Chapter 6

Direct Methods for Solving Linear Systems

6.1 Introduction

Systems of equations are used to represent physical problems that involve the interaction of various properties. The variables in the system represent the properties being studied, and the equations describe the interaction between the variables. The system is easiest to study when the equations are all linear. Often the number of equations is the same as the number of variables, for only in this case is it likely that a unique solution will exist.

Although not all physical problems can be reasonably represented using a linear system with the same number of equations as unknowns, the solutions to many problems either have this form or can be approximated by such a system. In fact, this is quite often the only approach that can give quantitative information about a physical problem.

In this chapter we consider direct methods for approximating the solution of a system of $n$ linear equations in $n$ unknowns. A direct method is one that gives the exact solution to the system, if it is assumed that all calculations can be performed without round-off error effects. This assumption is idealized. We will need to consider quite carefully the role of finite-digit arithmetic error in the approximation to the solution to the system and how to arrange the calculations to minimize its effect.

6.2 Gaussian Elimination

If you have studied linear algebra or matrix theory, you probably have been introduced to Gaussian elimination, the most elementary method for systematically determining the solution of a system of linear equations. Variables are eliminated from the equations until one equation involves only one variable, a second equation
CHAPTER 6. DIRECT METHODS FOR SOLVING LINEAR SYSTEMS

involves only that variable and one other, a third has only these two and one additional, and so on. The solution is found by solving for the variable in the single equation, using this to reduce the second equation to one that now contains a single variable, and so on, until values for all the variables are found.

Three operations are permitted on a system of equations ($E_n$).

[Operations on Systems of Equations]

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

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

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

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

EXAMPLE 1 The four equations

\[
\begin{align*}
E_1: & \quad x_1 + x_2 + 3x_4 = 4, \\
E_2: & \quad 2x_1 + x_2 - x_3 + x_4 = 1, \\
E_3: & \quad 3x_1 - x_2 - x_3 + 2x_4 = -3, \\
E_4: & \quad -x_1 + 2x_2 + 3x_3 - x_4 = 4,
\end{align*}
\]

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

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

where, for simplicity, the new equations are again labeled $E_1, E_2, E_3, \text{and } E_4$.

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

\[
\begin{align*}
E_1: & \quad x_1 + x_2 + 3x_4 = 4, \\
E_2: & \quad -x_2 - x_3 - 5x_4 = -7, \\
E_3: & \quad 3x_3 + 13x_4 = 13, \\
E_4: & \quad -13x_4 = -13.
\end{align*}
\]
The system of equations is now in **triangular** (or **reduced**) form and can be solved for the unknowns by a backward-substitution process. Noting that $E_4$ implies $x_4 = 1$, we can solve $E_3$ for $x_3$:

$$x_3 = \frac{1}{3}(13 - 13x_4) = \frac{1}{3}(13 - 13) = 0.$$  
Continuing, $E_2$ gives

$$x_2 = -(-7 + 5x_4 + x_3) = -(-7 + 5 + 0) = 2,$$
and $E_1$ gives

$$x_1 = 4 - 3x_4 - x_2 = 4 - 3 - 2 = -1.$$
The solution is, therefore, $x_1 = -1$, $x_2 = 2$, $x_3 = 0$, and $x_4 = 1$. It is easy to verify that these values solve the original system of equations.

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

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

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

**EXAMPLE 2**  
The matrix

$$A = \begin{bmatrix} 2 & -1 & 7 \\ 3 & 1 & 0 \end{bmatrix}$$
is a $2 \times 3$ matrix with $a_{11} = 2$, $a_{12} = -1$, $a_{13} = 7$, $a_{21} = 3$, $a_{22} = 1$, and $a_{23} = 0$.  

The $1 \times n$ matrix $A = [a_{11} \ a_{12} \ \cdots \ a_{1n}]$ is called an **n-dimensional row vector**, and an $n \times 1$ matrix

$$A = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}$$
is called an **n-dimensional column vector**. Usually the unnecessary subscript is omitted for vectors and a boldface lowercase letter is used for notation. So,

\[
x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}
\]

denotes a column vector, and \( y = [y_1 \ y_2 \ \cdots \ y_n] \) denotes a row vector.

A system of \( n \) linear equations in the \( n \) unknowns \( x_1, x_2, \ldots, x_n \) has the form

\[
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\
\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n.
\]

An \( n \times (n+1) \) matrix can be used to represent this linear system by first constructing

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

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

\[
[A, b] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix}.
\]

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

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

\[
\begin{bmatrix} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{bmatrix}
\]

Performing the operations as described in that example produces the matrices

\[
\begin{bmatrix} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & -4 & -1 & -7 & -15 \\ 0 & 3 & 3 & 2 & 8 \end{bmatrix} \quad \text{and} \quad
\begin{bmatrix} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & -13 & -13 \end{bmatrix}
\]
6.2. GAUSSIAN ELIMINATION

The latter matrix can now be transformed into its corresponding linear system and solutions for \( x_1, x_2, x_3, \) and \( x_4 \) obtained. The procedure involved in this process is called **Gaussian Elimination with Backward Substitution**.  

The general Gaussian elimination procedure applied to the linear system

\[
E_1: \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\
E_2: \quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\
\vdots \quad \vdots \\
E_n: \quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n,
\]

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

\[
\tilde{A} = [A, \mathbf{b}] = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn} & b_n
\end{bmatrix},
\]

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

Suppose that \( a_{11} \neq 0 \). To convert the entries in the first column, below \( a_{11} \), to zero, we perform the operations \( (E_k - m_{k1}E_1) \to (E_k) \) for each \( k = 2, 3, \ldots, n \) for an appropriate multiplier \( m_{k1} \). We first designate the diagonal element in the column, \( a_{11} \) as the **pivot element**. The **multiplier** for the \( k \)th row is defined by \( m_{k1} = a_{k1}/a_{11} \). Performing the operations \( (E_k - m_{k1}E_1) \to (E_k) \) for each \( k = 2, 3, \ldots, n \) eliminates (that is, change to zero) the coefficient of \( x_1 \) in each of these rows:

\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn} & b_n
\end{bmatrix} \quad \frac{E_2 - m_{21}E_1}{a_{21}} \to E_2 \quad \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn} & b_n
\end{bmatrix}
\]

Although the entries in rows \( 2, 3, \ldots, n \) are expected to change, for ease of notation, we again denote the entry in the \( i \)th row and the \( j \)th column by \( a_{ij} \).

If the pivot element \( a_{22} \neq 0 \), we form the multipliers \( m_{k2} = a_{k2}/a_{22} \) and perform the operations \( (E_k - m_{k2}E_2) \to E_k \) for each \( k = 3, \ldots, n \) obtaining

\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
0 & a_{22} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & a_{n2} & \cdots & a_{nn} & b_n
\end{bmatrix} \quad \frac{E_3 - m_{32}E_2}{a_{32}} \to E_3 \quad \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
0 & a_{22} & \cdots & a_{2n} & b_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & a_{n2} & \cdots & a_{nn} & b_n
\end{bmatrix}
\]

We then follow this sequential procedure for the rows \( i = 3, \ldots, n - 1 \). Define the multiplier \( m_{ki} = a_{ki}/a_{ii} \) and perform the operation

\( (E_k - m_{ki}E_i) \to (E_k) \)
for each \( k = i + 1, i + 2, \ldots, n \), provided the pivot element \( a_{ii} \) is nonzero. This eliminates \( x_i \) in each row below the \( i \)th for all values of \( i = 1, 2, \ldots, n - 1 \). The resulting matrix has the form

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

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

\[
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_{1,n+1}, \\
a_{22}x_2 + \cdots + a_{2n}x_n = a_{2,n+1}, \\
\vdots \\
a_{nn}x_n = a_{n,n+1},
\]

_backward substitution can be performed._ Solving the \( n \)th equation for \( x_n \) gives

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

Then solving the \((n-1)\)st equation for \( x_{n-1} \) and using the known value for \( x_n \) yields

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

Continuing this process, we obtain

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

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

The procedure will _fail_ if at the \( i \)th step the pivot element \( a_{ii} \) is zero, for then either the multipliers \( m_{ki} = a_{ki}/a_{ii} \) are not defined (this occurs if \( a_{ii} = 0 \) for some \( i < n \)) or the backward substitution cannot be performed (if \( a_{nn} = 0 \)). This does not necessarily mean that the system has no solution, but rather that the technique for finding the solution must be altered. An illustration is given in the following example.

**EXAMPLE 3** Consider the linear system

\[
E_1: \quad x_1 - x_2 + 2x_3 - x_4 = -8, \\
E_2: \quad 2x_1 - 2x_2 + 3x_3 - 3x_4 = -20, \\
E_3: \quad x_1 + x_2 + x_3 = -2, \\
E_4: \quad x_1 - x_2 + 4x_3 + 3x_4 = 4.
\]
The augmented matrix is
\[
\begin{bmatrix}
1 & -1 & 2 & -1 & -8 \\
2 & -2 & 3 & -3 & -20 \\
1 & 1 & 1 & 0 & -2 \\
1 & -1 & 4 & 3 & 4
\end{bmatrix}.
\]
Performing the operations
\[
(E_2 - 2E_1) \rightarrow (E_2), \quad (E_3 - E_1) \rightarrow (E_3), \quad \text{and} \quad (E_4 - E_1) \rightarrow (E_4),
\]
we have the matrix
\[
\begin{bmatrix}
1 & -1 & 2 & -1 & -8 \\
0 & 0 & -1 & -1 & -4 \\
0 & 2 & -1 & 1 & 6 \\
0 & 0 & 2 & 4 & 12
\end{bmatrix}.
\]
The element \(a_{22}\) in this matrix is zero, so the procedure cannot continue in its present form. But operations of the form \((E_i) \leftrightarrow (E_p)\) are permitted, so a search is made of the elements \(a_{32}\) and \(a_{42}\) for the first nonzero element. Since \(a_{32} \neq 0\), the operation \((E_2) \leftrightarrow (E_3)\) is performed to obtain a new matrix:
\[
\begin{bmatrix}
1 & -1 & 2 & -1 & -8 \\
0 & 2 & -1 & 1 & 6 \\
0 & 0 & -1 & -1 & -4 \\
0 & 0 & 2 & 4 & 12
\end{bmatrix}.
\]
The variable \(x_2\) is already eliminated from \(E_3\) and \(E_4\), so the computations continue with the operation \((E_4 + 2E_3) \rightarrow (E_4)\), giving
\[
\begin{bmatrix}
1 & -1 & 2 & -1 & -8 \\
0 & 2 & -1 & 1 & 6 \\
0 & 0 & -1 & -1 & -4 \\
0 & 0 & 0 & 2 & 4
\end{bmatrix}.
\]
Finally, the backward substitution is applied:
\[
x_4 = \frac{4}{2} = 2, \quad x_3 = \frac{-4 - (-1)x_4}{-1} = 2, \quad x_2 = \frac{6 - (-1)x_3 + x_4}{2} = 3, \quad x_1 = \frac{-8 - ((-1)x_2 + 2x_3 + (-1)x_4)}{1} = -7.
\]
To define matrices and perform Gaussian elimination using Maple, you must first access the linear algebra library using the command
\[
> \text{with(linalg)};
\]
To define the initial augmented matrix in Example 3, which we will call \( AA \), use the command

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

The first two parameters, 4 and 5, give the number of rows and columns, respectively, and the last parameter is a list, by rows, of the entries of \( AA \). The function \( \text{addrow}(AA, i, j, m) \) performs the operation \((E_j + mE_i) \rightarrow (E_j)\) and the function \( \text{swaprow}(AA, i, j) \) performs the operation \((E_i) \leftrightarrow (E_j)\). So, the sequence of operations

\[
> AA := \text{addrow}(AA, 1, 2, -2);
> AA := \text{addrow}(AA, 1, 3, -1);
> AA := \text{addrow}(AA, 1, 4, -1);
> AA := \text{swaprow}(AA, 2, 3);
> AA := \text{addrow}(AA, 3, 4, 2);
\]

gives the final reduction, which is again called \( AA \). Alternatively, the single command \( AA := \text{gausselim}(AA) \); returns the reduced matrix. A final operation,

\[
> x := \text{backsub}(AA);
\]

produces the solution \( x := [-7, 3, 2, 2] \).

Example 3 illustrates what is done if one of the pivot elements is zero. If the \( i \)th pivot element is zero, the \( i \)th column of the matrix is searched from the \( i \)th row downward for the first nonzero entry, and a row interchange is performed to obtain the new matrix. Then the procedure continues as before. If no nonzero entry is found the procedure stops, and the linear system does not have a unique solution; it might have no solution or an infinite number of solutions. The program \text{GAUSEL61} implements Gaussian Elimination with Backward Substitution and incorporates row interchanges when required.

The computations in the program are performed using only one \( n \times (n + 1) \) array for storage. This is done by replacing, at each step, the previous value of \( a_{ij} \) by the new one. In addition, the multipliers are stored in the locations of \( a_{ki} \) known to have zero values—that is, when \( i < n \) and \( k = i + 1, i + 2, \ldots, n \). Thus, the original matrix \( A \) is overwritten by the multipliers below the main diagonal and by the nonzero entries of the final reduced matrix on and above the main diagonal. We will see in Section 6.5 that these values can be used to solve other linear systems involving the original matrix \( A \).

Both the amount of time required to complete the calculations and the subsequent round-off error depend on the number of floating-point arithmetic operations needed to solve a routine problem. In general, the amount of time required to perform a multiplication or division on a computer is approximately the same and is considerably greater than that required to perform an addition or subtraction. Even though the actual differences in execution time depend on the particular computing system being used, the count of the additions/subtractions are kept separate from the count of the multiplications/divisions because of the time differential. The total
number of arithmetic operations depends on the size \( n \), as follows:

\[
\begin{align*}
\text{Multiplications/divisions:} & \quad \frac{n^3}{3} + n^2 - \frac{n}{3} \\
\text{Additions/subtractions:} & \quad \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}.
\end{align*}
\]

For large \( n \), the total number of multiplications and divisions is approximately \( n^3/3 \), that is, \( O(n^3) \), as is the total number of additions and subtractions. The amount of computation, and the time required to perform it, increases with \( n \) in approximate proportion to \( n^3/3 \), as shown in Table 6.1.

<table>
<thead>
<tr>
<th>( n )</th>
<th>Multiplications/Divisions</th>
<th>Additions/Subtractions</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>430</td>
<td>375</td>
</tr>
<tr>
<td>50</td>
<td>44,150</td>
<td>42,875</td>
</tr>
<tr>
<td>100</td>
<td>343,300</td>
<td>338,250</td>
</tr>
</tbody>
</table>
EXERCISE SET 6.2

1. Obtain a solution by graphical methods of the following linear systems, if possible.

(a) \(x_1 + 2x_2 = 3,\) \(x_1 - x_2 = 0.\)
(b) \(x_1 + 2x_2 = 0,\) \(x_1 - x_2 = 0.\)
(c) \(x_1 + 2x_2 = 3,\) \(2x_1 + 4x_2 = 6.\)
(d) \(x_1 + 2x_2 = 3,\) \(-2x_1 - 4x_2 = 6.\)
(e) \(x_1 + 2x_2 = 0,\) \(2x_1 + 4x_2 = 0.\)
(f) \(2x_1 + x_2 = -1,\) \(x_1 + x_2 = 2,\) \(x_1 - 3x_2 = 5.\)
(g) \(2x_1 + x_2 = -1,\) \(4x_1 + 2x_2 = -2,\) \(x_1 - 3x_2 = 5.\)
(h) \(2x_1 + x_2 + x_3 = 1,\) \(2x_1 + 4x_2 - x_3 = -1.\)

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

(a) \(4x_1 - x_2 + x_3 = 8,\) \(2x_1 + 5x_2 + 2x_3 = 3,\) \(x_1 + 2x_2 + 4x_3 = 11.\)
(b) \(4x_1 + x_2 + 2x_3 = 9,\) \(2x_1 + 4x_2 - x_3 = -5,\) \(x_1 + x_2 - 3x_3 = -9.\)

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

(a) \(x_1 - x_2 + 3x_3 = 2,\) \(3x_1 - 3x_2 + x_3 = -1,\) \(x_1 + x_2 = 3.\)
(b) \(2x_1 - 1.5x_2 + 3x_3 = 1,\) \(-x_1 + 2x_3 = 3,\) \(4x_1 - 4.5x_2 + 5x_3 = 1.\)

(c) \(2x_1 + 1.5x_2 = 2,\) \(-3x_2 + 0.5x_3 = -2,\) \(2x_1 - 2x_2 + x_3 + x_4 = 0.\)
(d) \(3x_1 - \frac{1}{2}x_2 + x_3 = 4,\) \(2x_1 - x_2 - x_3 + x_4 = 5,\) \(x_1 + 0.6 + x_2 = 2.\)
(e) \(x_1 + x_2 + x_4 = 2,\) \(2x_1 + x_2 - x_3 + x_4 = 1,\) \(2x_1 + x_2 - x_3 + x_4 = 1,\) \(4x_1 - x_2 - 2x_3 + 2x_4 = 0,\) \(-x_1 + 2x_2 + 3x_3 - x_4 = 4,\) \(3x_1 - x_2 - x_3 + 2x_4 = -3.\)
6.2. GAUSSIAN ELIMINATION

4. Use Maple with Digits set to 7 and Gaussian elimination to solve the following linear systems.

(a) \[
\frac{1}{4}x_1 + \frac{1}{5}x_2 + \frac{1}{6}x_3 = 9, \\
\frac{1}{7}x_1 + \frac{1}{8}x_2 + \frac{1}{9}x_3 = 8, \\
\frac{1}{2}x_1 + x_2 + 2x_3 = 8.
\]

(b) \[
3.333x_1 + 15920x_2 - 10.333x_3 = 15913, \\
2.222x_1 + 16.71x_2 + 9.612x_3 = 28.544, \\
1.5611x_1 + 5.1791x_2 + 1.6852x_3 = 8.4254.
\]

(c) \[
x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 + \frac{1}{4}x_4 = \frac{1}{5}, \\
\frac{1}{7}x_1 + \frac{1}{8}x_2 + \frac{1}{9}x_3 + \frac{1}{10}x_4 = \frac{1}{11}, \\
\frac{1}{12}x_1 + \frac{1}{13}x_2 + \frac{1}{14}x_3 + \frac{1}{15}x_4 = \frac{1}{16}, \\
\frac{1}{17}x_1 + \frac{1}{18}x_2 + \frac{1}{19}x_3 + \frac{1}{20}x_4 = \frac{1}{21}.
\]

(d) \[
2x_1 + x_2 - x_3 + x_4 - 3x_5 = 7, \\
x_1 + 2x_3 - x_4 + x_5 = 2, \\
-2x_2 - x_3 + x_4 - x_5 = -5, \\
3x_1 + x_2 - 4x_3 + 5x_5 = 6, \\
x_1 - x_2 - x_3 - x_4 + x_5 = 3.
\]

5. Given the linear system

\[
2x_1 - 6\alpha x_2 = 3, \\
3\alpha x_1 - x_2 = \frac{3}{2}.
\]

(a) Find value(s) of $\alpha$ for which the system has no solutions.

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

(c) Assuming a unique solution exists for a given $\alpha$, find the solution.

6. Given the linear system

\[
x_1 - x_2 + \alpha x_3 = -2, \\
-x_1 + 2x_2 - \alpha x_3 = 3, \\
\alpha x_1 + x_2 + x_3 = 2.
\]

(a) Find value(s) of $\alpha$ for which the system has no solutions.

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

(c) Assuming a unique solution exists for a given $\alpha$, find the solution.
7. Suppose that in a biological system there are \( n \) species of animals and \( m \) sources of food. Let \( x_j \) represent the population of the \( j \)th species for each \( j = 1, \ldots, n \); \( b_i \) represent the available daily supply of the \( i \)th food; and \( a_{ij} \) represent the amount of the \( i \)th food consumed on the average by a member of the \( j \)th species. The linear system

\[
\begin{align*}
 a_{11}x_1 &+ a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\
 a_{21}x_1 &+ a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\
 &\vdots & &\vdots & &\vdots \\
 a_{m1}x_1 &+ a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{align*}
\]

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

(a) Let

\[
 A = [a_{ij}] = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 1 & 0 & 2 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix},
\]

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

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

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

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

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

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

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

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

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

(a) Show that the linear system

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

must be solved when the Trapezoidal rule is used.

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

(c) Repeat part (b) using the Composite Simpson’s rule.
6.3 Pivoting Strategies

If all the calculations could be done using exact arithmetic, we could nearly end the chapter with the previous section. We now know how many calculations are needed to perform Gaussian elimination on a system and from this we should be able to determine whether our computational device can solve our problem in reasonable time. In a practical situation, however, we do not have exact arithmetic, and the large number of arithmetic computations, on the order of $O(n^3)$, makes the consideration of computational round-off error necessary. In fact, as we will see in our next example, even for certain very small systems round-off error can dominate the calculations. In this section we will see how the calculations in Gaussian elimination can be arranged to reduce the effect of this error.

In deriving the Gaussian elimination method, we found that a row interchange is needed when one of the pivot elements, $a_{ii}$, is zero. This row interchange has the form $(E_i) \leftrightarrow (E_p)$, where $p$ is the smallest integer greater than $i$ with $a_{pi} \neq 0$. To reduce the round-off error associated with finite-digit arithmetic, it is often necessary to perform row interchanges even when the pivot elements are not zero.

If $a_{ii}$ is small in magnitude compared to $a_{ki}$, the magnitude of the multiplier

$$m_{ki} = \frac{a_{ki}}{a_{ii}}$$

will be much larger than 1. A round-off error introduced in the computation of one of the terms $a_{il}$ is multiplied by $m_{ki}$ when computing $a_{kl}$, compounding the original error. Also, when performing the backward substitution for

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

with a small value of $a_{ii}$, any round-off error in the numerator is dramatically increased when dividing by $a_{ii}$. An illustration of this difficulty is given in the following example.

**Example 1** The linear system

\[
\begin{align*}
E_1: & \quad 0.003000x_1 + 59.14x_2 = 59.17, \\
E_2: & \quad 5.291x_1 - 6.130x_2 = 46.78
\end{align*}
\]

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

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

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

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

\[
\begin{align*}
0.003000x_1 + 59.14x_2 & = 59.17 \\
-104300x_2 & \approx -104400
\end{align*}
\]
instead of the precise values

\[
\begin{align*}
0.003000x_1 + 59.14x_2 &= 59.17 \\
-104309.376x_2 &= -104309.376.
\end{align*}
\]

The disparity in the magnitudes of \(m_{21}a_{13}\) and \(a_{23}\) has introduced round-off error, but the error has not yet been propagated. Backward substitution yields

\[x_2 \approx 1.001,\]

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

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

contains the small error of 0.001 multiplied by 59.14/0.003000 \(\approx 20000\). This ruins the approximation to the actual value \(x_1 = 10.00\). (See Figure 6.1.)

**Figure 6.1**

This is clearly a contrived example and the graph demonstrate why the error can so easily occur, but for only slightly larger systems it is much more difficult to predict in advance when devastating round-off error can occur.

Example 1 shows how difficulties arise when the pivot element \(a_{ii}\) is small relative to the entries \(a_{kj}\) for \(i \leq k \leq n\) and \(i \leq j \leq n\). To avoid this problem, pivoting is performed by selecting an element \(a_{pq}\) for the pivot that has a larger magnitude than \(a_{pp}\) and interchanging the \(i\)th and \(p\)th rows.

The simplest strategy is to select, at the \(i\)th step, the element in the same column that is below the diagonal and has the largest absolute value; that is, to determine the smallest \(p \geq i\) such that

\[|a_{pi}| = \max_{i \leq k \leq n} |a_{ki}|\]

and perform \((E_i) \leftrightarrow (E_p)\). In this case no interchange of columns is used.
EXAMPLE 2  Reconsider the system

\[
E_1: \quad 0.003000x_1 + 59.14x_2 = 59.17, \\
E_2: \quad 5.291x_1 - 6.130x_2 = 46.78.
\]

The pivoting procedure just described results in first finding

\[
\max\{|a_{11}|, |a_{21}|\} = \max\{|0.003000|, |5.291|\} = |5.291| = |a_{21}|.
\]

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

\[
E_1: \quad 5.291x_1 - 6.130x_2 = 46.78, \\
E_2: \quad 0.003000x_1 + 59.14x_2 = 59.17.
\]

The multiplier for this system is

\[
m_{21} = \frac{a_{21}}{a_{11}} = 0.0005670,
\]

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

\[
5.291x_1 - 6.130x_2 = 46.78, \\
59.14x_2 \approx 59.14.
\]

The four-digit answers resulting from the backward substitution are the correct values, \(x_1 = 10.00\) and \(x_2 = 1.000\).

The technique just described is called partial pivoting, or maximal column pivoting, and is implemented in the program GAUMPP62.

Although partial pivoting is sufficient for many linear systems, situations do arise when it is inadequate. For example, the linear system

\[
E_1: \quad 30.00x_1 + 591400x_2 = 591700, \\
E_2: \quad 5.291x_1 - 6.130x_2 = 46.78
\]

is the same as that in Examples 1 and 2 except that all entries in the first equation have been multiplied by \(10^4\). Partial pivoting with four-digit arithmetic leads to the same results as obtained in Example 1 since no row interchange would be performed. A technique known as scaled partial pivoting is needed for this system. The first step in this procedure is to define a scale factor \(s_k\) for each row:

\[
s_k = \max_{1 \leq j \leq n} |a_{kj}|.
\]

The appropriate row interchange to place zeros in the first column is determined by choosing the first integer \(p\) with

\[
\frac{|a_{p1}|}{s_p} = \max_{1 \leq k \leq n} \frac{|a_{k1}|}{s_k}
\]
and performing \((E_1) \leftrightarrow (E_p)\). The effect of scaling is to ensure that the largest element in each row has a \textit{relative} magnitude of 1 before the comparison for row interchange is performed.

In a similar manner, before eliminating variable \(x_i\) using the operations 
\[E_k - m_{ki}E_i \rightarrow E_k\] for \(k = i + 1, \ldots, n\)
we select the smallest integer \(p \geq i\) with
\[\frac{|a_{pi}|}{s_p} = \max_{i \leq k \leq n} \frac{|a_{ki}|}{s_k}\]
and perform the row interchange \(E_i \leftrightarrow E_p\) if \(i \neq p\). We must note that the scale factors \(s_1, \ldots, s_n\) are computed only once at the start of the procedure and must also be interchanged when row interchanges are performed.

In the program GAUSPP63 the scaling is done only for comparison purposes, so the division by scaling factors produces no round-off error in the system.

