]
Unfortunately, this formula is much harder to use than the original recurrence relation (30). That is, iteratively computing successive Fibonacci numbers by summing the previous two numbers is much easier than using (40). The one important fact we do get from (40) is that \( r_n \) grows at an exponential rate (faster than any polynomial in \( n \)).

There is one important similarity to note between Example 1 and Example 4. The largest root of the characteristic polynomial equation (35) is the growth rate of this model, just as the largest eigenvalue of the Leslie model is that model’s growth rate. To illustrate this link further, let us treat our original pair of recurrence relations for young and adults as a Leslie model:

\[
y_n = m_{n-1} \\
m_n = y_{n-1} + m_{n-1}
\]

The matrix of coefficients in (41) is

\[
L = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}
\]

The eigenvalues of this Leslie matrix \( L \) are the roots of \( \det(L - \lambda I) \).

\[
\det(L - \lambda I) = \begin{vmatrix} -\lambda & 1 \\ 1 & 1 - \lambda \end{vmatrix} = \lambda^2 - \lambda - 1
\]

But (43) is just the characteristic polynomial (35) for our second-order recurrence relation. So the roots of (43) are again \( (1 \pm \sqrt{5})/2 \).

Iterating the growth model \( x' = Lx \) is equivalent to iterating the recurrence relation. Using the eigenvector coordinate approach from Section 2.5, let us write \( x \) as a linear combination of the eigenvectors \( u_1, u_2 \) of \( L \):

\[
x = a_1 u_1 + a_2 u_2
\]

Then

\[
x^{(n)} = A^n x = a_1 A^n u_1 + a_2 A^n u_2
\]

or

\[
\begin{bmatrix} x_1^{(n)} \\ x_2^{(n)} \end{bmatrix} = a_1 \alpha_1^n \begin{bmatrix} u_{11} \\ u_{12} \end{bmatrix} + a_2 \alpha_2^n \begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix} \]

\[
= a_1 \alpha_1^n u_1 + a_2 \alpha_2^n u_2
\]

The sum \( x_1^{(n)} + x_2^{(n)} \) of the components of \( x^{(n)} \) is \( r_n \), the total number of rabbits. From (44), this sum is

\[
r_n = x_1^{(n)} + x_2^{(n)} = a_1 \alpha_1^n (u_{11} + u_{12}) + a_2 \alpha_2^n (u_{21} + u_{22})
\]
If we define constants $b_1$, $b_2$ as follows:

$$b_1 = a_1(u_{11} + u_{12}) \quad \text{and} \quad b_2 = a_2(u_{21} + u_{22})$$

then (45) becomes

$$r_n = b_1\alpha_1^n + b_2\alpha_2^n \quad (46)$$

Formula (46) is the same answer we got in (37). Then the $b_1$ and $b_2$ in (46) and in (37) must be the same constants. Thus the equations (38) for determining initial-condition constants $b_1$, $b_2$ are closely related to the equations for determining the weights $a_1$, $a_2$ in writing the initial vector $\mathbf{x}$ as a linear combination of the eigenvectors, $\mathbf{x} = a_1\mathbf{u}_1 + a_2\mathbf{u}_2$.

We close by noting that any recurrence relation can be recast as a system of first-order equations, the way the Fibonacci relation was in (41). For example,

$$r_n = a_1r_{n-1} + a_2r_{n-2} + a_3r_{n-3} + a_4r_{n-4} \quad (47)$$

becomes

$$r_n = a_1r_{n-1} + a_2s_{n-1} + a_3t_{n-1} + a_4u_{n-1}$$

$$s_n = r_{n-1} \quad \text{or} \quad r_n = A r_{n-1}$$

$$t_n = s_{n-1}$$

$$u_n = t_{n-1} \quad (48)$$

One can check that for the $A$ in (48), the characteristic polynomial $p(A, \lambda)$ is $\lambda^4 - a_1\lambda^3 - a_2\lambda^2 - a_3\lambda - a_4$.

In summary, the study of linear recurrence relations, when viewed as (48), is a special case of the study of linear growth models.

**Section 4.5 Exercises**

**Summary of Exercises**

Exercises 1 and 2 concern Leslie population growth models. Exercise 3 looks at eigenvalues of cyclic Markov chains. Exercises 4–10 deal with the harvesting model and variations. Exercises 11–21 involve recurrence relations, with Exercises 11–13 about building recurrence relations.

1. Find the long-term annual growth rate for the following Leslie growth models (use iteration; this growth rate is the size of the largest eigenvalue). Also find the long-term population distribution (percentages in each age group).

   (a) $y' = m + 2o$
   
   \begin{align*}
   m' &= y \\
   o' &= m
   \end{align*}

   (b) $y' = m + 2o$

   \begin{align*}
   m' &= .5y \\
   o' &= .5m
   \end{align*}
Sec. 4.5 Growth Models

(c) \[ y' = m + 4o \]
\[ m' = .5y \]
\[ o' = .5m \]

(d) \[ y' = 4a + 4o \]
\[ m' = .5y \]
\[ a' = .5m \]
\[ o' = .5a \]

(e) \[ y' = 4o \]
\[ m' = .5y \]
\[ a' = .5m \]
\[ o' = .5a \]

2. Consider the three-age-group Leslie matrix

\[
L = \begin{bmatrix}
0 & 0 & b \\
d_1 & 0 & 0 \\
0 & d_2 & 0 \\
\end{bmatrix}
\]

The characteristic equation, \( \det(L - \lambda I) = 0 \), for this matrix is \( \lambda^3 - bd_1d_2 = 0 \), or

\[ \lambda^3 = bd_1d_2 \]

Briefly describe the behavior of the system \( p' = Lp \) over time for the cases

(a) \( bd_1d_2 < 1 \)  
(b) \( bd_1d_2 = 1 \)  
(c) \( bd_1d_2 > 1 \)

3. What are the eigenvalues for the Markov chain in Exercise 1 of Section 4.4? Explain that Markov chain’s behavior in terms of the eigenvalues.

4. For the harvesting model in Example 3:
   (a) If \( hf = 50 \), how large can \( hm \) be (without negative herd values, as happened with \( hf = 50, \) \( hm = 100 \))?
   (b) Suppose that we harvest 100 yearling males and 100 yearling females. What is the stable herd vector now?
   (c) Suppose that we harvest 100 yearling males and 50 yearling females. What is the stable herd vector? Does it make sense?

5. Solve the rabbit harvesting system of equations (16) yourself by Gaussian elimination with \( h_m = h_f = 100 \).

6. In the harvesting model in Example 3, explain in words why the number of females and males is the same when we harvest the same number of adult males and adult females? Will this also be true if in addition we harvest equal numbers of yearling males and yearling females?

7. In the harvesting model in Example 3, explain in words why when \( hm = 100 \) and \( hf = 50 \) we get an impossible solution (involving a negative number of adult males).
8. Re-solve (16), using Gaussian elimination, with the survival probabilities of .6 changed to .5; again let \( h_m = h_f = 100 \). Explain in words the difference between your answer and the answer in (17) obtained with survival probabilities of .6.

9. (a) Suppose that we want the rabbit population to grow by 10% during the year. Rewrite the matrix equation (15) to reflect that fact that \( x' = 1.1x \). Solve the new version of (16) with \( h_m = h_f = 100 \).

(b) Find the inverse of the new coefficient matrix \((A - 1.1I)\) in (16).

Note: Requires a computer program for inverses.

10. Try to solve our harvesting model by iteration with \( h_m = h_f = 100 \). That is, guess an initial value for the population vector \( x \) and insert that vector of values on the right side of (13). Use the resulting left-side values as your next estimated \( x \) and continue iterating. Try several different starting vectors. Do you ever get convergence to a solution?

11. Suppose that \( a_n \), the level of radioactivity after \( n \) years from the element linearium, decreases by 20% a year. Write a recurrence relation that expresses this decay rate.

12. Let \( a_n \) be the number of dollars in a savings account after \( n \) years. Suppose that money earns 10% interest a year.

(a) Write a recurrence relation to represent this interest rate.

(b) If \( a_0 = 100 \), calculate \( a_5 \).

13. Let \( a_n \) be the number of different ways for an elf to climb a sequence of \( n \) stairs with steps of size 1 or 2. Explain why \( a_n = a_{n-1} + a_{n-2} \). What are \( a_1 \) and \( a_2 \)? Determine \( a_8 \).

14. Solve the following recurrence relations, given the initial values.

(a) \( a_n = 3a_{n-1} - 2a_{n-2}, \quad a_0 = a_1 = 2 \)

(b) \( a_n = 6a_{n-1} - 8a_{n-2}, \quad a_0 = 0, a_1 = 1 \)

(c) \( a_n = 3a_{n-1} + 4a_{n-2}, \quad a_0 = a_1 = 1 \)

(d) \( a_n = 2a_{n-1} - a_{n-2}, \quad a_0 = a_1 = 1 \)

Hint: See Exercise 21.

15. Give an approximate formula for \( a_{20} \) for each recurrence relation in Exercise 14 (use just the largest root of the characteristic equation).

16. Convert each recurrence relation in Exercise 14 into a pair of first-order recurrence relations \( x_n = Ax_{n-1} \) for \( x_n = [a_n, b_n] \).

Hint: Let \( b_n = a_{n-1}; \) see (48).

Recast the initial values from Exercise 14 into an initial-value vector \( x_1 \). Check that the eigenvalues of each \( A \) equal the roots of the characteristic equation for the corresponding original recurrence relation in Exercise 14.
17. Solve the recurrence relations you got in Exercise 16 using the method at the end of Section 3.1 to get a formula for \( a_n \).

18. Convert the following recurrence relations into a system of first-order recurrence relations.
   (a) \( a_n = 2a_{n-1} + a_{n-2} - a_{n-3} \)  \( \quad \) (b) \( a_n = a_{n-2} + a_{n-4} \)

19. Show that \( a_n = \alpha^n \), where \( \alpha^2 = c_1 \alpha + c_2 \), will always be a solution to the recurrence relation \( a_n = c_1a_{n-1} + c_2a_{n-2} \).

20. Show that any linear combination of solutions to the recurrence relation \( a_n = c_1a_{n-1} + c_2a_{n-2} \) is again a solution.

21. Verify that \( a_n = n\lambda^n \) is a solution to the recurrence relation \( a_n = c_1a_{n-1} + c_2a_{n-2} \) whose characteristic equation \( k^2 - c_1k - c_2 = 0 \) has \( \lambda \) as a double root.

   Note: If \( k^2 - c_1k - c_2 = 0 \) has \( \lambda \) as a double root, the characteristic equation can be factored as \((k - \lambda)^2 = 0\). This means that \( c_1 = 2\lambda \) and \( c_2 = -\lambda^2 \). Use these values for \( c_1 \) and \( c_2 \).

---

**Section 4.6 Linear Programming**

Studies have shown that about 25% of all scientific computing is devoted to solving linear programs. Linear programming is the principal tool of management science. The object of a linear program is to optimize—maximize or minimize—some linear function subject to a system of linear constraints. There are hundreds of different real-world problems that can be posed as linear programs. We start with a simple linear problem presented to maximize sales from furniture production.

**Example 1. Production of Chairs and Tables**

A factory can manufacture chairs and tables. Let \( x_1 \) be the number of chairs produced and \( x_2 \) the number of tables. Chairs sell for \$40 a piece and tables for \$200 a piece. The production of \( x_1 \) chairs and \( x_2 \) tables requires various amounts of raw materials whose supplies are limited. The following inequalities describe the requirements and supplies.

\[
\begin{align*}
\text{Wood:} & \quad x_1 + 4x_2 \leq 1400 \\
\text{Labor:} & \quad 2x_1 + 3x_2 \leq 2000 \\
\text{Braces:} & \quad x_1 + 12x_2 \leq 3600 \\
\text{Upholstery:} & \quad 2x_1 \leq 1800
\end{align*}
\]

Subject to these constraints we want to pick \( x_1 \) and \( x_2 \) so as to maximize the objective function of the total sales. Our model is
Maximize $40x_1 + 200x_2$  \hspace{1cm} (2)

subject to $x_1 \geq 0$, $x_2 \geq 0$ and (1)

Linear programming models of energy usage in the American economy have used thousands of variables and constraint equations and inequalities. Linear models with close to 1,000,000 variables and over 100,000 constraints have been developed and solved in private industry. Such a system of equations has $10,000,000,000$ (10 billion) coefficient terms. Of course, in these large problems almost all coefficients are zero. A large mathematical theory about linear programs has been developed with simplifying shortcuts that make it possible to solve huge linear programs.

In Figure 4.12, we have marked (the shaded area) the feasible region of $x_1 - x_2$ points that satisfy (1).

The key to solving a linear program is the following theorem.

**Theorem.** A linear objective function assumes its maximum and minimum values on the boundary of the feasible region (assuming that the feasible region is bounded). In fact, the optimal value is achieved at a corner point of this boundary.

**Proof.** Although true for all linear programs, we verify this proposition in the case of two variables (as in Example 1). Consider any line crossing through the feasible region. If we compute the values of a linear function along this line, we observe that as we move in one direction on the line, the linear function constantly increases and in the other direction constantly decreases. (There is one exception—the linear function could be constant along the line.) To find the maximum value of this linear function along the line, we should go as far as possible along the line in the direction of increasing values, that is, go to an end of the line segment where it meets the boundary. Thus the
maximum of a linear objective function occurs at the boundary of the feasible region.

Next repeat this argument using a boundary line of the feasible region. Again it pays to go to an end of the boundary line, that is, to a corner of the feasible region.

The theorem tells us to look at the corners of the feasible region for the optimal \((x_1, x_2)\)-value. In theory, any pair of constraint lines could intersect to form a corner of the feasible region, but by plotting the constraint lines, as in Figure 4.12, we can see which pairs of lines intersect along the boundary of the feasible region. To find the intersection point of two constraint lines, we solve for an \((x_1, x_2)\) point that lies on both lines—the same old problem of two equations in two unknowns. Table 4.5 lists the coordinates of the corners and the associated objective function values.

**Table 4.5**

<table>
<thead>
<tr>
<th>Corner Coordinates</th>
<th>Intersecting Constraints</th>
<th>Objective Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>(x_1 \geq 0) and (x_2 \geq 0)</td>
<td>0</td>
</tr>
<tr>
<td>(0, 300)</td>
<td>(x_1 \geq 0) and braces</td>
<td>60,000</td>
</tr>
<tr>
<td>(300, 275)</td>
<td>Braces and wood</td>
<td>67,000***</td>
</tr>
<tr>
<td>(760, 160)</td>
<td>Wood and labor</td>
<td>62,400</td>
</tr>
<tr>
<td>(900, 66.6)</td>
<td>Labor and upholstery</td>
<td>39,333</td>
</tr>
<tr>
<td>(900, 0)</td>
<td>Upholstery and (x_2 \geq 0)</td>
<td>36,000</td>
</tr>
</tbody>
</table>

So the optimal production schedule is to make 300 chairs and 275 tables, whose sales value will be $67,000.

Before giving a general procedure for solving linear programs, we present some examples that show how to build linear programming models. The reader may also want to refer back to the crop planting linear program presented in Section 1.4.

**Example 2. Dietician’s Problem**

Suppose that a meal must contain at least 500 units of vitamin A, 1000 units of vitamin C, 100 units of iron, and 50 grams of protein. A dietician has two foods for the meal, meat and fruit. Meat costs 50 cents a unit and fruit costs 40 cents a unit.

Each unit of meat has 20 units of vitamin A, 30 units of vitamin C, 10 units of iron, and 15 grams of protein. Each unit of fruit has 50 units of vitamin A, 100 units of vitamin C, 1 unit of iron, and 2 units of protein.
The dietician wants to have the cheapest meal that satisfies the four nutritional constraints. Let us formulate the dietician’s problem as a linear program. Make $x_1$ be the number of units of meat used and $x_2$ the number of units of fruit used. Then the objective is to minimize the cost of the meal

Minimize \[ .5x_1 + .4x_2 \]
subject to the nutritional lower-bound constraints

Vitamin A: \[ 20x_1 + 50x_2 \geq 500 \]
Vitamin C: \[ 30x_1 + 100x_2 \geq 1000 \]
Iron: \[ 10x_1 + 1x_1 \geq 100 \]
Protein: \[ 15x_1 + 2x_2 \geq 50 \]

$x_1 \geq 0, \quad x_2 \geq 0$

We should note that the constraints in linear programs do not have to be inequalities. Later in this section we shall see how to convert inequalities to equations and equations to inequalities. The next example is an important type of linear programming problem in which the constraints are equations.

**Example 3. A Transportation Problem**

Warehouses 1, 2, and 3 have 20, 30, and 15 tons, respectively, of chicken wings. Colleges 1 and 2 need 25 and 40 tons, respectively, of chicken wings (to serve to students). The following table indicates the cost of shipping a ton from a given warehouse to a given college.

<table>
<thead>
<tr>
<th>Warehouses</th>
<th>College A</th>
<th>College B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>65</td>
</tr>
</tbody>
</table>

Since the overall demand at both colleges is $25 + 40 = 65$ and the overall supply of all three warehouses is also $20 + 30 + 15 = 65$, all the supplies of each warehouse must be used. The constraints are that the total amount of chicken wings shipped from warehouse 1 must equal 20 tons; from warehouse 2, 30 tons; from warehouse 3, 15 tons; and the total amount shipped to college A must equal 25 tons and to college B 40 tons. If $x_{ij}$ is the number of tons shipped from warehouse $i$ to college $j$, then these constraints are

\[
\begin{align*}
\text{Warehouse equations} \quad & x_{11} + x_{12} = 20 \\
& x_{21} + x_{22} = 30 \\
& x_{31} + x_{32} = 15
\end{align*}
\]
Section 4.6 Linear Programming

College equations:

\[ x_{11} + x_{21} + x_{31} = 25 \]
\[ x_{12} + x_{22} + x_{32} = 40 \]

The objective is to minimize the transportation costs. In terms of the \( x_{ij} \), the transportation costs are

\[ 80x_{11} + 45x_{12} + 60x_{21} + 55x_{22} + 40x_{31} + 65x_{32} \] (4)

Thus our linear program is to minimize (4) subject to (3) and \( x_{ij} \geq 0 \).

Example 4. Running a Chicken Farm

We have a farm with 5000 chickens. Each year for 3 years we must decide how many of the chickens should lay eggs to be sold and how many chickens should be hatching eggs to produce more chickens next year (we assume that all chickens are hens; roosters are ignored). At the end of 3 years, all the chickens are sold for slaughter at $2 per bird (this includes chickens hatched during the third year). A chicken can hatch 30 eggs in a year. The eggs from one chicken in 1 year earn $7. It costs $2 a year in feed for chickens (no charge for chickens born during the year). The objective is to maximize income from eggs and from the final sale of the chickens. State this maximization problem as a linear program.

Let \( x_i \) and \( y_i \) be the number of chickens hatching and laying eggs for sale, respectively, in the \( i \)th year, \( i = 1, 2, 3 \). Then the following equations represent the fact that \( x_i + y_i \) equals the total number of chickens each year (the original number 5000 plus the numbers of new chickens born thus far).

\[ x_1 + y_1 = 5000 \]
\[ x_2 + y_2 = 5000 + 30x_1 \]
\[ x_3 + y_3 = 5000 + 30x_1 + 30x_2 \] (5)

Let us rewrite (5) with all the variables on the left side and each variable in a different column.

\[ x_1 + y_1 = 5000 \]
\[ -30x_1 + x_2 + y_2 = 5000 \]
\[ -30x_1 - 30x_2 + x_3 + y_3 = 5000 \] (6)

As usual, all variables must be nonnegative.

\[ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0 \] (7)

The objective function to be maximized is
Maximize  
\[ 2(5000 + 30x_1 + 30x_2 + 30x_3) + 7(y_1 + y_2 + y_3) - 2(15,000 + 60x_1 + 30x_2) \]  
(8)

The first factor in (8) represents the sale of all chickens after 3 years, the second factor the income from eggs, and the last factor the feeding cost [the total number of chicken-years of feeding is the sum of the right-hand sides in (7)].

The required linear program is to maximize (8) subject to (6) and (7).

We shall now develop a general procedure for solving linear programs. The discussion will be couched in terms of the chair-table linear program in Example 1. Our presentation will necessarily be sketchy. Readers interested in a more extensive treatment of linear programming should turn to any of the dozens of books on the subject (most colleges have several courses about linear programming, offered by mathematics, economics, and business departments).

Note that the method used in Example 1 of graphing and then checking the corner points of the feasible region for an optimal value is not feasible for larger problems. An \( n \)-variable problem with \( m \) constraints can have up to \( m^n \) corner points to check.

The theory of linear programming is centered about the following method, now called the **simplex algorithm**, which was developed over 30 years ago when the advent of digital computers first made it possible to try to solve moderate-sized linear programs. Intuitively, the simplex algorithm starts at the origin (a corner of the feasible region) and moves along a boundary edge of the feasible region to a better corner, where the objective function is larger, and continues in this way until it reaches an optimal corner.

We outline the simplex algorithm and then describe it in detail using Example 1 to illustrate the calculations.

---

**Simplex Algorithm**

Part 1. Let \( x_h \) be the variable with the largest positive coefficient in the objective function. Starting at the origin, increase \( x_h \) as much as possible until an inequality constraint is reached (while the other variables remain equal to 0).

Part 2. Make a partial change of coordinates by replacing \( x_h \) with \( x_h' \) so that the new corner, which we reached by increasing \( x_h \), becomes the origin in the new coordinates. If any coefficient in the new objective function is positive, go back to part 1; otherwise, the new origin is an optimal corner.
To find a better corner, the simplex algorithm looks at the objective function and checks which coefficient \( c_j \) is largest (most positive). Let \( x_h \) be the variable with the largest coefficient in the objective function. Then increasing \( x_h \) yields the greatest rate of increase in the objective function; in economic terms, activity \( h \) is the most profitable.

The simplex algorithm increases the value of \( x_h \) as much as possible, that is, until increasing \( x_h \) any more would violate one of the inequality constraints. If all \( c_j \leq 0 \), so that increasing any \( x_j \) cannot increase the objective function, the current origin must be an optimal corner.

In the chair–table problem, the simplex algorithm would start at the origin \((0, 0)\). The coefficient of \( x_2 \) is greater than the coefficient of \( x_1 \) in the objective function \((c_2 = 200 \text{ versus } c_1 = 40)\), so \( x_2 \) would be increased (while \( x_1 \) is kept fixed = 0). In words, we produce as many tables (variable \( x_2 \)) as possible because they are more profitable than chairs.

Recall the objective function and inequalities in this problem,

\[
\text{Maximize } 40x_1 + 200x_2 \\
x_1 \geq 0, \quad x_2 \geq 0
\]

Wood: \( x_1 + 4x_2 \leq 1400 \)

Labor: \( 2x_1 + 3x_2 \leq 2000 \) \hspace{1cm} \hspace{1cm} (9)

Braces: \( x_1 + 12x_2 \leq 3600 \)

Upholstery: \( 2x_1 \leq 1800 \)

Looking at Figure 4.12, we see that \( x_2 \) can be increased to \( x_2 = 300 \), where the objective function is 60,000. Any greater value of \( x_2 \) would violate the braces constraint.

Let us show how the new corner can be found algebraically (since in larger problems, we will not be able to draw a picture of the feasible region). We want to increase \( x_2 \) while keeping \( x_1 = 0 \). Substituting \( x_1 = 0 \) into the inequalities of (9), we obtain

Wood: \( 4x_2 \leq 1400 \)

Labor: \( 3x_2 \leq 2000 \) \hspace{1cm} \hspace{1cm} (10)

Braces: \( 12x_2 \leq 3600 \)

Upholstery: \( 0 \leq 1800 \)

We can ignore the upholstery inequality \( 0 \leq 1800 \)—it does not contain \( x_2 \) and will always be true. The other three inequalities in (10) are easily solved in terms of \( x_2 \) to become

Wood: \( x_2 \leq 350 \)

Labor: \( x_2 \leq 666\frac{2}{3} \) \hspace{1cm} \hspace{1cm} (11)

Braces: \( x_2 \leq 300 \)

The smallest of the bounds on \( x_2 \), namely 300, is the amount \( x_2 \) can be increased without violating any inequality. So again we find that \((0, 300)\) is
the new comer point we reach by increasing $x_2$ as much as possible. This corner $(0, 300)$ is formed by the intersection of constraints $x_1 = 0$ and $x_1 + 12x_2 = 3600$.

Next comes part 2 of the simplex algorithm. We perform a linear transformation that changes our coordinate system by replacing $x_2$ with a different coordinate variable, call it $x'_2$. The current corner $x_1 = 0, x_2 = 300$ will be transformed into the origin $x_1 = 0, x'_2 = 0$ in this new coordinate system.

To motivate this change of variables, we first need to show how our system of constraint inequalities can be recast into a system of equations. For this recasting, we must introduce additional variables. A **slack variable** equals the difference between the left- and right-hand sides of an inequality. With slack variables, the linear program in (9) becomes

Maximize $40x_1 + 200x_2$
subject to $x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0, \ x_4 \geq 0, \ x_5 \geq 0, \ x_6 \geq 0$ and

<table>
<thead>
<tr>
<th>Constraints</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wood:</td>
<td>$x_1 + 4x_2 + x_3 = 1400$</td>
</tr>
<tr>
<td>Labor:</td>
<td>$2x_1 + 3x_2 + x_4 = 2000$</td>
</tr>
<tr>
<td>Braces:</td>
<td>$x_1 + 12x_2 + x_5 = 3600$</td>
</tr>
<tr>
<td>Upholstery:</td>
<td>$2x_1 + x_6 = 1800$</td>
</tr>
</tbody>
</table>

where $x_3, x_4, x_5, x_6$ are the *slack variables*. We call $x_1, x_2$ **independent variables**—they are the original coordinate variables from (9). Like $x_1, x_2$, a slack variable must be $\geq 0$.

A slack variable is what the simplex algorithm uses as $x'_2$, the variable to replace $x_2$ in the change of coordinates mentioned above. In particular, we want to use $x_5$, the slack variable in the braces constraint $x_1 + 12x_2 + x_5 = 3600$. Remember that the corner $(0, 300)$ is the intersection point of $x_1 + 12x_2 = 3600$ with axis line $x_1 = 0$. Forcing the slack variable $x_5$ to be 0 is the same as forcing $x_1, x_2$ to satisfy the equation $x_1 + 12x_2 = 3600$. The corner $x_1 = 0, x_2 = 300$ where lines $x_1 = 0$ and $x_1 + 12x_2 = 3600$ meet is in $x_1, x_5$-coordinates the origin, $x_1 = 0, x_5 = 0$. This is exactly what we were looking for (see Figure 4.13).

We must rewrite the system of equations in (12) so that $x_5$ replaces $x_2$ as an independent variable, that is, as a coordinate variable. We do this by using the elimination-by-pivoting process to eliminate $x_2$ from other equations in (12). We subtract the appropriate multiple of the braces equation in (12), $x_1 + 12x_2 + x_5 = 3600$, from the other equations to eliminate $x_2$.

The wood equation $x_1 + 4x_2 + x_3 = 1400$ has an $x_2$-coefficient of 4 while the braces equation has an $x_2$-coefficient of 12. So we subtract $\frac{1}{3},$ or $\frac{1}{3}$, times the braces equation from the wood equation

\[
\text{Wood:} \quad x_1 + 4x_2 + x_3 = 1400 \\
-\frac{1}{3}(\text{Braces:} \quad x_1 + 12x_2 + x_5 = 3600) \quad (13a)
\]

\[
\text{New wood:} \quad \frac{2}{3}x_1 + x_3 - \frac{1}{3}x_5 = 200
\]

The labor constraint becomes
Figure 4.13 Feasible region with $x_1$, $x_5$ as independent variables.

Labor:
\[ 2x_1 + 3x_2 + x_4 = 2000 \]
\[ -\frac{1}{4}(\text{Braces:} \quad x_1 + 12x_2 + x_5 = 3600) \]  
New labor:
\[ \frac{7}{4}x_1 + x_4 - \frac{1}{4}x_5 = 1100 \]  

The upholstery equation is unchanged, since $x_2$ does not occur in it.

New upholstery:
\[ 2x_1 + x_6 = 1800 \]  

We also need to eliminate $x_2$ from the objective function. To do this, we write the braces equation as $x_1 + 12x_2 + x_5 = 0$.

Objective functions:
\[ 40x_1 + 200x_2 \]
\[ -\frac{50}{3} (\text{Braces:} \quad x_1 + 12x_2 + x_5 = 3600) \]  
New objective function:
\[ \frac{70}{3}x_1 - \frac{50}{3}x_5 + 60,000 \]  

We must also rewrite the original braces constraint $x_1 + 12x_2 + x_5 = 3600$ to make $x_2$ look like a slack variable; that is, $x_2$ should have a coefficient of 1. Dividing by 12, we have

New braces:
\[ \frac{1}{12}x_1 + x_2 \quad + \frac{1}{12}x_5 = 300 \]
Collecting (13a)-(13d) and (14), we have the desired transformed linear program.

\[
\begin{align*}
\text{Maximize } & \quad \frac{7}{3} x_1 - \frac{5}{3} x_5 + 60,000 \\
\text{subject to } & \quad x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0, \quad x_6 \geq 0, \text{ and} \\
& \quad \frac{2}{3} x_1 + x_3 - \frac{1}{3} x_5 = 200 \\
& \quad \frac{7}{4} x_1 + x_4 - \frac{1}{4} x_5 = 1100 \\
& \quad \frac{1}{2} x_1 + x_2 - \frac{1}{2} x_5 = 300 \\
& \quad 2x_1 + x_6 = 1800 \\
\end{align*}
\]  

(15)

Observe that (15) has the same general form as (12), except that \(x_2\) and \(x_5\) have interchanged roles as independent and slack variables. Since we have only restated the problem, a maximum for this problem is a maximum for our original problem (12). The feasible region for our linear program in the restated form is the "hashed" region in Figure 4.13; note that the coordinate axes are labeled \(x_1\) and \(x_5\).

This sequence of computations to interchange the independent and slack variable roles of \(x_2\) and \(x_5\) is called a pivot exchange. Recall from Section 3.2 about elimination by pivoting that we used the term pivot on entry \(a_{ij}\) to denote the process of using equation \(i\) to eliminate \(x_j\) from all other equations (and making the coefficient of \(x_j\) be 1 in equation \(i\)). If independent variable \(x_h\) is exchanged with the slack variable in equation \(g\), then we are pivoting on the coefficient of \(x_h\) in equation \(g\).

Using the concept of a pivot, we restate the simplex algorithm as follows.

**Simplex Algorithm**

Part 1. Let \(x_h\) be the (independent) variable with the largest positive coefficient in the current objective function. Increase \(x_h\) as much as possible until an inequality constraint is reached; call this the \(g\)th inequality.