**EXAMPLE 3** Applying scaled partial pivoting to the system in Example 1 gives
\[s_1 = \max\{|30.00|, |591400|\} = 591400 \quad \text{and} \quad s_2 = \max\{|5.291|, |-6.130|\} = 6.130.\]
Consequently,
\[\frac{|a_{11}|}{s_1} = \frac{30.00}{591400} = 0.5073 \times 10^{-4} \quad \text{and} \quad \frac{|a_{21}|}{s_2} = \frac{5.291}{6.130} = 0.8631\]
and the interchange \((E_1) \leftrightarrow (E_2)\) is made. Applying Gaussian elimination to the new system produces the correct results: \(x_1 = 10.00\) and \(x_2 = 1.000.\)

**EXAMPLE 4** Use scaled partial pivoting to solve the linear system using three-digit rounding arithmetic.
\[
\begin{align*}
2.11x_1 &- 4.21x_2 + 0.921x_3 = 2.01, \\
4.01x_1 &+ 10.2x_2 - 1.12x_3 = -3.09, \\
1.09x_1 &+ 0.987x_2 + 0.832x_3 = 4.21.
\end{align*}
\]
To obtain three-digit rounding arithmetic, enter

```plaintext
>Digits:=3;
```
We have \(s_1 = 4.21\), \(s_2 = 10.2\), and \(s_3 = 1.09\). So
\[\frac{|a_{11}|}{s_1} = \frac{2.11}{4.21} = 0.501, \quad \frac{|a_{21}|}{s_1} = \frac{4.01}{10.2} = 0.393, \quad \text{and} \quad \frac{|a_{31}|}{s_3} = \frac{1.09}{1.09} = 1.\]
The augmented matrix \(AA\) is defined by

```plaintext
>AA:=matrix(3,4,[2.11,-4.21,0.921,2.01,4.01,10.2,-1.12,-3.09,1.09,0.987,0.832,4.21]);
```
which gives
\[
AA := \begin{bmatrix}
2.11 & -4.21 & .921 & 2.01 \\
4.01 & 10.2 & -1.12 & -3.09 \\
1.09 & .987 & .832 & 4.21
\end{bmatrix}.
\]

Since \(|a_{31}|/s_3\) is largest, we perform \((E_1) \leftrightarrow (E_3)\) using
\[
>AA:=swaprow(AA,1,3);
\]
to obtain
\[
AA := \begin{bmatrix}
1.09 & .987 & .832 & 4.21 \\
4.01 & 10.2 & -1.12 & -3.09 \\
2.11 & -4.21 & .921 & 2.01
\end{bmatrix}.
\]

We compute the multipliers \(m_{21} = 3.68\) and \(m_{31} = 1.94\) using
\[
>m21:=4.01/1.09;
\]
and
\[
>m31:=2.11/1.09;
\]

We perform the first two eliminations using
\[
>AA:=addrow(AA,1,2,-m21);
\]
and
\[
>AA:=addrow(AA,1,3,-m31);
\]
to obtain
\[
AA := \begin{bmatrix}
1.09 & .987 & .832 & 4.21 \\
0 & 6.57 & -4.18 & -18.6 \\
0 & -6.12 & -.689 & -6.16
\end{bmatrix}.
\]

Since
\[
\frac{|a_{22}|}{s_2} = \frac{6.57}{10.2} = 0.644 < \frac{|a_{32}|}{s_3} = \frac{6.12}{4.21} = 1.45,
\]
we perform
\[
>AA:=swaprow(AA,2,3);
\]
giving
\[
AA := \begin{bmatrix}
1.09 & .987 & .832 & 4.21 \\
0 & -6.12 & -.689 & -6.16 \\
0 & 6.57 & -4.18 & -18.6
\end{bmatrix}.
\]

The multiplier \(m_{32} = -1.07\) is computed by
6.3. PIVOTING STRATEGIES

\[ m_{32} := \frac{6.57}{-6.12}; \]

The elimination step

\[ AA := \text{addrow}(AA, 2, 3, -m_{32}); \]

gives

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

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

\[ AA[3, 2] := 0; \]

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

\[ \texttt{evalm}(AA); \]

which displays the matrix \( AA \). Finally,

\[ x := \text{backsub}(AA); \]

gives the solution

\[ x := [-.431, .430, 5.12]. \]

The scaled partial pivoting procedure adds a total of

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

comparisons

and

\[ \frac{n(n + 1)}{2} - 1 \]

divisions to the Gaussian elimination procedure. The time required to perform a comparison is slightly more than that of an addition/subtraction. The total time to perform the basic Gaussian elimination procedure is the time required for approximately \( n^3/3 \) multiplications/divisions and \( n^3/3 \) additions/subtractions. So scaled partial pivoting does not add significantly to the computational time required to solve a system for large values of \( n \).

If a system has computation difficulties that scaled partial pivoting cannot resolve, \textit{maximal} (also called \textit{total} or \textit{full}) pivoting can be used. Maximal pivoting at the \( i \)th step searches all the entries \( a_{kj} \), for \( k = i, i + 1, \ldots, n \) and \( j = i, i + 1, \ldots, n \), to find the entry with the largest magnitude. Both row and column interchanges are
performed to bring this entry to the pivot position. The additional time required to incorporate maximal pivoting into Gaussian elimination is

\[ \frac{n(n - 1)(2n + 5)}{6} \]

comparisons.

This approximately doubles the amount of addition/subtraction time over ordinary Gaussian elimination.
6.3. PIVOTING STRATEGIES

EXERCISE SET 6.3

1. Use standard Gaussian elimination to find the row interchanges that are required to solve the following linear systems.

(a) \[ x_1 - 5x_2 + x_3 = 7 \]
\[ 10x_1 + 20x_3 = 6 \]
\[ 5x_1 - x_3 = 4 \]

(b) \[ x_1 + x_2 - x_3 = 1 \]
\[ 2x_1 - x_2 + 2x_3 = 3 \]

(c) \[ 2x_1 - 3x_2 + 2x_3 = 5 \]
\[ -4x_1 + 2x_2 - 6x_3 = 14 \]
\[ 2x_1 + 2x_2 + 4x_3 = 8 \]

(d) \[ x_1 + x_2 + 4x_3 = 2 \]
\[ 5x_1 - x_3 = 4 \]
\[ 2x_1 - 2x_2 - x_3 = 3 \]

2. Repeat Exercise 1 using Gaussian elimination with partial pivoting.

3. Repeat Exercise 1 using Gaussian elimination with scaled partial pivoting.

4. Repeat Exercise 1 using Gaussian elimination with complete pivoting.

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

(a) \[ 0.03x_1 + 58.9x_2 = 59.2 \]
\[ 5.31x_1 - 6.10x_2 = 47.0 \]
Actual solution \( x_1 = 10, x_2 = 1 \).

(b) \[ 58.9x_1 + 0.03x_2 = 59.2 \]
\[ -6.10x_1 + 5.31x_2 = 47.0 \]
Actual solution \( x_1 = 1, x_2 = 10 \).

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

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

(e) \[ 1.19x_1 + 2.11x_2 - 100x_3 + x_4 = 1.12 \]
\[ 14.2x_1 - 0.122x_2 + 12.2x_3 - x_4 = 3.44 \]
\[ 100x_2 - 99.9x_3 + x_4 = 2.15 \]
\[ 15.3x_1 + 0.110x_2 - 13.1x_3 - x_4 = 4.16 \]
Actual solution \( x_1 = 0.17682530, x_2 = 0.01269269, x_3 = -0.02065405, x_4 = -1.18260870 \).
(f) \[ \begin{align*}
\pi x_1 & - e x_2 + \sqrt{2} x_3 - \sqrt{3} x_4 = \sqrt{11} \\
\pi^2 x_1 & + e x_2 - e^2 x_3 + \frac{1}{2} x_4 = 0 \\
\sqrt{5} x_1 & - \sqrt{6} x_2 + x_3 - \sqrt{2} x_4 = \pi \\
\pi^3 x_1 & + e^2 x_2 - \sqrt{7} x_3 + \frac{1}{2} x_4 = \sqrt{2}
\end{align*} \]

Actual solution \( x_1 = 0.78839378, x_2 = -3.12541367, x_3 = 0.16759660, x_4 = 4.55700252. \)

6. Repeat Exercise 5 using three-digit rounding arithmetic.
7. Repeat Exercise 5 using Gaussian elimination with partial pivoting.
8. Repeat Exercise 5 using Gaussian elimination with scaled partial pivoting.
9. Suppose that

\[ \begin{align*}
2x_1 + x_2 + 3x_3 &= 1 \\
4x_1 + 6x_2 + 8x_3 &= 5 \\
6x_1 + \alpha x_2 + 10x_3 &= 5
\end{align*} \]

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

\[ \allowdisplaybreaks
\begin{align*}
\text{(a) } \alpha &= 6 & \text{(b) } \alpha &= 9 & \text{(c) } \alpha &= -3
\end{align*} \]
6.4 Linear Algebra and Matrix Inversion

Early in this chapter we illustrated the convenience of matrix notation for the study of linear systems of equations, but there is a wealth of additional material in linear algebra that finds application in the study of approximation techniques. In this section we introduce some basic notation and results that are needed for both theory and application. All the topics discussed here should be familiar to anyone who has studied matrix theory at the undergraduate level. This section could be omitted, but it is advisable to read the section to see the results from linear algebra that will be frequently called upon for service.

Two matrices $A$ and $B$ are equal if both are of the same size, say, $n \times m$, and if $a_{ij} = b_{ij}$ for each $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

This definition means, for example, that

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

since they differ in dimension.

If $A$ and $B$ are $n \times m$ matrices and $\lambda$ is a real number, then the sum of $A$ and $B$, denoted $A + B$, is the $n \times m$ matrix whose entries are $a_{ij} + b_{ij}$, and the scalar product of $\lambda$ and $A$, denoted $\lambda A$, is the $n \times m$ matrix whose entries are $\lambda a_{ij}$.

If $A$ is an $n \times m$ matrix and $B$ is an $m \times p$ matrix, the matrix product of $A$ and $B$, denoted $AB$, is an $n \times p$ matrix $C$ whose entries $c_{ij}$ are given by

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{im} b_{mj},$$

for each $i = 1, 2, \ldots n$ and $j = 1, 2, \ldots, p$.

The computation of $c_{ij}$ can be viewed as the multiplication of the entries of the $i$th row of $A$ with corresponding entries in the $j$th column of $B$, followed by a summation; that is,

$$\begin{bmatrix} a_{i1}, a_{i2}, \ldots, a_{im} \end{bmatrix} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{mj} \end{bmatrix} = [c_{ij}],$$

where

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{im} b_{mj} = \sum_{k=1}^{m} a_{ik} b_{kj}.$$
EXAMPLE 1 Let
\[
A = \begin{bmatrix}
2 & 1 & -1 \\
3 & 1 & 2 \\
0 & -2 & -3
\end{bmatrix}, \quad B = \begin{bmatrix}
3 & 2 \\
-1 & 1 \\
6 & 4
\end{bmatrix}, \\
C = \begin{bmatrix}
2 & 1 & 0 \\
-1 & 3 & 2
\end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix}
1 & -1 & 1 \\
2 & -1 & 2 \\
3 & 0 & 3
\end{bmatrix}.
\]

Then,
\[
AD = \begin{bmatrix}
1 & -3 & 1 \\
11 & -4 & 11 \\
-13 & 2 & -13
\end{bmatrix} \neq \begin{bmatrix}
-1 & -2 & -6 \\
1 & -3 & -10 \\
6 & -3 & -12
\end{bmatrix} = DA.
\]

Further,
\[
BC = \begin{bmatrix}
4 & 9 & 4 \\
-3 & 2 & 2 \\
8 & 18 & 8
\end{bmatrix} \quad \text{and} \quad CB = \begin{bmatrix}
5 & 5 \\
6 & 9
\end{bmatrix}
\]

are not even the same size. Finally,
\[
AB = \begin{bmatrix}
-1 & 1 \\
20 & 15 \\
-16 & -14
\end{bmatrix}
\]

but $BA$ cannot be computed.

A square matrix has the same number of rows as columns. A diagonal matrix is a square matrix $D = [d_{ij}]$ with $d_{ij} = 0$ whenever $i \neq j$. The identity matrix of order $n$, $I_n = [\delta_{ij}]$, is a diagonal matrix with entries
\[
\delta_{ij} = \begin{cases}
1, & \text{if } i = j, \\
0, & \text{if } i \neq j.
\end{cases}
\]

When the size of $I_n$ is clear, this matrix is generally written simply as $I$. For example, the identity matrix of order three is
\[
I = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

If $A$ is any $n \times n$ matrix, then $AI = IA = A$.

An $n \times n$ upper-triangular matrix $U = [u_{ij}]$ has, for each $j = 1, 2, \ldots, n$, the entries
\[
u_{ij} = 0, \quad \text{for each } i = j + 1, j + 2, \ldots, n;
\]

and a lower-triangular matrix $L = [l_{ij}]$ has, for each $j = 1, 2, \ldots, n$, the entries
\[
l_{ij} = 0, \quad \text{for each } i = 1, 2, \ldots, j - 1.
\]
(A diagonal matrix is both upper and lower triangular.)

In Example 1 we found that, in general, \( AB \neq BA \), even when both products are defined. However, the other arithmetic properties associated with multiplication do hold. For example, when \( A, B, \) and \( C \) are matrices of the appropriate size and \( \lambda \) is a scalar, we have

\[
A(BC) = (AB)C, \quad A(B + C) = AB + AC, \quad \text{and} \quad \lambda(AB) = (\lambda A)B = A(\lambda B).
\]

Certain \( n \times n \) matrices have the property that another \( n \times n \) matrix, which we will denote \( A^{-1} \), exists with \( AA^{-1} = A^{-1}A = I \). In this case \( A \) is said to be nonsingular, or invertible, and the matrix \( A^{-1} \) is called the inverse of \( A \). A matrix without an inverse is called singular, or noninvertible.

**EXAMPLE 2**

Let

\[
A = \begin{bmatrix}
1 & 2 & -1 \\
2 & 1 & 0 \\
-1 & 1 & 2
\end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix}
-2 & 5 & -1 \\
4 & -1 & 2 \\
-3 & 3 & 3
\end{bmatrix}.
\]

Since

\[
AB = \begin{bmatrix}
1 & 2 & -1 \\
2 & 1 & 0 \\
-1 & 1 & 2
\end{bmatrix} \cdot \begin{bmatrix}
-2 & 5 & -1 \\
4 & -1 & 2 \\
-3 & 3 & 3
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

and

\[
BA = \begin{bmatrix}
-2 & 5 & -1 \\
4 & -1 & 2 \\
-3 & 3 & 3
\end{bmatrix} \cdot \begin{bmatrix}
1 & 2 & -1 \\
2 & 1 & 0 \\
-1 & 1 & 2
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix},
\]

\( A \) and \( B \) are nonsingular with \( B = A^{-1} \) and \( A = B^{-1} \). \( \square \)

The reason for introducing this matrix operation at this time is that the linear system

\[
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\
\vdots \quad \vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n,
\]

can be viewed as the matrix equation \( Ax = b \), where

\[
A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}, \quad x = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{bmatrix}.
\]

If \( A \) is a nonsingular matrix, then the solution \( x \) to the linear system \( Ax = b \) is given by \( x = A^{-1}(Ax) = A^{-1}b \). In general, however, it is more difficult
to determine $A^{-1}$ than it is to solve the system $Ax = b$, because the number of operations involved in determining $A^{-1}$ is larger. Even so, it is useful from a conceptual standpoint to describe a method for determining the inverse of a matrix.

**EXAMPLE 3**

To determine the inverse of the matrix

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

let us first consider the product $AB$, where $B$ is an arbitrary $3 \times 3$ matrix.

$$AB = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & 0 \\ -1 & 1 & 2 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} b_{11} + 2b_{21} - b_{31} & b_{12} + 2b_{22} - b_{32} & b_{13} + 2b_{23} - b_{33} \\ 2b_{11} + b_{21} & 2b_{12} + b_{22} & 2b_{13} + b_{23} \\ -b_{11} + b_{21} + 2b_{31} & -b_{12} + b_{22} + 2b_{32} & -b_{13} + b_{23} + 2b_{33} \end{bmatrix}.$$

If $B = A^{-1}$, then $AB = I$, so we must have

$$b_{11} + 2b_{21} - b_{31} = 1, \quad b_{12} + 2b_{22} - b_{32} = 0, \quad b_{13} + 2b_{23} - b_{33} = 1,$$

$$2b_{11} + b_{21} = 0, \quad 2b_{12} + b_{22} = 1, \quad 2b_{13} + b_{23} = 0,$$

$$-b_{11} + b_{21} + 2b_{31} = 0, \quad -b_{12} + b_{22} + 2b_{32} = 0, \quad -b_{13} + b_{23} + 2b_{33} = 0.$$

Notice that the coefficients in each of the systems of equations are the same; the only change in the systems occurs on the right side of the equations. As a consequence, the computations can be performed on the larger augmented matrix, formed by combining the matrices for each of the systems

$$\begin{bmatrix} 1 & 2 & -1 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 2 & 0 & 0 & 1 \end{bmatrix}.$$
to eventually give
\[
\begin{align*}
  b_{11} &= -\frac{2}{9}, & b_{12} &= \frac{5}{9}, & b_{13} &= \frac{1}{9}, \\
  b_{21} &= \frac{4}{9}, & b_{22} &= \frac{1}{3}, & b_{23} &= \frac{2}{9}, \\
  b_{31} &= -\frac{1}{3}, & b_{32} &= \frac{1}{3}, & b_{33} &= \frac{1}{3}.
\end{align*}
\]
These are the entries of \(A^{-1}\):
\[
A^{-1} = \begin{bmatrix}
  \frac{-2}{9} & \frac{5}{9} & \frac{-1}{9} \\
  \frac{4}{9} & \frac{-1}{3} & \frac{2}{3} \\
  \frac{-1}{3} & \frac{1}{3} & \frac{1}{3}
\end{bmatrix} = \frac{1}{9} \begin{bmatrix}
  -2 & 5 & -1 \\
  4 & -1 & 2 \\
  -3 & 3 & 3
\end{bmatrix}.
\tag{6.1}
\]

The transpose of an \(n \times m\) matrix \(A = [a_{ij}]\) is the \(m \times n\) matrix \(A^t = [a_{ji}]\). A square matrix \(A\) is symmetric if \(A = A^t\).

**Example 4**

The matrices
\[
A = \begin{bmatrix}
  7 & 2 & 0 \\
  3 & 5 & -1 \\
  0 & 5 & -6
\end{bmatrix}, \quad B = \begin{bmatrix}
  2 & 4 & 7 \\
  3 & -5 & -1 \\
  -3 & 0 & 1
\end{bmatrix}, \quad C = \begin{bmatrix}
  6 & 4 & -3 \\
  4 & -2 & 0 \\
  -3 & 0 & 1
\end{bmatrix}
\]

have transposes
\[
A^t = \begin{bmatrix}
  7 & 3 & 0 \\
  2 & 5 & 5 \\
  0 & -1 & -6
\end{bmatrix}, \quad B^t = \begin{bmatrix}
  2 & 3 \\
  4 & -5 \\
  7 & -1
\end{bmatrix}, \quad C^t = \begin{bmatrix}
  6 & 4 & -3 \\
  4 & -2 & 0 \\
  -3 & 0 & 1
\end{bmatrix}.
\]
The matrix \(C\) is symmetric, since \(C^t = C\). The matrices \(A\) and \(B\) are not symmetric.

The following operations involving the transpose of a matrix hold whenever the operation is possible.

[Transpose Facts]
\[
\begin{align*}
  (i) & \quad (A^t)^t = A. \\
  (ii) & \quad (A + B)^t = A^t + B^t. \\
  (iii) & \quad (AB)^t = B^t A^t. \\
  (iv) & \quad \text{If } A^{-1} \text{ exists, } (A^{-1})^t = (A^t)^{-1}.
\end{align*}
\]

The determinant of a square matrix is a number that can be useful in determining the existence and uniqueness of solutions to linear systems. We denote the determinant of a matrix \(A\) by \(\det A\), but it is also common to use the notation \(|A|\).
CHAPTER 6. DIRECT METHODS FOR SOLVING LINEAR SYSTEMS

Determinant of a Matrix

(i) If $A = [a]$ is a $1 \times 1$ matrix, then $\det A = a$.

(ii) If $A$ is an $n \times n$ matrix, the minor $M_{ij}$ is the determinant of the $(n-1) \times (n-1)$ submatrix of $A$ obtained by deleting the $i$th row and $j$th column of the matrix $A$.

Then the determinant of $A$ is given either by

$$\det A = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} M_{ij} \quad \text{for any } i = 1, 2, \ldots, n,$$

or by

$$\det A = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} M_{ij} \quad \text{for any } j = 1, 2, \ldots, n.$$

To calculate the determinant of a general $n \times n$ matrix by expanding by minors requires $O(n!)$ multiplications/divisions and additions/subtractions. Even for relatively small values of $n$, the number of calculations becomes unwieldy. Fortunately, the precise value of the determinant is seldom needed, and there are efficient ways to approximate its value.

Although it appears that there are $2^n$ different definitions of $\det A$, depending on which row or column is chosen, all definitions give the same numerical result. The flexibility in the definition is used in the following example. It is most convenient to compute $\det A$ across the row or down the column with the most zeros.

**Example 5**

Let

$$A = \begin{bmatrix} 2 & -1 & 3 & 0 \\ 4 & -2 & 7 & 0 \\ -3 & -4 & 1 & 5 \\ 6 & -6 & 8 & 0 \end{bmatrix}.$$ 

To compute $\det A$, it is easiest to expand about the fourth column:

$$\det A = -a_{14} M_{14} + a_{24} M_{24} - a_{34} M_{34} + a_{44} M_{44} = -5M_{34}$$

$$= -5 \det \begin{bmatrix} 2 & -1 & 3 \\ 4 & -2 & 7 \\ 6 & -6 & 8 \end{bmatrix}$$

$$= -5 \left\{ 2 \det \begin{bmatrix} -2 & 7 \\ -6 & 8 \end{bmatrix} - (-1) \det \begin{bmatrix} 4 & 7 \\ 6 & 8 \end{bmatrix} + 3 \det \begin{bmatrix} 4 & -2 \\ 6 & -6 \end{bmatrix} \right\}$$

$$= -5 \{ 2(-16 + 42) - (-1)(32 - 42) + 3(-24 + 12) \} = -30.$$  

(6.2)
The following properties of determinants are useful in relating linear systems and Gaussian elimination to determinants.

[Determinant Facts] Suppose $A$ is an $n \times n$ matrix:

(i) If any row or column of $A$ has only zero entries, then $\det A = 0$.

(ii) If $\tilde{A}$ is obtained from $A$ by the operation $(E_i) \leftrightarrow (E_k)$, with $i \neq k$, then $\det \tilde{A} = -\det A$.

(iii) If $A$ has two rows or two columns the same, then $\det A = 0$.

(iv) If $\tilde{A}$ is obtained from $A$ by the operation $(\lambda E_i) \rightarrow (E_i)$, then $\det \tilde{A} = \lambda \det A$.

(v) If $\tilde{A}$ is obtained from $A$ by the operation $(E_i + \lambda E_k) \rightarrow (E_i)$ with $i \neq k$, then $\det \tilde{A} = \det A$.

(vi) If $B$ is also an $n \times n$ matrix, then $\det AB = \det A \det B$.

(vii) $\det A^t = \det A$.

(viii) If $A^{-1}$ exists, then $\det A^{-1} = \frac{1}{\det A}$.

(ix) If $A$ is an upper triangular, lower triangular, or diagonal matrix, then $\det A = a_{11} \cdot a_{22} \cdots a_{nn}$.

**EXAMPLE 6** We will compute the determinant of the matrix

\[
A = \begin{bmatrix}
2 & 1 & -1 & 1 \\
1 & 1 & 0 & 3 \\
-1 & 2 & 3 & -1 \\
3 & -1 & -1 & 2
\end{bmatrix}
\]

using Determinant Facts (ii), (iv), and (vi) and Maple. Matrix $A$ is defined by

>`A:=matrix(4,4,[2,1,-1,1,1,1,0,3,-1,2,3,-1,3,-1,-1,2]);`

The sequence of operations in Table 6.2 produces the matrix

\[
A^8 = \begin{bmatrix}
1 & \frac{1}{7} & -\frac{1}{7} & \frac{1}{2} \\
0 & 1 & 1 & 5 \\
0 & 0 & 3 & 13 \\
0 & 0 & 0 & -13
\end{bmatrix}.
\]

By fact (ix), $\det A^8 = (1)(1)(3)(-13) = -39$, so $\det A = -\det A^8 = -39$. \qed
### Table 6.2

<table>
<thead>
<tr>
<th>Operation</th>
<th>Maple</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{2}E_1 \rightarrow E_1 )</td>
<td>( A1:= \text{mulrow}(A,1,0.5) )</td>
<td>( \text{det } A1 = \frac{1}{2} \text{ det } A )</td>
</tr>
<tr>
<td>( E_2 - E_1 \rightarrow E_2 )</td>
<td>( A2:= \text{addrow}(A1,1,2,-1) )</td>
<td>( \text{det } A2 = \text{det } A1 = \frac{1}{2} \text{ det } A )</td>
</tr>
<tr>
<td>( E_3 + E_1 \rightarrow E_3 )</td>
<td>( A3:= \text{addrow}(A2,1,3,1) )</td>
<td>( \text{det } A3 = \text{det } A2 = \frac{1}{2} \text{ det } A )</td>
</tr>
<tr>
<td>( E_4 - 3E_1 \rightarrow E_4 )</td>
<td>( A4:= \text{addrow}(A3,1,4,-3) )</td>
<td>( \text{det } A4 = \text{det } A3 = \frac{1}{2} \text{ det } A )</td>
</tr>
<tr>
<td>( 2E_2 \rightarrow E_2 )</td>
<td>( A5:= \text{mulrow}(A,2,2) )</td>
<td>( \text{det } A5 = 2 \text{ det } A4 = \text{det } A )</td>
</tr>
<tr>
<td>( E_3 - \frac{3}{2}E_2 \rightarrow E_3 )</td>
<td>( A6:= \text{addrow}(A,2,3,-2.5) )</td>
<td>( \text{det } A6 = \text{det } A5 = \text{det } A )</td>
</tr>
<tr>
<td>( E_4 + \frac{5}{2}E_2 \rightarrow E_4 )</td>
<td>( A7:= \text{addrow}(A,2,4,2.5) )</td>
<td>( \text{det } A7 = \text{det } A6 = \text{det } A )</td>
</tr>
<tr>
<td>( E_3 \leftrightarrow E_4 )</td>
<td>( A8:= \text{swaprow}(A,3,4) )</td>
<td>( \text{det } A8 = - \text{det } A7 = - \text{det } A )</td>
</tr>
</tbody>
</table>

The key result relating nonsingularity, Gaussian elimination, linear systems, and determinants is that the following statements are equivalent.

---

**[Equivalent Statements about an \( n \times n \) Matrix \( A \)]**

(i) The equation \( Ax = 0 \) has the unique solution \( x = 0 \).

(ii) The system \( Ax = b \) has a unique solution for any \( n \)-dimensional column vector \( b \).

(iii) The matrix \( A \) is nonsingular; that is, \( A^{-1} \) exists.

(iv) \( \text{det } A \neq 0 \).

(v) Gaussian elimination with row interchanges can be performed on the system \( Ax = b \) for any \( n \)-dimensional column vector \( b \).

---

Maple can be used to perform the arithmetic operations on matrices. Matrix addition is done with \text{matadd}(A,B)\ or \text{evalm}(A+B)\). Scalar multiplication is defined by \text{scalarmul}(A,C)\ or \text{evalm}(C*A)\). Matrix multiplication is done using \text{multiply}(A,B)\ or \text{evalm}(A&B)\). The matrix operation of transposition is achieved with \text{transpose}(A)\, matrix inversion with \text{inverse}(A)\, and the determinant with \text{det}(A)\.
EXERCISE SET 6.4

1. Compute the following matrix products.

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

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

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

(d) \[
\begin{bmatrix}
2 & -1 & 4 \\
0 & -1 & 2 \\
0 & 0 & 3
\end{bmatrix}
\begin{bmatrix}
3 & -3 & 4 \\
0 & 1 & 1 \\
0 & 0 & 2
\end{bmatrix}
\]

2. For the matrices given below:

i. Find the transpose of the matrix.

ii. Determine which matrices are nonsingular and compute their inverses.

(a) \[
\begin{bmatrix}
4 & 2 & 6 \\
3 & 0 & 7 \\
-2 & -1 & -3
\end{bmatrix}
\]

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

(c) \[
\begin{bmatrix}
4 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 3
\end{bmatrix}
\]

(d) \[
\begin{bmatrix}
1 & 1 & -1 \\
1 & 2 & -4 \\
2 & 1 & 1 \\
-1 & 0 & -2 \\
0 & 1 & 5
\end{bmatrix}
\]

(e) \[
\begin{bmatrix}
4 & 0 & 0 \\
6 & 7 & 0 \\
9 & 11 & 1 \\
5 & 4 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

(f) \[
\begin{bmatrix}
2 & 0 & 1 & 2 \\
1 & 1 & 0 & 2 \\
2 & -1 & 3 & 1 \\
3 & -1 & 4 & 3
\end{bmatrix}
\]

3. Compute the determinants of the matrices in Exercise 2 and the determinants of the inverse matrices of those that are nonsingular.