Part 2. Perform a pivot exchange between \(x_h\) and the slack variable in the \(g\)th constraint. If any coefficient in the new objective function is positive, go back to part 1; otherwise, the new origin is an optimal corner.

We can convert the constraint equations in (15) back to inequality constraints by dropping the slack variables \(x_2, x_3, x_4,\) and \(x_6\):

\[
\begin{align*}
\text{Maximize } & \quad \frac{7}{3} x_1 - \frac{5}{3} x_5 + 60,000 \\
& \quad \frac{2}{3} x_1 - \frac{1}{3} x_5 \leq 200 \\
& \quad \frac{7}{4} x_1 - \frac{1}{4} x_5 \leq 1100 \\
& \quad \frac{1}{2} x_1 + \frac{1}{2} x_5 \leq 300 \\
& \quad 2x_1 \leq 1800 \\
\end{align*}
\]  

(16)
Sec. 4.6 Linear Programming

Now we apply the two steps of the simplex algorithm to our problem in its new form (16). In step 1 we increase $x_1$, which has the only positive coefficient in the objective function, while keeping $x_5 = 0$. (As an aside, we note that back in Figure 4.12, keeping $x_5 = 0$ means that we move along the brace constraint line $x_1 + 12x_2 = 3600$). When we set $x_5 = 0$ in (16), we obtain the inequalities

\[
\begin{align*}
\frac{8}{3}x_1 & \leq 200 \quad \text{or} \quad x_1 \leq 300 \\
\frac{7}{4}x_1 & \leq 1100 \quad \text{or} \quad x_1 \leq 4400 \\
\frac{1}{2}x_1 & \leq 300 \quad \text{or} \quad x_1 \leq 600 \\
2x_1 & \leq 1800 \quad \text{or} \quad x_1 \leq 900
\end{align*}
\]

(17)

The smallest of these constraints is the first. So we can increase $x_1$ to 300, and our new corner is at $x_1 = 300$, $x_5 = 0$, where the first constraint $2x_1/3 - x_5/3 = 200$ and constraint $x_5 = 0$ meet (see Figure 4.13). To see where we are in the original problem, we can use the third constraint equation in (15) to compute $x_2$’s value when $x_1 = 300$ and $x_5 = 0$—we get $x_2 = 275$; so our new corner corresponds to the point $x_1 = 300$, $x_2 = 275$ in Figure 4.12.

We return to the equation form (15) of the constraints. The first constraint is $2x_1/3 + x_3 - x_5/3 = 200$. Then $x_3$ is the slack variable for this constraint, so in part 2 of the simplex algorithm, we perform a pivot exchange between independent variable $x_1$ and slack variable $x_3$. [Note that the computations in (17) could be done with the constraints in equations form, as in (15): we simply divide the right-side value in each equation by the coefficient of $x_1$ and pick the smallest positive value.]

Using the first constraint to eliminate $x_1$ from the other equations in (15) and from the objective function, we obtain the new linear program.

Maximize $-35x_3 - 5x_5 + 67,000$

subject to $x_i \geq 0$, $i = 1, 2, 3, 4, 5, 6$, and

\[
\begin{align*}
x_1 & + \frac{3}{2}x_3 - \frac{1}{2}x_5 = 300 \\
- \frac{2}{3}x_3 + x_4 + \frac{5}{8}x_5 & = 575 \\
x_2 & - \frac{1}{8}x_3 + \frac{1}{8}x_5 = 275 \\
- 3x_3 & + x_5 + x_6 = 1200
\end{align*}
\]

(18)

Since all coefficients in the current objective function are negative, the current origin is the maximum corner. The value of the objective function at the origin $x_3 = x_5 = 0$ is simply the constant term in the objective function, 67,000. By setting $x_3 = x_5 = 0$ in (18), we can also directly read off the values of $x_1$ (= 300) and $x_2$ (= 275) as well as the values of the other slack variables.

This finishes our example of how the simplex algorithm works. Any linear program of the general form
Maximize $c \cdot x + d$ subject to $Ax \leq b, \ x \geq 0$ \hspace{1cm} (19)

can be solved by the method we have just presented.

The key decision in the simplex algorithm is the choice of the pivot entry (part 1). We pick the variable $x_k$ with the largest coefficient in the objective function. Then we divide the right-side value in each constraint equation by the coefficient of $x_k$ and pick the equation with the smallest quotient. The coefficient of $x_k$ in that equation is the pivot entry.

Just as in Gaussian elimination, we can display the computations in the simplex algorithm in matrix notation (omitting the variables). For the linear program in (19) we use the augmented matrix

$$
\begin{bmatrix}
c & -d \\
A & b
\end{bmatrix}
$$

(20)

Note that the constant $d$ is written with a minus sign, $-d$. This is a technicality based on the fact that $b$ is the vector of right sides of equations, while $d$ is on the left side (if fictitiously $d$ were put in the "right side" of the objective function, the constant would become $-d$).

Let us repeat the stages of the simplex algorithm for our chair–table program using matrix notation. At the outset we have (where $x_1$, $x_2$ are the initial independent variables).

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wood</td>
<td>40</td>
<td>200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1400</td>
</tr>
<tr>
<td>Labor</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2000</td>
</tr>
<tr>
<td>Braces</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3600</td>
</tr>
<tr>
<td>Upholstery</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1800</td>
</tr>
</tbody>
</table>

We pick $x_2$ (200 is largest coefficient in first row) and then divide each positive entry in $x_2$'s column into the corresponding entry in the last column. The smallest quotient is in the braces equation, so we pivot on entry (4, 2), which is circled in (21). After pivoting on entry (4, 2), we obtain

$$
\begin{bmatrix}
\begin{array}{cccccc|c}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & 0 \\
\hline
0 & 0 & 0 & -\frac{50}{3} & 0 & -60,000 \\
\end{array}
\end{bmatrix}
$$

(22)

Next we pick $x_1$ and then the wood equation. So entry (2, 1) is the second pivot. After pivoting on entry (2, 1), we obtain
Since the objective function has no positive coefficients, we now have an optimum with objective function value 67,000. The values of the current independent variables, \( x_3 \), \( x_5 \), are zero and from (23) we read off the values of the other variables to be

\[
x_1 = 300, \quad x_2 = 275, \quad x_4 = 575, \quad x_6 = 1200
\]

The name given to these augmented matrices, (21), (22), and (23), is simplex tableaus.

Let us now go quickly through the solution of another linear program.

**Example 5. Simplex Algorithm Applied to a Linear Program**

Consider the following linear program for the production of sugar \( (x_1) \), syrup \( (x_2) \), and molasses \( (x_3) \):

Maximize \( 3x_1 + 4x_2 + 2x_3 \),

subject to \( x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \) and

\[
\begin{align*}
\text{Transportation:} & \quad 2x_1 + x_2 + x_3 \leq 6 \\
\text{Labor:} & \quad x_1 + 2x_2 + 3x_3 \leq 7 \\
\text{Machinery:} & \quad 3x_1 + 2x_2 + 4x_3 \leq 15
\end{align*}
\]

In simplex tableau with slack variables added, we have (where \( x_1, x_2, x_3 \) are the initial independent variables)

\[
\begin{bmatrix}
\mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & \mathbf{X}_4 & \mathbf{X}_5 & \mathbf{X}_6 \\
\hline
3 & 4 & 2 & 0 & 0 & 0 \\
2 & 1 & 1 & 1 & 0 & 0 \\
1 & \underline{2} & 3 & 0 & 1 & 0 \\
3 & 2 & 4 & 0 & 0 & 1
\end{bmatrix}
\]

The pivot will be in \( x_2 \)'s column. Picking the minimum of \( \frac{6}{2}, \frac{7}{2}, \frac{15}{2} \), we take the second, from labor's row. So the pivot entry is \( (3, 2) \).

After pivoting there, we obtain
The next pivot will be in $x_1$'s column. Picking the minimum of \( \frac{2}{3} \), \( \frac{2}{3} \), \( \frac{2}{3} \), we take the first, from transportation's row. So the pivot entry is (2, 1). After pivoting there, we obtain

\[
\begin{bmatrix}
0 & 0 & -1 & -\frac{1}{3} & -\frac{2}{3} & -\frac{5}{3} & 0 & -\frac{47}{3}
\end{bmatrix}
\]

Now there are no positive coefficients in the objective function, so we have a maximum, where the objective function equals \( \frac{47}{3} \), the independent variables \( x_3, x_4, x_5 \) are zero, and the other variables are

\[
x_1 = \frac{5}{3}, \quad x_2 = \frac{8}{3}, \quad x_6 = \frac{14}{3}
\]

The simplex algorithm was invented in the earliest days of linear programming by G. Dantzig as an intuitive scheme for “walking” along the boundary of the feasible regions in search of better and better corners. Many more sophisticated methods have been proposed—in the 1970s Khachian’s algorithm made the front pages of major newspapers for its theoretical advantages over the simplex algorithm, and in 1984 Karmarkar’s algorithm gained publicity for being faster than the simplex algorithm for some linear programs—but the simplex algorithm is still the best general-purpose way to solve a linear program. The explanation of why it works so well requires advanced mathematical analysis.

There are many important variations in the basic theory of the simplex algorithm. We mention a few here and give some more in the Exercises and in an appendix to this section.

First, if the problem involves minimization, we can convert it to a maximization problem by multiplying the objective function by \(-1\) (maximizing \(-c \cdot x\) is the same as minimizing \(c \cdot x\)).

Second, if an inequality constraint is of the form \(a \cdot x \geq b\), convert it to a constraint with a \(\leq\) sign by multiplying both sides by \(-1\) (to get \(-a \cdot x \leq -b\)).

Third, we consider linear programs whose constraints are equations. We converted the inequalities in the chair-table program to equations by introducing slack variables [see system (12)]. For a system of equations to
be in slack variable form, we require that each equation should have a slack variable which has a coefficient of +1 and occurs only in that equation. If a system of constraint equations is not in slack variable form, we can use pivoting to get the system in this form. We illustrate this process with the transportation problem in Example 3.

**Example 6. Putting Transportation Constraint Equations in Slack Variable Form**

In Example 3 we obtained the following mathematical formulation of the problem of minimizing the cost of shipping chicken wings from three warehouses to two colleges.

Minimize $80x_{11} + 45x_{12} + 60x_{21} + 55x_{22} + 40x_{31} + 65x_{32}$

subject to $x_{ij} \geq 0$ and

Warehouse

- $x_{11} + x_{12} = 20$
- $x_{21} + x_{22} = 30$
- $x_{31} + x_{32} = 15$  \hspace{1cm} (28)

College

- $x_{11} + x_{21} + x_{31} = 25$
- $x_{12} + x_{22} + x_{32} = 40$

We want to use pivoting to convert the five constraint equations in (28) into an equivalent system of equations with slack variables (variables that each occur in just one equation).

Let us make $x_{12}$ the slack variable for the first equation. To eliminate $x_{12}$ from the fifth equation, we simply subtract the first equation from the fifth equation (i.e., we pivot on entry (1, 2), the coefficient of $x_{12}$ in the first equation). In a similar fashion we make $x_{22}$ the slack variable for the second equation by pivoting on entry (2, 4), and we make $x_{32}$ the slack variable for the third equation by pivoting on entry (3, 6). After these three pivots (which involve subtracting the first, second and third equations from the fifth equation), we have

$$
\begin{align*}
    x_{11} + x_{12} &= 20 \\
    x_{21} + x_{22} &= 30 \\
    x_{31} + x_{32} &= 15 \\
    x_{11} + x_{21} + x_{31} &= 25 \\
    -x_{11} - x_{21} - x_{31} &= -25 \\
\end{align*}
$$

In (29), $x_{12}$, $x_{22}$, and $x_{32}$ have the form of slack variables as required. These pivots had the effect of converting the last equation into the negative of the fourth equation—the last equation is redundant and can be eliminated (the reason for this redundancy is explained in Exercise 20). Let us make $x_{31}$ the slack variable for the fourth equation. Pivoting on entry (4, 5), that is, subtracting the fourth equation from the third equation, we obtain
\[ x_{11} + x_{12} + x_{21} + x_{22} - x_{11} - x_{21} + x_{31} = 20 \]  
\[ + x_{21} + x_{22} + x_{32} = 30 \]  
\[ - x_{11} - x_{21} + x_{31} = -10 \]  

(30)

Now each equation has a slack variable, and we can rewrite (30) as the following family of inequalities:

\[ x_{11} \leq 20 \]
\[ x_{21} \leq 30 \]
\[ -x_{11} - x_{21} \leq -10 \quad (\Leftrightarrow x_1 + x_{21} \geq 10) \]
\[ x_{11} + x_{21} \leq 25 \]  

(31)

In the pivoting process, we also have to restate the objective function in terms of just \( x_{11} \) and \( x_{21} \) (see Exercise 21 for details). Note that the origin in \( x_{11}, x_{21} \) coordinates is not in the feasible region.

We have reduced this problem in five equations and six unknowns to an easy problem of four inequalities in two variables.

---

**Sensitivity Analysis**

We conclude this section with a brief discussion of the sensitivity of the solution of our linear program to changes in the input values in the constraints. In many economics applications, this sensitivity analysis to changes in the input is almost as important as solving the linear program.

**Example 7. Sensitivity Analysis in Chair–Table Production**

The final (slack-variable) form of our linear program when an optimum was obtained by the simplex algorithm was

Maximize \(-35x_3 - 5x_5 + 67,000\)

subject to \(x_i \geq 0, i = 1, 2, 3, 4, 5, 6\), and

\[ x_1 + \frac{3}{4}x_3 - \frac{1}{3}x_5 = 300 \]
\[ - \frac{3}{4}x_3 + x_4 + \frac{3}{4}x_5 = 575 \]
\[ x_2 - \frac{1}{3}x_3 + \frac{1}{3}x_5 = 275 \]
\[ - 3x_3 + x_5 + x_6 = 1200 \]

(32)

Here \( x_3 \) and \( x_5 \) are the current independent variables whose origin \( x_3 = x_5 = 0 \) is the optimal corner. These are the slack variables for the original wood and braces constraints. So \( x_3 = x_5 = 0 \) in the optimal solution means that the optimal production schedule will use all the wood and all the braces. To determine the values of the original inde-
pendent variables and other slack variables, we simply set $x_3 = x_5 = 0$ in (32) and read off the values of the remaining variables in each equation:

$$x_1 = 300, \quad x_2 = 275, \quad x_4 = 575, \quad x_6 = 1200 \quad (33)$$

As we had found earlier, we make 300 chairs ($x_1$) and 275 tables ($x_2$). The labor slack variable ($x_4$) is 575 and the upholstery slack variable ($x_6$) is 1200. These slack-variable values are the amounts of these two inputs that are unused in the optimal solution. We have an excess supply of labor and upholstery. Thus moderate decreases in the amount of labor or upholstery available will not affect our solution.

What about changes in the input materials we do use, wood and braces? This is a place where the simplex algorithm really shines. First it is convenient to solve for $x_1$ and $x_2$ in the first and third equations, which involve $x_1$ and $x_2$, respectively.

$$x_1 = 300 - \frac{3}{2}x_3 + \frac{1}{2}x_5$$
$$x_2 = 275 + \frac{1}{8}x_3 - \frac{1}{8}x_5 \quad (34)$$

Having 1 less unit of wood (1399 units instead of 1400) is equivalent to increasing the wood slack variable $x_3$ from 0 to 1. To determine the effects of 1 less unit of wood, we simply set $x_3 = 1$, while $x_5 = 0$, in (34). From (34) we have $x_1 = 300 - \frac{3}{2}$ and $x_2 = 275 + \frac{1}{8}$. The new value of the objective function with 1 less unit of wood is also obtained by setting $x_3 = 1$ in the objective function (while $x_5 = 0$). We have $-35 + 67,000$.

In summary, the coefficients of $x_3$ in (34) and in the objective function give the effect of 1 less unit of wood: chair production will decrease by $\frac{3}{2}$, table production $x_2$ will increase by $\frac{1}{8}$, and profit will decrease by $35$. If we had 1 more unit of wood ($x_3 = -1$), the opposite occurs. Chair production increases by $\frac{3}{2}$, table production decreases by $\frac{1}{8}$, and profit increases by $35$. It is left as an exercise to the reader to evaluate the effects of changing the number of braces.

In economics, the increase in the profit caused by using 1 more unit of an input is called the marginal value of the input. In this case it means that we should be willing to pay $35 for 1 additional unit of wood because that is the value of wood to us in increasing our sales.

### Section 4.6 Exercises

**Summary of Exercises**

Exercises 1–12 involve converting a “word problem” into a linear program (some of these problems previously appeared in Section 1.4); solutions, if requested, are to be obtained by graphing. Exercises 13–17 require one to solve linear programs. Exercises 18–21 ask for transportation problems to be converted into slack-variable form. Exercises 22–25 involve sensitivity analysis. Exercise 26 illustrates duality theory.
Now each equation has a slack variable, and we can rewrite (30) as the following family of inequalities:

\[
\begin{align*}
& x_{11} + x_{12} + x_{21} + x_{22} = 20 \\
& -x_{11} - x_{21} + x_{32} = -10 \\
& x_{11} + x_{21} + x_{31} = 25 
\end{align*}
\] (30)

In the pivoting process, we also have to restate the objective function in terms of just \(x_{11}\) and \(x_{21}\) (see Exercise 21 for details). Note that the origin in \(x_{11}, x_{21}\) coordinates is not in the feasible region.

We have reduced this problem in five equations and six unknowns to an easy problem of four inequalities in two variables.

\[\text{Sensitivity Analysis}\]

We conclude this section with a brief discussion of the sensitivity of the solution of our linear program to changes in the input values in the constraints. In many economics applications, this sensitivity analysis to changes in the input is almost as important as solving the linear program.

\[\text{Example 7. Sensitivity Analysis in Chair–Table Production}\]

The final (slack-variable) form of our linear program when an optimum was obtained by the simplex algorithm was

\[
\begin{align*}
\text{Maximize } & -35x_3 - 5x_5 + 67,000 \\
\text{subject to } & x_i \geq 0, \ i = 1, 2, 3, 4, 5, 6, \text{ and} \\
& x_1 + \frac{3}{2}x_3 - \frac{1}{2}x_5 = 300 \\
& -\frac{21}{8}x_3 + x_4 + \frac{5}{8}x_5 = 575 \\
& x_2 - \frac{1}{8}x_3 + \frac{1}{8}x_5 = 275 \\
& -3x_3 + x_5 + x_6 = 1200 
\end{align*}
\] (32)

Here \(x_3\) and \(x_4\) are the current independent variables whose origin \(x_3 = x_5 = 0\) is the optimal corner. These are the slack variables for the original wood and braces constraints. So \(x_3 = x_5 = 0\) in the optimal solution means that the optimal production schedule will use all the wood and all the braces. To determine the values of the original inde-
pendent variables and other slack variables, we simply set \( x_3 = x_5 = 0 \) in (32) and read off the values of the remaining variables in each equation:

\[
x_1 = 300, \quad x_2 = 275, \quad x_4 = 575, \quad x_6 = 1200
\]

(33)

As we had found earlier, we make 300 chairs \((x_1)\) and 275 tables \((x_2)\). The labor slack variable \((x_4)\) is 575 and the upholstery slack variable \((x_6)\) is 1200. These slack-variable values are the amounts of these two inputs that are unused in the optimal solution. We have an excess supply of labor and upholstery. Thus moderate decreases in the amount of labor or upholstery available will not affect our solution.

What about changes in the input materials we do use, wood and braces? This is a place where the simplex algorithm really shines. First it is convenient to solve for \( x_1 \) and \( x_2 \) in the first and third equations, which involve \( x_1 \) and \( x_2 \), respectively.

\[
x_1 = 300 - \frac{3}{2}x_3 + \frac{1}{2}x_5 \\
x_2 = 275 + \frac{3}{2}x_3 - \frac{1}{2}x_5
\]

(34)

Having 1 less unit of wood (1399 units instead of 1400) is equivalent to increasing the wood slack variable \( x_3 \) from 0 to 1. To determine the effects of 1 less unit of wood, we simply set \( x_3 = 1 \), while \( x_5 = 0 \), in (34). From (34) we have \( x_1 = 300 - \frac{3}{2} \) and \( x_2 = 275 + \frac{1}{2} \). The new value of the objective function with 1 less unit of wood is also obtained by setting \( x_3 = 1 \) in the objective function (while \( x_5 = 0 \)). We have \(-35 + 67,000\).

In summary, the coefficients of \( x_3 \) in (34) and in the objective function give the effect of 1 less unit of wood: chair production will decrease by \( \frac{3}{2} \), table production \( x_2 \) will increase by \( \frac{1}{2} \), and profit will decrease by \$35. If we had 1 more \( x_3 \) (\( x_3 = -1 \)), the opposite occurs. Chair production increases by \( \frac{3}{2} \), table production decreases by \( \frac{1}{2} \), and profit increases by \$35. It is left as an exercise to the reader to evaluate the effects of changing the number of braces.

In economics, the increase in the profit caused by using 1 more unit of an input is called the marginal value of the input. In this case it means that we should be willing to pay \$35 for 1 additional unit of wood because that is the value of wood to us in increasing our sales.

Section 4.6 Exercises

Summary of Exercises

Exercises 1–12 involve converting a "word problem" into a linear program (some of these problems previously appeared in Section 1.4); solutions, if requested, are to be obtained by graphing. Exercises 13–17 require one to solve linear programs. Exercises 18–21 ask for transportation problems to be converted into slack-variable form. Exercises 22–25 involve sensitivity analysis. Exercise 26 illustrates duality theory.
1. Suppose that a building supervisor must hire 18-year-olds and 35-year-olds to do the following jobs: clean 500 windows, empty 800 wastepaper baskets, and mop 8000 square feet of floors. In 1 day, an 18-year-old can clean 50 windows, empty 100 baskets, and mop 500 square feet of floors. A 35-year-old can clean 100 windows, empty 100 baskets, and mop 700 square feet of floors. An 18-year-old gets $40 a day and a 35-year-old gets $55 a day. Formulate the problem of minimizing the cost of hiring workers to do the required work as a linear program. (Set up; do not solve.)

2. The Arizona tile company manufactures three types of tiles, plain, regular, and fancy, in two different factories. Factory A produces 3000 plain, 2000 regular, and 1000 fancy tiles a day and costs $2000 a day to operate. Factory B produces 2000 plain, 4000 regular, and 2000 fancy tiles a day and costs $3000 a day to operate. Write down, but do not solve, a linear program for determining the least-cost way to produce at least 20,000 plain, at least 30,000 regular, and at least 10,000 fancy tiles.

3. A farmer has 400 acres on which he can plant any combination of two crops, barley and rye. Barley requires 5 worker-days and $15 of capital for each acre planted, while rye requires 3 worker-days and $20 of capital for each acre planted. Suppose that barley yields $40 per acre and rye $30 per acre. The farmer has $4000 of capital and 500 worker-days of labor available for the year. He wants to determine the most profitable planting strategy.
   (a) Formulate this problem as a linear program.
   (b) Plot the feasible region and find the corner point that maximizes profit.

4. Suppose that a Bored Motor Company factory requires 7 units of metal, 20 units of labor, 3 units of paint, and 8 units of plastic to build a car, while it requires 10 units of metal, 24 units of labor, 3 units of paint, and 4 units of plastic to build a truck. A car sells for $6000 and a truck for $8000. The following resources are available: 2000 units of metal, 5000 units of labor, 1000 units of paint, and 1500 units of plastic.
   (a) State the problem of maximizing the value of the vehicles produced with these resources as a linear program.
   (b) Plot the feasible region of this linear program and solve by checking the objective function at the corners (by looking at the objective function, you should be able to tell which corners are good candidates for the maximum).

5. Consider the two-refinery problem.

\[
\text{Heating oil: } 20x_1 + 6x_2 = 500 \\
\text{Diesel oil: } 8x_1 + 15x_2 = 750 \\
\text{Gasoline: } 4x_1 + 6x_2 = 1000
\]
Suppose that it costs $30 to refine a barrel in refinery 1 and $25 a barrel in refinery 2. What is the production schedule (i.e., values of \( x_1, x_2 \)) that minimizes the cost while producing \emph{at least} the amounts demanded of each product (i.e., at least 500 gallons of heating oil, etc.)? Solve by the method in Exercise 3.

6. Plot the boundary of the feasible region for the linear program in Example 2 and solve this linear program as discussed in Exercise 3.

7. Formulate the following transportation problem as a linear program (see Example 3); \emph{do not solve}. There are three factories \( A, B, \) and \( C \) that ship motors to three stores 1, 2, and 3. Factory \( A \) makes 1000 motors, \( B \) makes 2000 motors, and \( C \) makes 3000 motors. Store 1 needs 1500 motors, store 2 needs 2000 motors, and store 3 needs 2500 motors. The following matrix gives the costs of shipping a motor from a given factory to a given store.

\[
\begin{array}{ccc}
A & B & C \\
1 & 1 & 2 \\
2 & 1 & 3 \\
3 & 2 & 4 \\
\end{array}
\]

8. There are four boys and four girls and we wish to pair them off in a fashion that minimizes the sum of the personality conflicts in the matches. Entry \((i, j)\) in the following matrix gives a measure of conflict when girl \( i \) is matched with boy \( j \). Set up this problem as a linear problem (see the hint at the end of Exercise 9).

\[
\begin{array}{cccc}
A & B & C & D \\
1 & 2 & 3 & 4 \\
2 & 0 & 3 & 7 \\
C & 2 & 2 & 3 \\
D & 3 & 1 & 5 \\
\end{array}
\]

9. We wish to assign each person to a different job so as to minimize the total amount of time that must be spent to get all the jobs done. Entry \((i, j)\) in the following matrix tells how many hours it takes person \( i \) to do job \( j \) (a dash “—” means that the person cannot do the job). Set up this problem as a linear program.

\[\text{Boys} \quad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 3 \\ 0 & 3 & 7 & 9 \\ 2 & 2 & 3 & 3 \\ 3 & 1 & 5 & 1 \end{bmatrix}\]

\[\text{Girls} \quad \begin{bmatrix} \end{bmatrix}\]

\emph{Hint:} This is a special form of transportation problem in which the “demands” of jobs and “supplies” of people are all equal to 1.
10. We have a farm with 200 cows and capital of $5000 with the following idealized conditions. Cows produce milk for sale or milk to nurse two yearlings (which in a year become cows). A cow can generate $500 worth of milk in a year if not nursing. It costs $300 to feed a cow for a year (no matter what its milk is used for). Write a linear program to maximize the total income over 3 years for the farm (be sure not to spend more money in a year than you currently have).

11. An investor has money-making activities A and B available at the start of each of the next 5 years. Each dollar invested at the start of a year in A returns $1.40 two years later (in time for immediate reinvestment). Each dollar invested in B returns $1.70 three years later. There are in addition activities C and D that are only available once. Each dollar invested in C at the start of the second year returns $2.00 four years later, and each dollar invested in D at the start of the fifth year returns $1.30 in 1 year. The investor begins with $10,000 and she wants an investment plan that maximizes the gain at the end of 5 years. Give a linear program model for this problem.

12. The Expando Manufacturing Co. wishes to enlarge its capacity over the next six periods to produce umbrellas so as to maximize available capacity at the beginning of the seventh period. Each umbrella produced in a period requires d dollars input and one unit of plant capacity; an umbrella yields r dollars revenue at the start of the next period. In each period, Expando can expand capacity using two construction methods, A and B. A requires b dollars per unit and takes one period; B requires c dollars per unit and takes two periods. Expando has D dollars initially to finance production and expansion (in no period can more money be spent than is available). The capacity initially is K. Formulate a linear program to maximize production capacity in period 7.

13. Work through the simplex algorithm for the linear program in Example 5 to verify systems (26) and (27).

14. Solve the following linear programs using the simplex algorithm.
Sec. 4.6 Linear Programming

(a) Maximize $3x_1 + 2x_2$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0 \\
    3x_1 + 4x_2 & \leq 12 \\
    4x_1 + 3x_2 & \leq 12 \\
    x_1 + 2x_2 & \leq 8
\end{align*}
\]

(b) Maximize $4x_1 + 6x_2$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0 \\
    3x_1 + 4x_2 & \leq 12 \\
    x_1 + 2x_2 & \leq 8 \\
    3x_1 + x_2 & \leq 6
\end{align*}
\]

(c) Maximize $x_1 + x_2$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0 \\
    3x_1 + x_2 & \leq 10 \\
    2x_1 + 3x_2 & \leq 15 \\
    2x_1 + 2x_2 & \leq 12
\end{align*}
\]

(d) Maximize $3x_1 + 2x_2$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0 \\
    2x_1 - x_2 & \leq 6 \\
    x_1 + x_2 & \leq 4 \\
    -3x_1 + x_2 & \leq 3
\end{align*}
\]

15. Solve the following linear programs using the simplex algorithm.
(a) Maximize $2x_1 + 3x_2 + 4x_3$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \\
    x_1 + 2x_2 + x_3 & \leq 10 \\
    3x_1 - x_2 + 2x_3 & \leq 12 \\
    x_1 + x_2 + x_3 & \leq 6
\end{align*}
\]

(b) Maximize $3x_1 + x_2 + 2x_3$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \\
    2x_1 + 3x_2 + x_3 & \leq 15 \\
    x_1 + x_2 + 2x_3 & \leq 9 \\
    x_1 + 2x_2 + x_3 & \leq 10
\end{align*}
\]

(c) Maximize $3x_1 + 4x_2 + 2x_3$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \\
    3x_1 + 2x_2 + 4x_3 & \leq 15 \\
    x_1 + 2x_2 + 3x_3 & \leq 7 \\
    2x_1 + x_2 + x_3 & \leq 6
\end{align*}
\]

(d) Maximize $2x_1 + x_2 + x_3$
subject to
\[
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \\
    4x_1 + 3x_2 + 3x_3 & \leq 12 \\
    x_1 + 2x_2 - x_3 & \leq 4 \\
    x_1 + x_2 + 2x_3 & \leq 6
\end{align*}
\]

16. Solve the following linear program using the simplex algorithm.
\[
\text{Maximize } 2x_1 + 4x_2 + x_3 + x_4 \\
\text{subject to} \\
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0 \\
    2x_1 + x_2 + 2x_3 + 3x_4 & \leq 12 \\
    2x_2 + x_3 + 2x_4 & \leq 20 \\
    2x_1 + x_2 + 4x_3 & \leq 16
\end{align*}
\]

17. Solve the following linear program using the simplex algorithm.
\[
\text{Maximize } 15x_1 + 28x_2 + 19x_3 + 24x_4 + 34x_5 \\
\text{subject to} \\
\begin{align*}
    x_1 & \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0 \\
    2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 & \leq 12 \\
    2x_2 + x_3 + 2x_4 + 4x_5 & \leq 20 \\
    2x_1 + x_2 + 4x_3 + 5x_4 + 6x_5 & \leq 16
\end{align*}
\]
\[
x_1 \geq 0, \ x_2 \geq 0, \ x_3 \geq 0, \ x_4 \geq 0, \ x_5 \geq 0
\]
\[
x_1 + 2x_2 + x_3 + 3x_4 + 2x_5 \leq 90
\]
\[
2x_1 + x_2 + x_3 + 4x_4 + x_5 \leq 70
\]
\[
2x_1 + x_2 + x_3 + x_4 + 2x_5 \leq 80
\]
\[
3x_2 + 2x_3 + x_4 + 3x_5 \leq 150
\]

18. Put the transportation problem in Exercise 7 in slack variable form as shown in Example 6. Also express the problem in inequality constraints.