4. Consider the four $3 \times 3$ linear systems having the same coefficient matrix:

\[
\begin{align*}
2x_1 - 3x_2 + x_3 &= 2, & 2x_1 - 3x_2 + x_3 &= 6, \\
x_1 + x_2 - x_3 &= -1, & x_1 + x_2 - x_3 &= 4, \\
-x_1 + x_2 - 3x_3 &= 0, & -x_1 + x_2 - 3x_3 &= 5, \\
2x_1 - 3x_2 + x_3 &= 0, & 2x_1 - 3x_2 + x_3 &= -1, \\
x_1 + x_2 - x_3 &= 1, & x_1 + x_2 - x_3 &= 0, \\
-x_1 + x_2 - 3x_3 &= -3, & -x_1 + x_2 - 3x_3 &= 0.
\end{align*}
\]
(a) Solve the linear systems by applying Gaussian elimination to the augmented matrix

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

(b) Solve the linear systems by finding and multiplying by the inverse of

\[
A = \begin{bmatrix}
2 & -3 & 1 \\
1 & 1 & -1 \\
-1 & 1 & -3
\end{bmatrix}.
\]

(c) Which method requires more operations?

5. Show that the following statements are true or provide counterexamples to show they are not.

(a) The product of two symmetric matrices is symmetric.

(b) The inverse of a nonsingular symmetric matrix is a nonsingular symmetric matrix.

(c) If \( A \) and \( B \) are \( n \times n \) matrices, then \((AB)^t = A^tB^t\).

6. (a) Show that the product of two \( n \times n \) lower triangular matrices is lower triangular.

(b) Show that the product of two \( n \times n \) upper triangular matrices is upper triangular.

(c) Show that the inverse of a nonsingular \( n \times n \) lower triangular matrix is lower triangular.

7. The solution by Cramer’s rule to the linear system

\[
\begin{align*}
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} &= \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},
\end{align*}
\]

has

\[
x_1 = \frac{1}{D} \det \begin{bmatrix}
b_1 & a_{12} & a_{13} \\
b_2 & a_{22} & a_{23} \\
b_3 & a_{32} & a_{33}
\end{bmatrix} = \frac{D_1}{D},
\]

where \( D \) is the determinant of the coefficient matrix and \( D_1 \) is the determinant of the matrix obtained by replacing the first column of the coefficient matrix with the constants \( b_1, b_2, b_3 \).
6.4. LINEAR ALGEBRA AND MATRIX INVERSION

\[ x_2 = \frac{1}{D} \det \begin{bmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{bmatrix} = \frac{D_2}{D}, \]

and

\[ x_3 = \frac{1}{D} \det \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{bmatrix} = \frac{D_3}{D}, \]

where

\[ D = \det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \]

(a) Use Cramer’s rule to find the solution to the linear system

\[
\begin{align*}
2x_1 + 3x_2 - x_3 &= 4, \\
x_1 - 2x_2 + x_3 &= 6, \\
x_1 - 12x_2 + 5x_3 &= 10.
\end{align*}
\]

(b) Show that the linear system

\[
\begin{align*}
2x_1 + 3x_2 - x_3 &= 4, \\
x_1 - 2x_2 + x_3 &= 6, \\
x_1 - 2x_2 - 12x_3 &= 9
\end{align*}
\]
does not have a solution. Compute \( D_1, D_2, \) and \( D_3. \)

(c) Show that the linear system

\[
\begin{align*}
2x_1 + 3x_2 - x_3 &= 4, \\
x_1 - 2x_2 + x_3 &= 6, \\
x_1 - 12x_2 + 5x_3 &= 10
\end{align*}
\]

has an infinite number of solutions. Compute \( D_1, D_2, \) and \( D_3. \)

(d) Suppose that a \( 3 \times 3 \) linear system with \( D = 0 \) has solutions. Explain why we must also have \( D_1 = D_2 = D_3 = 0. \)

8. In a paper entitled “Population Waves,” Bernadelli [Ber] hypothesizes a type of simplifed beetle, which has a natural life span of 3 years. The female of this species has a survival rate of \( \frac{1}{2} \) in the first year of life, has a survival rate of \( \frac{1}{4} \) from the second to third years, and gives birth to an average of six new females before expiring at the end of the third year. A matrix can be used to show the contribution an individual female beetle makes, in a probabilistic
sense, to the female population of the species by letting $a_{ij}$ in the matrix $A = [a_{ij}]$ denote the contribution that a single female beetle of age $j$ will make to the next year’s female population of age $i$; that is,

$$
A = \begin{bmatrix}
0 & 0 & 6 \\
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{4} & 0
\end{bmatrix}.
$$

(a) The contribution that a female beetle makes to the population 2 years hence is determined from the entries of $A^2$, of 3 years hence from $A^3$, and so on. Construct $A^2$ and $A^3$, and try to make a general statement about the contribution of a female beetle to the population in $n$ years’ time for any positive integral value of $n$.

(b) Use your conclusions from part (a) to describe what will occur in future years to a population of these beetles that initially consists of 6000 female beetles in each of the three age groups.

(c) Construct $A^{-1}$ and describe its significance regarding the population of this species.

9. The study of food chains is an important topic in the determination of the spread and accumulation of environmental pollutants in living matter. Suppose that a food chain has three links. The first link consists of vegetation of types $v_1, v_2, \ldots, v_n$, which provide all the food requirements for herbivores of species $h_1, h_2, \ldots, h_m$ in the second link. The third link consists of carnivorous animals $c_1, c_2, \ldots, c_k$, which depend entirely on the herbivores in the second link for their food supply. The coordinate $a_{ij}$ of the matrix

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

represents the total number of plants of type $v_i$ eaten by the herbivores in the species $h_j$, whereas $b_{ij}$ in

$$
B = \begin{bmatrix}
b_{11} & b_{12} & \cdots & b_{1k} \\
b_{21} & b_{22} & \cdots & b_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
b_{m1} & b_{m2} & \cdots & b_{mk}
\end{bmatrix}
$$

describes the number of herbivores in species $h_i$ that are devoured by the animals of type $c_j$.

(a) Show that the number of plants of type $v_i$ that eventually end up in the animals of species $c_j$ is given by the entry in the $i$th row and $j$th column of the matrix $AB$. 
(b) What physical significance is associated with the matrices $A^{-1}, B^{-1},$ and $(AB)^{-1} = B^{-1}A^{-1}$?

10. In Section 3.6 we found that the parametric form $(x(t), y(t))$ of the cubic Hermite polynomials through $(x(0), y(0)) = (x_0, y_0)$ and $(x(1), y(1)) = (x_1, y_1)$ with guidepoints $(x_0 + \alpha_0, y_0 + \beta_0)$ and $(x_1 - \alpha_1, y_1 - \beta_1)$, respectively, is given by

\[
\begin{align*}
x(t) &= [2(x_0 - x_1) + (\alpha_0 + \alpha_1)]t^3 + [3(x_1 - x_0) - \alpha_1 - 2\alpha_0]t^2 + \alpha_0 t + x_0 \\
y(t) &= [2(y_0 - y_1) + (\beta_0 + \beta_1)]t^3 + [3(y_1 - y_0) - \beta_1 - 2\beta_0]t^2 + \beta_0 t + y_0.
\end{align*}
\]

The Bézier cubic polynomials have the form

\[
\begin{align*}
\hat{x}(t) &= [2(x_0 - x_1) + 3(\alpha_0 + \alpha_1)]t^3 + [3(x_1 - x_0) - 3(\alpha_1 + 2\alpha_0)]t^2 + 3\alpha_0 t + x_0 \\
\hat{y}(t) &= [2(y_0 - y_1) + 3(\beta_0 + \beta_1)]t^3 + [3(y_1 - y_0) - 3(\beta_1 + 2\beta_0)]t^2 + 3\beta_0 t + y_0.
\end{align*}
\]

(a) Show that the matrix

\[
A = \begin{bmatrix}
7 & 4 & 4 & 0 \\
-6 & -3 & -6 & 0 \\
0 & 0 & 3 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

maps the Hermite polynomial coefficients onto the Bézier polynomial coefficients.

(b) Determine a matrix $B$ that maps the Bézier polynomial coefficients onto the Hermite polynomial coefficients.

11. Consider the $2 \times 2$ linear system $(A + iB)(x + iy) = c + id$ with complex entries in component form:

\[
\begin{align*}
(a_{11} + ib_{11})(x_1 + iy_1) + (a_{12} + ib_{12})(x_2 + iy_2) &= c_1 + id_1, \\
(a_{21} + ib_{21})(x_1 + iy_1) + (a_{22} + ib_{22})(x_2 + iy_2) &= c_2 + id_2.
\end{align*}
\]

(a) Use the properties of complex numbers to convert this system to the equivalent $4 \times 4$ real linear system

\[
\begin{align*}
\text{Real part:} & \quad Ax - By = c, \\
\text{Imaginary part:} & \quad Bx + Ay = d.
\end{align*}
\]

(b) Solve the linear system

\[
\begin{align*}
(1 - 2i)(x_1 + iy_1) + (3 + 2i)(x_2 + iy_2) &= 5 + 2i, \\
(2 + i)(x_1 + iy_1) + (4 + 3i)(x_2 + iy_2) &= 4 - i.
\end{align*}
\]
6.5 Matrix Factorization

Gaussian elimination is the principal tool for the direct solution of linear systems of equations, so it should come as no surprise to learn that it can be used for other purposes. In this section we will see that the steps used to solve a system of the form \( Ax = b \) by Gaussian Elimination can be used to factor the matrix \( A \) into a product of matrices that are easier to manipulate. The factorization is particularly useful when it has the form \( A = LU \), where \( L \) is lower triangular and \( U \) is upper triangular. Although not all matrices have this type of representation, many do that occur frequently in the application of numerical techniques.

In Section 6.2 we found that Gaussian elimination applied to an arbitrary nonsingular system requires \( O(n^3) \) operations to determine \( x \). However, to solve a linear system that involves an upper-triangular system requires only backward substitution, which takes \( O(n^2) \) operations. The situation is similar for lower-triangular systems. So if \( A \) has been factored into the triangular form \( A = LU \), then we can solve for \( x \) more easily by using a two-step process. First we let \( y = Ux \) and solve the system \( Ly = b \) for \( y \). Since \( L \) is lower triangular, determining \( y \) from this equation requires only \( O(n^2) \) operations. Once \( y \) is known, the upper triangular system \( Ux = y \) requires only an additional \( O(n^2) \) operations to determine the solution \( x \). This means that the total number of operations needed to solve the system \( Ax = b \) is reduced from \( O(n^3) \) to \( O(n^2) \). In systems greater than 100 by 100, this can reduce the amount of calculation by more than 99%, since \( 100^2 = 10,000 = (0.01)(1,000,000) = (0.01)(100)^3 \).

Not surprisingly, the reductions from the factorization do not come free; determining the specific matrices \( L \) and \( U \) requires \( O(n^3) \) operations. But once the factorization is determined, systems involving the matrix \( A \) can be solved in this simplified manner for any number of vectors \( b \).

**Example 1**

The linear system

\[
\begin{align*}
2x_1 + x_2 - x_3 + x_4 &= 1, \\
3x_1 - x_2 - x_3 + 2x_4 &= -3, \\
x_1 + x_2 + 3x_3 - x_4 &= 4
\end{align*}
\]

was considered in Section 6.2. The sequence of operations \( (E_2 - 2E_1) \rightarrow (E_2), \ (E_3 - 3E_1) \rightarrow (E_3), \ (E_4 - (-1)E_1) \rightarrow (E_4), \ (E_3 - 4E_2) \rightarrow (E_3), \ (E_4 - (-3)E_2) \rightarrow (E_4) \) converts the system to one that has the upper triangular form

\[
\begin{align*}
x_1 + x_2 + 3x_4 &= 4, \\
x_2 + x_3 - 5x_4 &= -7, \\
3x_3 + 13x_4 &= 13, \\
-13x_4 &= -13.
\end{align*}
\]

Let \( U \) be the upper triangular matrix with these coefficients as its entries and \( L \) be the lower triangular matrix with 1s along the diagonal and the multipliers \( m_{kj} \) as
6.5. MATRIX FACTORIZATION

entries below the diagonal. Then we have the factorization

\[
A = \begin{bmatrix}
1 & 1 & 0 & 3 \\
2 & 1 & -1 & 1 \\
3 & -1 & -1 & 2 \\
-1 & 2 & 3 & -1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 4 & 1 & 0 \\
-1 & -3 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 0 & 3 \\
0 & -1 & -1 & -5 \\
0 & 0 & 3 & 13 \\
0 & 0 & 0 & -13
\end{bmatrix} = LU.
\]

This factorization permits us to easily solve any system involving the matrix \(A\). For example, to solve

\[
\begin{bmatrix}
10 & 0 & 0 \\
21 & 0 & 0 \\
34 & 1 & 0 \\
-1 & -3 & 0 
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix}
= \begin{bmatrix}
8 \\
7 \\
14 \\
-7
\end{bmatrix},
\]

we first introduce the substitution \(y = Ux\). Then \(Ly = b\); that is,

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 4 & 1 & 0 \\
-1 & -3 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= \begin{bmatrix}
8 \\
7 \\
14 \\
-7
\end{bmatrix}.
\]

This system is solved for \(y\) by a simple forward substitution process:

\[
y_1 = 8, \\
2y_1 + y_2 = 7, \ \text{so} \ y_2 = 7 - 2y_1 = -9, \\
3y_1 + 4y_2 + y_3 = 14, \ \text{so} \ y_3 = 14 - 3y_1 - 4y_2 = 26, \\
-y_1 - 3y_2 + y_4 = -7, \ \text{so} \ y_4 = -7 + y_1 + 3y_2 = -26.
\]

We then solve \(Ux = y\) for \(x\), the solution of the original system; that is,

\[
\begin{bmatrix}
1 & 1 & 0 & 3 \\
0 & -1 & -1 & -5 \\
0 & 0 & 3 & 13 \\
0 & 0 & 0 & -13
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4
\end{bmatrix}
= \begin{bmatrix}
8 \\
-9 \\
26 \\
-26
\end{bmatrix}.
\]

Using backward substitution we obtain \(x_4 = 2, x_3 = 0, x_2 = -1, x_1 = 3\).

In the factorization of \(A = LU\), we generate the first column of \(L\) and the first row of \(U\) using the equations

\[
l_{11}u_{11} = a_{11}
\]

and, for each \(j = 2, 3, \ldots, n\),

\[
l_{j1} = \frac{a_{j1}}{u_{11}} \quad \text{and} \quad u_{1j} = \frac{a_{1j}}{l_{11}}.
\]

For each \(i = 2, 3, \ldots, n - 1\), we select the diagonal entries \(u_{ii}\) and \(l_{ii}\) and generate the remaining entries in the \(i\)th column of \(L\) and the \(i\)th row of \(U\). The required equations are

\[
l_{ij}u_{ji} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj};
\]
and, for each $j = i + 1, \ldots, n$,

\[
l_{ji} = \frac{1}{u_{ii}} \left[ a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right] \quad \text{and} \quad u_{ij} = \frac{1}{l_{ii}} \left[ a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right].
\]

Finally, $l_{nn}$ and $u_{nn}$ are selected to satisfy

\[
l_{nn} u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn}.
\]

A general procedure for factoring matrices into a product of triangular matrices is performed in the program LUFAC764. Although new matrices $L$ and $U$ are constructed, the values generated replace the corresponding entries of $A$ that are no longer needed. Thus, the new matrix has entries $a_{ij} = l_{ij}$ for each $i = 2, 3, \ldots, n$ and $j = 1, 2, \ldots, i - 1$ and $a_{ij} = u_{ij}$ for each $i = 1, 2, \ldots, n$ and $j = i + 1, i + 2, \ldots, n$.

The factorization is particularly useful when a number of linear systems involving $A$ must be solved, since the bulk of the operations need to be performed only once. To solve $LUx = b$, we first solve $Ly = b$ for $y$. Since $L$ is lower triangular, we have

\[
y_1 = \frac{b_1}{l_{11}}
\]

and

\[
y_i = \frac{1}{l_{ii}} \left[ b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right], \quad \text{for each } i = 2, 3, \ldots, n.
\]

Once $y$ is calculated by this forward substitution process, the upper triangular system $Ux = y$ is solved by backward substitution using the equations

\[
x_n = \frac{y_n}{u_{nn}} \quad \text{and} \quad x_i = \frac{1}{u_{ii}} \left[ y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right].
\]

In the previous discussion we assumed that $A$ is such that a linear system of the form $Ax = b$ can be solved using Gaussian elimination that does not require row interchanges. From a practical standpoint, this factorization is useful only when row interchanges are not required to control the round-off error resulting from the use of finite-digit arithmetic. Although many systems we encounter when using approximation methods are of this type, factorization modifications must be made when row interchanges are required. We begin the discussion with the introduction of a class of matrices that are used to rearrange, or permute, rows of a given matrix.

An $n \times n$ permutation matrix $P$ is a matrix with precisely one entry whose value is 1 in each column and each row and all of whose other entries are 0.

**EXAMPLE 2** The matrix

\[
P = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]
is a $3 \times 3$ permutation matrix. For any $3 \times 3$ matrix $A$, multiplying on the left by $P$ has the effect of interchanging the second and third rows of $A$:

$$PA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}. $$

Similarly, multiplying on the right by $P$ interchanges the second and third columns of $A$.

There are two useful properties of permutation matrices that relate to Gaussian elimination. The first of these is illustrated in the previous example and states that if $k_1, \ldots, k_n$ is a permutation of the integers $1, \ldots, n$ and the permutation matrix $P = [p_{ij}]$ is defined by

$$p_{ij} = \begin{cases} 1, & \text{if } j = k_i \\ 0, & \text{otherwise,} \end{cases}$$

then

$$PA = \begin{bmatrix} a_{k_1,1} & a_{k_1,2} & \cdots & a_{k_1,n} \\ a_{k_2,1} & a_{k_2,2} & \cdots & a_{k_2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k_n,1} & a_{k_n,2} & \cdots & a_{k_n,n} \end{bmatrix}. $$

The second is that if $P$ is a permutation matrix, then $P^{-1}$ exists and $P^{-1} = P^t$.

**EXAMPLE 3**

Since $a_{11} = 0$, the matrix

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

does not have an $LU$ factorization. However, using the row interchange $(E_1) \leftrightarrow (E_2)$, followed by $(E_3 + E_1) \rightarrow E_3$ and $(E_4 - E_1) \rightarrow E_4$, produces

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

Then the row interchange $(E_3) \leftrightarrow (E_4)$, followed by $(E_3 - E_2) \rightarrow E_3$, gives the matrix

$$U = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 2 \end{bmatrix}. $$
The permutation matrix associated with the row interchanges \((E_1) \leftrightarrow (E_2)\) and \((E_3) \leftrightarrow (E_4)\) is

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

Gaussian elimination can be performed on \(PA\) without row interchanges to give the \(LU\) factorization

\[
PA = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
-1 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
1 & 1 & -1 & -2 \\
0 & 1 & -1 & 1 \\
0 & 0 & 2 & -1 \\
0 & 0 & 0 & 2 \\
\end{bmatrix} = LU.
\]

So

\[
A = P^{-1}(LU) = P^t(LU) = (P^tL)U = \begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{bmatrix} \begin{bmatrix}
1 & 1 & -1 & 2 \\
0 & 1 & -1 & 1 \\
0 & 0 & 2 & -1 \\
0 & 0 & 0 & 2 \\
\end{bmatrix}.
\]

Maple has the command `LUdecomp` to compute a factorization of the form \(A = PLU\) of the matrix \(A\). If the matrix \(A\) has been created, the function call

\[
> \text{U:=LUdecomp}(A,P='G',L='H');
\]

returns the upper triangular matrix \(U\) as the value of the function and returns the lower triangular matrix \(L\) in \(H\) and the permutation matrix \(P\) in \(G\).
EXERCISE SET 6.5

1. Solve the following linear systems.

   (a) \[
   \begin{bmatrix}
   1 & 0 & 0 \\
   2 & 1 & 0 \\
   -1 & 0 & 1
   \end{bmatrix}
   \begin{bmatrix}
   2 & 3 & -1 \\
   0 & -2 & 1 \\
   0 & 0 & 3
   \end{bmatrix}
   \begin{bmatrix}
   x_1 \\
   x_2 \\
   x_3
   \end{bmatrix}
   =
   \begin{bmatrix}
   2 \\
   -1 \\
   1
   \end{bmatrix}
   \]

   (b) \[
   \begin{bmatrix}
   2 & 0 & 0 \\
   -1 & 1 & 0 \\
   3 & 2 & -1
   \end{bmatrix}
   \begin{bmatrix}
   1 & 1 & 1 \\
   0 & 1 & 2 \\
   0 & 0 & 1
   \end{bmatrix}
   \begin{bmatrix}
   x_1 \\
   x_2 \\
   x_3
   \end{bmatrix}
   =
   \begin{bmatrix}
   -1 \\
   3 \\
   0
   \end{bmatrix}
   \]

2. Factor the following matrices into the \( LU \) decomposition with \( l_{ii} = 1 \) for all \( i \).

   (a) \[
   \begin{bmatrix}
   2 & -1 & 1 \\
   3 & 3 & 9 \\
   3 & 3 & 5
   \end{bmatrix}
   \]

   (b) \[
   \begin{bmatrix}
   1.012 & -2.132 & 3.104 \\
   -2.132 & 4.096 & -7.013 \\
   3.104 & -7.013 & 0.014
   \end{bmatrix}
   \]

   (c) \[
   \begin{bmatrix}
   2 & 0 & 0 & 0 \\
   1 & 1.5 & 0 & 0 \\
   0 & -3 & 0.5 & 0 \\
   2 & -2 & 1 & 1
   \end{bmatrix}
   \]

   (d) \[
   \begin{bmatrix}
   2.1756 & 4.0231 & -2.1732 & 5.1967 \\
   -4.0231 & 6.0000 & 0 & 1.1973 \\
   -1.0000 & -5.2107 & 1.1111 & 0 \\
   6.0235 & 7.0000 & 0 & -4.1561
   \end{bmatrix}
   \]

3. Obtain factorizations of the form \( A = P^tLU \) for the following matrices.

   (a) \( A = \begin{bmatrix}
   0 & 2 & 3 \\
   1 & 1 & -1 \\
   0 & -1 & 1
   \end{bmatrix} \)

   (b) \( A = \begin{bmatrix}
   1 & 2 & -1 \\
   1 & 2 & 3 \\
   2 & -1 & 4
   \end{bmatrix} \)

   (c) \( A = \begin{bmatrix}
   1 & -2 & 3 & 0 \\
   3 & -6 & 9 & 3 \\
   2 & 1 & 4 & 1 \\
   1 & -2 & 2 & -2
   \end{bmatrix} \)

   (d) \( A = \begin{bmatrix}
   1 & -2 & 3 & 0 \\
   1 & -2 & 3 & 1 \\
   1 & -2 & 2 & -2 \\
   2 & 1 & 3 & -1
   \end{bmatrix} \)

4. Suppose \( A = P^tLU \), where \( P \) is a permutation matrix, \( L \) is a lower-triangular matrix with 1s on the diagonal, and \( U \) is an upper-triangular matrix.

   (a) Count the number of operations needed to compute \( P^tLU \) for a given matrix \( A \).
(b) Show that if $P$ contains $k$ row interchanges, then

$$\det P = \det P^t = (-1)^k.$$  

(c) Use $\det A = \det P^t \det L \det U = (-1)^k \det U$ to count the number of operations for determining $\det A$ by factoring.

(d) Compute $\det A$ and count the number of operations when

$$A = \begin{bmatrix}
0 & 2 & 1 & 4 & -1 & 3 \\
1 & 2 & -1 & 3 & 4 & 0 \\
0 & 1 & 1 & -1 & 2 & -1 \\
2 & 3 & -4 & 2 & 0 & 5 \\
1 & 1 & 1 & 3 & 0 & 2 \\
-1 & -1 & 2 & -1 & 2 & 0
\end{bmatrix}.$$
6.6 Techniques for Special Matrices

Although this chapter has been concerned primarily with the effective application of Gaussian elimination for finding the solution to a linear system of equations, many of the results have wider application. It might be said that Gaussian elimination is the hub about which the chapter revolves, but the wheel itself is of equal interest and has application in many forms in the study of numerical methods. In this section we consider some matrices that are of special types, forms that will be used in other chapters of the book.

The \( n \times n \) matrix \( A \) is said to be \textbf{strictly diagonally dominant} when

\[
|a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ij}|
\]

holds for each \( i = 1, 2, \ldots, n \).

**EXAMPLE 1** Consider the matrices

\[
A = \begin{bmatrix}
7 & 2 & 0 \\
3 & 5 & -1 \\
0 & 5 & -6
\end{bmatrix}
\text{ and } B = \begin{bmatrix}
6 & 4 & -3 \\
4 & -2 & 0 \\
-3 & 0 & 1
\end{bmatrix}.
\]

The nonsymmetric matrix \( A \) is strictly diagonally dominant, since \(|7| > |2| + |0|, |5| > |3| + |-1|, \) and \(|-6| > |0| + |5|\). The symmetric matrix \( B \) is not strictly diagonally dominant, because, for example, in the first row the absolute value of the diagonal element is \(|6| < |4| + |-3| = 7\). It is interesting to note that \( A^t \) is not strictly diagonally dominant, since the middle row of \( A^t \) is \( 2 \ 5 \ 0 \), nor, of course, is \( B^t \) since \( B^t = B \). \( \square \)

[Strictly Diagonally Dominant Matrices] A strictly diagonally dominant matrix \( A \) has an inverse. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form \( Ax = b \) to obtain its unique solution without row or column interchanges, and the computations are stable with respect to the growth of round-off error.

A matrix \( A \) is \textbf{positive definite} if it is symmetric and if \( x^tAx > 0 \) for every \( n \)-dimensional column vector \( x \neq 0 \). Using the definition to determine whether a matrix is positive definite can be difficult. Fortunately, there are more easily verified criteria for identifying members that are and are not of this important class.
[Positive Definite Matrix Properties] If $A$ is an $n \times n$ positive definite matrix, then

(i) $A$ has an inverse;
(ii) $a_{ii} > 0$ for each $i = 1, 2, \ldots, n$;
(iii) $\max_{1 \leq k, j \leq n} |a_{kj}| \leq \max_{1 \leq i \leq n} |a_{ii}|$;
(iv) $(a_{ij})^2 < a_{ii}a_{jj}$ for each $i \neq j$.

Our definition of positive definite requires the matrix to be symmetric, but not all authors make this requirement. For example, Golub and Van Loan [GV], a standard reference in matrix methods, requires only that $x^tAx > 0$ for each nonzero vector $x$. Matrices we call positive definite are called symmetric positive definite in [GV]. Keep this discrepancy in mind if you are using material from other sources.

The next result parallels the strictly diagonally dominant results presented previously.

[Positive Definite Matrix Equivalences] The following are equivalent for any $n \times n$ symmetric matrix $A$:

(i) $A$ is positive definite.
(ii) Gaussian elimination without row interchanges can be performed on the linear system $Ax = b$ with all pivot elements positive. (This ensures that the computations are stable with respect to the growth of round-off error.)
(iii) $A$ can be factored in the form $LL^t$, where $L$ is lower triangular with positive diagonal entries.
(iv) $A$ can be factored in the form $LDL^t$, where $L$ is lower triangular with 1s on its diagonal and $D$ is a diagonal matrix with positive diagonal entries.
(v) For each $i = 1, 2, \ldots, n$, we have
$$\det \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} \\ a_{21} & a_{22} & \cdots & a_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ii} \end{bmatrix} > 0.$$
Maple also has a useful command to determine the positive definiteness of a matrix. The command
\[
\text{definite(A,positive_def)};
\]
returns true or false as an indication. Consistent with our definition, symmetry is required for a true result to be produced.