19. Put the dating problem in Exercise 8 in slack-variable form as shown in Example 6. Also express the problem in inequality constraints.

20. (a) Show that the last equation in the transportation problem in Example 6 is redundant by verifying that the sum of the warehouse equations equals the sum of the college equations; hence the sum of the warehouse equations minus the first college equation will equal the second college equation.
   (b) Explain in words why any nonnegative solution to the first four equations in (28) would also have to satisfy the fifth equation.

21. (a) Use the equations in (30) to rewrite the objective function for the transportation problem in Example 6 in terms of just \( x_{11} \) and \( x_{22} \).
   (b) Now solve the transportation problem graphically using the linear program in (31) with the objective function from part (a).

22. Repeat the sensitivity analysis in Example 7 of the chair–table production problem for braces: How would profit change with 1 less unit of braces, how would the number of chairs and number of tables change?

23. Perform sensitivity analysis on the farming linear program in Example 4 of Section 1.4. What are the effects on profit and on amounts of corn and wheat planted if 1 less acre is planted? If 1 less dollar of capital is available?

24. Solve the farming problem in Exercise 3 using the simplex algorithm and perform a sensitivity analysis on all constraints that are fully used. For example, determine the affect of having 1 less dollar of capital.

25. Solve the car–truck production problem in Exercise 4 using the simplex algorithm and perform a sensitivity analysis on all constraints that are fully used.

26. Consider the following two linear programs.
   (i) Maximize \( 3x_1 + 3x_2 \)
   subject to
   \[
x_1 \geq 0, \ x_2 \geq 0
\]
   \[
3x_1 + 2x_2 \leq 10
\]
   \[
4x_1 + x_2 \leq 8
\]
   (ii) Minimize \( 10x_1 + 8x_2 \)
   subject to
   \[
x_1 \geq 0, \ x_2 \geq 0
\]
   \[
3x_1 + 4x_2 \geq 3
\]
   \[
2x_1 + x_1 \geq 3
\]
Solve them by graphing and show that the optimum values of these two objective functions are the same. What relations link the input data in the two problems?

**Appendix to Section 4.6: Linear Programming Details**

**Simplex Algorithm for a General Linear Program**

We use the following notation involving \( n \) variables and \( m \) inequality constraints.

**General Linear Program—Inequality Form**

Maximize \( c_1x_1 + c_2x_2 + \cdots + c_nx_n + d \)

subject to \( x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0 \)

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_2 + \cdots + a_{1h}x_h + \cdots + a_{1n}x_n & \leq b_1 \\
    a_{21}x_1 + a_{22}x_2 + \cdots + a_{2h}x_h + \cdots + a_{2n}x_n & \leq b_2 \\
    & \vdots \\
    a_{g1}x_1 + a_{g2}x_2 + \cdots + a_{gh}x_h + \cdots + a_{gn}x_n & \leq b_g \\
    & \vdots \\
    a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mh}x_h + \cdots + a_{mn}x_n & \leq b_m
\end{align*}
\]

or in matrix notation,

Maximize \( \mathbf{c} \cdot \mathbf{x} + d \)

subject to

\[
\mathbf{Ax} \leq \mathbf{b}, \quad \mathbf{x} \geq 0
\]

Upon restating the inequality constraints in (1a) as equations with slack variables, the system is

**General Linear Program—Equation Form**

Maximize \( c_1x_1 + c_2x_2 + \cdots + c_hx_h + \cdots + c_nx_n + d \)

subject to \( x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0, x_{n+1} \geq 0, \ldots, x_{n+m} \geq 0 \), and

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_2 + \cdots + a_{1h}x_h + \cdots + a_{1n}x_n + x_{n+1} & = b_1 \\
    a_{21}x_1 + a_{22}x_2 + \cdots + a_{2h}x_h + \cdots + a_{2n}x_n + x_{n+2} & = b_2 \\
    & \vdots \\
    a_{g1}x_1 + a_{g2}x_2 + \cdots + a_{gh}x_h + \cdots + a_{gn}x_n + x_{n+g} & = b_g \\
    & \vdots \\
    a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mh}x_h + \cdots + a_{mn}x_n + x_{n+m} & = b_m
\end{align*}
\]
The first $n$ variables are our independent variables from (1a) and the last $m$ variables are the slack variables. *Keep in mind that after iterations of the simplex algorithm, the positions of variables that play the role of slack variables become totally scrambled.* (In linear programming texts, the set of slack variables is called the *basis* of a linear program in equation form.)

If we let $\mathbf{x}^* = [\mathbf{x}, \mathbf{x}']$, where $\mathbf{x}' = [x_{n+1}, \ldots, x_{n+m}]$ is the vector of slack variables, and $\mathbf{A}^* = [\mathbf{A} \ I]$, then (2a) can be written

Maximize $\mathbf{c} \cdot \mathbf{x} + d$
subject to
\[ \mathbf{A}^* \mathbf{x}^* = \mathbf{b}, \quad \mathbf{x}^* \geq 0 \] (2b)

It is important to note that the simplex algorithm assumes that the origin of the current independent variables is a feasible corner. This is the solution obtained by setting all independent variables equal to 0 and setting the slack variable for row $i$ equal to $b_i$. If the origin is not feasible, see the section "Finding an Initial Feasible Solution."

Now we state the general simplex algorithm. In part I we find the independent variable $x_h$ with the largest positive coefficient $c_h$ in the objective function. This is the variable whose increase provides the greatest rate of increase in the objective function. If no coefficient $c_j$ is positive, the current corner is optimal.

Next we determine how much we can increase $x_h$ (while all other independent variables remain $= 0$). The inequalities in (1a) reduce to

\[
\begin{align*}
    a_{1h}x_h &\leq b_1 \Rightarrow x_h \leq \frac{b_1}{a_{1h}} \\
    a_{2h}x_h &\leq b_2 \Rightarrow x_h \leq \frac{b_2}{a_{2h}} \\
    \vdots & \quad \vdots \quad \vdots \\
    a_{ih}x_h &\leq b_i \Rightarrow x_h \leq \frac{b_i}{a_{ih}} \\
    \vdots & \quad \vdots \quad \vdots \\
    a_{nh}x_h &\leq b_m \Rightarrow x_h \leq \frac{b_m}{a_{nh}}
\end{align*}
\] (3)

Then we can increase $x_h$ by the minimum of these bounds. Let $t_h$ be this amount. If some $a_{ih} \leq 0$, then $x_h$ can increase indefinitely without violating the $i$th constraint. So we only want to examine constraints with $a_{ih} > 0$. If there is no positive $a_{ih}$, then $x$ can increase indefinitely without violating any constraint—this "pathological" situation is mentioned later. Summarizing, we have
Part 1 of Simplex Algorithm

1. Let $x_h$ be the independent variable with the largest positive coefficient $c_h$ in the objective function. If all $c_j \leq 0$, the current corner is optimal.
2. Determine the row index $i$ that achieves the minimum value for the ratio $b_i / a_{ih}$ when $a_{ih} > 0$. Let $g$ be the minimizing $i$.
3. (a) If all $a_{ih} \leq 0$, $x_h$ can be increased infinitely and the problem is unbounded. See Abnormal Possibilities II below.
   (b) If the minimizing ratio $b_i / a_{gh}$ equals 0, that is, $b_i = 0$ (and every other row $i$ has $b_i \leq 0$ or $a_{ih} \leq 0$), special steps must be taken. See Abnormal Possibilities III below.

The intersection of the $g$th constraint with the constraints $x_j = 0$, $j \neq h$, forms the new corner $(0, 0, \ldots, t_h, \ldots, 0)$.

Part 2 of the simplex algorithm requires us to rewrite the equations in (2) to make this new corner become the origin of the new independent variables. We do this by performing a pivot exchange between independent variable $x_h$ and the slack variable $x'_g$ in equation $g$; that is, we pivot on entry $a_{gh}$.

Part 2 of Simplex Algorithm: Pivoting on $a_{gh}$

1. The old $g$th constraint equation is rewritten to make $x_h$ become a slack variable, that is, make $x_h$ have coefficient 1. This is accomplished by dividing the equation by $a^*$.
2. Set the column of coefficients for $x_h$ equal to $e_g$ (all 0's except in the $g$th equation). This is accomplished by the standard elimination by pivoting process.
3. The objective function undergoes the same change as in step 2 ($x_h$ drops out and $x'_g$ comes in).

The following diagram illustrates steps 1, 2, and 3. Suppose that $a^* = a_{gh}$ is again the pivot entry, $p = a_{gj}$ is another coefficient in the pivot row, $q = a_{ij}$ is another coefficient in the pivot column, and $r = a_{ij}$ is a coefficient in $p$'s row and $q$'s column (possibly $q$ and $r$ are in the objective function, or possibly $p$ and $r$ are $b_j$'s). Then part 2 of the simplex algorithm has the form

Columns

<table>
<thead>
<tr>
<th>Before pivoting</th>
<th>$h$</th>
<th>$j$</th>
<th>$g'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g$</td>
<td>$\ldots$</td>
<td>$a^*$</td>
<td>$\ldots$</td>
</tr>
<tr>
<td>$i$</td>
<td>$\ldots$</td>
<td>$q$</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>
If \( r \) is the constant \(-d\) in the objective function, pivoting changes \(-d\) to \(-d - pq/a^*\) (the objective function constant increases from \(d\) to \(d + pq/a^*\)). In the display above we have listed column \( g' \), the column of the new independent variable \( x'_g \), which previously was the slack variable in row \( g \).

After pivoting, we return to part 1 of the simplex algorithm for the linear program in this new form and continue doing part 1 and part 2 until no improvement in the objective function is possible in part 1. Then we shall have found the maximum.

### Abnormal Possibilities

**No Feasible Points.** The feasible region for a linear program may be empty. That is, no point lies on the correct side of all the inequalities, for example, to satisfy \( x_1 + x_2 \leq -1 \), either \( x_1 \) or \( x_2 \) must be negative—but \( x_j \geq 0 \) is required in all linear programs. A linear program is called *infeasible* in this case.

**Unbounded Feasible Region.** When we increase \( x_h \) in part 1, perhaps \( x_h \) can be increased without limit (to infinity). See step 3(a) of part 1 in the simplex algorithm. This means that the feasible region is unbounded along the \( x_h \)-axis and the maximum value of the objective function is infinite. If a practical problem has been misformulated (or data not entered correctly), this difficulty can arise. When this happens, the linear program is called *unbounded*.

**Degeneracy.** In very rare cases, it can happen that in part 1 replacing an independent variable by a slack variable does not increase the objective function because some constraint has a zero right-hand side (see step 3(b) of part 1 in the simplex algorithm). This can happen without being at an optimal point. This phenomenon is called *degeneracy*. Since it is rare and requires special methods to handle, we shall ignore degeneracy in this book.
How do we start the simplex algorithm if the origin [setting all the independent variables in (1a)] is not a corner of the feasible region? For example, consider a two-variable problem with constraints

\[ \begin{align*}
2x_1 + x_2 & \geq 4 \\
x_1 - 3x_2 & \leq -6
\end{align*} \]  

(6)

or, multiplying the first inequality by \(-1\),

\[ \begin{align*}
-2x_1 - x_2 & \leq -4 \\
x_1 - 3x_2 & \leq -6
\end{align*} \]  

(7)

Neither of these inequalities is satisfied by \(x_1 = x_2 = 0\). Let us put the inequalities of (7) in equation form with slack variables \(x_3, x_4\).

\[ \begin{align*}
-2x_1 - x_2 + x_3 & = -4 \\
x_1 - 3x_2 + x_4 & = -6
\end{align*} \]  

(8)

As in (7), setting \(x_1 = x_2 = 0\) cannot yield a solution of (8), since we require \(x_3 \geq 0, x_4 \geq 0\).

There is a standard "trick" for finding a starting feasible corner (and associated set of independent variables) when the origin, \(x_1 = x_2 = 0\), is infeasible. We add a new equality-violation variable on the left-hand side of each constraint with a negative right-hand side. For the system (8), we introduce \(x_5, x_6\).

\[ \begin{align*}
-2x_1 - x_2 + x_3 - x_5 & = -4 \\
x_1 - 3x_2 + x_4 - x_6 & = -6
\end{align*} \]  

(9)

where \(x_5 \geq 0, x_6 \geq 0\). Let us multiply (9) by \(-1\) to get

\[ \begin{align*}
2x_1 + x_2 - x_3 + x_5 & = 4 \\
x_1 + 3x_2 - x_4 + x_6 & = 6
\end{align*} \]  

(10)

Now \(x_1 = x_2 = x_3 = x_4 = 0\) yields a feasible solution to (10), with \(x_5 = 4\) and \(x_6 = 6\). We can apply the simplex algorithm to (10).

Next we define a contrived objective function that makes us look for a corner where the equality-violation variables become zero, that is, a corner satisfying our original equations (8). The linear program we want to solve, with the simplex algorithm, is
Maximize \(- x_5 - x_6\)  
subject to (10) and \(x_i \geq 0\)  

Since \(x_5, x_6\) are nonnegative, the best possible maximum for (11) would be 0, occurring when \(x_5 = x_6 = 0\). The values of \(x_1, x_2, x_3,\) and \(x_4\) when \(x_5 = x_6 = 0\) will be the starting feasible solution we need to use the simplex algorithm on the original problem. If we solve the linear problem (11) and do not get a maximum of 0, then the original system of constraints was infeasible.

Note that since \(x_5\) and \(x_6\) are slack variables, we must rewrite the objective function in (11) in terms of the independent variables \(x_1, x_2, x_3, x_4\). Reading off expressions for \(x_5\) and \(x_6\) from (10), we have the linear program

\[
\text{Maximize } (2x_1 + x_2 - x_3 - 4) + (-x_1 + 3x_2 - x_4 - 6) \\
= x_1 + 4x_2 - x_3 - x_4 - 10
\]

subject to (10) and \(x_i \geq 0\)  

Matrix Representation of Pivoting and Revised Simplex Method

The pivoting operation of the simplex algorithm can be expressed as a matrix product. Let the linear program be written in equation form with slack variables as in (2):

Maximize \(c \cdot x + d\)  
subject to \(A^* x^* = b, \ x^* \geq 0\)

We can obtain the new coefficient matrix after pivoting on entry \(a_{gh}\) as the matrix product \(PA^*\), where \(P\) is the \(m\)-by-\(m\) "pivot" matrix. Since \(A^* = [A \ I]\), where \(I\) is the \(m\)-by-\(m\) identity matrix, we can write

\[
PA^* = P[A \ I] = [PA \ P]
\]

From (13) we see that the matrix \(P\), assuming that \(P\) exists, is just the \(m\)-by-\(m\) submatrix formed by the last \(m\) columns (the columns of the slack variables) after pivoting is performed. As noted earlier, the columns in the slack-variable submatrix are unchanged by pivoting except for column \(n + g\) [which is as shown in (5)]. Thus if \(a^* = a_{gh}\) is the pivot entry, then
P is the identity matrix with column g replaced by the hth column of A* divided by \( a^* \), except that entry \((g, g)\) of P is \( 1/a^* \), the inverse of the pivot value \( a^* = a_{gh} \).

As a check, let us compute entry \((i, j)\), \( j \neq g \), in the product \( PA \). This entry will be the scalar product \( p_i \cdot a_j \) (where \( p_i \) denotes the ith row of P and \( a_j \) the jth column of \( A^* \)). Row vector \( p_i \) has just two nonnegative entries, \( p_{ii} = 1 \) and \( p_{ig} = -a_{ih}/a^* \). Thus

\[
\begin{align*}
\text{entry } (i, j) \text{ of } PA &= p_i \cdot a_j = p_{ii}a_{ij} + p_{ig}a_{gj} \\
&= 1a_{ij} - \left( \frac{a_{ih}}{a^*} \right) a_{gj} \\
&= a_{ij} - \frac{a_{gj}a_{ih}}{a^*}
\end{align*}
\]

This expression corresponds to the value of entry \((i, j)\) in table (5), since \( a_{ij} = r, a_{gj} = p \) and \( a_{ih} = q \). So P is, as advertised, the desired pivot matrix.

As mentioned above for table (5), the column vector \( b \) and objective row vector \( c \) are changed in pivoting the same way \( A^* \) is. So we can expand \( A^* \) into an \((m + n + 1)\)-by-\((m + 1)\) matrix \( A^+ \) of all the data.

\[
A^+ = \begin{bmatrix} c & 0 & -d \\ A & 1 & b \end{bmatrix}
\]
(The minus sign for \( d \) is a technicality based on the fact that the constant in the objective function is not on the right-hand side of an equation, the way the \( b_i \) are.) The \((m + 1)\)-by-\( m \) pivot matrix \( P^+ \) for \( A^+ \) has a corresponding additional top row with \(-ch/a^*\) in its \( g \)th entry and 0's elsewhere.

Let us use the notation \( P_{gh} \) to denote the pivot matrix when we pivot on entry \((g, h)\) (to exchange independent variable \( x_h \) and the slack variable for row \( g \)). Then a sequence of \( k \) pivots on entries \((g_1, h_1), (g_2, h_2), \ldots, (g_k, h_k)\) would produce the new linear program with matrix \( A_k^+ \):

\[
A_k^+ = P_{g_1h_1} P_{g_2h_2} \cdots P_{g_kh_k} A^+
\]  

(17)

The revised simplex algorithm uses (17) to compute just the cost coefficients \( c_j \) in \( A_k^+ \), needed to select the pivot column. We determine which column has the largest \( c_j \); call its index \( h' \). Next (17) is used to compute \( b \) and the \( h' \)th column of the constraint matrix in \( A_k^+ \), needed to select the pivot row. The other entries in \( A \) are never computed. The pivot row is the row that achieves the minimum positive value of \( b_j/a_{jh} \); call it row \( g' \). The next pivot matrix \( P_{g'h'} \) can be constructed using the entries of the \( h' \)th column [see (14)]. This completes one iteration of the revised simplex algorithm.

The pivot matrices are easy to store and multiply. Only column \( g' \) needs to be stored, and even this vector is likely to be very sparse, since the coefficient matrices in large linear programs are very sparse. Because we only calculate one row and two columns of the current linear program, a tremendous savings in time is achieved over the standard simplex algorithm.

Section 4.7 Linear Models for Differentiation and Integration

Although calculus deals with highly nonlinear functions, both the theory and computations associated with calculus are built on linear models. It is the computation that will be our primary interest in this section. We will show how linear approximations to arbitrary functions allow one to solve numerically almost any calculus or differential equation problem.

We first point out the central role of linearity in calculus. The derivative of a function is the essence of a linear model. The derivative \( f'(x) \) of a function \( y = f(x) \) at a point \((x_0, y_0)\) is the slope of \( f(x) \) at \((x_0, y_0)\). That is, \( f'(x_0) \) gives the slope of a line through \((x_0, y_0)\) that coincides with \( f(x) \) when \( x \) is very close to \( x_0 \) (see Figure 4.14). In other words, the derivative gives the slope of a linear approximation to \( f(x) \) at a point.

An equally important aspect of linearity in calculus is the fact that differentiation is a linear operation, in the sense that if \( f(x) = a g(x) + b h(x) \), where \( a, b \) are constants, then \( f'(x) = a g'(x) + b h'(x) \) [the same way that \( A(au + bv) = aAu + bAv \)]. For example, we compute the derivative of \( f(x) = 5x^3 + 6x^2 \) by knowing that \( 3x^2 \) is the derivative of \( x^3 \) and \( 2x \) is the derivative of \( x_2 \) and then \( f'(x) = 5(3x^2) + 6(2x) \). Integration
Sec. 4.7 Linear Models for Differentiation and Integration

Figure 4.14 Slope of line tangent to $y = f(x)$ is the derivative.

is also a linear operation. Without this linearity, computing the derivative or integral of a polynomial would be a very complicated process.

We now show how to use the linear approximation of a function given by the derivative to build an iterative scheme to find zeros of a function. The scheme is called **Newton’s method**. For $x$-values close to $x = x_0$, we have

$$f(x) = f'(x_0)x + c$$

for an appropriate constant $c$ \hspace{1cm} (1)

Pretend that (1) is a good approximation for $f(x)$ over a wide interval around $x = x_0$. We shall use (1) to approximate where $f(x)$ has a zero [where $f(x) = 0$]. If $f(x)$ has value $f(x_0)$ at $x_0$ and has slope $f'(x_0)$, then the linear approximation (1) decreases by $f'(x_0)$ for each unit we decrease $x$ and will be zero if we decrease $x$ by $f(x_0)/f'(x_0)$. Thus we estimate the $x$-value $x_1$ that makes $f(x) = 0$ to be

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

(see Figure 4.15a). Note that if $f(x_0)/f'(x_0) < 0$, then $x_1 > x_0$.

Normally, $f(x_1) \neq 0$, because the derivative $f'(x_0)$ only approximates the slope of $f(x)$ when $x$ is very near $x_0$. However, there is a fair chance that $x_1$ is close to a zero of $f(x)$. Let us use the approximation (1) again, now at the point $x_1$, to estimate where $f(x)$ is zero. Similar to (2), we get

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

We continue in this method to approximate the true zero of $f(x)$. The general formula is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

(3)

We stop when $f(x_n)$ is very close to 0. Once we get an $x$-value close to a zero of $f(x)$, the approximation (1) is quite accurate and our method converges quickly to a zero of $f(x)$. 
We should mention that if one cannot compute $f'(x)$ by a differentiation formula, then $f'(x)$ must be estimated by the approximation $[f(x + h) - f(x)]/h$ (for some small $h$).

**Example 1. Newton’s Method**

Consider the function $f(x) = x^5 - 5x^4 - 4x^3 - 3x^2 - 2x - 1$. We use Newton’s method to find a zero of this function. First we note that this type of problem arises in determining eigenvalues. Suppose that we want to find the eigenvalues of a matrix $A = \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix}$. Then we must find the zeros of $\det(A - \lambda I)$.
\[
\det(A - \lambda I) = \begin{vmatrix} 3 - \lambda & 2 \\ 1 & 2 - \lambda \end{vmatrix} = (3 - \lambda)(2 - \lambda) - 2 \cdot 1 = \lambda^2 - 5\lambda + 4
\]

For this \( f(x), f'(x) = 2x - 5 \). Suppose that we start with \( x_0 = 10 \). We calculate that \( f(10) = 54 \) and \( f'(10) = 15 \). So Newton’s method estimates the zero to be at

\[
x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 10 - \frac{54}{15} = 6.4
\]

Next we calculate \( f(6.4) = 12.96 \) and \( f'(6.4) = 7.8 \). Then

\[
x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 6.4 - \frac{12.96}{7.8} = 4.74
\]

Calculating \( f(4.74) = 2.77 \) and \( f'(4.74) = 4.48 \) gives us

\[
x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 4.74 - \frac{2.77}{4.48} = 4.12
\]

and

\[
x_4 = x_3 - \frac{f(x_3)}{f'(x_3)} = 4.12 - \frac{.37}{3.24} = 4.006
\]

\[
x_5 = x_4 - \frac{f(x_4)}{f'(x_4)} = 4.006 - \frac{.018}{3.012} = 4.000024
\]

Clearly, we are converging to 4. That is, \( f(4) = 0 \).

**Example 2. Bad Performance by Newton’s Method**

Use Newton’s method to find a zero of the function \( y = x^5 - 5x^4 - 4x^3 - 3x^2 - 2x - 1 \). We have plotted this function in Figure 4.15b with the y-axis magnified near the origin. This curve gets close to a zero at \( x = -.4 \) but only has a relative maximum with \( y = -.56 \). It decreases awhile and then increases sharply with \( f(5.5) = -310, f(6) = 300 \), and \( f(7) = 3000 \). If we start Newton’s method with \( x_0 = 0 \), we get the sequence of points shown in Table 4.6.

We see that our sequence of points gets caught around the relative minimum at \( x = -.4 \) and tends to swing back and forth from one side of \( -.4 \) to the other. It finally gets very near the minimum on the 107th iteration. Here the slope \( f'(x_{107}) \) is almost zero, so dividing by \( f'(x_{107}) \) (\( = .04 \)) in (3) brings a large change in \( x \) that gets us away from this

If we had started with $x_0 > 5$, the procedure would have converged quickly. Although it did finally escape from the region of the relative minimum, a better scheme would have been to pick a new starting value when the method had not converged after 20 iterations.
Next let us consider the definition of the integral of a function $f(x)$. The integral of $f(x)$ is the area under a curve from 0 to $x$ (sometimes 0 is replaced by another $x$-value $x_0$). The formal definition of an integral is the limit of the sum of areas of approximating rectangles

$$
\int_0^x f(x) \, dx = \lim_{n \to \infty} \sum_{i=1}^{n} (t_i - t_{i-1}) f(t_i)
$$

(4)

where $t_i = i(x/n)$. The $t_i$ subdivide the interval $[0, x]$ into equal subintervals of length $x/n$, with $t_0 = 0$ and $t_n = x$ (see Figure 4.16a). We are approximating the area under the curve $f(x)$ with the area under the piecewise approximation to $f(x)$:

$$
f_n(x) = f(t_i) \quad \text{when } t_{i-1} < x \leq t_i, \quad i = 1, 2, \ldots, n
$$

(5)

That is, the sum on the right-hand side of (4) is the area under $f_n(x)$. The function $f_n(x)$ is a piecewise constant function, since it is made up of constant functions with a different constant on each subinterval of $[0, x]$.

Figure 4.16 (a) Set of rectangles whose areas approximate the area under the curve. (b) Piecewise linear approximation to curve $f(x)$ in part (a). Set of trapezoids approximate the area under $f(x)$. 

(a) 

(b)
The points \( t_i \) in (4) are called \textbf{mesh points}. The collection of mesh points and the subintervals they form is called the \textbf{mesh}. As the mesh becomes finer (as \( n \) increases), it is intuitively clear that the area under \( f_n(x) \) will approach the area under the curve \( f(x) \).

Let us consider how we might try to calculate the area under a function \( f(x) \) from 0 to \( x^o \) when we do not know how to integrate exactly [i.e., no integration formulas apply to \( f(x) \), and integration by parts, etc., all fail]. One obvious approach is to use the definition of the integral given in (4). That is, we pick some number of mesh points, say \( n = 50 \), for the interval \((0, x^o)\). Then we calculate the value of \( f(x) \) at the mesh points and compute the sum on the right-hand side of (4). A better estimate for the area under \( f(x) \) should be obtained by using linear approximations to \( f(x) \) in each subinterval that agree with the values of \( f(x) \) at the subinterval endpoints (see Figure 4.16b). It is left to the reader to check that the following piecewise linear approximation does this.

\[
g_n(x) = \frac{f(t_i) - f(t_{i-1})}{t_i - t_{i-1}} (x - t_{i-1}) + f(t_{i-1})
\]

for \( t_{i-1} < x \leq t_i \), \( i = 1, 2, \ldots, n \) \hspace{1cm} (6)

By looking at Figure 4.16b, one sees that the region in each subinterval using (6) is a trapezoid. The area of this trapezoid is the same as the area of a rectangle with height halfway between \( f(t_{i-1}) \) and \( f(t_i) \). That is, using (6) gives the same area as using the piecewise constant function

\[
g_n^*(x) = \frac{f(t_i) + f(t_{i-1})}{2}
\]

for \( t_{i-1} < x \leq t_i \) \hspace{1cm} (7)

Let \( h = t_i - t_{i-1} = x^o/n \). Then the integral of \( f(x) \) from 0 to \( x^o \) is approximated by the area under \( g_n^*(x) \),

\[
\int g_n^*(x) \, dx = h \sum_{i=1}^{n} \frac{f(t_i) + f(t_{i-1})}{2}
\]

\[
= \frac{hf(t_0)}{2} + h \sum_{i=1}^{n-1} f(t_i) + \frac{hf(t_n)}{2}
\]

Using (8) to approximate an integral is an integration scheme called the \textbf{trapezoidal rule} (named after the trapezoids in Figure 4.16b). Note that this rule weights the two endpoint values \( f(t_0) \) and \( f(t_n) \) half as much as the other \( f(t_i) \). Various schemes have been developed that give different sets of weights to the \( f(t_i) \).

\textbf{Example 3. Piecewise Approximation of Area Under Curve}

Consider the function \( f(x) = 1/(x^2 + 1) \). We want to compute the area under \( f(x) \) on the interval \([0, 2] \) using the piecewise approximations given in (5) and (6). To make calculations easy, let the mesh
points be $t_0 = 0, t_1 = .5, t_2 = 1, t_3 = 1.5, t_4 = 2$. Then we get the table of function values

<table>
<thead>
<tr>
<th>$t_i$</th>
<th>0</th>
<th>.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(t_i)$</td>
<td>1</td>
<td>.8</td>
<td>.5</td>
<td>-.3</td>
<td>.2</td>
</tr>
</tbody>
</table>

The approximation of $f(x)$ by the function in (5),

$$f_4(x) = f(t_i) \quad \text{for } t_{i-1} \leq x \leq t_i, \quad i = 1, 2, 3, 4$$

gives

$$f_4(x) = \begin{cases} 
.8 \text{ on } [0, .5] \\
.5 \text{ on } [.5, 1] \\
.3 \text{ on } [1, 1.5] \\
.2 \text{ on } [1.5, 1]
\end{cases}$$

and the approximation of $f(x)$ by the function in (6),

$$g_4(x) = \frac{f(t_i) - f(t_{i-1})}{t_i - t_{i-1}} (x - t_{i-1}) + f(t_{i-1})$$

for $t_{i-1} < x \leq t_i, \quad i = 1, 2, 3, 4$
gives

$$g_4(x) = \begin{cases} 
-.4x + 1 \text{ on } [0, .5] \\
-.6x + .8 \text{ on } [.5, 1] \\
-.4x + .5 \text{ on } [1, 1.5] \\
-.2x + .3 \text{ on } [1.5, 2]
\end{cases}$$

The area under $f_4(x)$ equals $.5(.8 + .5 + .3 + .2) = .9$ and the area under $g_4(x)$ equals [using (8)] $.5\{1/2 + (.8 + .5 + .3) + .2/2\} = 1.2$. The actual area under $1/(x^2 + 1)$ from 0 to 2 is about 1.11. So, as expected, $g_4(x)$ led to a better estimate of the area.