The factorization in part (iv) can be obtained by Choleski’s factorization method. Set \( l_{11} = \sqrt{a_{11}} \) and generate the remainder of the first column of \( L \) using the equation
\[
l_{j1} = \frac{a_{j1}}{l_{11}} \quad \text{for each} \quad j = 2, 3, \ldots, n.
\]
For each \( i = 2, 3, \ldots, n - 1 \), determine the \( i \)th column of \( L \) by
\[
l_{ii} = \left( a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2}
\]
and, for each \( j = i + 1, i + 2, \ldots, n \), by
\[
l_{ji} = \frac{1}{l_{ii}} \left( a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik} l_{ik} \right).
\]
Finally,
\[
l_{nn} = \left( a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2 \right)^{1/2}.
\]
These equations can be derived by writing out the system associated with \( A = LL^t \). Choleski’s method gives the \( LL^t \) factorization and can be implemented using the program CHOLFC65.

The Choleski factorization of \( A \) is computed in Maple using the statement
\[
\text{L:=cholesky(A)};
\]
In a similar manner to the general \( LU \) factorization, the factorization \( A = LDL^t \) uses the equations \( d_1 = a_{11} \) and \( l_{j1} = a_{j1}/d_1 \) for each \( j = 2, 3, \ldots, n \), to generate the first column of \( L \). For each \( i = 2, 3, \ldots, n - 1 \), compute \( d_i \) and the \( i \)th column of \( L \) as follows:
\[
d_i = a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2 d_j
\]
and
\[
l_{ji} = \frac{1}{d_i} \left[ a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik} d_k \right].
\]
for each \( j = i + 1, i + 2, \ldots, n \). The last entry in \( D \) is

\[
d_n = a_{nn} - \sum_{j=1}^{n-1} l_{nj}^2 d_j.
\]

The \( LDL^t \) factorization can be accomplished with the program LDLFCT66.

**EXAMPLE 2**

The matrix

\[
A = \begin{bmatrix}
4 & -1 & 1 \\
-1 & 4.25 & 2.75 \\
1 & 2.75 & 3.5
\end{bmatrix}
\]

is positive definite. The factorization \( LDL^t \) of \( A \) is

\[
A = LDL^t = \begin{bmatrix}
1 & 0 & 0 \\
-0.25 & 1 & 0 \\
0.25 & 0.75 & 1
\end{bmatrix} \begin{bmatrix}
4 & 0 & 0 \\
0 & 4 & 0 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
1 & -0.25 & 0.25 \\
0 & 1 & 0.75 \\
0 & 0 & 1
\end{bmatrix},
\]

and Choleski’s method produces the factorization

\[
A = LL^t = \begin{bmatrix}
2 & 0 & 0 \\
-0.5 & 2 & 0 \\
0.5 & 1.5 & 1
\end{bmatrix} \begin{bmatrix}
2 & -0.5 & 0.5 \\
0 & 2 & 1.5 \\
0 & 0 & 1
\end{bmatrix}.
\]

We can solve the linear system \( Ax = b \) when \( A \) is positive definite by using Choleski’s method to factor \( A \) into the form \( LL^t \). Then \( LL^t x = y \), and to solve this system we first let \( y = L^t x \). The linear system \( Ly = b \) is solved using forward substitution, as

\[
y_1 = \frac{b_1}{l_{11}},
\]

and, for \( i = 2, 3, \ldots, n \),

\[
y_i = \frac{1}{l_{ii}} \left[ b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right].
\]

Then the solution to the original system is obtained by using backward substitution to solve \( L^t x = y \) with the equations

\[
x_n = \frac{y_n}{l_{nn}}
\]

and, for \( i = n - 1, n - 2, \ldots, 1 \),

\[
x_i = \frac{1}{l_{ii}} \left[ y_i - \sum_{j=i+1}^{n} l_{ji} x_j \right].
\]

If \( Ax = b \) is to be solved and the factorization \( A = LDL^t \) is known, then we let \( y = DL^t x \) and solve the system \( Ly = b \) using forward substitution

\[
y_1 = b_1
\]
and, for \( i = 2, 3, \ldots, n \),
\[
y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j.
\]

The system \( Dz = y \) is solved as
\[
z_i = \frac{y_i}{d_i}, \quad \text{for each } i = 1, 2, \ldots, n.
\]

Then the system \( L^t x = z \) is solved by backward substitution
\[
x_n = z_n
\]
and, for \( i = n - 1, n - 2, \ldots, 1 \),
\[
x_i = z_i - \sum_{j=i+1}^{n} l_{ji} x_j.
\]

Any symmetric matrix \( A \) for which Gaussian elimination can be applied without row interchanges can be factored into the form \( LDL^t \). In this general case, \( L \) is lower triangular with 1’s on its diagonal, and \( D \) is the diagonal matrix with the Gaussian elimination pivots on its diagonal. This result is widely applied, since symmetric matrices are common and easily recognized.

The last matrices considered are band matrices. In many applications the band matrices are also strictly diagonally dominant or positive definite. This combination of properties is very useful.

An \( n \times n \) matrix is called a band matrix if integers \( p \) and \( q \), with \( 1 < p, q < n \), exist with the property that \( a_{ij} = 0 \) whenever \( p \leq j - i \) or \( q \leq i - j \). The number \( p \) describes the number of diagonals above, and including, the main diagonal on which nonzero entries may lie. The number \( q \) describes the number of diagonals below, and including, the main diagonal on which nonzero entries may lie. The bandwidth of the band matrix is \( w = p + q - 1 \), which tells us how many of the diagonals can contain nonzero entries. The 1 is subtracted from the sum of \( p \) and \( q \) since both of these numbers count the main diagonal.

For example, the matrix
\[
A = \begin{bmatrix}
7 & 2 & 1 & 0 \\
3 & 5 & -3 & -2 \\
0 & 4 & 6 & -1 \\
0 & 0 & 5 & 8
\end{bmatrix}
\]
is a band matrix with \( p = 3 \) and \( q = 2 \), so it has bandwidth \( 3 + 2 - 1 = 4 \).

Band matrices concentrate all their nonzero entries about the diagonal. Two special cases of band matrices that occur often have \( p = q = 2 \) and \( p = q = 4 \). Matrices of bandwidth three that occur when \( p = q = 2 \) are called tridiagonal,
because they have the form

\[
A = \begin{bmatrix}
\alpha_1 & \gamma_1 & 0 & 0 \\
\beta_2 & \alpha_2 & \gamma_2 & 0 \\
0 & \beta_3 & \alpha_3 & 0 \\
0 & 0 & \beta_n & \gamma_{n-1} \\
\end{bmatrix}.
\]

Since the entries of tridiagonal matrices are predominantly zero, it is common to avoid the double subscript notation by relabeling the entries as indicated.

Tridiagonal matrices will appear in Chapter 11 in connection with the study of piecewise linear approximations to boundary-value problems. The case of \(p = q = 4\) will also be used in that chapter for the solution of boundary-value problems, when the approximating functions assume the form of cubic splines.

The factorization methods can be simplified considerably in the case of band matrices, because a large number of zeros appear in regular patterns. It is particularly interesting to observe the form the Crout (where \(u_{ii} = 1\)) and Doolittle (where \(l_{ii} = 1\)) methods assume in this case. To illustrate the situation, suppose a tridiagonal matrix

\[
A = \begin{bmatrix}
\alpha_1 & \gamma_1 & 0 & 0 \\
\beta_2 & \alpha_2 & \gamma_2 & 0 \\
0 & \beta_3 & \alpha_3 & 0 \\
0 & 0 & \beta_n & \gamma_{n-1} \\
\end{bmatrix},
\]

can be factored into the triangular matrices \(L\) and \(U\) in the Crout form

\[
L = \begin{bmatrix}
l_1 & 0 & 0 \\
\beta_2 & l_2 & 0 \\
0 & \beta_3 & l_3 \\
0 & 0 & \beta_n \\
\end{bmatrix}
\quad\text{and}\quad
U = \begin{bmatrix}
1 & u_1 & 0 & 0 \\
0 & 1 & u_2 & 0 \\
0 & 1 & 0 & u_{n-1} \\
0 & 0 & 1 & u_n \\
\end{bmatrix}.
\]

The entries are given by the following equations:

\[
l_1 = \alpha_1 \quad\text{and}\quad u_1 = \frac{\gamma_1}{l_1};
\]

for each \(i = 2, 3, \ldots, n - 1,\)

\[
l_i = \alpha_i - \beta_i u_{i-1} \quad\text{and}\quad u_i = \frac{\gamma_i}{l_i};
\]

and

\[
l_n = \alpha_n - \beta_n u_{n-1}.
\]

The linear system \(Ax = LUx = b\) is solved using the equations

\[
y_1 = \frac{b_1}{l_1}
\]
and, for each \( i = 2, 3, \ldots, n \),
\[
y_i = \frac{1}{l_i} \left[ b_i - \beta_i y_{i-1} \right],
\]
which determines \( y \) in the linear system \( L y = b \). The linear system \( U x = y \) is solved using the equations
\[
x_n = y_n
\]
and, for each \( i = n-1, n-2, \ldots, 1 \),
\[
x_i = y_i - u_i x_{i+1}.
\]
The Crout factorization of a tridiagonal matrix can be performed with the program CRTRLS67.

**EXAMPLE 3** To illustrate the procedure for tridiagonal matrices, consider the tridiagonal system of equations
\[
\begin{align*}
2x_1 - x_2 &= 1, \\
-x_1 + 2x_2 - x_3 &= 0, \\
-x_2 + 2x_3 - x_4 &= 0, \\
-x_3 + 2x_4 &= 1,
\end{align*}
\]
whose augmented matrix is
\[
\begin{bmatrix}
2 & -1 & 0 & 0 & 1 \\
-1 & 2 & -1 & 0 & 0 \\
0 & -1 & 2 & -1 & 0 \\
0 & 0 & -1 & 2 & 1
\end{bmatrix}.
\]
The \( LU \) factorization is given by
\[
\begin{bmatrix}
2 & -1 & 0 & 0 \\
-1 & 2 & -1 & 0 \\
0 & -1 & 2 & -1 \\
0 & 0 & -1 & 2
\end{bmatrix} = \begin{bmatrix}
2 & 0 & 0 & 0 \\
-1 & \frac{2}{3} & 0 & 0 \\
0 & -1 & \frac{4}{3} & 0 \\
0 & 0 & -1 & \frac{5}{3}
\end{bmatrix} \begin{bmatrix}
1 & -\frac{1}{2} & 0 & 0 \\
0 & 1 & -\frac{2}{3} & 0 \\
0 & 0 & 1 & -\frac{3}{4} \\
0 & 0 & 0 & 1
\end{bmatrix} = LU.
\]
Solving the system \( L y = b \) gives \( y = (\frac{1}{2}, \frac{1}{3}, \frac{1}{3}, 1)^t \), and the solution of \( U x = y \) is \( x = (1, 1, 1, 1)^t \).

The tridiagonal factorization can be applied whenever \( l_i \neq 0 \) for each \( i = 1, 2, \ldots, n \). Two conditions, either of which ensure that this is true, are that the coefficient matrix of the system is positive definite or that it is strictly diagonally dominant. An additional condition that ensures this method can be applied is as follows.
Suppose that $A$ is tridiagonal with $\beta_i \neq 0$ and $\gamma_i \neq 0$ for each $i = 2, 3, \ldots, n - 1$. If $|\alpha_1| > |\gamma_1|$, $|\alpha_n| > |\beta_n|$, and $|\alpha_i| \geq |\beta_i| + |\gamma_i|$ for each $i = 2, 3, \ldots, n - 1$, then $A$ is nonsingular, and the values of $l_i$ are nonzero for each $i = 1, 2, \ldots, n$.
EXERCISE SET 6.6

1. Determine which of the following matrices are (i) symmetric, (ii) singular, (iii) strictly diagonally dominant, (iv) positive definite.

(a) \[
\begin{bmatrix}
2 & 1 \\
1 & 3
\end{bmatrix}
\]

(b) \[
\begin{bmatrix}
-2 & 1 \\
1 & -3
\end{bmatrix}
\]

(c) \[
\begin{bmatrix}
2 & 1 & 0 \\
0 & 3 & 0 \\
1 & 0 & 4
\end{bmatrix}
\]

(d) \[
\begin{bmatrix}
2 & 1 & 0 \\
0 & 3 & 2 \\
1 & 2 & 4
\end{bmatrix}
\]

(e) \[
\begin{bmatrix}
4 & 2 & 6 \\
3 & 0 & 7 \\
-2 & -1 & -3
\end{bmatrix}
\]

(f) \[
\begin{bmatrix}
2 & -1 & 0 \\
-1 & 4 & 2 \\
0 & 2 & 2
\end{bmatrix}
\]

(g) \[
\begin{bmatrix}
4 & 0 & 0 & 0 \\
6 & 7 & 0 & 0 \\
9 & 11 & 1 & 0 \\
5 & 4 & 1 & 1
\end{bmatrix}
\]

(h) \[
\begin{bmatrix}
2 & 3 & 1 & 2 \\
-2 & 4 & -1 & 5 \\
3 & 7 & 1.5 & 1 \\
6 & -9 & 3 & 7
\end{bmatrix}
\]

2. Find a factorization of the form \(A = LDL^t\) for the following symmetric matrices:

(a) \[
A = \begin{bmatrix}
2 & -1 & 0 \\
-1 & 2 & -1 \\
0 & -1 & 2
\end{bmatrix}
\]

(b) \[
A = \begin{bmatrix}
4 & 1 & 1 & 1 \\
1 & 3 & -1 & 1 \\
1 & -1 & 2 & 0 \\
1 & 1 & 0 & 2
\end{bmatrix}
\]

(c) \[
A = \begin{bmatrix}
4 & 1 & -1 & 0 \\
1 & 3 & -1 & 0 \\
-1 & -1 & 5 & 2 \\
0 & 0 & 2 & 4
\end{bmatrix}
\]

(d) \[
A = \begin{bmatrix}
6 & 2 & 1 & -1 \\
2 & 4 & 1 & 0 \\
1 & 1 & 4 & -1 \\
-1 & 0 & -1 & 3
\end{bmatrix}
\]

3. Find a factorization of the form \(A = LL^t\) for the matrices in Exercise 2.

4. Use the factorization in Exercise 2 to solve the following linear systems.

(a) \[
\begin{align*}
2x_1 - x_2 &= 3, \\
-x_1 + 2x_2 - x_3 &= -3, \\
-x_2 + 2x_3 &= 1.
\end{align*}
\]

(b) \[
\begin{align*}
4x_1 + x_2 + x_3 + x_4 &= 0.65, \\
x_1 + 3x_2 - x_3 + x_4 &= 0.05, \\
x_1 - x_2 + 2x_3 &= 0, \\
x_1 + x_2 + 2x_4 &= 0.5.
\end{align*}
\]
5. Use Crout factorization for tridiagonal systems to solve the following linear systems.

(a) \[ x_1 - x_2 = 0, \quad -2x_1 + 4x_2 - 2x_3 = -1, \quad x_2 + 2x_3 = 1.5. \]

(b) \[ 3x_1 + x_2 = -1, \quad 2x_1 + 4x_2 + x_3 = 7, \quad 2x_2 + 5x_3 = 9. \]

(c) \[ 2x_1 - x_2 = 3, \quad x_1 + 2x_2 - x_3 = -3, \quad x_2 + 2x_3 = 1. \]

(d) \[ 0.5x_1 + 0.25x_2 = 0.35, \quad 0.35x_1 + 0.8x_2 + 0.4x_3 = 0.77, \quad 0.25x_2 + x_3 + 0.5x_4 = -0.5, \quad x_3 - 2x_4 = -2.25. \]

6. Let \( A \) be the \( 10 \times 10 \) tridiagonal matrix given by \( a_{ii} = 2, a_{i,i+1} = a_{i,i-1} = -1, \) for each \( i = 2, \ldots, 9, \) and \( a_{11} = a_{10,10} = 2, a_{12} = a_{10,9} = -1. \) Let \( b \) be the 10-dimensional column vector given by \( b_1 = b_{10} = 1 \) and \( b_i = 0 \) for each \( i = 2, 3, \ldots, 9. \) Solve \( Ax = b \) using the Crout factorization for tridiagonal systems.

7. Suppose that \( A \) and \( B \) are positive definite \( n \times n \) matrices.

(a) Is \(-A\) positive definite?

(b) Is \( A^t\) positive definite?

(c) Is \( A + B\) positive definite?

(d) Is \( A^2\) positive definite?

(e) Is \( A - B\) positive definite?

8. Let

\[
A = \begin{bmatrix}
1 & 0 & -1 \\
0 & 1 & 1 \\
-1 & 1 & \alpha
\end{bmatrix}.
\]

Find all values of \( \alpha \) for which
6.6. TECHNIQUES FOR SPECIAL MATRICES

(a) $A$ is singular.
(b) $A$ is strictly diagonally dominant.
(c) $A$ is symmetric.
(d) $A$ is positive definite.

9. Let

\[
A = \begin{bmatrix}
\alpha & 1 & 0 \\
\beta & 2 & 1 \\
0 & 1 & 2
\end{bmatrix}
\]

Find all values of $\alpha$ and $\beta$ for which

(a) $A$ is singular.
(b) $A$ is strictly diagonally dominant.
(c) $A$ is symmetric.
(d) $A$ is positive definite.

10. Suppose $A$ and $B$ commute; that is, $AB = BA$. Must $A^t$ and $B^t$ also commute?

11. In a paper by Dorn and Burdick [DB], it is reported that the average wing length that resulted from mating three mutant varieties of fruit flies ($Drosophila melanogaster$) can be expressed in the symmetric matrix form

\[
A = \begin{bmatrix}
1.59 & 1.69 & 2.13 \\
1.69 & 1.31 & 1.72 \\
2.13 & 1.72 & 1.85
\end{bmatrix},
\]

where $a_{ij}$ denotes the average wing length of an offspring resulting from the mating of a male of type $i$ with a female of type $j$.

(a) What physical significance is associated with the symmetry of this matrix?

(b) Is this matrix positive definite? If so, prove it; if not, find a nonzero vector $x$ for which $x^tAx \leq 0$. 

6.7 Survey of Methods and Software

In this chapter we have looked at direct methods for solving linear systems. A linear system consists of \( n \) equations in \( n \) unknowns expressed in matrix notation as \( Ax = b \). These techniques use a finite sequence of arithmetic operations to determine the exact solution of the system subject only to roundoff error. We found that the linear system \( Ax = b \) has a unique solution if and only if \( A^{-1} \) exists, which is equivalent to \( \det A \neq 0 \). The solution of the linear system is the vector \( x = A^{-1}b \).

Pivoting techniques were introduced to minimize the effects of roundoff error, which can dominate the solution when using direct methods. We studied partial pivoting, scaled partial pivoting, and total pivoting. We recommend the partial or scaled partial pivoting methods for most problems since these decrease the effects of roundoff error without adding much extra computation. Total pivoting should be used if roundoff error is suspected to be large. In Section 7.6 we will see some procedures for estimating this roundoff error.

Gaussian elimination with minor modifications was shown to yield a factorization of the matrix \( A \) into \( LU \), where \( L \) is lower triangular with 1s on the diagonal and \( U \) is upper triangular. This process is called Doolittle’s factorization. Not all nonsingular matrices can be factored this way, but a permutation of the rows will always give a factorization of the form \( PA = LU \), where \( P \) is the permutation matrix used to rearrange the rows of \( A \). The advantage of the factorization is that the work is reduced when solving linear systems \( Ax = b \) with the same coefficient matrix \( A \) and different vectors \( b \).

Factorizations take a simpler form when the matrix \( A \) is positive definite. For example, the Choleski factorization has the form \( A = LL^t \), where \( L \) is lower triangular. A symmetric matrix that has an LU factorization can also be factored in the form \( A = LDL^t \), where \( D \) is diagonal and \( L \) is lower triangular with 1s on the diagonal. With these factorizations, manipulations involving \( A \) can be simplified. If \( A \) is tridiagonal, the \( LU \) factorization takes a particularly simple form, with \( L \) having 1s on the main diagonal and 0s elsewhere, except on the diagonal immediately below the main diagonal. In addition, \( U \) has its only nonzero entries on the main diagonal and one diagonal above.

The direct methods are the methods of choice for most linear systems. For tridiagonal, banded, and positive definite matrices, the special methods are recommended. For the general case, Gaussian elimination or \( LU \) factorization methods, which allow pivoting, are recommended. In these cases, the effects of roundoff error should be monitored. In Section 7.6 we discuss estimating errors in direct methods.

Large linear systems with primarily 0 entries occurring in regular patterns can be solved efficiently using an iterative procedure such as those discussed in Chapter 7. Systems of this type arise naturally, for example, when finite-difference techniques are used to solve boundary-value problems, a common application in the numerical solution of partial-differential equations.

It can be very difficult to solve a large linear system that has primarily nonzero entries or one where the 0 entries are not in a predictable pattern. The matrix associated with the system can be placed in secondary storage in partitioned form and portions read into main memory only as needed for calculation. Methods that
require secondary storage can be either iterative or direct, but they generally require techniques from the fields of data structures and graph theory. The reader is referred to [BuR] and [RW] for a discussion of the current techniques.

The software for matrix operations and the direct solution of linear systems implemented in IMSL and NAG is based on LAPACK, a subroutine package in the public domain. There is excellent documentation available with it and from the books written about it.

Accompanying LAPACK is a set of lower-level operations called Basic Linear Algebra Subprograms (BLAS). Level 1 of BLAS generally consists of vector-vector operations with input data and operation counts of $O(n)$. Level 2 consists of the matrix-vector operations with input data and operation counts of $O(n^2)$. Level 3 consists of the matrix-matrix operations with input data and operation counts of $O(n^3)$.

The subroutines in LAPACK for solving linear systems first factor the matrix $A$. The factorization depends on the type of matrix in the following way:

- General matrix $PA = LU$;
- Positive definite matrix $A = LL^t$;
- Symmetric matrix $A = LDL^t$;
- Tridiagonal matrix $A = LU$ (in banded form).

Linear systems are then solved based on factorization. It is also possible to compute determinants and inverses and to estimate the round-off error involved.

Many of the subroutines in LINPACK, and its successor LAPACK, can be implemented using MATLAB. A nonsingular matrix $A$ is factored using the command
\[
[L, U, P] = lu(A)
\]
into the form $PA = LU$, where $P$ is the permutation matrix defined by performing partial pivoting to solve a linear system involving $A$. If the nonsingular matrix $A$ and the vector $b$ have been defined in MATLAB, the command
\[
x = A \backslash b
\]
solves the linear system by first using the $PA = LU$ factoring command. Then it solves the lower-triangular system $Lz = b$ for $z$ using its command,
\[
z = L \backslash b
\]
This is followed by a solution to the upper-triangular system $Ux = z$ using the command
\[
x = U \backslash z
\]
Other MATLAB commands include computing the inverse, transpose, and determinant of matrix $A$ by issuing the commands $inv(A)$, $A'$, and $det(A)$, respectively.

Further information on the numerical solution of linear systems and matrices can be found in Golub and Van Loan [GV], Forsythe and Moler [FM], and Stewart [St]. The use of direct techniques for solving large sparse systems is discussed in detail in George and Liu [GL] and in Pissanetzky [Pi]. Coleman and Van Loan [CV] consider the use of BLAS, LINPACK, and MATLAB.
Chapter 7

Iterative Methods for Solving Linear Systems

7.1 Introduction

The previous chapter considered the approximation of the solution of a linear system using direct methods, techniques that would produce the exact solution if all the calculations were performed using exact arithmetic. In this chapter we describe some popular iterative techniques which require an initial approximation to the solution. These methods will not be expected to return the exact solution even if all the calculations could be performed using exact arithmetic. In many instances, however, they are more effective than the direct methods, since they can require far less computational effort and round-off error is reduced. This is particularly true when the matrix is sparse—that is, when it has a high percentage of zero entries.

Some additional material from linear algebra is needed to describe the convergence of the iterative methods. Principally, we need to have a measure of how close two vectors are to one another, since the object of an iterative method is to determine an approximation that is within a certain tolerance of the exact solution. In Section 7.2, the notion of a norm is used to show how various forms of distance between vectors can be described. We will also see how this concept can be extended to describe the norm of—and, consequently, the distance between—matrices. In Section 7.3, matrix eigenvalues and eigenvectors are described, and we consider the connection between these concepts and the convergence of an iterative method.

Section 7.4 describes the elementary Jacobi and Gauss-Seidel iterative methods. By analyzing the size of the largest eigenvalue of a matrix associated with an iterative method, we can determine conditions that predict the likelihood of convergence of the method. In Section 7.5 we introduce the SOR method. This is a commonly applied iterative technique, since it reduces the approximation errors faster than the Jacobi and Gauss-Seidel methods.

The conjugate gradient method is presented in Section 7.6. This method, with
preconditioning, is the technique most often used for sparse, positive-definite matrices.

The final section in the chapter discusses some of the concerns that should be addressed when applying either an iterative or direct technique for approximating the solution to a linear system.

7.2 Convergence of Vectors

The distance between the real numbers $x$ and $y$ is $|x - y|$. In Chapter 2 we saw that the stopping techniques for the iterative root-finding techniques used this measure to estimate the accuracy of approximate solutions and to determine when an approximation was sufficiently accurate. The iterative methods for solving systems of equations use similar logic, so the first step is to determine a way to measure the distance between $n$-dimensional vectors, since this is the form that is taken by the solution to a system of equations.

Let $\mathbb{R}^n$ denote the set of all $n$-dimensional column vectors with real number coefficients. It is a space-saving convenience to use the transpose notation presented in Section 6.4 when such a vector is represented in terms of its components. For example, the vector

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

is generally written $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$.

[Vector Norm on $\mathbb{R}^n$] A vector norm on $\mathbb{R}^n$ is a function, $\| \cdot \|$, from $\mathbb{R}^n$ into $\mathbb{R}$ with the following properties:

1. $\| \mathbf{x} \| \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$,
2. $\| \mathbf{x} \| = 0$ if and only if $\mathbf{x} = (0, 0, \ldots, 0)^t \equiv \mathbf{0}$,
3. $\| \alpha \mathbf{x} \| = |\alpha| \| \mathbf{x} \|$ for all $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$,
4. $\| \mathbf{x} + \mathbf{y} \| \leq \| \mathbf{x} \| + \| \mathbf{y} \|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

For our purposes, we need only two specific norms on $\mathbb{R}^n$. (A third is presented in Exercise 2.)

The $l_2$ and $l_\infty$ norms for the vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ are defined by

$$
\| \mathbf{x} \|_2 = \left( \sum_{i=1}^{n} x_i^2 \right)^{1/2} \quad \text{and} \quad \| \mathbf{x} \|_\infty = \max_{1 \leq i \leq n} |x_i|.
$$
The $l_2$ norm is called the **Euclidean norm** of the vector $x$, since it represents the usual notion of distance from the origin in case $x$ is in $\mathbb{R}^1 \equiv \mathbb{R}$, $\mathbb{R}^2$, or $\mathbb{R}^3$. For example, the $l_2$ norm of the vector $x = (x_1, x_2, x_3)^t$ gives the length of the straight line joining the points $(0, 0, 0)$ and $(x_1, x_2, x_3)$; that is, the length of the shortest path between those two points. Figure 7.1 shows the boundary of those vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$ that have $l_2$ norm less than 1. Figure 7.2 gives a similar illustration for the $l_\infty$ norm.
The vectors in $\mathbb{R}^2$ with $l$ norm less than 1 are inside this figure.

The vectors in the first octant of $\mathbb{R}^3$ with $l$ norm less than 1 are inside this figure.