In (5) we approximated $f(x)$ with a piecewise constant function $f_n(x)$, and in (6) we approximated $f(x)$ with a piecewise linear function $g^*_n(n)$. A better approximation to the area under $f(x)$ can be sought by using piecewise approximations to $f(x)$ that are quadratic or cubic functions. It turns out that cubic functions have several good properties that quadratics lack, so piecewise cubic functions are frequently used to approximate functions in integration and other calculations. The cubic function in the $i$th mesh interval would be

$$s_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \quad \text{for } t_{i-1} \leq x \leq t_i$$
We want each cubic piece to equal \( f(x) \) at the mesh points \( t_{i-1} \) and \( t_i \), that is,

\[
s_i(t_{i-1}) = f(t_{i-1}) \quad \text{and} \quad s_i(t_i) = f(t_i), \quad i = 1, 2, \ldots, n \quad (12)
\]

Another desirable property for an approximating function is to be smooth at the mesh points \( t_i \), where the pieces meet. By this we mean that the first and second derivatives of two successive cubics coincide at their common mesh point. Then we require

\[
s'_i(t_i) = s'_{i+1}(t_i) \quad \text{and} \quad s''_i(t_i) = s''_{i+1}(t_i) \quad \text{for } i = 1, 2, \ldots, n - 1 \quad (13)
\]

The term \textit{spline} is the name given to a smooth piecewise function. Cubic splines are used to approximate curves in thousands of applications. For example, automotive designers use splines to approximate car contours in computer models that simulate a car's wind resistance. Splines are widely used in computer graphics to generate the forms of complex figures; they are used to join points made with a light pen in tracing out a figure on a terminal screen. Once a figure is represented by mathematical equations, it is an easy matter to rotate, shrink, and perform other transformations of the figure, as discussed in Section 4.1.

The problem with splines is determining the coefficients in each of the cubic pieces. However, the system of equations (12) and (13) determining the coefficients in the splines can be collapsed to a tridiagonal system of \( n - 1 \) equations in \( n - 1 \) unknowns. See the Appendix to this section for details. Such tridiagonal systems can be solved very quickly (see Section 3.5), so spline approximations can be computed very quickly.

Another important approach, called functional approximation, for approximating a function for integration and other purposes is to use one linear combination of nice functions (e.g., whose integrals are easily computed) to approximate \( f(x) \) over the entire interval from 0 to \( x^0 \). This approach is discussed in Section 5.4.

We next consider linear approximations to solutions of differential equations. The derivative \( f'(x) \), the slope of function \( f(x) \) at \( x \), is approximated by

\[
f'(x) \approx \frac{f(x + h) - f(x)}{h} \quad (14)
\]

Indeed, the limit of (14) as \( h \) goes to zero is by definition \( f'(x) \). First and second derivatives of complicated functions are often needed in differential equations computations.

\textbf{Example 4. Discrete Approximation to a Differential Equation}

Consider \( y(x) \), the temperature of a rod, as a function of \( x \), the distance from the left end of the rod. If a heat source is applied to the rod with
a temperature \( f(x) \) at position \( x \), the following differential equation describes how \( f(x) \) affects \( y(x) \).

\[
\frac{d^2y}{dx^2} = -f(x) \quad 0 \leq x \leq d \quad (d = \text{length of rod}) \quad (15)
\]

For example, \( f(x) \) might be zero everywhere except at two short segments of the rod. It is too difficult to solve this differential equation analytically for most interesting \( f(x) \). Instead, one seeks an approximate solution by computing values of \( y(x) \) at a set of mesh points \( x_0, x_1, \ldots, x_n \) on the interval \((0, d)\). If \( h = x_i - x_{i-1} \), then at \( x = x_i \), (14) becomes

\[
y'(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} \quad (16)
\]

We could just as well use

\[
y'(x_i) = \frac{y(x_i) - y(x_{i-1})}{h} \quad (16')
\]

Next, using the fact that \( y''(x) \) is the derivative of the derivative, we can estimate \( y''(x_i) \) using (16) or (16') twice. To make the result symmetric about \( x_i \), we use (16) for \( y'(x_i) \) and (16') to get \( y''(x_i) \).

\[
y''(x_i) = \frac{\{y'(x_i) - y'(x_{i-1})\}}{h} = \frac{\{[y(x_{i+1}) - y(x_i)]/h - [y(x_i) - y(x_{i-1})]/h\}}{h} = \frac{\{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})\}}{h^2} \quad (17)
\]

Substituting (17) into (15), we obtain a system of equations for the values \( y(x_i) \). Letting \( y_i = y(x_i) \) and \( f_i = f(x_i) \), we have

\[
\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = -f_i \quad i = 1, 2, \ldots, n - 1 \quad (18)
\]

We have \( n - 1 \) equations for the \( n + 1 \) unknowns \( y_0, y_1, \ldots, y_n \). Like the differential equations in Section 3.3, we need to specify two starting, or boundary, conditions. Typically, these have the form \( y_0 = a \) and \( y_n = b \). Suppose, for later reference, that we choose the boundary conditions
The physical interpretation of (19) is that both ends of the rod are attached to objects that dissipate away all the heat at the ends. Now we drop \( y_0 \) and \( y_n \) from (18), since by (19) they are 0. This reduces (18) to a system of \( n - 1 \) equations in \( n - 1 \) unknowns.

In matrix terms, (18) becomes

\[
Dy = f^* 
\]

(20)

where \( y = (y_1, y_2, \ldots, y_{n-1}) \)—remember that \( y_0 \) and \( y_n \) are dropped, since they are 0—\( f^* = (h^2f_1, h^2f_2, \ldots, h^2f_{n-1}) \), and \( D \) is

\[
D = 
\begin{bmatrix}
-2 & 1 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 \\
0 & 0 & 1 & -2 & 1 \\
1 & -2 & 1 & -2 & 1 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & -2 & 1 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 \\
0 & 0 & 1 & -2 & 1 \\
0 & 0 & 0 & 1 & -2 \\
\end{bmatrix}
\]

(21)

Observe that \( D \) is a tridiagonal matrix, so \( Dy = f \) can be solved very quickly by Gaussian elimination (see Section 3.5). Let us use (20) to get an approximate solution to our differential equation (15) on the interval \([0, 1]\) with 100 mesh points. So \( n = 100 \) and \( h = 1/n = .01 \). Suppose that \( f(x) \) represents a heat source applied to a point at the middle of the rod:

\[
f_{50} = 10,000 \quad \text{and} \quad \text{all other } f_i = 0 
\]

(22)

Then, if \( f^*, -h^2f_{50} = -(.01)^2 \times 10,000 = -1 \). So \( f^* \) is all 0’s except -1 in its fiftieth entry.

Applying Gaussian elimination to (21) plus the right-side vector \( f^* \), we obtain the following matrix.
Back substitution yields

\[ y_k = \frac{k}{2} \quad k = 1, 2, \ldots, 49 \]  
\[ y_k = \frac{100 - k}{2} \quad k = 50, 51, \ldots, 99 \]  

The continuous function \( y(x) \) that (24) approximates is clearly

\[ y(x) = \begin{cases} 
50x, & 0 \leq x \leq .5 \\
50 - 50x, & .5 \leq x \leq 1 
\end{cases} \]  

The solution (25) says that if the middle of the rod is heated and the ends are kept at temperature 0, the temperature will decrease at a uniform rate along the rod toward the ends (as opposed to an exponential decay or some other nonlinear decrease). This uniform decrease is indeed what occurs in nature.

By letting the number of mesh points grow very large, our solution vector \( y \) in (24) becomes a better and better approximation to the true solution \( y(x) \). In the limit, \( y \) becomes an infinite vector that "is" \( y(x) \). Thus any continuous function can be thought of as an infinite-length vector. In Section 5.4 we show another way to view continuous functions as vectors.

Most methods of solving differential equations by discrete approximation are called finite difference schemes. The other basic approach to approximating solutions to differential equations is finite element methods. Here the function \( f(x) \) is approximated as a sum of special functions for which the differential equation is easily solved.
Section 4.7 Exercises

Summary of Exercises
Exercises 1–3 involve finding zeros with Newton’s method. Exercises 4–9 involve approximating areas under curves. Exercises 10 and 11 involve finite difference methods for solving differential equations. Problems about splines appear in the Exercises in the appendix to this section.

1. Use Newton’s method to find a zero of the following functions; let \( x_0 = 2 \).
   (a) \( f(x) = 3x - 5 \)
   (b) \( f(x) = x^2 + 6x + 5 \)
   (c) \( f(x) = x^3 - 3x^2 + 3x - 1 \)
   (d) \( f(x) = x^3 + x^2 - 2x \)
   (e) \( f(x) = \cos(x) \) (x in radians)
   (f) \( f(x) = x^2 - 2x + 8 \)
   (g) \( f(x) = x^4 - 4x^3 + 5x^2 - x - 15 \)
   (h) \( f(x) = x^3 + 4x^3 - 8x - 3 \)

2. Use Newton’s method to find all zeros of the functions in Exercise 1 in the interval from \(-2\) to 6.
   Hint: If \( \alpha \) is a zero of \( f(x) \), the other zeros of \( f(x) \) are also zeros of \( f(x)/(x - \alpha) \).

3. Use Newton’s method to find all eigenvalues of the following matrices. (See the hint in Exercise 2 and see Section 3.1 for determinant formulas.)
   (a) \[
   \begin{bmatrix}
   4 & 0 \\
   2 & 2
   \end{bmatrix}
   \]
   (b) \[
   \begin{bmatrix}
   4 & -1 & 0 \\
   1 & 0 & 2 \\
   0 & 2 & 1
   \end{bmatrix}
   \]
   (c) \[
   \begin{bmatrix}
   0 & 4 & 1 \\
   .4 & 0 & 0 \\
   0 & .6 & 0
   \end{bmatrix}
   \]
   (d) \[
   \begin{bmatrix}
   0 & 0 & 4 & 8 \\
   .5 & 0 & 0 & 0 \\
   0 & .5 & 0 & 0 \\
   0 & 0 & .5 & 0
   \end{bmatrix}
   \]

4. (a) Use the trapezoidal rule, equation (8), to estimate the area under the curve \( f(x) = x^2 - 2x + 1 \) from 0 to 4 with the mesh \{0, 1, 2, 3, 4\}. Determine the area exactly by integration. Also plot this function and the approximation given by the piecewise linear function in (6).
   (b) See how accurate your answer gets with a denser mesh. Compute with the trapezoidal rule again with meshes.
   (i) \{0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4.0\}
   (ii) \{0, .25, .5, .75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4\}.
5. (a) Use the trapezoidal rule, equation (8), to estimate the area under the curve \( f(x) = x^3 - 2x + 4 \) from 0 to 4 with the mesh \( \{0, 1, 2, 3, 4\} \). Determine the area exactly by integration. Also plot this function and the approximation given by the piecewise linear function in (6).

(b) See how accurate your answer gets with a denser mesh. Compute with the trapezoidal rule again with the mesh \( \{0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4\} \).

6. (a) Use the trapezoidal rule, equation (8), to estimate the area under the curve \( f(x) = e^x \) from 0 to 4 with the mesh \( \{0, 1, 2, 3, 4\} \). Determine the area exactly by integration. Also plot this function and the approximation given by the piecewise linear function in (6).

(b) See how accurate your answer gets with a denser mesh. Compute with the trapezoidal rule again with meshes:
   (i) \( \{0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4\} \)
   (ii) \( \{0, .25, .5, .75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4\} \)

7. (a) Use the trapezoidal rule, equation (8), to estimate the area under the curve \( f(x) = 10/(x + 1)^2 \) from 0 to 4 with the mesh \( \{0, 1, 2, 3, 4\} \). Determine the area exactly by integration. Also plot this function and the approximation given by the piecewise linear function in (6).

(b) See how accurate your answer gets with a denser mesh. Compute with the trapezoidal rule again with meshes:
   (i) \( \{0, .5, 1, 1.5, 2, 2.5, 3, 3.5, 4\} \)
   (ii) \( \{0, .25, .5, .75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4\} \)

8. (a) Use the trapezoidal rule, equation (8), to estimate the area under the curve \( f(x) = \sin(2\pi/x) \) from \( \frac{1}{2} \) to 1 with the mesh \( \{.25, .5, .75, 1\} \). Also plot this function and the approximation given by the piecewise linear function in (6). This function cannot be integrated by any standard integration technique.

(b) See how much your answer changes with a denser mesh. Compute with the trapezoidal rule with the variable mesh \( \{.25, .26, .27, .28, .29, .3, .32, .34, .36, .39, .42, .45, .5, .6, .7, .8, .9, 1\} \).

9. Simpson’s rule approximates an integral by

\[
\int_a^b f(x) \, dx = \frac{h}{3} \left\{ f(a) + 4f(t_1) + 2f(t_2) + 4f(t_3) + 2f(t_4) + \cdots + 4f(t_{n-1}) + f(b) \right\}
\]

where \( t_i = a + hi, \ i = 1, 2, \ldots, n - 1, \ n \) even, and \( h = (b - a)/n \).
(a) Simpson’s rule finds the exact integral of quadratic functions. Verify this by using Simpson’s rule to approximate the given integrals and check by formal integration.

(i) \( f(x) = x^2 + 1 \) from 0 to 1 with \( n = 4 \).

(ii) \( f(x) = x^2 - 2x + 4 \) from 1 to 3 with \( n = 2 \).

(b) Repeat Exercise 5, part (a) with Simpson’s rule. Is the Simpson’s rule approximation more accurate?

(c) Repeat Exercise 6, part (a) with Simpson’s rule. Is the Simpson’s rule approximation more accurate?

(d) Repeat Exercise 7, part (a) with Simpson’s rule. Is the Simpson’s rule approximation more accurate?

10. Repeat the method for solving \( \frac{d^2y}{dx^2} = -f(x), \ 0 \leq x \leq 1 \) approximately in Example 4 with \( y(0) = y(1) = 0 \) using the following \( f(x) \).

(a) \( f(x) = \{f_{25} = -10,000 \text{ and all other } f_i = 0\} \)

(b) \( f(x) = \{f_1 = -10,000 \text{ and all other } f_i = 0\} \)

11. Use the method for solving a differential equation in Example 4 to solve approximately

\[
\frac{d^2y}{dx^2} = -10,000, \quad 0 \leq x \leq 1
\]

with \( h = .01 \) and \( y(0) = y(1) = 0 \).

### Appendix to Section 4.7: Computing Cubic Spline Approximations

In this appendix we show how the equations for determining the coefficients in a cubic spline can be greatly simplified and reduced to a tridiagonal system of \( n - 1 \) equations in \( n - 1 \) unknowns, where \( n \) is the number of subintervals in the cubic spline.

Suppose that we divide the interval \( [a, b] \), into \( n \) subintervals. Let \( a = t_0 < t_1 < \cdots < t_n = b \) be the mesh points. Recall that a cubic spline \( s(x) \) for a function \( f(x) \) is a piecewise cubic function such that

(a) \( s(t_i) \) equals the function \( f(t_i) \) at each \( t_i \).

(b) The first and second derivatives \( s'(x) \) and \( s''(x) \) are continuous.

We only know the values of \( f(x) \) at the mesh points \( t_0, t_1, \ldots, t_n \) and want to interpolate values for \( f(x) \) over the whole interval.