EXAMPLE 1  The vector $\mathbf{x} = (-1, 1, -2)^t$ in $\mathbb{R}^3$ has norms

$$\|\mathbf{x}\|_2 = \sqrt{(-1)^2 + (1)^2 + (-2)^2} = \sqrt{6} \quad \text{and} \quad \|\mathbf{x}\|_\infty = \max\{|-1|, |1|, |-2|\} = 2.$$

Showing that $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$ satisfies the conditions necessary for a norm on $\mathbb{R}^n$ follows directly from the truth of similar statements concerning absolute values of real numbers. In the case of the $l_2$ norm, it is also easy to demonstrate the first three of the required properties, but the fourth,

$$\|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2,$$

is more difficult to show. To demonstrate this inequality we need the Cauchy-Buniakowsky-Schwarz inequality, which states that for any $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$,

$$\sum_{i=1}^{n} |x_i y_i| \leq \left( \sum_{i=1}^{n} x_i^2 \right)^{1/2} \left( \sum_{i=1}^{n} y_i^2 \right)^{1/2}.$$

With this it follows that $\|\mathbf{x} + \mathbf{y}\|_2 \leq \|\mathbf{x}\|_2 + \|\mathbf{y}\|_2$ since

$$\|\mathbf{x} + \mathbf{y}\|_2^2 = \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \leq \sum_{i=1}^{n} x_i^2 + 2 \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \leq \sum_{i=1}^{n} x_i^2 + 2 \left( \sum_{i=1}^{n} x_i^2 \right)^{1/2} \left( \sum_{i=1}^{n} y_i^2 \right)^{1/2} + \sum_{i=1}^{n} y_i^2 = (\|\mathbf{x}\|_2 + \|\mathbf{y}\|_2)^2.$$
The norm of a vector gives a measure for the distance between the vector and
the origin, so the distance between two vectors is the norm of the difference of the
vectors.

\[
\|x - y\|_2 = \left( \sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2} \quad \text{and} \quad \|x - y\|_\infty = \max_{1 \leq i \leq n} |x_i - y_i|.
\]

**EXAMPLE 2**  The linear system

\[
\begin{align*}
3.3330x_1 + 15920x_2 - 10.333x_3 &= 15913, \\
2.2220x_1 + 16.710x_2 + 9.6120x_3 &= 28.544, \\
1.5611x_1 + 5.1791x_2 + 1.6852x_3 &= 8.4254
\end{align*}
\]

has solution \((x_1, x_2, x_3)^t = (1.0000, 1.0000, 1.0000)^t\). If Gaussian elimination is per-
formed in five-digit rounding arithmetic using partial pivoting, the solution obtained
is \(\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)^t = (1.2001, 0.99991, 0.92538)^t\).

Measurements of the accuracy of the approximation \(\tilde{x}\) to \(x\) are given by

\[
\|x - \tilde{x}\|_\infty = \max\{|1.0000 - 1.2001|, |1.0000 - 0.99991|, |1.0000 - 0.92538|\} = 0.2001
\]

and

\[
\|x - \tilde{x}\|_2 = \left[ (1.0000 - 1.2001)^2 + (0.0000 - 0.99991)^2 + (1.0000 - 0.92538)^2 \right]^{1/2} = 0.21356.
\]

Although the components \(\tilde{x}_2\) and \(\tilde{x}_3\) are good approximations to \(x_2\) and \(x_3\), the
component \(\tilde{x}_1\) is a poor approximation to \(x_1\), and \(|x_1 - \tilde{x}_1|\) dominates both norms.

The distance concept in \(\mathbb{R}^n\) is used to define a limit of a sequence of vectors. A
sequence \(\{x^{(k)}\}_{k=1}^\infty\) of vectors in \(\mathbb{R}^n\) is said to **converge** to \(x\) with respect to the
norm \(\| \cdot \|\) if, given any \(\varepsilon > 0\), there exists an integer \(N(\varepsilon)\) such that

\[
\|x^{(k)} - x\| < \varepsilon \quad \text{for all } k \geq N(\varepsilon).
\]
EXAMPLE 3 Let $x^{(k)} \in \mathbb{R}^4$ be defined by

$$x^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)})^t = \left(1, 2 + \frac{1}{k}, \frac{3}{k^2}, e^{-k} \sin k\right)^t.$$ 

We have

$$\lim_{k \to \infty} 1 = 1, \quad \lim_{k \to \infty} (2 + 1/k) = 2, \quad \lim_{k \to \infty} 3/k^2 = 0, \quad \text{and} \quad \lim_{k \to \infty} e^{-k} \sin k = 0.$$ 

So for any given $\varepsilon$ an integer $N(\varepsilon)$ can be found so that the largest of $|x_1^{(k)} - 1|$, $|x_2^{(k)} - 2|$, $|x_3^{(k)} - 0|$, and $|x_4^{(k)} - 0|$ is less than $\varepsilon$. This implies that the sequence $\{x^{(k)}\}$ converges to $(1, 2, 0, 0)^t$ with respect to $\|\cdot\|$. \qed

In Example 3 we implicitly used the fact that a sequence of vectors $\{x^{(k)}\}_{k=1}^\infty$ converges in the norm $\|\cdot\|_\infty$ to the vector $x$ if and only if for each $i = 1, 2, \ldots, n$ the sequence $\{x_i^{(k)}\}_{k=1}^\infty$ converges to $x_i$, the $i$th component of $x$. This makes the determination of convergence for the norm $\|\cdot\|_\infty$ relatively easy.

To show directly that the sequence in Example 3 converges to $(1, 2, 0, 0)^t$ with respect to the $l_2$ norm is quite complicated. However, suppose that $j$ is an index with the property that

$$\|x\|_\infty = \max_{i=1,\ldots,n} |x_i| = |x_j|.$$ 

Then

$$\|x\|_\infty^2 = |x_j|^2 = x_j^2 \leq \sum_{i=1}^n x_i^2 = \|x\|_2^2$$

and also

$$\|x\|_2^2 = \sum_{i=1}^n x_i^2 \leq \sum_{i=1}^n x_j^2 = n x_j^2 = n \|x\|_\infty^2.$$ 

These inequalities imply that the sequence of vectors $\{x^{(k)}\}$ also converges to $x$ in $\mathbb{R}^n$ with respect to $\|\cdot\|_2$ if and only if $\lim_{k \to \infty} x_i^{(k)} = x_i$ for each $i = 1, 2, \ldots, n$, since this is when the sequence converges in the $l_\infty$ norm.

In fact, it can be shown that all norms on $\mathbb{R}^n$ are equivalent with respect to convergence; that is, if $\|\cdot\|$ and $\|\cdot\|'$ are any two norms on $\mathbb{R}^n$ and $\{x^{(k)}\}_{k=1}^\infty$ has the limit $x$ with respect to $\|\cdot\|$, then $\{x^{(k)}\}_{k=1}^\infty$ has the limit $x$ with respect to $\|\cdot\|'$. Since a vector sequence converges in the $l_\infty$ norm precisely when each of its component sequences converges, we have the following.

**[Vector Sequence Convergence]** The following statements are equivalent:

(i) The sequence of vectors $\{x^{(k)}\}$ converges to $x$ in some norm.

(ii) The sequence of vectors $\{x^{(k)}\}$ converges to $x$ in every norm.

(iii) $\lim_{k \to \infty} x_i^{(k)} = x_i$, the $i$th component of $x$, for each $i = 1, 2, \ldots, n$. 

In the subsequent sections, we will need methods for determining the distance between \( n \times n \) matrices. This again requires the use of a norm.

**[Matrix Norm]** A matrix norm on the set of all \( n \times n \) matrices is a real-valued function, \( \| \cdot \| \), defined on this set, satisfying for all \( n \times n \) matrices \( A \) and \( B \) and all real numbers \( \alpha \):

1. \( \| A \| \geq 0 \),
2. \( \| A \| = 0 \), if and only if \( A \) is \( O \), the matrix with all zero entries,
3. \( \| \alpha A \| = |\alpha| \| A \| \),
4. \( \| A + B \| \leq \| A \| + \| B \| \),
5. \( \| AB \| \leq \| A \| \| B \| \).

A **distance between \( n \times n \) matrices** \( A \) and \( B \) with respect to this matrix norm is \( \| A - B \| \). Although matrix norms can be obtained in various ways, the only norms we consider are those that are natural consequences of a vector norm.

**[Natural Matrix Norm]** If \( \| \cdot \| \) is a vector norm on \( \mathbb{R}^n \), the natural matrix norm on the set of \( n \times n \) matrices given by \( \| \cdot \| \) is defined by

\[
\| A \| = \max_{\| x \| = 1} \| A x \| .
\]

As a consequence, the matrix norms we will consider have the forms

\[
\| A \|_2 = \max_{\| x \|_2 = 1} \| A x \|_2, \quad (\text{the } l_2 \text{ norm})
\]

and

\[
\| A \|_\infty = \max_{\| x \|_\infty = 1} \| A x \|_\infty, \quad (\text{the } l_\infty \text{ norm}).
\]

When \( n = 2 \) these norms have the geometric representations shown in Figures 7.3 and 7.4.

**Figure 7.3**
The $l_\infty$ norm of a matrix has a representation with respect to the entries of the matrix that makes it particularly easy to compute. The $l_2$ norm of a matrix is not as easily determined, but in the next section we will discover an alternative method for finding this norm.

\[
\| A \|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|.
\]
EXAMPLE 4  If 

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & -1 \\ 5 & -1 & 1 \end{bmatrix},$$

then 

$$\sum_{j=1}^{3} |a_{1j}| = |1| + |2| + |-1| = 4, \quad \sum_{j=1}^{3} |a_{2j}| = |0| + |3| + |-1| = 4,$$

and 

$$\sum_{j=1}^{3} |a_{3j}| = |5| + |-1| + |1| = 7.$$

So 

$$\|A\|_{\infty} = \max\{4, 4, 7\} = 7. \quad \Box$$
EXERCISE SET 7.2

1. Find $\|x\|_\infty$ and $\|x\|_2$ for the following vectors.

   (a) $x = (3, -4, 0, \frac{3}{2})^t$
   (b) $x = (2, 1, -3, 4)^t$
   (c) $x = (\sin k, \cos k, 2^k)^t$ for a fixed positive integer $k$
   (d) $x = (4/(k + 1), 2/k^2, k^2 e^{-k})^t$ for a fixed positive integer $k$

2. (a) Verify that $\|\cdot\|_1$ is a norm for $\mathbb{R}^n$ (called the $l_1$ norm), where

\[ \|x\|_1 = \sum_{i=1}^{n} |x_i| \]

   (b) Find $\|x\|_1$ for the vectors given in Exercise 1.

3. Show that the following sequences are convergent, and find their limits.

   (a) $x^{(k)} = (1/k, e^{1-k}, -2/k^2)^t$
   (b) $x^{(k)} = (e^{-k} \cos k, k \sin(1/k), 3 + k^{-2})^t$
   (c) $x^{(k)} = (ke^{-k^2}, \cos k/k, \sqrt{k^2 + k} - k)^t$
   (d) $x^{(k)} = (e^{1/k}, (k^2 + 1)/(1 - k^2), (1/k^2)(1 + 3 + 5 + \cdots + (2k - 1)))^t$

4. Find $\|\cdot\|_\infty$ for the following matrices.

   (a) $\begin{bmatrix} 10 & 15 \\ 0 & 1 \end{bmatrix}$
   (b) $\begin{bmatrix} 10 & 0 \\ 15 & 1 \end{bmatrix}$
   (c) $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$
   (d) $\begin{bmatrix} 4 & -1 & 7 \\ -1 & 4 & 0 \\ -7 & 0 & 4 \end{bmatrix}$

5. The following linear systems $Ax = b$ have $x$ as the actual solution and $\tilde{x}$ as an approximate solution. Compute $\|x - \tilde{x}\|_\infty$ and $\|A\tilde{x} - b\|_\infty$.

   (a) $\begin{array}{c} \frac{1}{2}x_1 + \frac{1}{3}x_2 = \frac{1}{6}, \\ \frac{1}{5}x_1 + \frac{1}{4}x_2 = \frac{1}{16} \end{array}$
   $\begin{align*}
   x &= \left(\frac{1}{7}, -\frac{1}{6}\right)^t, \\
   \tilde{x} &= (0.142, -0.166)^t.
   \end{align*}$
7.2. **CONVERGENCE OF VECTORS**

(b) \[ x_1 + 2x_2 + 3x_3 = 1, \]
\[ 2x_1 + 3x_2 + 4x_3 = -1, \]
\[ 3x_1 + 4x_2 + 6x_3 = 2, \]
\[ \mathbf{x} = (0, -7, 5)^t, \]
\[ \tilde{\mathbf{x}} = (-0.33, -7.9, 5.8)^t. \]

(c) \[ x_1 + 2x_2 + 3x_3 = 1, \]
\[ 2x_1 + 3x_2 + 4x_3 = -1, \]
\[ 3x_1 + 4x_2 + 6x_3 = 2, \]
\[ \mathbf{x} = (0, -7, 5)^t, \]
\[ \tilde{\mathbf{x}} = (-0.2, -7.5, 5.4)^t. \]

(d) \[ 0.04x_1 + 0.01x_2 - 0.01x_3 = 0.06, \]
\[ 0.2x_1 + 0.5x_2 - 0.2x_3 = 0.3, \]
\[ x_1 + 2x_2 + 4x_3 = 11, \]
\[ \mathbf{x} = (1.827586, 0.6551724, 1.965517)^t, \]
\[ \tilde{\mathbf{x}} = (1.8, 0.64, 1.9)^t. \]

6. The **$l_1$ matrix norm**, defined by \[ \| A \|_1 = \max_{\| \mathbf{x} \|_1 = 1} \| A\mathbf{x} \|_1, \] can be computed using the formula
\[ \| A \|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|, \]
where the **$l_1$ vector norm** is defined in Exercise 2. Find the $l_1$ norm of the matrices in Exercise 4.

7. Show by example that \[ \| \cdot \|_{\infty}, \] defined by \[ \| A \|_{\infty} = \max_{1 \leq i,j \leq n} |a_{ij}|, \] does not define a matrix norm.

8. Show that \[ \| \cdot \|_{\Omega}, \] defined by
\[ \| A \|_{\Omega} = \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|, \]
is a matrix norm. Find \[ \| \cdot \|_{\Omega} \] for the matrices in Exercise 4.

9. Show that if \[ \| \cdot \| \] is a vector norm on $\mathbb{R}^n$, then \[ \| A \| = \max_{\| \mathbf{x} \|_1 = 1} \| A\mathbf{x} \| \] is a matrix norm.
7.3 Eigenvalues and Eigenvectors

An \( n \times m \) matrix can be considered as a function that uses matrix multiplication to take \( m \)-dimensional vectors into \( n \)-dimensional vectors. So an \( n \times n \) matrix \( A \) takes the set of \( n \)-dimensional vectors into itself. In this case certain nonzero vectors have \( x \) and \( Ax \) parallel, which means that a constant \( \lambda \) exists with \( Ax = \lambda x \), or that \( (A - \lambda I)x = 0 \). There is a close connection between these numbers \( \lambda \) and the likelihood that an iterative method will converge. We will consider this connection in this section.

For a square \( n \times n \) matrix \( A \), the characteristic polynomial of \( A \) is defined by

\[
p(\lambda) = \det(A - \lambda I).
\]

Because of the way the determinant of a matrix is defined, \( p \) is an \( n \)th-degree polynomial and, consequently, has at most \( n \) distinct zeros, some of which may be complex. These zeros of \( p \) are called the eigenvalues of the matrix \( A \).

If \( \lambda \) is an eigenvalue, then \( \det(A - \lambda I) = 0 \), and the equivalence result at the end of Section 6.4 implies that \( A - \lambda I \) is a singular matrix. As a consequence, the linear system defined by \( (A - \lambda I)x = 0 \) has a solution other than the zero vector. If \( (A - \lambda I)x = 0 \) and \( x \neq 0 \), then \( x \) is called an eigenvector of \( A \) corresponding to the eigenvalue \( \lambda \).

If \( x \) is an eigenvector associated with the eigenvalue \( \lambda \), then \( Ax = \lambda x \), so the matrix \( A \) takes the vector \( x \) into a scalar multiple of itself. When \( \lambda \) is a real number and \( \lambda > 1 \), \( A \) has the effect of stretching \( x \) by a factor of \( \lambda \). When \( 0 < \lambda < 1 \), \( A \) shrinks \( x \) by a factor of \( \lambda \). When \( \lambda < 0 \), the effects are similar, but the direction is reversed (see Figure 7.5).

**Figure 7.5**

(a) \( \lambda > 1 \) \hspace{1cm} (b) \( 1 > \lambda > 0 \) \hspace{1cm} (c) \( \lambda < -1 \) \hspace{1cm} (d) \( -1 < \lambda < 0 \)

![Diagram of eigenvalues and eigenvectors]

**EXAMPLE 1**

Let

\[
A = \begin{bmatrix}
0 & -1 \\
2 & 3
\end{bmatrix}.
\]

To compute the eigenvalues of \( A \), consider

\[
p(\lambda) = \det(A - \lambda I) = \det\left[\begin{array}{cc}
0 - \lambda & -1 \\
2 & 3 - \lambda
\end{array}\right] = -\lambda(3 - \lambda) + 2 = (\lambda - 1)(\lambda - 2).
\]
The eigenvalues of $A$ are $\lambda_1 = 1$ and $\lambda_2 = 2$.

An eigenvector $x \neq 0$ of $A$ associated with $\lambda_1 = 1$ is a solution of the system $(A - 1 \cdot I)x = 0$, so

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ and } x_2 = -x_1.$$

Any nonzero value of $x_1$ produces an eigenvector for the eigenvalue $\lambda_1 = 1$. For example, when $x_1 = 1$ we have the eigenvector $(1, -1)^t$.

Similarly, an eigenvector $x \neq 0$ of $A$ associated with $\lambda_2 = 2$ is a solution of the system $(A - 2 \cdot I)x = 0$, so

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \text{ and } x_2 = -2x_1.$$

When $x_1 = 1$ we have the eigenvector $(1, -2)^t$ for the eigenvalue $\lambda_2 = 2$.

**EXAMPLE 2** Let

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

To compute the eigenvalues of $A$, consider

$$p(\lambda) = \det(A - \lambda I) = \det \begin{bmatrix} 1 - \lambda & 0 & 2 \\ 0 & 1 - \lambda & -1 \\ -1 & 1 & 1 - \lambda \end{bmatrix} = (1 - \lambda)(\lambda^2 - 2\lambda + 4).$$

The eigenvalues of $A$ are the solutions of $p(\lambda) = 0$: $\lambda_1 = 1, \lambda_2 = 1 + \sqrt{3}i, \text{ and } \lambda_3 = 1 - \sqrt{3}i$. Notice that there are two complex conjugate eigenvalues for this matrix.

An eigenvector $x$ of $A$ associated with $\lambda_1 = 1$ is a solution of the system $(A - \lambda_1 I)x = 0$:

$$\begin{bmatrix} 0 & 0 & 2 \\ 0 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Thus

$$2x_3 = 0, \quad -x_3 = 0, \quad \text{and} \quad -x_1 + x_2 = 0,$$

which implies that

$$x_3 = 0, \quad x_2 = x_1, \quad \text{and} \quad x_1 \text{ is arbitrary.}$$

The choice $x_1 = 1$ produces the eigenvector $(1, 1, 0)^t$ with $\|(1, 1, 0)_\infty = 1$, corresponding to the eigenvalue $\lambda_1 = 1$. The choice $x_1 = \frac{\sqrt{2}}{2}$ produces an eigenvector corresponding to $\lambda_1$ with

$$\left\| \left( \frac{1}{2} \sqrt{2}, \frac{1}{2} \sqrt{2}, 0 \right) \right\|_2 = 1.$$
Since $\lambda_2$ and $\lambda_3$ are complex numbers, their corresponding eigenvectors are also complex. To find an eigenvector for $\lambda_2$, we solve the system

$$
\begin{bmatrix}
1 - (1 + \sqrt{3}i) & 0 & 2 \\
0 & 1 - (1 + \sqrt{3}i) & -1 \\
-1 & 1 & 1 - (1 + \sqrt{3}i)
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}.
$$

One solution to this system is the vector

$$
\begin{pmatrix}
-2/3\sqrt{3}i, \\
1/3\sqrt{3}i, \\
1
\end{pmatrix}^t.
$$

Similarly, the vector

$$
\begin{pmatrix}
2/3\sqrt{3}i, \\
-1/3\sqrt{3}i, \\
1
\end{pmatrix}^t
$$

is an eigenvector corresponding to the eigenvalue $\lambda_3 = 1 - \sqrt{3}i$. □

Maple provides a number of methods to compute the eigenvalues and eigenvectors of a matrix. To find the eigenvalues we can use the function `Eigenvals`. For the matrix in Example 2 we enter

> with(linalg);
> A:=matrix(3,3,[1,0,2,0,1,-1,-1,1,1]);
> evalf(Eigenvals(A));

$$
[1.000000000 + 1.732050807I, \\
1.000000000 - 1.732050807I, \\
1.000000000]
$$

This computes the eigenvalues

$$
\lambda_2 = 1 + \sqrt{3}i, \\
\lambda_3 = 1 - \sqrt{3}i, \\
\lambda_3 = 1.
$$

To compute both the eigenvalues and eigenvectors, we can use the Maple function `eigenvects`. For example,

> eigenvects(A);

produces the output

$$
[1 + I\sqrt{3}, 1, \{-2, 1, -I\sqrt{3}\}], \\
[1 - I\sqrt{3}, 1, \{-2, 1, I\sqrt{3}\}], \\
[1, 1, \{1, 1, 0\}]
$$

The first set of bracketed information states that the eigenvalue $1 + \sqrt{3}i$ is an eigenvalue of multiplicity 1 and that an associated eigenvector is $(-2, 1, -\sqrt{3}i)^t$. The other sets of bracketed information tell the multiplicity and eigenvectors for the eigenvalues $1 - \sqrt{3}i$ and $1$.

Any nonzero multiple of an eigenvector is also an eigenvector, so multiplying each coordinate by $(\sqrt{3}/3)i$ gives the previously determined eigenvector of $\lambda_2 = 1 + \sqrt{3}i$, which was

$$
\begin{pmatrix}
-2/3\sqrt{3}i, \\
1/3\sqrt{3}i, \\
1
\end{pmatrix}^t.
$$
A similar manner multiplication will give use the previously determined eigenvectors for the eigenvalues $\lambda_3 = 1 - \sqrt{3}i$ and $\lambda_1 = 1$.

The notions of eigenvalues and eigenvectors are introduced here for a specific computational convenience, but these concepts arise frequently in the study of physical systems. In fact, they are of sufficient interest that Chapter 9 is devoted to their numerical approximation.

The spectral radius $\rho(A)$ of a matrix $A$ is defined by

$$\rho(A) = \max |\lambda|, \text{ where } \lambda \text{ is an eigenvalue of } A.$$ 

*(Note: For complex $\lambda = \alpha + \beta i$, we have $|\lambda| = (\alpha^2 + \beta^2)^{1/2}$.)

**EXAMPLE 3** For the matrix considered in Example 2,

$$\rho(A) = \max\{|1 + \sqrt{3}i|, |1 - \sqrt{3}i|\} = \max\{1, 2\} = 2. \quad \Box$$

The spectral radius is closely related to the norm of a matrix.

**[l_2 Matrix Norm Characterization]** If $A$ is an $n \times n$ matrix, then

(i) $\|A\|_2 = (\rho(A^t A))^{1/2}$;

(ii) $\rho(A) \leq \|A\|$ for any natural norm.

The first part of this result is the computational method for determining the $l_2$ norm of matrices that we mentioned at the end of the previous section.

**EXAMPLE 4** If $A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{bmatrix}$,

then $A^t A = \begin{bmatrix} 3 & 2 & -1 \\ 2 & 6 & 4 \\ -1 & 4 & 5 \end{bmatrix}$.

To calculate $\rho(A^t A)$, we need the eigenvalues of $A^t A$:

$$0 = \det(A^t A - \lambda I) = \det \begin{bmatrix} 3 - \lambda & 2 & -1 \\ 2 & 6 - \lambda & 4 \\ -1 & 4 & 5 - \lambda \end{bmatrix}$$

$$= -\lambda^3 + 14\lambda^2 - 42\lambda = -\lambda(\lambda^2 - 14\lambda + 42).$$

So the eigenvalues are

$$\lambda = 0, \quad \lambda = 7 + \sqrt{7}, \quad \text{and} \quad \lambda = 7 - \sqrt{7}.$$
Hence
\[ \|A\|_2 = \sqrt{\rho(A^t A)} = \sqrt{\max\{0, 7 - \sqrt{7}, 7 + \sqrt{7}\}} = \sqrt{7 + \sqrt{7}} \approx 3.106. \]

The operations in Example 4 can also be performed using Maple:
\[
\begin{align*}
&> \text{with(linalg)}; \\
&> A := \text{matrix}(3, 3, [1, 1, 0, 1, 2, 1, -1, 1, 2]); \\
&> B := \text{transpose}(A); \\
&> C := \text{multiply}(A, B); \\
&> \text{evalf(Eigenvals(C))};
\end{align*}
\]
giving
\[ \begin{bmatrix} 0.3195356929 \times 10^{-8}, 4.354248691, 9.645751310 \end{bmatrix} \]
Since \( \|A\|_2 = \sqrt{\rho(A^t A)} = \sqrt{\rho(C)} \), we have
\[ \|A\|_2 = \sqrt{9.645751310} = 3.105760987. \]

Maple also permits us to obtain \( \|A\|_2 = \sqrt{7 + \sqrt{7}} \) directly with the command
\[ > \text{norm}(A, 2); \]
The \( l_\infty \) norm of \( A \) can be determined with \text{norm}(A) \) or \( \text{norm}(A, \text{infinity}) \). For the matrix \( A \) in Example 4 we have \( \|A\|_\infty = 4 \).

In studying iterative matrix techniques, it is of particular importance to know when powers of a matrix become small (that is, when all of the entries approach zero). We call an \( n \times n \) matrix \( A \) \textbf{convergent} if for each \( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots, n \) we have
\[ \lim_{k \to \infty} (A^k)_{ij} = 0. \]

EXAMPLE 5 Let
\[ A = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} \end{bmatrix}. \]
Computing powers of \( A \), we obtain:
\[ A^2 = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} \end{bmatrix}, \quad A^3 = \begin{bmatrix} \frac{1}{8} & 0 \\ \frac{3}{16} & \frac{1}{8} \end{bmatrix}, \quad A^4 = \begin{bmatrix} \frac{1}{16} & 0 \\ \frac{3}{16} & \frac{1}{16} \end{bmatrix}, \]
and, in general,
\[ A^k = \begin{bmatrix} \left(\frac{1}{2}\right)^k & 0 \\ \frac{k}{2^{k+1}} & \left(\frac{1}{2}\right)^k \end{bmatrix}. \]
Since
\[ \lim_{k \to \infty} \left(\frac{1}{2}\right)^k = 0 \quad \text{and} \quad \lim_{k \to \infty} \frac{k}{2^{k+1}} = 0, \]
\( A \) is a convergent matrix. Note that \( \rho(A) = \frac{1}{2} \), since \( \frac{1}{2} \) is the only eigenvalue of \( A \).
7.3. EIGENVALUES AND EIGENVECTORS

The following important connection exists between the spectral radius of a matrix and the convergence of the matrix.

[Convergent Matrix Equivalences] The following are equivalent statements:

(i) \(A\) is a convergent matrix.

(ii) \(\lim_{n \to \infty} \|A^n\| = 0\), for some natural norm.

(iii) \(\lim_{n \to \infty} \|A^n\| = 0\), for all natural norms.

(iv) \(\rho(A) < 1\).

(v) \(\lim_{n \to \infty} A^n x = 0\), for every \(x\).
EXERCISE SET 7.3

1. Compute the eigenvalues and associated eigenvectors of the following matrices.

   (a) \[
   \begin{bmatrix}
   2 & -1 \\
   -1 & 2
   \end{bmatrix}
   \]

   (b) \[
   \begin{bmatrix}
   0 & 1 \\
   1 & 1
   \end{bmatrix}
   \]

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

   (d) \[
   \begin{bmatrix}
   1 & -1 \\
   -2 & -2
   \end{bmatrix}
   \]

   (e) \[
   \begin{bmatrix}
   2 & 1 & 0 \\
   1 & 2 & 0 \\
   0 & 0 & 3
   \end{bmatrix}
   \]

   (f) \[
   \begin{bmatrix}
   -1 & 2 & 0 \\
   0 & 3 & 4 \\
   0 & 0 & 7
   \end{bmatrix}
   \]

   (g) \[
   \begin{bmatrix}
   2 & 1 & 1 \\
   2 & 3 & 2 \\
   1 & 1 & 2
   \end{bmatrix}
   \]

   (h) \[
   \begin{bmatrix}
   3 & 2 & -1 \\
   1 & -2 & 3 \\
   2 & 0 & 4
   \end{bmatrix}
   \]

2. Find the spectral radius for each matrix in Exercise 1.

3. Show that

   \[
   A_1 = \begin{bmatrix}
   1 & 0 \\
   \frac{1}{4} & \frac{1}{2}
   \end{bmatrix}
   \]

   is not convergent, but

   \[
   A_2 = \begin{bmatrix}
   \frac{1}{2} & 0 \\
   16 & \frac{1}{2}
   \end{bmatrix}
   \]

   is convergent.

4. Which of the matrices in Exercise 1 are convergent?

5. Find the \( \| \cdot \|_2 \) norms of the matrices in Exercise 1.

6. Show that if \( \lambda \) is an eigenvalue of a matrix \( A \) and \( \| \cdot \| \) is a vector norm, then an eigenvector \( x \) associated with \( \lambda \) exists with \( \| x \| = 1 \).

7. Find matrices \( A \) and \( B \) for which \( \rho(A + B) > \rho(A) + \rho(B) \). (This shows that \( \rho(A) \) cannot be a matrix norm.)

8. Show that if \( A \) is symmetric, then \( \| A \|_2 = \rho(A) \).

9. Let \( \lambda \) be an eigenvalue of the \( n \times n \) matrix \( A \) and \( x \neq 0 \) be an associated eigenvector.
(a) Show that $\lambda$ is also an eigenvalue of $A^t$.

(b) Show that for any integer $k \geq 1$, $\lambda^k$ is an eigenvalue of $A^k$ with eigenvector $\mathbf{x}$.

(c) Show that if $A^{-1}$ exists, then $1/\lambda$ is an eigenvalue of $A^{-1}$ with eigenvector $\mathbf{x}$.

(d) Let $\alpha \neq \lambda$ be given. Show that if $(A - \alpha I)^{-1}$ exists, then $1/(\lambda - \alpha)$ is an eigenvalue of $(A - \alpha I)^{-1}$ with eigenvector $\mathbf{x}$.

10. In Exercise 8 of Section 6.4, we assumed that the contribution a female beetle of a certain type made to the future years’ beetle population could be expressed in terms of the matrix

$$
A = \begin{bmatrix}
0 & 0 & 6 \\
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{3} & 0
\end{bmatrix},
$$

where the entry in the $i$th row and $j$th column represents the probabilistic contribution of a beetle of age $j$ onto the next year’s female population of age $i$.

(a) Does the matrix $A$ have any real eigenvalues? If so, determine them and any associated eigenvectors.

(b) If a sample of this species was needed for laboratory test purposes that would have a constant proportion in each age group from year to year, what criteria could be imposed on the initial population to ensure that this requirement would be satisfied?


CHAPTER 7. ITERATIVE METHODS FOR SOLVING LINEAR SYSTEMS

7.4 The Jacobi and Gauss-Seidel Methods

In this section we describe the elementary Jacobi and Gauss-Seidel iterative methods. These are classic methods that date to the late eighteenth century, but they find current application in problems where the matrix is large and has mostly zero entries in predictable locations. Applications of this type are common, for example, in the study of large integrated circuits and in the numerical solution of boundary-value problems and partial-differential equations.

An iterative technique for solving the \( n \times n \) linear system \( A\mathbf{x} = \mathbf{b} \) starts with an initial approximation \( \mathbf{x}(0) \) to the solution \( \mathbf{x} \) and generates a sequence of vectors \( \{\mathbf{x}(k)\}_{k=1}^{\infty} \) that converges to \( \mathbf{x} \). These iterative techniques involve a process that converts the system \( A\mathbf{x} = \mathbf{b} \) into an equivalent system of the form \( \mathbf{x} = T\mathbf{x} + \mathbf{c} \) for some \( n \times n \) matrix \( T \) and vector \( \mathbf{c} \).

After the initial vector \( \mathbf{x}(0) \) is selected, the sequence of approximate solution vectors is generated by computing \( \mathbf{x}(k) = T\mathbf{x}(k-1) + \mathbf{c} \) for each \( k = 1, 2, 3, \ldots \).

The following result provides an important connection between the eigenvalues of the matrix \( T \) and the expectation that the iterative method will converge.

[Convergence and the Spectral Radius] The sequence

\[
\mathbf{x}(k) = T\mathbf{x}(k-1) + \mathbf{c}
\]

converges to the unique solution of \( \mathbf{x} = T\mathbf{x} + \mathbf{c} \) for any \( \mathbf{x}(0) \) in \( \mathbb{R}^n \) if and only if \( \rho(T) < 1 \).

**EXAMPLE 1** The linear system \( A\mathbf{x} = \mathbf{b} \) given by

\[
\begin{align*}
E_1: & \quad 10x_1 - x_2 + 2x_3 = 6, \\
E_2: & \quad -x_1 + 11x_2 - x_3 + 3x_4 = 25, \\
E_3: & \quad 2x_1 - x_2 + 10x_3 - x_4 = -11, \\
E_4: & \quad 3x_2 - x_3 + 8x_4 = 15
\end{align*}
\]

has solution \( \mathbf{x} = (1, 2, -1, 1)^t \). To convert \( A\mathbf{x} = \mathbf{b} \) to the form \( \mathbf{x} = T\mathbf{x} + \mathbf{c} \), solve equation \( E_i \) for \( x_i \) obtaining

\[
\begin{align*}
x_1 &= \frac{1}{10}x_2 - \frac{1}{5}x_3 + \frac{3}{5}, \\
x_2 &= \frac{1}{11}x_1 + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11}, \\
x_3 &= \frac{1}{5}x_1 + \frac{1}{10}x_2 + \frac{1}{10}x_4 - \frac{11}{10}, \\
x_4 &= \frac{3}{8}x_2 + \frac{1}{8}x_3 + \frac{15}{8}.
\end{align*}
\]
7.4. THE JACOBI AND GAUSS-SEIDEL METHODS

Then \( Ax = b \) has the form \( x = Tx + c \), with

\[
T = \begin{bmatrix}
0 & 1 & -1 & 0 \\
1 & 0 & \frac{1}{11} & -3 \\
-\frac{1}{5} & 1 & 0 & \frac{1}{10} \\
0 & -\frac{3}{8} & \frac{1}{8} & 0
\end{bmatrix}
\quad \text{and} \quad
\begin{bmatrix}
3 \\
\frac{25}{11} \\
\frac{11}{10} \\
\frac{15}{8}
\end{bmatrix}
\]

For an initial approximation, suppose \( x^{(0)} = (0, 0, 0, 0)^t \). Then \( x^{(1)} \) is given by

\[
x_1^{(1)} = \frac{1}{10} x_2^{(0)} - \frac{1}{5} x_3^{(0)} + \frac{3}{5} = 0.6000,
\]
\[
x_2^{(1)} = \frac{1}{11} x_1^{(0)} + \frac{1}{11} x_3^{(0)} - \frac{3}{11} x_4^{(0)} + \frac{25}{11} = 2.2727,
\]
\[
x_3^{(1)} = -\frac{1}{5} x_1^{(0)} + \frac{1}{10} x_2^{(0)} - \frac{1}{10} x_4^{(0)} - \frac{11}{10} = -1.1000,
\]
\[
x_4^{(1)} = -\frac{3}{8} x_2^{(0)} + \frac{1}{8} x_3^{(0)} + \frac{15}{8} = 1.8750.
\]

Additional iterates, \( x^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)})^t \), are generated in a similar manner and are presented in Table 7.1. The decision to stop after 10 iterations was based on the criterion

\[
\|x^{(10)} - x^{(9)}\|_\infty = 8.0 \times 10^{-4} < 10^{-3}.
\]

Since we know that \( x = (1, 2, -1, 1)^t \), we have \( \|x^{(10)} - x\|_\infty \approx 0.0002 \).

---

**Table 7.1**

<table>
<thead>
<tr>
<th>( k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1^{(k)} )</td>
<td>0.000</td>
<td>0.6000</td>
<td>1.0473</td>
<td>0.9326</td>
<td>1.0152</td>
<td>0.9890</td>
<td>1.0032</td>
<td>0.9981</td>
<td>1.0006</td>
<td>0.9997</td>
<td>1.0001</td>
</tr>
<tr>
<td>( x_2^{(k)} )</td>
<td>0.0000</td>
<td>2.2727</td>
<td>1.7159</td>
<td>2.053</td>
<td>1.9537</td>
<td>2.0114</td>
<td>1.9922</td>
<td>2.0023</td>
<td>1.9987</td>
<td>2.0004</td>
<td>1.9998</td>
</tr>
<tr>
<td>( x_3^{(k)} )</td>
<td>0.0000</td>
<td>-1.1000</td>
<td>-0.8052</td>
<td>-1.0493</td>
<td>-0.9681</td>
<td>-1.0103</td>
<td>-0.9945</td>
<td>-1.0020</td>
<td>-0.9990</td>
<td>-1.0004</td>
<td>-0.9998</td>
</tr>
<tr>
<td>( x_4^{(k)} )</td>
<td>0.0000</td>
<td>1.8750</td>
<td>0.8852</td>
<td>1.1309</td>
<td>0.9739</td>
<td>1.0214</td>
<td>0.9944</td>
<td>1.0036</td>
<td>0.9989</td>
<td>1.0006</td>
<td>0.9998</td>
</tr>
</tbody>
</table>

---

**Example 1** uses the **Jacobi** iterative method. It consists of solving the \( i \)th equation in \( Ax = b \) for \( x_i \) to obtain, provided \( a_{ii} \neq 0 \),

\[
x_i = \sum_{\substack{j=1 \atop j \neq i}}^{n} \left( -\frac{a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \quad \text{for } i = 1, 2, \ldots, n,
\]
and generating each $x_i^{(k)}$ from components of $x^{(k-1)}$, for $k \geq 1$, by

$$
x_i^{(k)} = \frac{\sum_{j=1}^{n} (-a_{ij}x_j^{(k-1)}) + b_i}{a_{ii}}, \quad \text{for } i = 1, 2, \ldots, n.
$$

The method is written in the form $x^{(k)} = Tx^{(k-1)} + c$ by splitting $A$ into its diagonal and off-diagonal parts. To see this, let $D$ be the diagonal matrix whose diagonal entries are those of $A$, $-L$ be the strictly lower-triangular part of $A$, and $-U$ be the strictly upper triangular part of $A$. With this notation,

$$
A = \begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} \\
  a_{21} & a_{22} & \cdots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
$$

is split into

$$
A = \begin{bmatrix}
  a_{11} & 0 & \cdots & 0 \\
  0 & a_{22} & \cdots & 0 \\
  0 & 0 & \cdots & a_{nn} \\
  0 & a_{n2} & \cdots & -a_{1n}
\end{bmatrix} - \begin{bmatrix}
  0 & 0 & \cdots & 0 \\
  -a_{21} & -a_{n1} & \cdots & 0 \\
  -a_{n1} & -a_{n,n-1} & \cdots & 0 \\
  0 & 0 & \cdots & -a_{n-1,n}
\end{bmatrix}

= D - L - U.
$$

The equation $Ax = b$ or $(D - L - U)x = b$ is then transformed into

$$
Dx = (L + U)x + b,
$$

and, if $D^{-1}$ exists—that is, if $a_{ii} \neq 0$ for each $i$—then

$$
x = D^{-1}(L + U)x + D^{-1}b.
$$

This results in the matrix form of the Jacobi iterative technique:

$$
x^{(k)} = T_jx^{(k-1)} + c_j,
$$

where $T_j = D^{-1}(L + U)$ and $c_j = D^{-1}b$.

The program JACITR71 implements the Jacobi method. If $a_{ii} = 0$ for some $i$ and the system is nonsingular, a reordering of the equations is performed so that no $a_{ii} = 0$. To speed convergence, the equations should be arranged so that $a_{ii}$ is as large as possible.

A likely improvement on the Jacobi method can be seen by reconsidering Eq.(7.1). In this equation all the components of $x^{(k-1)}$ are used to compute each of the
EXAMPLE 2 In Example 1 we used the Jacobi method to solve the linear system. That is, we can use
\[ x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)}}{a_{ii}} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i, \]
for each \( i = 1, 2, \ldots, n \). This modification is called the \textbf{Gauss-Seidel} iterative technique and is illustrated in the following example.

\[ \begin{align*}
x_1^{(k)} &= \frac{1}{10} x_2^{(k-1)} - \frac{1}{5} x_3^{(k-1)} + \frac{3}{10}, \\
x_2^{(k)} &= \frac{1}{10} x_1^{(k)} + \frac{1}{10} x_3^{(k-1)} - \frac{3}{10} x_4^{(k-1)} + \frac{25}{10}, \\
x_3^{(k)} &= -\frac{1}{5} x_1^{(k)} + \frac{1}{10} x_2^{(k)} + \frac{1}{10} x_4^{(k-1)} - \frac{10}{10}, \\
x_4^{(k)} &= -\frac{3}{8} x_2^{(k)} + \frac{1}{8} x_3^{(k)} + \frac{1}{8}.
\end{align*} \]

Letting \( x^{(0)} = (0, 0, 0, 0)^t \), we generate the \textbf{Gauss-Seidel} iterates in Table 7.2. Since
\[ \|x^{(5)} - x^{(4)}\|_\infty = 0.0008 < 10^{-3}, \]
\( x^{(5)} \) is accepted as a reasonable approximation to the solution. Note that Jacobi’s method in Example 1 required twice as many iterations for the same accuracy. \( \square \)

<table>
<thead>
<tr>
<th>( k )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1^{(k)} )</td>
<td>0.0000</td>
<td>0.6000</td>
<td>1.030</td>
<td>1.0065</td>
<td>1.0009</td>
<td>1.0001</td>
</tr>
<tr>
<td>( x_2^{(k)} )</td>
<td>0.0000</td>
<td>2.3272</td>
<td>2.037</td>
<td>2.0036</td>
<td>2.0003</td>
<td>2.0000</td>
</tr>
<tr>
<td>( x_3^{(k)} )</td>
<td>0.0000</td>
<td>-0.9873</td>
<td>-1.014</td>
<td>-1.0025</td>
<td>-1.0003</td>
<td>-1.0000</td>
</tr>
<tr>
<td>( x_4^{(k)} )</td>
<td>0.0000</td>
<td>0.8789</td>
<td>0.9844</td>
<td>0.9983</td>
<td>0.9999</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

To write the Gauss-Seidel method in matrix form, multiply both sides of Eq. (7.2) by \( a_{ii} \) and collect all \( k \)th iterate terms, to give
\[ a_{i1} x_1^{(k)} + a_{i2} x_2^{(k)} + \cdots + a_{ii} x_i^{(k)} = -a_{i,i+1} x_{i+1}^{(k-1)} - \cdots - a_{ii} x_n^{(k-1)} + b_i, \]
for each \( i = 1, 2, \ldots, n \). Writing all \( n \) equations gives

\[
\begin{align*}
    a_{11}x_1^{(k)} &= -a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1, \\
    a_{21}x_1^{(k)} + a_{22}x_2^{(k)} &= -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2, \\
    &\vdots \\
    a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \cdots + a_{nn}x_n^{(k)} &= b_n.
\end{align*}
\]

With the definitions of \( D, L \), and \( U \) that we used previously, we have the Gauss-Seidel method represented by

\[
(D - L)x^{(k)} = Ux^{(k-1)} + b
\]

or, if \((D - L)^{-1}\) exists, by

\[
x^{(k)} = T_gx^{(k-1)} + c_g, \quad \text{for each } k = 1, 2, \ldots,
\]

where \( T_g = (D - L)^{-1}U \) and \( c_g = (D - L)^{-1}b \). Since

\[
\det(D - L) = a_{11} \cdot a_{22} \cdots a_{nn},
\]

the lower-triangular matrix \( D - L \) is nonsingular precisely when \( a_{ii} \neq 0 \) for each \( i = 1, 2, \ldots, n \). The Gauss-Seidel method is performed by the program GSEITR72.

The preceding discussion and the results of Examples 1 and 2 seem to imply that the Gauss-Seidel method is superior to the Jacobi method. This is almost always true, but there are linear systems for which the Jacobi method converges and the Gauss-Seidel method does not. However, if \( A \) is strictly diagonally dominant, then for any \( b \) and any choice of \( x^{(0)} \), the Jacobi and Gauss-Seidel methods will both converge to the unique solution of \( Ax = b \).
EXERCISE SET 7.4

1. Find the first two iterations of the Jacobi method for the following linear systems, using \( x^{(0)} = 0 \):

(a) \[
\begin{align*}
3x_1 - x_2 + x_3 &= 1, \\
3x_1 + 6x_2 + 2x_3 &= 0, \\
3x_1 + 3x_2 + 7x_3 &= 4.
\end{align*}
\]

(b) \[
\begin{align*}
10x_1 - x_2 &= 9, \\
x_1 + 10x_2 - 2x_3 &= 7, \\
-2x_2 + 10x_3 &= 6.
\end{align*}
\]

(c) \[
\begin{align*}
5x_1 + 10x_2 - 4x_3 &= 25, \\
-4x_2 + 8x_3 - x_4 &= -4, \\
x_2 + 5x_3 + x_4 &= 0.
\end{align*}
\]

(d) \[
\begin{align*}
6x_1 + x_2 - x_3 + x_4 &= -2, \\
x_2 - x_3 - x_4 &= -1, \\
x_2 + 3x_3 &= 1.
\end{align*}
\]

(e) \[
\begin{align*}
4x_1 + x_2 + x_3 + x_5 &= 6, \\
x_2 - x_3 + x_4 &= 6, \\
x_2 - x_3 + 4x_4 &= 6, \\
x_2 - x_3 + 4x_4 &= 6.
\end{align*}
\]

(f) \[
\begin{align*}
4x_1 - x_2 - x_4 &= 0, \\
x_1 + 4x_2 - x_3 - x_5 &= 5, \\
x_1 - 4x_2 - x_3 - x_5 &= 6, \\
x_1 - 4x_2 - x_3 - x_5 &= 6.
\end{align*}
\]

2. Repeat Exercise 1 using the Gauss-Seidel method.

3. Use the Jacobi method to solve the linear systems in Exercise 1, with \( TOL = 10^{-3} \) in the \( l_\infty \) norm.

4. Repeat Exercise 3 using the Gauss-Seidel method.

5. The linear system

\[
\begin{align*}
x_1 - x_3 &= 0.2, \\
-\frac{1}{2}x_1 + x_2 - \frac{1}{4}x_3 &= -1.425, \\
x_1 - \frac{1}{2}x_2 + x_3 &= 2.
\end{align*}
\]

has the solution \((0.9, -0.8, 0.7)^t\).

(a) Is the coefficient matrix

\[
\begin{bmatrix}
1 & 0 & -1 \\
-\frac{1}{2} & 1 & -\frac{1}{4} \\
1 & -\frac{1}{2} & 1
\end{bmatrix}
\]
strictly diagonally dominant?

(b) Compute the spectral radius of the Jacobi matrix $T_j$.

(c) Use the Jacobi iterative method to approximate the solution to the linear system with a tolerance of $10^{-2}$ and a maximum of 300 iterations.

(d) What happens in part (c) when the system is changed to

\[
\begin{align*}
    x_1 - 2x_3 &= 0.2, \\
    -\frac{1}{2}x_1 + x_2 - \frac{1}{4}x_3 &= -1.425, \\
    x_1 - \frac{1}{2}x_2 + x_3 &= 2.
\end{align*}
\]

6. Repeat Exercise 5 using the Gauss-Seidel method.

7. Show that if $A$ is strictly diagonally dominant, then $\|T_j\|_\infty < 1$. 

7.5 The SOR Method

The SOR method is similar to the Jacobi and Gauss-Seidel methods, but it uses a scaling factor to more rapidly reduce the approximation error. In contrast to the classic methods discussed in the previous section, the SOR technique is a more recent innovation.

The SOR technique is one of a class of relaxation methods that compute approximations $x^{(k)}$ by the formula

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

where $\omega$ is the scaling factor.

When $\omega = 1$, we have the Gauss-Seidel method. When $0 < \omega < 1$, the procedures are called under-relaxation methods and can be used to obtain convergence of some systems that are not convergent by the Gauss-Seidel method.

When $1 < \omega$, the procedures are called over-relaxation methods, which are used to accelerate the convergence for systems that are convergent by the Gauss-Seidel technique. These methods are abbreviated SOR for Successive Over-Relaxation and are used for solving the linear systems that occur in the numerical solution of certain partial-differential equations.

To determine the matrix form of the SOR method, we rewrite the preceding equation as

$$a_{ii}x_i^{(k)} + \omega \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = (1 - \omega)a_{ii}x_i^{(k-1)} - \omega \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} + \omega b_i,$$

so that in vector form we have

$$(D - \omega L)x^{(k)} = [(1 - \omega)D + \omega U]x^{(k-1)} + \omega b.$$

If $(D - \omega L)^{-1}$ exists, then

$$x^{(k)} = T_\omega x^{(k-1)} + c_\omega,$$

where $T_\omega = (D - \omega L)^{-1}[(1 - \omega)D + \omega U]$ and $c_\omega = \omega(D - \omega L)^{-1}b$. The SOR technique can be applied using the program SORITR73.

**EXAMPLE 1** The linear system $Ax = b$ given by

$$
\begin{align*}
4x_1 + 3x_2 &= 24, \\
3x_1 + 4x_2 - x_3 &= 30, \\
-x_2 + 4x_3 &= -24
\end{align*}
$$

has the solution $(3, 4, -5)^t$. The Gauss-Seidel method and the SOR method with $\omega = 1.25$ will be used to solve this system, using $x^{(0)} = (1, 1, 1)^t$ for both methods.
For each \( k = 1, 2, \ldots \), the equations for the Gauss-Seidel method are
\[
\begin{align*}
x_1^{(k)} &= -0.75x_2^{(k-1)} + 6, \\
x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5, \\
x_3^{(k)} &= 0.25x_2^{(k)} - 6,
\end{align*}
\]
and the equations for the SOR method with \( \omega = 1.25 \) are
\[
\begin{align*}
x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5, \\
x_2^{(k)} &= -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375, \\
x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5.
\end{align*}
\]

The first seven iterates for each method are listed in Tables 7.3 and 7.4. To be accurate to seven decimal places, the Gauss-Seidel method required 34 iterations, as opposed to only 14 iterations for the SOR method with \( \omega = 1.25 \).

The obvious question to ask is how the appropriate value of \( \omega \) is chosen. Although no complete answer to this question is known for the general \( n \times n \) linear system, the following result can be used in certain situations.
7.5. THE SOR METHOD

[SOR Method Convergence] If $A$ is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate solution vector $x^{(0)}$.

If, in addition, $A$ is tridiagonal, then $\rho(T_j) = [\rho(T_j)]^2 < 1$, and the optimal choice of $\omega$ for the SOR method is

$$
\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}.
$$

With this choice of $\omega$, we have $\rho(T_\omega) = \omega - 1$.

EXAMPLE 2

The matrix

$$
A = \begin{bmatrix}
4 & 3 & 0 \\
3 & 4 & -1 \\
0 & -1 & 4
\end{bmatrix}
$$

given in Example 1 is positive definite and tridiagonal. Since

$$
T_j = D^{-1}(L+U) = \begin{bmatrix}
\frac{1}{4} & 0 & 0 \\
0 & \frac{1}{4} & 0 \\
0 & 0 & \frac{1}{4}
\end{bmatrix} \begin{bmatrix}
0 & -3 & 0 \\
-3 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix} = \begin{bmatrix}
0 & -0.75 & 0 \\
-0.75 & 0 & 0.25 \\
0 & 0.25 & 0
\end{bmatrix},
$$

we have

$$
\det(T_j - \lambda I) = -\lambda(\lambda^2 - 0.625) \quad \text{and} \quad \rho(T_j) = \sqrt{0.625}.
$$

Hence, the optimal choice of $\omega$ is

$$
\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}} = \frac{2}{1 + \sqrt{1 - 0.625}} \approx 1.24.
$$

This explains the rapid convergence obtained in Example 1 by using $\omega = 1.25$. □
EXERCISE SET 7.5

1. Find the first two iterations of the SOR method with $\omega = 1.1$ for the following linear systems, using $x^{(0)} = 0$:

   (a) $3x_1 - x_2 + x_3 = 1$, $3x_1 + 6x_2 + 2x_3 = 0$, $3x_1 + 3x_2 + 7x_3 = 4$.
   (b) $10x_1 - x_2 = 9$, $-x_1 + 10x_2 - 2x_3 = 7$, $-2x_2 + 10x_3 = 6$.
   (c) $10x_1 + 5x_2 = 6$, $-4x_2 + 8x_3 - x_4 = -4$, $-x_3 + 5x_4 = -4$.
   (d) $6x_1 + x_2 - x_3 + x_4 = -2$, $2x_2 - x_3 - x_4 + 5x_5 + x_4 = 0$, $-x_3 + 5x_4 = -4$.
   (e) $4x_1 + x_2 + x_3 + x_5 = 6$, $-x_1 - 3x_2 + x_3 + x_4 = 6$, $2x_1 + x_2 + 5x_3 - x_4 - x_5 = 6$.
   (f) $4x_1 - x_2 - x_4 = 0$, $-x_1 + 4x_2 - x_3 - x_5 = 5$, $-x_2 + 4x_3 - x_6 = 0$.

2. Use the SOR method with $\omega = 1.2$ to solve the linear systems in Exercise 1 with a tolerance $TOL = 10^{-3}$ in the $l_\infty$ norm.

3. For those matrices in Exercise 1 that are both tridiagonal and positive definite, use the SOR method with the optimal choice of $\omega$.

4. Suppose that an object can be at any one of $n + 1$ equally spaced points $x_0$, $x_1, \ldots, x_n$. When an object is at location $x_i$, it is equally likely to move to either $x_{i-1}$ or $x_{i+1}$ and cannot directly move to any other location. Consider the probabilities $\{P_i\}_{i=0}^n$ that an object starting at location $x_i$ will reach the left endpoint $x_0$ before reaching the right endpoint $x_n$. Clearly, $P_0 = 1$ and $P_n = 0$. Since the object can move to $x_i$ only from $x_{i-1}$ or $x_{i+1}$ and does so with probability $\frac{1}{2}$ for each of these locations,

$$P_i = \frac{1}{2}P_{i-1} + \frac{1}{2}P_{i+1}, \quad \text{for each } i = 1, 2, \ldots, n - 1.$$
7.5. **THE SOR METHOD**

(a) Show that
\[
\begin{bmatrix}
1 & -\frac{1}{2} & 0 & 0 \\
\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\
0 & -\frac{1}{2} & 1 & 0 \\
-\frac{1}{2} & 1 & -\frac{1}{2} & 1 \\
0 & 0 & -\frac{1}{2} & 1
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
\vdots \\
P_{n-1}
\end{bmatrix}
= 
\begin{bmatrix}
\frac{1}{2} \\
0 \\
\vdots \\
0
\end{bmatrix}.
\]

(b) Solve this system using \( n = 10, 50, \) and 100.

(c) Change the probabilities to \( \alpha \) and \( 1 - \alpha \) for movement to the left and right, respectively, and derive the linear system similar to the one in part (a).

(d) Repeat part (b) with \( \alpha = \frac{1}{3} \).