Let \( s_i(x) \) be the cubic polynomial in subinterval \( [t_i, t_{i+1}] \), \( i = 0, 1, \ldots, n - 1 \). Then condition (a) becomes

\[
s_i(t_i) = f(t_i), \quad i = 0, 1, \ldots, n - 1 \tag{1}
\]

\[
s_i(t_{i+1}) = f(t_{i+1}), \quad i = 0, 1, \ldots, n - 1 \tag{2}
\]
and condition (b) becomes

\[ s'_i(t_{i+1}) = s'_{i+1}(t_{i+1}), \quad i = 0, 1, \ldots, n - 2 \]
\[ s''_i(t_{i+1}) = s''_{i+1}(t_{i+1}), \quad i = 0, 1, \ldots, n - 2 \]

This gives us \( 4n - 2 \) equations in \( 4n \) unknowns. As in the discrete approximation of a differential equation in Example 4 of Section 4.7, we need two extra constraints. The ones that work out the best are

\[ s''_0(t_0) = 0 \quad \text{and} \quad s''_{n-1}(t_n) = 0 \]

Let us assume that the mesh points are equally spaced with \( t_{i+1} - t_i = h \). The first step is to express \( s_i(x) \) in the translated form

\[ s_i(x) = a_i + b_i(x - t_i) + c_i(x - t_i)^2 + d_i(x - t_i)^3, \quad i = 1, \ldots, n \]

As a result of (6), when \( x = t_i \), then \( s_i(t_i) = a_i \), \( s'_i(t_i) = b_i \) and \( s''_i(t_i) = 2c_i \). Observe also that

\[ s_i(t_{i+1}) = a_i + b_i h + c_i h^2 + d_i h^3 \]
\[ s'_i(t_{i+1}) = b_i + 2c_i h + 3d_i h^2 \]
\[ s''_i(t_{i+1}) = 2c_i + 6d_i h \]

Using (7), we rewrite conditions (1)–(5) as

\[ a_i = f(t_i), \quad i = 0, 1, \ldots, n \] (1')
\[ a_{i+1} = a_i + b_i h + c_i h^2 + d_i h^3, \quad i = 0, 1, \ldots, n - 1 \] (2')
\[ b_{i+1} = b_i + 2c_i h + 3d_i h^2, \quad i = 0, 1, \ldots, n - 2 \] (3')
\[ 2c_{i+1} = 2c_i + 6d_i h \] or
\[ c_{i+1} = c_i + 3d_i h, \quad i = 0, 1, \ldots, n - 2 \] (4')
\[ c_0 = 0 \quad \text{and} \quad c_n = 0 \] (5')

Here we have "invented" \( a_n = f(t_n) \) in (1') and \( c_n = 0 \) in (5'). There is no \( s_i(x) \) but the terms \( a_n \) and \( c_n \) are a useful way to represent conditions on the spline at \( t_n \).

From (1') we see that the \( a_i \)'s can be determined immediately from the values \( f(t_i) \). We shall now proceed to show how to express the \( d_i \)'s and \( b_i \)'s in terms of the \( c_i \)'s and the \( a_i \)'s, and then we obtain a tridiagonal system to solve for the \( c_i \)'s.

Solving (4') for \( d_i \), we have

\[ d_i = \frac{c_{i+1} - c_i}{3h}, \quad i = 0, 1, \ldots, n - 2 \] (8)
Substituting this value for \( d_i \) in \((2')\) and \((3')\), we obtain

\[
\begin{align*}
ad_{i+1} &= a_i + h b_i + \frac{h^2}{3} (2c_i + c_{i+1}), & i &= 0, 1, \ldots, n - 1 \\
b_{i+1} &= b_i + h(c_{i+1} + c_i), & i &= 0, 1, \ldots, n - 2
\end{align*}
\]

Solving \((2'')\) for \( b_i \), we obtain

\[
b_i = \frac{a_{i+1} - a_i}{h} - \frac{h(2c_i + c_{i+1})}{3}, & i &= 0, 1, \ldots, n - 1
\]

Reducing the subscript by 1, \((9)\) becomes

\[
b_{i-1} = \frac{a_{i} - a_{i-1}}{h} - \frac{h(2c_{i-1} + c_i)}{3}, & i &= 1, 2, \ldots, n
\]

Let us also rewrite \((3'')\) with reduced subscript

\[
b_i = b_{i-1} + h(c_i + c_{i-1}), & i &= 1, 2, \ldots, n - 1
\]

Now we substitute \((9)\) for \( b_i \) and \((9')\) for \( b_{i-1} \) in \((3'')\) to obtain

\[
\frac{a_{i+1} - a_i}{h} - \frac{h(2c_i + 1 c_{i+1})}{3} = \frac{a_i - a_{i-1}}{h}
\]

\[
- \frac{h(2c_{i-1} + c_i)}{3} + h(c_i + c_{i-1})
\]

Multiplying by 3, dividing by \( h \), and collecting the \( a_i \)'s on the right side, we have

\[
(2c_i + c_{i+1}) - (2c_{i-1} + c_i) + 3(c_i + c_{i-1}) = \frac{3(a_{i+1} - a_i)}{h^2}
\]

\[
- \frac{3(a_i - a_{i-1})}{h^2}
\]

or

\[
c_{i-1} + 4c_i + c_{i+1} = \frac{3}{h^2} (a_{i-1} - 2a_i + a_{i+1}), & i &= 1, \ldots, n - 1
\]

Recall from \((1')\) that \( a_i = f(t_i) \). Once we solve \((11)\) for the \( c_i \)'s, then by \((8)\) we can determine the \( d_i \)'s and by \((9)\) we can determine the \( b_i \)'s. Thus, to determine the \( 4n \) coefficients in the \( n \) cubic polynomials of our cubic spline, we only need to solve the \((n - 1)\)-by-\((n - 1)\) tridiagonal system \((11)\) (recall that \( c_0 \) and \( c_n \) equal 0).
Note that it would have been possible to develop the foregoing equations without equally spaced subintervals; unequal spacing is natural for curves that are straight most of the time but change rapidly over few short intervals. If \( h_i = t_{i+1} - t_i \), the tridiagonal system (11) can be shown to be

\[
h_{j-1}c_i + 2(h_{j-1} + h_j)c_i + hjc_{i+1} = \frac{3}{h_i} (a_{i+1} - a_i) - \frac{3}{h_{i+1}} (a_i - a_{i-1})
\]

**Example 1. Cubic Spline Approximation of** \( x \cdot \sin(\pi x) \)

Let us use a cubic spline to approximate the function \( f(x) = x \cdot \sin(\pi x) \) over the interval \([1, 3]\). We use the values of \( f(x) \) at nine mesh points: 1, 1.25, 1.5, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0 (\( h = .025 \)), yielding eight subintervals. We have the following table of values at these points.

<table>
<thead>
<tr>
<th>Mesh Point ( t_i )</th>
<th>Function Value ( f(x) = x \cdot \sin(\pi x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_0 = 1.0 )</td>
<td>( f(t_0) = 0 )</td>
</tr>
<tr>
<td>( t_1 = 1.25 )</td>
<td>( f(t_1) = -.884 )</td>
</tr>
<tr>
<td>( t_2 = 1.5 )</td>
<td>( f(t_2) = -1.5 )</td>
</tr>
<tr>
<td>( t_3 = 1.75 )</td>
<td>( f(t_3) = -1.237 )</td>
</tr>
<tr>
<td>( t_4 = 2.0 )</td>
<td>( f(t_4) = 0 )</td>
</tr>
<tr>
<td>( t_5 = 2.25 )</td>
<td>( f(t_5) = 1.591 )</td>
</tr>
<tr>
<td>( t_6 = 2.5 )</td>
<td>( f(t_6) = 2.5 )</td>
</tr>
<tr>
<td>( t_7 = 2.75 )</td>
<td>( f(t_7) = 1.945 )</td>
</tr>
</tbody>
</table>
| \( t_8 = 3.0 \)       | \( f(t_8) = 0 \)                             | (13)

By (1'), \( a_i = f(t_i) \), so the second column of (13) gives the values of the \( a_i \)’s. Next we write the system of seven equations in seven unknowns given by equation (11) (note that \( 3/h^2 = 3/.25^2 = 48 \)).

\[
\begin{align*}
4c_1 &+ c_2 = 48(0 + 2 \cdot .884 - 1.5) = 12.864 \\
c_1 + 4c_2 + c_3 & = 48(- .884 + 2 \cdot 1.5 - 1.237) = 42.192 \\
c_2 + 4c_3 + c_4 & = 48(- 1.5 + 2 \cdot 1.237 + 0) = 46.752 \\
c_3 + 4c_4 + c_5 & = 48(- 1.237 - 2 \cdot 0 + 1.591) = 16.992 \\
c_4 + 4c_5 + c_6 & = 48(0 - 2 \cdot 1.591 + 2.5) = -32.736 \\
c_5 + 4c_6 + c_7 & = 48(1.591 - 2 \cdot 2.5 + 1.945) = -70.272 \\
c_6 + 4c_7 & = 48(2.5 - 2 \cdot 1.945 + 0) = -66.720
\end{align*}
\]
Solving (14) by Gaussian elimination, we obtain

\[ c_1 = 1.204, \quad c_2 = 8.048, \quad c_3 = 8.797, \quad c_4 = 3.520, \]
\[ c_5 = -5.883, \quad c_6 = -12.722, \quad c_7 = -13.499 \]

Using (8) to determine the \( d_i \)'s and (9) to determine the \( b_i \)'s, we obtain the cubic polynomials \( s_i(x) \):

\[
\begin{align*}
  s_0(x) &= 0 - 3.636(x - 1) + 0(x - 1)^2 + 1.605(x - 1)^3 \\
  s_1(x) &= -0.884 - 3.335(x - 1.25) + 1.204(x - 1.25)^2 + 9.125(x - 1.25)^3 \\
  s_2(x) &= -1.5 - 1.022(x - 1.5) + 8.048(x - 1.5)^2 + 0.999(x - 1.5)^3 \\
  s_3(x) &= -1.237 + 3.189(x - 1.75) + 8.797(x - 1.75)^2 - 7.036(x - 1.75)^3 \\
  s_4(x) &= 0 + 6.268(x - 2.0) + 3.520(x - 2.0)^2 - 12.537(x - 2.0)^3 \\
  s_5(x) &= 1.591 + 5.677(x - 2.25) - 5.883(x - 2.25)^2 - 9.119(x - 2.25)^3 \\
  s_6(x) &= 2.5 + 1.025(x - 2.5) - 12.722(x - 2.5)^2 - 1.036(x - 2.5)^3 \\
  s_7(x) &= 1.945 - 5.530(x - 2.75) - 13.499(x - 2.75)^2 + 17.999(x - 2.75)^3
\end{align*}
\]

Finally, we give a table comparing the values of the spline approximation \( s(x) \) with the original function \( f(x) = x \cdot \sin(\pi x) \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>( f(x) )</th>
<th>( s(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.1</td>
<td>-0.340</td>
<td>-0.362</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.705</td>
<td>-0.714</td>
</tr>
<tr>
<td>1.3</td>
<td>-1.052</td>
<td>-1.046</td>
</tr>
<tr>
<td>1.4</td>
<td>-1.331</td>
<td>-1.326</td>
</tr>
<tr>
<td>1.5</td>
<td>-1.5</td>
<td>-1.5</td>
</tr>
<tr>
<td>1.6</td>
<td>-1.522</td>
<td>-1.521</td>
</tr>
<tr>
<td>1.7</td>
<td>-1.375</td>
<td>-1.375</td>
</tr>
<tr>
<td>1.8</td>
<td>-1.058</td>
<td>-1.056</td>
</tr>
<tr>
<td>2.0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.1</td>
<td>0.649</td>
<td>0.649</td>
</tr>
<tr>
<td>2.2</td>
<td>1.293</td>
<td>1.294</td>
</tr>
<tr>
<td>2.3</td>
<td>1.861</td>
<td>1.859</td>
</tr>
<tr>
<td>2.4</td>
<td>2.282</td>
<td>2.279</td>
</tr>
<tr>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>2.6</td>
<td>2.473</td>
<td>2.474</td>
</tr>
<tr>
<td>2.7</td>
<td>2.184</td>
<td>2.188</td>
</tr>
<tr>
<td>2.8</td>
<td>1.646</td>
<td>1.637</td>
</tr>
<tr>
<td>2.9</td>
<td>0.896</td>
<td>0.873</td>
</tr>
<tr>
<td>3.0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(15) A very good fit.
There are other ways to obtain a tridiagonal system for determining the coefficients in a cubic spline; see Cheney and Kincaid (mentioned in the References) for a method starting with the second derivative of \( s(x) \) and integrating twice.

**Section 4.7 Appendix Exercises**

**Summary of Exercises**

These exercises require the construction of cubic splines to approximate various functions. The reader should mimic the steps in Example 1.

1. Repeat the calculations in Example 1 but now use the interval \([3, 4]\) with mesh points 3, 3.25, 3.5, 3.75, 4. Compare the values of your approximation with the true values of \( f(x) \) at \( x = 3.1 \) and 3.9.

2. (a) Use a cubic spline to approximate the function \( f(x) = \sin (\pi x) \) over the interval \([0, 1]\) with mesh points 0, .25, .5, .75, 1. Compare the values of your approximation with the true values of \( f(x) \) at \( x = .1 \) and .65.
   
   (b) Use your cubic spline to approximate the integral of \( \sin (\pi x) \) from 0 to 1. Note the exact answer is \( 2/\pi \).

3. Repeat Exercise 2 using mesh points 0, .2, .4, .6, .8, 1.

4. (a) Use a cubic spline to approximate the function \( f(x) = e^{-x} \) over the interval \([0, 4]\) with mesh points 0, 1, 2, 3, 4. Compare the values of your approximation with the true values of \( f(x) \) at \( x = .5 \) and 2.5.
   
   (b) Use your cubic spline to approximate the integral of \( e^{-x} \) from 0 to 4. Integrate to compute the exact answer.

5. (a) Use a cubic spline to approximate the function \( f(x) = \log e^x \) over the interval \([.5, 1.5]\) with mesh points .5, .75, 1, 1.25, 1.5.
   
   (b) Use your cubic spline to approximate the integral of \( \log e^x \) from .5 to 1.5. Integrate by parts to determine the exact answer.
Using (1) to determine the $p_{AB}$ of the presence of each parameter $i$, we can estimate the probability of the parameters $i$ being present. The parameters $i$ with the highest probability are the ones that are most likely to be present. Therefore, we can use the presence or absence of these parameters to classify the presence or absence of other parameters.

1. **Presence of Parameters:**

   - **Parameter A:**
     - $p_{AB} = 0.749$ (Present)
     - $p_{A} = 0.051$ (Not Present)

   - **Parameter B:**
     - $p_{AB} = 0.649$ (Present)
     - $p_{B} = 0.351$ (Not Present)

2. **Absence of Parameters:**

   - **Parameter C:**
     - $p_{AC} = 0.288$ (Present)
     - $p_{C} = 0.712$ (Not Present)

   - **Parameter D:**
     - $p_{AD} = 0.027$ (Present)
     - $p_{D} = 0.973$ (Not Present)

3. **Presence of Other Parameters:**

   - **Parameter E:**
     - $p_{AE} = 0.001$ (Present)
     - $p_{E} = 0.999$ (Not Present)

   - **Parameter F:**
     - $p_{AF} = 0.001$ (Present)
     - $p_{F} = 0.999$ (Not Present)

**Conclusion:**

A very good fit.