5. The forces on the bridge truss shown here satisfy the equations in the following table:

<table>
<thead>
<tr>
<th>Joint</th>
<th>Horizontal Component</th>
<th>Vertical Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-F_1 + \frac{\sqrt{2}}{2} f_1 + f_2 = 0)</td>
<td>(\frac{\sqrt{2}}{2} f_1 - F_2 = 0)</td>
</tr>
<tr>
<td>2</td>
<td>(-\frac{\sqrt{2}}{2} f_1 + \frac{\sqrt{3}}{2} f_4 = 0)</td>
<td>(-\frac{\sqrt{2}}{2} f_1 - f_3 + \frac{1}{2} f_4 = 0)</td>
</tr>
<tr>
<td>3</td>
<td>(-f_2 + f_5 = 0)</td>
<td>(f_3 - 10,000 = 0)</td>
</tr>
<tr>
<td>4</td>
<td>(-\frac{\sqrt{3}}{2} f_4 - f_5 = 0)</td>
<td>(\frac{1}{2} f_4 - F_3 = 0)</td>
</tr>
</tbody>
</table>
This linear system can be placed in the matrix form

\[
\begin{bmatrix}
-1 & 0 & 0 & \frac{\sqrt{3}}{2} & 1 & 0 & 0 & 0 \\
0 & -1 & 0 & \frac{\sqrt{3}}{2} & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & \frac{1}{2} & 0 \\
0 & 0 & 0 & -\frac{\sqrt{2}}{2} & 0 & -1 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -\frac{\sqrt{3}}{2} & 0 & 0 & \frac{\sqrt{3}}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{\sqrt{3}}{2} & -1
\end{bmatrix}
\begin{bmatrix}
F_1 \\
F_2 \\
F_3 \\
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
0 \\
0 \\
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5
\end{bmatrix}.
\]

(a) Explain why the system of equations was reordered.

(b) Approximate the solution of the resulting linear system to within $10^{-2}$ in the $l_\infty$ norm using as initial approximation the vector all of whose entries are 1s and (i) the Gauss-Seidel method, (ii) the Jacobi method, and (iii) the SOR method with $\omega = 1.25$. 
7.6 Error Bounds and Iterative Refinement

This section considers the errors in approximation that are likely to occur when solving linear systems by both direct and iterative methods. There is no universally superior technique for approximating the solution to linear systems, but some methods will give better results than others when certain conditions are satisfied by the matrix.

It seems intuitively reasonable that if \( \tilde{x} \) is an approximation to the solution \( x \) of \( Ax = b \) and the residual vector, defined by \( b - A\tilde{x} \), has the property that \( \|b - A\tilde{x}\| \) is small, then \( \|x - \tilde{x}\| \) should be small as well. This is often the case, but certain systems, which occur quite often in practice, fail to have this property.

**EXAMPLE 1**

The linear system \( Ax = b \) given by

\[
\begin{bmatrix}
1 & 1 \\
1.0001 & 2 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\end{bmatrix} =
\begin{bmatrix}
3 \\
3.0001 \\
\end{bmatrix}
\]

has the unique solution \( x = (1, 1)^t \). The poor approximation \( \tilde{x} = (3, 0)^t \) has the residual vector

\[
b - A\tilde{x} = \begin{bmatrix}
3 \\
3.0001 \\
\end{bmatrix} - \begin{bmatrix}
1 & 2 \\
1.0001 & 2 \\
\end{bmatrix}\begin{bmatrix}
3 \\
0 \\
\end{bmatrix} = \begin{bmatrix}
0 \\
-0.0002 \\
\end{bmatrix},
\]

so \( \|b - A\tilde{x}\|_\infty = 0.0002 \). Although the norm of the residual vector is small, the approximation \( \tilde{x} = (3, 0)^t \) is obviously quite poor; in fact, \( \|x - \tilde{x}\|_\infty = 2 \). \( \square \)

The difficulty in Example 1 is explained quite simply by noting that the solution to the system represents the intersection of the lines

\( l_1: x_1 + 2x_2 = 3 \) and \( l_2: 1.0001x_1 + 2x_2 = 3.0001 \).

The point \( (3, 0) \) lies on \( l_1 \), and the lines are nearly parallel. This implies that \( (3, 0) \) also lies close to \( l_2 \), even though it differs significantly from the solution of the system, which is the intersection point \( (1, 1) \). (See Figure 7.6.)

**Figure 7.6**
Example 1 was clearly constructed to show the difficulties that might—and, in fact, do—arise. Had the lines not been nearly coincident, we would expect a small residual vector to imply an accurate approximation. In the general situation, we cannot rely on the geometry of the system to give an indication of when problems might occur. We can, however, obtain this information by considering the norms of the matrix and its inverse.

[Residual Vector Error Bounds] If \( \tilde{x} \) is an approximation to the solution of \( Ax = b \) and \( A \) is a nonsingular matrix, then for any natural norm,

\[
\| x - \tilde{x} \| \leq \| b - A\tilde{x} \| \cdot \| A^{-1} \|
\]

and

\[
\frac{\| x - \tilde{x} \|}{\| x \|} \leq \| A \| \cdot \| A^{-1} \| \frac{\| b - A\tilde{x} \|}{\| b \|}, \quad \text{provided } x \neq 0 \text{ and } b \neq 0.
\]

This implies that \( \| A^{-1} \| \) and \( \| A \| \cdot \| A^{-1} \| \) provide an indication of the connection between the residual vector and the accuracy of the approximation. In general, the relative error \( \| x - \tilde{x} \| / \| x \| \) is of most interest, and this error is bounded by the product of \( \| A \| \cdot \| A^{-1} \| \) with the relative residual for this approximation, \( \| b - A\tilde{x} \| / \| b \| \).

Any convenient norm can be used for this approximation; the only requirement is that it be used consistently throughout.

The condition number, \( K(A) \), of the nonsingular matrix \( A \) relative to a norm \( \| \cdot \| \) is

\[
K(A) = \| A \| \cdot \| A^{-1} \|.
\]

With this notation, we can reexpress the inequalities in the previous result as

\[
\| x - \tilde{x} \| \leq K(A) \frac{\| b - A\tilde{x} \|}{\| A \|} \quad \text{and} \quad \frac{\| x - \tilde{x} \|}{\| x \|} \leq K(A) \frac{\| b - A\tilde{x} \|}{\| b \|}.
\]

For any nonsingular matrix \( A \) and natural norm \( \| \cdot \| \),

\[
1 = \| I \| = \| A \cdot A^{-1} \| \leq \| A \| \cdot \| A^{-1} \| = K(A).
\]

A matrix \( A \) is well-behaved (called well-conditioned) if \( K(A) \) is close to 1 and is not well-behaved (called ill-conditioned) when \( K(A) \) is significantly greater than 1. Conditioning in this instance refers to the relative security that a small residual vector implies a correspondingly accurate approximate solution.

**Example 2** The matrix for the system considered in Example 1 was

\[
A = \begin{bmatrix}
1 & 2 \\
1.0001 & 2
\end{bmatrix}.
\]
7.6. ERROR BOUNDS AND ITERATIVE REFINEMENT

which has $\|A\|_{\infty} = 3.0001$. This norm would not be considered large. However,

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 5000.5 & -5000 \end{bmatrix}, \text{ so } \|A^{-1}\|_{\infty} = 20000,$$

and for the infinity norm, $K(A) = (20000)(3.0001) = 60002$. The size of the condition number for this example should certainly keep us from making hasty accuracy decisions based on the residual of an approximation.

In Maple the condition number $K_{\infty}$ for the matrix in Example 2 can be computed as follows:

```maple
>with(linalg);
>with(LinearAlgebra);
>A:=matrix(2,2,[1,2,1.0001,2]);
>cond(A);
```

$$60002.00000$$

The residual of an approximation can also be used to improve the accuracy of the approximation. Suppose that $\tilde{x}$ is an approximation to the solution of the linear system $Ax = b$ and that $r = b - Ax$ is the residual associated with $\tilde{x}$. Consider $\tilde{y}$, the approximate solution to the system $Ay = r$. Then

$$\tilde{y} \approx A^{-1}r = A^{-1}(b - A\tilde{x}) = A^{-1}b - A^{-1}A\tilde{x} = x - \tilde{x}.$$

So

$$x \approx \tilde{x} + \tilde{y}.$$

This new approximation $\tilde{x} + \tilde{y}$ is often much closer to the solution of $Ax = b$ than is $\tilde{x}$, and $\tilde{y}$ is easy to determine since it involves the same matrix, $A$, as the original system. This technique is called **iterative refinement**, or **iterative improvement**, and is illustrated in the following example. To increase accuracy, the residual vector is computed using double-digit arithmetic. The method can also be implemented with the program ITREF74.

**EXAMPLE 3** The linear system given by

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 15913 \\ 28.544 \\ 8.4254 \end{bmatrix}$$

has the exact solution $x = (1,1,1)^t$.

Using Gaussian elimination and five-digit rounding arithmetic leads successively to the augmented matrices

$$\begin{bmatrix} 3.3330 & 15920 & -10.333 & 15913 \\ 0 & -10596 & 16.501 & 10580 \\ 0 & -7451.4 & 6.5250 & -7444.9 \end{bmatrix}$$
The approximate solution to this system is
\[ \tilde{x} = (1.2001, 0.99991, 0.92538)^t. \]

When computed in 10-digit arithmetic, the residual vector corresponding to \( \tilde{x} \) is
\[
\begin{bmatrix}
15913 \\
28.544 \\
8.4254
\end{bmatrix} - \begin{bmatrix}
3.3330 & 15920 & -10.333 & 15913 \\
0 & -10596 & 16.501 & -10580 \\
0 & 0 & -5.0790 & -4.7000
\end{bmatrix} \begin{bmatrix}
1.20001 \\
0.99991 \\
0.92538
\end{bmatrix} = \begin{bmatrix}
15913.00518 \\
28.26987086 \\
8.611560367
\end{bmatrix} - \begin{bmatrix}
-0.00518 \\
0.27413 \\
-0.18616
\end{bmatrix}.
\]

Using five-digit arithmetic and Gaussian elimination, the approximate solution \( \tilde{y} \) to the equation \( A\tilde{y} = r \) is
\[ \tilde{y} = (-0.20008, 8.9987 \times 10^{-5}, 0.074607)^t \]
and we have a much better approximation to the system \( Ax = b \),
\[ \tilde{x} + \tilde{y} = (1.2001, 0.99991, 0.92538)^t + (-0.20008, 8.9987 \times 10^{-5}, 0.074607)^t = (1.0000, 1.0000, 0.99999)^t, \]
than we had with the original approximation \( \tilde{x} = (1.2001, 0.99991, 0.92538)^t \). If we were continuing the iteration processes, we would, of course, use \( \tilde{x} + \tilde{y} \) as our starting values rather than \( \tilde{x} \).
EXERCISE SET 7.6

1. Compute the condition numbers of the following matrices relative to $\| \cdot \|_\infty$.

   (a) $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
   (b) $\begin{bmatrix} 3.9 & 1.6 \\ 6.8 & 2.9 \end{bmatrix}$
   (c) $\begin{bmatrix} 1 & 2 \\ 1.0001 & 2 \end{bmatrix}$
   (d) $\begin{bmatrix} 1.003 & 58.09 \\ 5.550 & 321.8 \end{bmatrix}$
   (e) $\begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$
   (f) $\begin{bmatrix} 0.04 & 0.01 & -0.01 \\ 0.2 & 0.5 & -0.2 \\ 1 & 2 & 4 \end{bmatrix}$

2. The following linear systems $Ax = b$ have $x$ as the actual solution and $\tilde{x}$ as an approximate solution. Using the results of Exercise 1, compute $\| x - \tilde{x} \|_\infty$ and

   $K_\infty(A) \frac{\| b - A\tilde{x} \|_\infty}{\| A \|_\infty}$.

   (a) $\frac{1}{2} x_1 + \frac{1}{3} x_2 = \frac{1}{63}$
   $\frac{1}{4} x_1 + \frac{1}{168} x_2 = \frac{1}{63}$
   $x = \left( \frac{1}{7} - \frac{1}{6} \right)^t$
   $\tilde{x} = (0.142, -0.166)^t$

   (b) $3.9 x_1 + 1.6 x_2 = 5.5$
   $6.8 x_3 + 2.9 x_2 = 9.7$
   $x = (1, 1)^t$
   $\tilde{x} = (0.98, 1.1)^t$

   (c) $x_1 + 2 x_2 = 3$
   $1.0001 x_1 + 2 x_2 = 3.0001$
   $x = (1, 1)^t$
   $\tilde{x} = (0.96, 1.02)^t$

   (d) $1.003 x_1 + 58.09 x_2 = 68.12$
   $5.550 x_1 + 321.8 x_2 = 377.3$
   $x = (10, 1)^t$
   $\tilde{x} = (-10, 1)^t$

   (e) $x_1 - x_2 - x_3 = 2\pi$,
   $x_2 - x_3 = 0$,
   $-x_3 = \pi.$
   $x = (0, -\pi, -\pi)^t$
   $\tilde{x} = (-0.1, -3.15, -3.14)^t$

   (f) $0.04 x_1 + 0.01 x_2 - 0.01 x_3 = 0.06$
   $0.2 x_1 + 0.5 x_2 - 0.2 x_3 = 0.3$
   $x_1 + 2 x_2 + 4 x_3 = 11$
   $x = (1.827586, 0.6551724, 1.965517)^t$
   $\tilde{x} = (1.8, 0.64, 1.9)^t$
3. The linear system
\[
\begin{bmatrix}
1 & 2 \\
1.0001 & 2
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
3 \\
3.0001
\end{bmatrix}
\]
has solution \((1, 1)^t\). Change \(A\) slightly to
\[
\begin{bmatrix}
1 & 2 \\
0.9999 & 2
\end{bmatrix}
\]
and consider the linear system
\[
\begin{bmatrix}
1 & 2 \\
0.9999 & 2
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
3 \\
3.0001
\end{bmatrix}
\].
Compute the new solution using five-digit rounding arithmetic, and compare the change in \(A\) to the change in \(x\).

4. The linear system \(Ax = b\) given by
\[
\begin{bmatrix}
1 & 2 \\
1.00001 & 2
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
3 \\
3.00001
\end{bmatrix}
\]
has solution \((1, 1)^t\). Use seven-digit rounding arithmetic to find the solution of the perturbed system
\[
\begin{bmatrix}
1 & 2 \\
1.000011 & 2
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
3.00001 \\
3.00003
\end{bmatrix}
\],
and compare the change in \(A\) and \(b\) to the change in \(x\).

5. (i) Use Gaussian elimination and three-digit rounding arithmetic to approximate the solutions to the following linear systems. (ii) Then use one iteration of iterative refinement to improve the approximation, and compare the approximations to the actual solutions.

(a) \(0.03x_1 + 58.9x_2 = 59.2\)
\[5.31x_1 - 6.10x_2 = 47.0\]
Actual solution \((10, 1)^t\).

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

(c) \(1.19x_1 + 2.11x_2 - 100x_3 + x_4 = 1.12\)
\(14.2x_1 - 0.122x_2 + 12.2x_3 - x_4 = 3.44\)
\(100x_2 - 99.9x_3 + x_4 = 2.15\)
\(15.3x_1 + 0.110x_2 - 13.1x_3 - x_4 = 4.16\)
Actual solution \((0.17682530, 0.01269269, -0.02065405, -1.18260870)^t\).
7.6. ERROR BOUNDS AND ITERATIVE REFINEMENT

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

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

7. (a) Use Maple with Digits set to 7 to solve the following linear system using Gaussian elimination.
\[
\begin{array}{ccc}
\frac{1}{3} x_1 - \frac{1}{3} x_2 - \frac{1}{3} x_3 - \frac{1}{3} x_4 - \frac{1}{3} x_5 & = & 1 \\
\frac{1}{3} x_2 - \frac{1}{3} x_3 - \frac{1}{3} x_4 - \frac{1}{3} x_5 & = & 0 \\
\frac{1}{3} x_3 - \frac{1}{3} x_4 - \frac{1}{3} x_5 & = & -1 \\
\frac{1}{3} x_4 - \frac{1}{3} x_5 & = & 2 \\
\frac{1}{3} x_5 & = & 7 \\
\end{array}
\]

(b) Compute the condition number of the matrix for the system relative to \(\| \cdot \|_{\infty}\).

(c) Find the exact solution to the linear system.

8. The \(n \times n\) Hilbert matrix, \(H^{(n)}\), defined by
\[ H_{ij}^{(n)} = \frac{1}{i + j - 1}, \quad 1 \leq i, j \leq n \]
is an ill-conditioned matrix that arises in solving for the coefficients of least squares polynomials (see Section 8.3).

(a) Show that
\[
[H^{(4)}]^{-1} = \begin{bmatrix}
16 & -120 & 240 & -140 \\
-120 & 1200 & -2700 & 1680 \\
240 & -2700 & 6480 & -4200 \\
-140 & 1680 & -4200 & 2800 \\
\end{bmatrix},
\]
and compute \(K_\infty(H^{(4)})\).

(b) Show that
\[
[H^{(5)}]^{-1} = \begin{bmatrix}
25 & -300 & 1050 & -1400 & 630 \\
-300 & 4800 & -18900 & 26880 & -12600 \\
1050 & -18900 & 79380 & -117600 & 56700 \\
-1400 & 26880 & -117600 & 179200 & -88200 \\
630 & -12600 & 56700 & -88200 & 44100 \\
\end{bmatrix},
\]
and compute $K_\infty(H^{(5)})$.

(c) Solve the linear system

\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\end{bmatrix}
= \begin{bmatrix}
1 \\
0 \\
0 \\
1 \\
\end{bmatrix}
\]

using Maple with \texttt{Digits} set to 3, and compare the actual error to the residual vector error bound.

9.  (a) Use Maple with \texttt{Digits} set to 4 to compute the inverse $H^{-1}$ of the $3 \times 3$ Hilbert matrix $H$.

(b) Use Maple with \texttt{Digits} set to 4 to compute $\hat{H} = (H^{-1})^{-1}$.

(c) Determine $\|H - \hat{H}\|_\infty$. 
7.7. The Conjugate Gradient Method

The conjugate gradient method of Hestenes and Stiefel [HS] was originally developed as a direct method designed to solve an $n \times n$ positive definite linear system. As a direct method it is generally inferior to Gaussian elimination with pivoting since both methods require $n$ major steps to determine a solution, and the steps of the conjugate gradient method are more computationally expensive than those in Gaussian elimination.

However, the conjugate gradient method is very useful when employed as an iterative approximation method for solving large sparse systems with nonzero entries occurring in predictable patterns. These problems frequently arise in the solution of boundary-value problems. When the matrix has been preconditioned to make the calculations more effective, good results are obtained in only about $\sqrt{n}$ steps. Employed in this way, the method is preferred over Gaussian elimination and the previously-discussed iterative methods.

Throughout this section we assume that the matrix $A$ is positive definite. We will use the inner product notation

$$\langle x, y \rangle = x^t y,$$  \hspace{1cm} (7.3)

where $x$ and $y$ are $n$-dimensional vectors. We will also need some additional standard results from linear algebra. A review of this material is found in Section 9.2.

The following result follows easily from the properties of transposes.

<table>
<thead>
<tr>
<th>Inner Product Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>For any vectors $x$, $y$, and $z$ and any real number $\alpha$, we have</td>
</tr>
<tr>
<td>(i) $\langle x, y \rangle = \langle y, x \rangle$;</td>
</tr>
<tr>
<td>(ii) $\langle \alpha x, y \rangle = \langle x, \alpha y \rangle = \alpha \langle x, y \rangle$;</td>
</tr>
<tr>
<td>(iii) $\langle x + z, y \rangle = \langle x, y \rangle + \langle z, y \rangle$;</td>
</tr>
<tr>
<td>(iv) $\langle x, x \rangle \geq 0$;</td>
</tr>
<tr>
<td>(v) $\langle x, x \rangle = 0$ if and only if $x = 0$.</td>
</tr>
</tbody>
</table>

When $A$ is positive definite, $\langle x, Ax \rangle = x^t A x > 0$ unless $x = 0$. Also, since $A$ is symmetric, we have $x^t A y = x^t A^t y = (Ax)^t y$, so in addition to the inner product results, we have for each $x$ and $y$,

$$\langle x, A y \rangle = \langle A x, y \rangle.$$  \hspace{1cm} (7.4)

The following result is a basic tool in the development of the conjugate gradient method.
CHAPTER 7. ITERATIVE METHODS FOR SOLVING LINEAR SYSTEMS

[Minimization Condition]
The vector $x^*$ is a solution to the positive definite linear system $Ax = b$ if and only if $x^*$ minimizes

$$g(x) = \langle x, Ax \rangle - 2\langle x, b \rangle.$$

In addition, for any $x$ and $v \neq 0$ the function $g(x + tv)$ has its minimum when $t = \langle v, b - Ax \rangle / \langle v, Av \rangle$.

To begin the conjugate gradient method, we choose $x$, an approximate solution to $Ax^* = b$, and $v \neq 0$, which gives a search direction in which to move away from $x$ to improve the approximation. Let $r = b - Ax$ be the residual vector associated with $x$ and

$$t = \frac{\langle v, b - Ax \rangle}{\langle v, Av \rangle} = \frac{\langle v, r \rangle}{\langle v, Av \rangle}.$$

If $r \neq 0$ and if $v$ and $r$ are not orthogonal, then $x + tv$ gives a smaller value for $g$ than $g(x)$ and is presumably closer to $x^*$ than is $x$. This suggests the following method.

Let $x^{(0)}$ be an initial approximation to $x^*$, and let $v^{(1)} \neq 0$ be an initial search direction. For $k = 1, 2, 3, \ldots$, we compute

$$t_k = \frac{\langle v^{(k)}, b - Ax^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle},$$

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}$$

and choose a new search direction $v^{(k+1)}$. The object is to make this selection so that the sequence of approximations $\{x^{(k)}\}$ converges rapidly to $x^*$.

To choose the search directions, we view $g$ as a function of the components of $x = (x_1, x_2, \ldots, x_n)^t$. Thus,

$$g(x_1, x_2, \ldots, x_n) = \langle x, Ax \rangle - 2\langle x, b \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}x_i x_j - 2 \sum_{i=1}^{n} x_i b_i.$$

Taking partial derivatives with respect to the component variables $x_k$ gives

$$\frac{\partial g}{\partial x_k}(x) = 2 \sum_{i=1}^{n} a_{ki} x_i - 2b_k.$$

Therefore, the gradient of $g$ is

$$\nabla g(x) = \left( \frac{\partial g}{\partial x_1}(x), \frac{\partial g}{\partial x_2}(x), \ldots, \frac{\partial g}{\partial x_n}(x) \right)^t = 2(Ax - b) = -2r,$$

where the vector $r$ is the residual vector for $x$. 
From multivariable calculus, we know that the direction of greatest decrease in the value of \( g(x) \) is the direction given by \( -\nabla g(x) \); that is, in the direction of the residual \( r \). The method that chooses \( v^{(k+1)} = r^{(k)} = b - Ax^{(k)} \) is called the method of steepest descent. Although we will see in Section 10.4 that this method has merit for nonlinear systems and optimization problems, it is not used for linear systems because of slow convergence.

An alternative approach uses a set of nonzero direction vectors \( \{v^{(1)}, \ldots, v^{(n)}\} \) that satisfy
\[
\langle v^{(i)}, Av^{(j)} \rangle = 0, \quad \text{if} \quad i \neq j.
\]
This is called an \( A \)-orthogonality condition, and the set of vectors \( \{v^{(1)}, \ldots, v^{(n)}\} \) is said to be \( A \)-orthogonal. It is not difficult to show that a set of \( A \)-orthogonal vectors associated with the positive definite matrix \( A \) is linearly independent. (See Exercise 13(a).) This set of search directions gives
\[
t_k = \frac{\langle v^{(k)}, b - Ax^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle} = \frac{\langle v^{(k)}, r^{(k-1)} \rangle}{\langle v^{(k)}, Av^{(k)} \rangle},
\]
and \( x^{(k)} = x^{(k-1)} + t_k v^{(k)} \).

The following result shows that this choice of search directions gives convergence in at most \( n \)-steps, so as a direct method it produces the exact solution, assuming that the arithmetic is exact.

---

**EXAMPLE 1** Consider the positive definite matrix
\[
A = \begin{bmatrix}
4 & 3 & 0 \\
3 & 4 & -1 \\
0 & -1 & 4
\end{bmatrix}.
\]

Let \( v^{(1)} = (1, 0, 0)^t \), \( v^{(2)} = (-3/4, 1, 0)^t \), and \( v^{(3)} = (-3/7, 4/7, 1)^t \). By direct
calculation,
\[ \langle v^{(1)}, Av^{(2)} \rangle = v^{(1)^t}A v^{(2)} = (1, 0, 0) \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} -\frac{3}{7} \\ 1 \\ 0 \end{bmatrix} = 0, \]
\[ \langle v^{(1)}, Av^{(3)} \rangle = (1, 0, 0) \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} -\frac{2}{7} \\ \frac{4}{7} \\ 1 \end{bmatrix} = 0, \]
and
\[ \langle v^{(2)}, Av^{(3)} \rangle = \left( -\frac{3}{4}, 1, 0 \right) \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} -\frac{2}{7} \\ \frac{4}{7} \\ 1 \end{bmatrix} = 0. \]
Thus, \( \{ v^{(1)}, v^{(2)}, v^{(3)} \} \) is an \( A \)-orthogonal set.

The linear system
\[ \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ 30 \\ -24 \end{bmatrix}, \]
has the exact solution \( x^* = (3, 4, -5)^t \). To approximate this solution, let \( x^{(0)} = (0, 0, 0)^t \). Since \( b = (24, 30, -24)^t \), we have
\[ r^{(0)} = b - Ax^{(0)} = b = (24, 30, -24)^t, \]
so
\[ \langle v^{(1)}, r^{(0)} \rangle = v^{(1)^t}r^{(0)} = 24, \quad \langle v^{(1)}, Av^{(1)} \rangle = 4, \quad \text{and} \quad t_0 = \frac{24}{4} = 6. \]
Thus,
\[ x^{(1)} = x^{(0)} + t_0 v^{(1)} = (0, 0, 0)^t + 6(1, 0, 0)^t = (6, 0, 0)^t. \]
Continuing, we have
\[ r^{(1)} = b - Ax^{(1)} = (0, 12, -24)^t; \quad t_1 = \frac{\langle v^{(2)}, r^{(1)} \rangle}{\langle v^{(2)}, Av^{(2)} \rangle} = \frac{12}{\frac{7}{4}} = \frac{48}{7}; \]
\[ x^{(2)} = x^{(1)} + t_1 v^{(2)} = (6, 0, 0)^t + \frac{48}{7} \left( -\frac{3}{4}, 1, 0 \right)^t = \left( \frac{6}{7}, \frac{48}{7}, 0 \right)^t; \]
\[ r^{(2)} = b - Ax^{(2)} = \left( 0, 0, -\frac{120}{7} \right)^t; \quad t_2 = \frac{\langle v^{(3)}, r^{(2)} \rangle}{\langle v^{(3)}, Av^{(3)} \rangle} = -\frac{120/7}{24/7} = -5; \]
and
\[ x^{(3)} = x^{(2)} + t_2 v^{(3)} = \left( \frac{6}{7}, \frac{48}{7}, 0 \right)^t + (-5) \left( -\frac{3}{7}, \frac{4}{7}, 1 \right)^t = (3, 4, -5)^t. \]
Since we applied the technique \( n = 3 \) times, this is the actual solution.
7.7. THE CONJUGATE GRADIENT METHOD

Before discussing how to determine the \( A \)-orthogonal set, we will continue the development. The use of an \( A \)-orthogonal set \( \{v^{(1)}, \ldots, v^{(n)}\} \) of direction vectors gives what is called a conjugate direction method. The following result concerns the orthogonality of the residual vectors \( r^{(k)} \) and the direction vectors \( v^{(j)} \).

[Orthogonal Residual Vectors]
The residual vectors \( r^{(k)} \), where \( k = 1, 2, \ldots, n \), for a conjugate direction method, satisfy the equations

\[
\langle r^{(k)}, v^{(j)} \rangle = 0, \quad \text{for each } j = 1, 2, \ldots, k.
\]

The conjugate gradient method of Hestenes and Stiefel chooses the search directions \( \{v^{(k)}\} \) during the iterative process so that the residual vectors \( \{r^{(k)}\} \) are mutually orthogonal. To construct the direction vectors \( \{v^{(1)}, v^{(2)}, \ldots\} \) and the approximations \( \{x^{(1)}, x^{(2)}, \ldots\} \), we start with an initial approximation \( x^{(0)} \) and use the steepest descent direction \( r^{(0)} = b - Ax^{(0)} \) as the first search direction \( v^{(1)} \).

Assume that the conjugate directions \( v^{(1)}, \ldots, v^{(k-1)} \) and the approximations \( x^{(1)}, \ldots, x^{(k-1)} \) have been computed with

\[
x^{(k-1)} = x^{(k-2)} + t_{k-1} v^{(k-1)},
\]

where

\[
\langle v^{(i)}, Av^{(j)} \rangle = 0 \quad \text{and} \quad \langle r^{(i)}, r^{(j)} \rangle = 0, \quad \text{for } i \neq j.
\]

If \( x^{(k-1)} \) is the solution to \( Ax = b \), we are done. Otherwise, \( r^{(k-1)} = b - Ax^{(k-1)} \neq 0 \) and the orthogonality implies that \( \langle r^{(k-1)}, v^{(i)} \rangle = 0, \) for \( i = 1, 2, \ldots, k-1 \). We then use \( r^{(k-1)} \) to generate \( v^{(k)} \) by setting

\[
v^{(k)} = r^{(k-1)} + s_{k-1} v^{(k-1)}.
\]

We want to choose \( s_{k-1} \) so that

\[
\langle v^{(k-1)}, Av^{(k)} \rangle = 0.
\]

Since

\[
Av^{(k)} = Ar^{(k-1)} + s_{k-1} Av^{(k-1)}
\]

and

\[
\langle v^{(k-1)}, Av^{(k)} \rangle = \langle v^{(k-1)}, Ar^{(k-1)} \rangle + s_{k-1} \langle v^{(k-1)}, Av^{(k-1)} \rangle,
\]

we will have \( \langle v^{(k-1)}, Av^{(k)} \rangle = 0 \) when

\[
s_{k-1} = -\frac{\langle v^{(k-1)}, Ar^{(k-1)} \rangle}{\langle v^{(k-1)}, Av^{(k-1)} \rangle}.
\]

It can also be shown that with this choice of \( s_{k-1} \) we have \( \langle v^{(k)}, Av^{(i)} \rangle = 0 \), for each \( i = 1, 2, \ldots, k-2 \). Thus, \( \{v^{(1)}, \ldots, v^{(k)}\} \) is an \( A \)-orthogonal set.
Having chosen \( \mathbf{v}^{(k)} \), we compute

\[
 t_k = \frac{\langle \mathbf{v}^{(k)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle} = \frac{\langle \mathbf{r}^{(k-1)} + s_{k-1} \mathbf{v}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle}
 = \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{v}^{(k)} \mathbf{v}^{(k)} \rangle + s_{k-1} \langle \mathbf{v}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle}.
\]

By the orthogonality result, \( \langle \mathbf{v}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle = 0 \), so

\[
 t_k = \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle}. \quad (7.5)
\]

Thus,

\[
 \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k \mathbf{v}^{(k)}.
\]

To compute \( \mathbf{r}^{(k)} \), we multiply by \( A \) and subtract \( \mathbf{b} \) to obtain

\[
 A \mathbf{x}^{(k)} - \mathbf{b} = A \mathbf{x}^{(k-1)} - \mathbf{b} + t_k A \mathbf{v}^{(k)}
\]

or

\[
 \mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - t_k A \mathbf{v}^{(k)}.
\]

Thus,

\[
 \langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle = \langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle - t_k \langle \mathbf{A} \mathbf{v}^{(k)}, \mathbf{r}^{(k)} \rangle = -t_k \langle \mathbf{r}^{(k)}, A \mathbf{v}^{(k)} \rangle.
\]

Further, from Eq. (7.5),

\[
 \langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle = t_k \langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle,
\]

so

\[
 s_k = \frac{\langle \mathbf{v}^{(k)}, A \mathbf{r}^{(k)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle} = \frac{\langle \mathbf{r}^{(k)}, A \mathbf{v}^{(k)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle}
 = \frac{(1/t_k) \langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{(1/t_k) \langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle} = \frac{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}.
\]

In summary, we have the formulas:

\[
 \mathbf{r}^{(0)} = \mathbf{b} - A \mathbf{x}^{(0)}; \quad \mathbf{v}^{(1)} = \mathbf{r}^{(0)};
\]

and, for \( k = 1, 2, \ldots, n \),

\[
 t_k = \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A \mathbf{v}^{(k)} \rangle},
 \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k \mathbf{v}^{(k)},
 \mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - t_k A \mathbf{v}^{(k)},
 s_k = \frac{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle},
 \mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} + s_k \mathbf{v}^{(k)}.
\]
We will now extend the conjugate gradient method to include preconditioning. If the matrix $A$ is ill-conditioned, the conjugate gradient method is highly susceptible to rounding errors. So, although the exact answer should be obtained in $n$ steps, this is not usually the case. As a direct method the conjugate gradient method is not as good as Gaussian elimination with pivoting. The main use of the conjugate gradient method is as an iterative method applied to a better-conditioned system. In this case an acceptable approximate solution is often obtained in about $\sqrt{n}$ steps.

To apply the method to a better-conditioned system, we want to select a non-singular conditioning matrix $C$ so that

$$\tilde{A} = C^{-1} A (C^{-1})^t$$

is better conditioned. To simplify the notation, we will use the matrix $C^{-1}$ to refer to $(C^{-1})^t$.

Consider the linear system

$$\tilde{A} \tilde{x} = \tilde{b},$$

where $\tilde{x} = C^t x$ and $\tilde{b} = C^{-1} b$. Then

$$\tilde{A} \tilde{x} = (C^{-1} A C^{-t}) (C^t x) = C^{-1} A x.$$

Thus, we could solve $\tilde{A} \tilde{x} = \tilde{b}$ for $\tilde{x}$ and then obtain $x$ by multiplying by $C^{-t}$. However, instead of rewriting equations (7.6) using $\tilde{r}^{(k)}$, $\tilde{v}^{(k)}$, $\tilde{t}^{(k)}$, $\tilde{x}^{(k)}$, and $\tilde{s}^{(k)}$, we incorporate the preconditioning implicitly.

Since

$$\tilde{x}^{(k)} = C^t x^{(k)},$$

we have

$$\tilde{r}^{(k)} = \tilde{b} - \tilde{A} \tilde{x}^{(k)} = C^{-1} b - (C^{-1} A C^{-t}) C^t x^{(k)} = C^{-1} (b - A x^{(k)}) = C^{-1} r^{(k)}.$$

Let $\tilde{v}^{(k)} = C^t v^{(k)}$ and $w^{(k)} = C^{-1} r^{(k)}$. Then

$$\tilde{s}^{(k)} = \frac{\langle \tilde{r}^{(k)}, \tilde{v}^{(k)} \rangle}{\langle \tilde{v}^{(k-1)}, \tilde{v}^{(k-1)} \rangle} = \frac{\langle C^{-1} r^{(k)}, C^{-1} r^{(k)} \rangle}{\langle C^{-1} r^{(k-1)}, C^{-1} r^{(k-1)} \rangle},$$

so

$$\tilde{s}^{(k)} = \frac{\langle w^{(k)}, w^{(k)} \rangle}{\langle w^{(k-1)}, w^{(k-1)} \rangle}. \tag{7.7}$$

Thus,

$$\tilde{t}^{(k)} = \frac{\langle \tilde{r}^{(k-1)}, \tilde{v}^{(k-1)} \rangle}{\langle \tilde{v}^{(k)}, A \tilde{v}^{(k)} \rangle} = \frac{\langle C^{-1} r^{(k-1)}, C^{-1} r^{(k-1)} \rangle}{\langle C^t v^{(k)}, C^{-1} A C^{-t} C^t v^{(k)} \rangle} = \frac{\langle w^{(k-1)}, w^{(k-1)} \rangle}{\langle C^t v^{(k)}, C^{-1} A v^{(k)} \rangle},$$

and

$$\tilde{t}^{(k)} = \frac{\langle w^{(k-1)}, w^{(k-1)} \rangle}{\langle v^{(k)}, A v^{(k)} \rangle}. \tag{7.8}$$

Further,

$$\tilde{x}^{(k)} = \tilde{x}^{(k-1)} + \tilde{t}^{(k)} \tilde{v}^{(k)}, \quad \text{so} \quad C^t \tilde{x}^{(k)} = C^t \tilde{x}^{(k-1)} + \tilde{t}^{(k)} C^t \tilde{v}^{(k)}.$$
and

\[ x^{(k)} = x^{(k-1)} + \tilde{t}_k v^{(k)}. \]  

(7.9)

Continuing,

\[ \tilde{r}^{(k)} = \tilde{r}^{(k-1)} - \tilde{t}_k \tilde{A} \tilde{v}^{(k)}, \]

so

\[ C^{-1} r^{(k)} = C^{-1} r^{(k-1)} - \tilde{t}_k C^{-1} A C^{-t} \tilde{v}^{(k)}, \quad r^{(k)} = r^{(k-1)} - \tilde{t}_k A C^{-t} C^t v^{(k)}, \]

and

\[ r^{(k)} = r^{(k-1)} - \tilde{t}_k A v^{(k)}. \]  

(7.10)

Finally,

\[ \tilde{v}^{(k+1)} = \tilde{r}^{(k)} + \tilde{s}_k \tilde{v}^{(k)} \quad \text{and} \quad C^t v^{(k+1)} = C^{-1} r^{(k)} + \tilde{s}_k C^t v^{(k)}, \]

so

\[ v^{(k+1)} = C^{-t} C^{-1} r^{(k)} + \tilde{s}_k v^{(k)} = C^{-t} w^{(k)} + \tilde{s}_k v^{(k)}. \]  

(7.11)

The preconditioned conjugate gradient method is based on using equations (7.7)–(11) in the order (7.8), (7.9), (7.10), (7.7), and (7.11). The next example illustrates the calculations in an easy problem.

**EXAMPLE 2** The linear system \( Ax = b \) given by

\[
\begin{align*}
4x_1 + 3x_2 &= 24, \\
3x_1 + 4x_2 - x_3 &= 30, \\
-2x_2 + 4x_3 &= -24
\end{align*}
\]

has solution \((3, 4, -5)^t\) and was considered in Example 1 of Section 7.5. In that example, both the Gauss-Seidel method and SOR method were used. We will use the conjugate gradient method with no preconditioning, so \(C = C^{-1} = I\). Let \(x^{(0)} = (0, 0, 0)^t\). Then

\[
\begin{align*}
r^{(0)} &= b - Ax^{(0)} = b = (24, 30, -24)^t; \\
w &= C^{-1} r^{(0)} = (24, 30, -24)^t; \\
v^{(1)} &= C^{-t} w = (24, 30, -24)^t; \\
\alpha &= \langle w, w \rangle = 2052.
\end{align*}
\]
We start the first iteration with \( k = 1 \). Then
\[
\begin{align*}
\mathbf{u} &= A\mathbf{v}^{(1)} = (186.0, 216.0, -126.0)^t; \\
\alpha &= \frac{\mathbf{v}^{(1)} \mathbf{u}}{\mathbf{v}^{(1)} \mathbf{u}} = 0.1469072165; \\
\mathbf{x}^{(1)} &= \mathbf{x}^{(0)} + t_1 \mathbf{v}^{(1)} = (3.525773196, 4.407216495, -3.525773196)^t; \\
\mathbf{r}^{(1)} &= \mathbf{r}^{(0)} - t_1 \mathbf{u} = (-3.32474227, -1.73195876, -5.4869072)^t; \\
\mathbf{w} &= C^{-1}\mathbf{r}^{(1)} = \mathbf{r}^{(1)}; \\
\beta &= \langle \mathbf{w}, \mathbf{w} \rangle = 44.19029651; \\
s_1 &= \frac{\beta}{\alpha} = 0.0215352322; \\
\mathbf{v}^{(2)} &= C^{-1} \mathbf{w} + s_1 \mathbf{v}^{(1)} = (-2.807896697, -1.085901793, -6.006536293)^t).
\end{align*}
\]
Set
\( \alpha = \beta = 44.19029651 \).

We are now ready to begin the second iteration. We have
\[
\begin{align*}
\mathbf{u} &= A\mathbf{v}^{(2)} = (-14.48929217, -6.760760967, -22.94024338)^t; \\
t_2 &= 0.2378157558; \\
\mathbf{x}^{(2)} &= (2.858011121, 4.148971939, -4.95422164)^t; \\
\mathbf{r}^{(2)} &= (0.121039698, -0.124143281, -0.034139402)^t; \\
\mathbf{w} &= C^{-1}\mathbf{r}^{(2)} = \mathbf{r}^{(2)}; \\
\beta &= 0.03122766148; \\
s_2 &= 0.0007066633163; \\
\mathbf{v}^{(3)} &= (0.1190554504, -0.1249106480, -0.03838400086)^t).
\end{align*}
\]
Set
\( \alpha = \beta = 0.03122766148 \).

Finally, the third iteration gives
\[
\begin{align*}
\mathbf{u} &= A\mathbf{v}^{(3)} = (0.1014898976, -0.1040922099, -0.0286253554)^t; \\
t_3 &= 1.192628008; \\
\mathbf{x}^{(3)} &= (2.999999998, 4.000000002, -4.999999998)^t; \\
\mathbf{r}^{(3)} &= (0.36 \times 10^{-8}, 0.39 \times 10^{-8}, -0.141 \times 10^{-8})^t.
\end{align*}
\]

Since \( \mathbf{x}^{(3)} \) is nearly the exact solution, rounding error did not significantly effect the result. In Example 1 of Section 7.5, the Gauss-Seidel method required 34 iterations, and the SOR method, with \( \omega = 1.25 \), required 14 iterations for an accuracy of \( 10^{-7} \). It should be noted, however, that in this example, we are really comparing a direct method to iterative methods.
The next example illustrates the effect of preconditioning on a poorly conditioned matrix. In this example and subsequently, we use $D^{-1/2}$ to represent the diagonal matrix whose entries are the reciprocals of the square roots of the diagonal entries of the coefficient matrix $A$.

**EXAMPLE 3** The linear system $Ax = b$ with

$$A = \begin{bmatrix} 0.2 & 0.1 & 1 & 1 & 0 \\ 0.1 & 4 & -1 & 1 & -1 \\ 1 & -1 & 60 & 0 & -2 \\ 1 & 1 & 0 & 8 & 4 \\ 0 & -1 & -2 & 4 & 700 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

has the solution $x^* = (7.859713071, 0.4229264082, -0.0735923906, -0.5406430164, 0.0106216286)^t$.

The matrix $A$ is symmetric and positive definite but is ill-conditioned with condition number $K_\infty(A) = 13961.71$. We will use tolerance 0.01 and compare the results obtained from the Jacobi, Gauss-Seidel, and SOR (with $\omega = 1.25$) iterative methods and from the conjugate gradient method with $C^{-1} = I$. Then we precondition by choosing $C^{-1}$ as $D^{-1/2}$, the diagonal matrix whose diagonal entries are the reciprocal of the positive square roots of the diagonal entries of the positive definite matrix $A$. The results are presented in Table 7.5. The preconditioned conjugate gradient method gives the most accurate approximation with the smallest number of iterations.

### Table 7.5

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Iterations</th>
<th>$x^{(k)}$</th>
<th>$|x^* - x^{(k)}|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi</td>
<td>49</td>
<td>$(7.86277141, 0.42320802, -0.07348669, -0.53975964, 0.01062847)^t$</td>
<td>0.00305834</td>
</tr>
<tr>
<td>Gauss-Seidel</td>
<td>15</td>
<td>$(7.83525748, 0.42257868, -0.07319124, -0.53753055, 0.01060903)^t$</td>
<td>0.02445559</td>
</tr>
<tr>
<td>SOR ($\omega = 1.25$)</td>
<td>7</td>
<td>$(7.85152706, 0.42277371, -0.07348303, -0.53978369, 0.01062286)^t$</td>
<td>0.00818607</td>
</tr>
<tr>
<td>Conjugate Gradient</td>
<td>5</td>
<td>$(7.85341523, 0.42298677, -0.07347963, -0.53987920, 0.008628916)^t$</td>
<td>0.00629785</td>
</tr>
<tr>
<td>Conjugate Gradient</td>
<td>4</td>
<td>$(7.85968827, 0.42288329, -0.07359878, -0.54063200, 0.01064344)^t$</td>
<td>0.00009312</td>
</tr>
</tbody>
</table>

The preconditioned conjugate gradient method is often used in the solution of large linear systems in which the matrix is sparse and positive definite. These
systems must be solved to approximate solutions to boundary-value problems in ordinary-differential equations (Sections 11.3, 11.5, 11.6). The larger the system, the more impressive the conjugate gradient method becomes since it significantly reduces the number of iterations required. In these systems, the preconditioning matrix $C$ is approximately equal to $L$ in the Choleski factorization $LL^t$ of $A$. Generally, small entries in $A$ are ignored and Choleski’s method is applied to obtain what is called an incomplete $LL^t$ factorization of $A$. Thus, $C^{-1}C^{-1} \approx A^{-1}$ and a good approximation is obtained. More information about the conjugate gradient method can be found in Kelley [Kelley].
EXERCISE SET 7.7

1. The linear system

\[
\begin{align*}
    x_1 + \frac{1}{2}x_2 &= 5, \\
    \frac{1}{2}x_1 + \frac{1}{3}x_2 &= \frac{11}{84}
\end{align*}
\]

has solution \((x_1, x_2)^t = (1/6, 1/7)^t\).

(a) Solve the linear system using Gaussian elimination with two-digit rounding arithmetic.

(b) Solve the linear system using the conjugate gradient method \((C = C^{-1} = I)\) with two-digit rounding arithmetic.

(c) Which method gives the better answer?

(d) Choose \(C^{-1} = D^{-1/2}\). Does this choice improve the conjugate gradient method?

2. The linear system

\[
\begin{align*}
    0.1x_1 + 0.2x_2 &= 0.3, \\
    0.2x_1 + 113x_2 &= 113.2
\end{align*}
\]

has solution \((x_1, x_2)^t = (1, 1)^t\). Repeat the directions for Exercise 1 on this linear system.

3. The linear system

\[
\begin{align*}
    x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 &= \frac{5}{6}, \\
    \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 &= \frac{5}{12}, \\
    \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 &= \frac{17}{60}
\end{align*}
\]

has solution \((1, -1, 1)^t\).

(a) Solve the linear system using Gaussian elimination with three-digit rounding arithmetic.

(b) Solve the linear system using the conjugate gradient method with three-digit rounding arithmetic.

(c) Does pivoting improve the answer in (a)?

(d) Repeat part (b) using \(C^{-1} = D^{-1/2}\). Does this improve the answer in (b)?
4. Repeat Exercise 3 using single-precision arithmetic on a computer.

5. Perform only two steps of the conjugate gradient method with $C = C^{-1} = I$ on each of the following linear systems. Compare the results to those obtained in Exercises 1 and 2 of Section 7.4 and Exercise 1 of Section 7.5.

   (a) $3x_1 - x_2 + x_3 = 1,$
       $-x_1 + 6x_2 + 2x_3 = 0,$
       $x_1 + 2x_2 + 7x_3 = 4.$

   (b) $10x_1 - x_2 = 9,$
       $-x_1 + 10x_2 - 2x_3 = 7,$
       $-2x_2 + 10x_3 = 6.$

   (c) $10x_1 + 5x_2 = 6,$
       $5x_1 + 10x_2 - 4x_3 = 25,$
       $-4x_2 + 8x_3 - x_4 = -11,$
       $-x_3 + 5x_4 = -11.$

   (d) $4x_1 + x_2 - x_3 + x_4 = -2,$
       $x_1 + 4x_2 - x_3 - x_4 = -1,$
       $-x_1 - x_2 + 5x_3 + x_4 = 0,$
       $x_1 - x_2 + x_3 + 3x_4 = 1.$

   (e) $4x_1 + x_2 + x_3 + x_5 = 6,$
       $x_1 + 3x_2 + x_3 + x_4 = 6,$
       $x_1 + x_2 + 5x_3 - x_4 - x_5 = 6,$
       $x_2 - x_3 + 4x_4 = 6,$
       $x_1 - x_3 + 4x_5 = 6.$

   (f) $4x_1 - x_2 - x_4 = 0,$
       $-x_1 + 4x_2 - x_3 - x_5 = 5,$
       $-x_2 + 4x_3 - x_6 = 0,$
       $-x_1 + 4x_4 - x_5 = 6,$
       $-x_2 - x_4 + 4x_5 - x_6 = -2,$
       $-x_3 - x_5 + 4x_6 = 6.$

6. Repeat Exercise 5 using $C^{-1} = D^{-1/2}.$

7. Repeat Exercise 5 with $TOL = 10^{-3}$ in the $l_\infty$ norm. Compare the results to those obtained in Exercises 3 and 4 of Section 7.4 and Exercise 2 of Section 7.5.

8. Repeat Exercise 7 using $C^{-1} = D^{-1/2}.$
9. Use (i) the Jacobi Method, (ii) the Gauss-Seidel method, (iii) the SOR method with $\omega = 1.3$, and (iv) the conjugate gradient method and preconditioning with $C^{-1} = D^{-1/2}$ to find solutions to the linear system $Ax = b$ to within $10^{-5}$ in the $l_{\infty}$ norm.

(a)

$$a_{i,j} = \begin{cases} 
4, & \text{when } j = i \text{ and } i = 1,2, \ldots, 16, \\
-1, & \text{when } j = i + 1 \text{ and } i = 1,2,3,5,6,7,9,10,11,13,14,15, \\
& j = i - 1 \text{ and } i = 2,3,4,6,7,8,10,11,12,14,15,16, \\
& j = i + 4 \text{ and } i = 1,2, \ldots, 12, \\
& j = i - 4 \text{ and } i = 5,6, \ldots, 16, \\
0, & \text{otherwise}
\end{cases}$$

and

$$b = (1.902207, 1.051143, 1.175689, 3.480083, 0.819600, -0.264419, \\
-0.412789, 1.175689, 0.913337, -0.150209, -0.264419, 1.051143, \\
1.966694, 0.913337, 0.819600, 1.902207)^t$$

(b)

$$a_{i,j} = \begin{cases} 
4, & \text{when } j = i \text{ and } i = 1,2, \ldots, 25, \\
-1, & \text{when } j = i + 1 \text{ and } i = \{1,2,3,4,6,7,8,9,11,12,13,14, \\
& 16,17,18,19,21,22,23,24, \\
& j = i - 1 \text{ and } i = \{2,3,4,5,7,8,9,10,12,13,14,15, \\
& 17,18,19,20,22,23,24,25, \\
& j = i + 5 \text{ and } i = 1,2, \ldots, 20, \\
& j = i - 5 \text{ and } i = 6,7, \ldots, 25, \\
0, & \text{otherwise}
\end{cases}$$

and

$$b = (1,0,-1,0,2,1,0,-1,0,2,1,0,-1,0,2,1,0,-1,0,2,1,0,-1,0,2)^t$$

(c)

$$a_{i,j} = \begin{cases} 
2i, & \text{when } j = i \text{ and } i = 1,2, \ldots, 40, \\
-1, & \text{when } j = i + 1 \text{ and } i = 1,2, \ldots, 39, \\
& j = i - 1 \text{ and } i = 2,3, \ldots, 40, \\
0, & \text{otherwise}
\end{cases}$$

and $b_i = 1.5i - 6$, for each $i = 1,2, \ldots, 40$
10. Solve the linear system in Exercise 4(b) of Section 7.5 using the conjugate gradient method with $C^{-1} = I$.

11. Let

$$A_1 = \begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix}, \quad -I = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

and $0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

Form the $16 \times 16$ matrix $A$ in partitioned form,

$$A = \begin{bmatrix} A_1 & -I & 0 & 0 \\ -I & A_1 & -I & 0 \\ 0 & -I & A_1 & -I \\ 0 & 0 & -I & A_1 \end{bmatrix}.$$

Let $b = (1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6)^t$.

(a) Solve $Ax = b$ using the conjugate gradient method with tolerance 0.05.

(b) Solve $Ax = b$ using the preconditioned conjugate gradient method with $C^{-1} = D^{-1/2}$ and tolerance 0.05.

(c) Is there any tolerance for which the methods of part (a) and part (b) require a different number of iterations?

12. Use the transpose properties given in Section 6.4 to show the Inner Product Properties given on the opening page of the Section.

13. (a) Show that an $A$-orthogonal set of nonzero vectors associated with a positive definite matrix is linearly independent.

(b) Show that if $\{v^{(1)}, v^{(2)}, \ldots, v^{(n)}\}$ is a set of $A$-orthogonal nonzero vectors in $\mathbb{R}$ and $z^t v^{(i)} = 0$, for each $i = 1, 2, \ldots, n$, then $z = 0$. 


7.8 Survey of Methods and Software

In this chapter we have studied iterative techniques to approximate the solution of linear systems. We began with the Jacobi method and the Gauss-Seidel method to introduce the iterative methods. Both methods require an arbitrary initial approximation \( x^{(0)} \) and generate a sequence of vectors \( x^{(i+1)} \) using an equation of the form

\[
x^{(i+1)} = T x^{(i)} + c.
\]

It was noted that the method will converge if and only if the spectral radius of the iteration matrix \( \rho(T) < 1 \), and the smaller the spectral radius, the faster the convergence. Analysis of the residual vectors of the Gauss-Seidel technique led to the SOR iterative method, which involves a parameter \( \omega \) to speed convergence. The preconditioned conjugate gradient method was introduced in Section 7.7.

These iterative methods and modifications are used extensively in the solution of linear systems which arise in the numerical solution of boundary value problems and partial differential equations (see Chapters 11 and 12). These systems are often very large, on the order of 10000 equations in 10000 unknowns, and are sparse with their nonzero entries in predictable positions. The iterative methods are also useful for other large sparse systems and are easily adapted for efficient use on parallel computers.

Almost all commercial and public domain packages that contain iterative methods for the solution of a linear system of equations require a preconditioner to be used with the method. Faster convergence of iterative solvers is often achieved by using a preconditioner. A preconditioner produces an equivalent system of equations that hopefully exhibits better convergence characteristics than the original system. The IMSL Library has a preconditioned conjugate gradient method. The NAG Library has several subroutines for the iterative solution of linear systems. All of the subroutines are based on Krylov subspaces. Saad [Sa2] has a detailed description of Krylov subspace methods. The packages LINPACK and LAPACK contain only direct methods for the solution of linear systems; however, the packages do contain many subroutines that are used by the iterative solvers. The public domain packages IML++, ITPACK, SLAP, and Templates, contain iterative methods. MATLAB contains several iterative methods that are also based on Krylov subspaces. For example, the command \( x = \text{PCG}(A, b) \) executes the preconditioned conjugate gradient method to solve the linear system \( Ax = b \). Some optional input parameters for PCG are, TOL a tolerance for convergence, MAXIT the maximum number of iterations, and \( M \) a preconditioner.

The concepts of condition number and poorly conditioned matrices were introduced in Section 7.6. Many of the subroutines for solving a linear system or for factoring a matrix into an \( LU \) factorization include checks for ill-conditioned matrices and also give an estimate of the condition number. LAPACK, LINPACK, the IMSL Library, and the NAG Library have subroutines that improve on a solution to a linear system that is poorly conditioned. The subroutines test the condition number and then use iterative refinement to obtain the most accurate solution possible given the precision of the computer.
More information on the use of iterative methods for solving linear systems can be found in Varga [Var], Young [Y], Hageman and Young [HY], and in the recent book by Axelsson [Ax]. Iterative methods for large sparse systems are discussed in Barrett et al [BBEPVR], Hackbusch [Hac], Kelley [Kelley], and Saad [Sa2].