9. Let λ be an eigenvalue of the $n \times n$ matrix A and $\mathbf{x} \neq \mathbf{0}$ be an associated eigenvector.

a. Show that λ is also an eigenvalue of A^t.

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

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

d. Generalize parts (b) and (c) to $(A^{-1})^k$ for integers $k \geq 2$.

e. Given the polynomial $q(x) = q_0 + q_1 x + \cdots + q_k x^k$, define $q(A)$ to be the matrix $q(A) = q_0 I + q_1 A + \cdots + q_k A^k$. Show that $q(\lambda)$ is an eigenvalue of $q(A)$ with eigenvector $\mathbf{x}$.

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

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

11. In Exercise 11 of Section 6.3, 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 ith row and jth 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?

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

13. Show that if $|| \cdot ||$ is any natural norm, then $(1/||A^{-1}||) \leq |\lambda| \leq ||A||$ for any eigenvalue λ of the nonsingular matrix A.

7.3 Iterative Techniques for Solving Linear Systems

In this section we describe the Jacobi and the Gauss-Seidel iterative methods, classic methods that date to the late eighteenth century. Iterative techniques are seldom used for solving linear systems of small dimension since the time required for sufficient accuracy exceeds that required for direct techniques such as Gaussian elimination. For large systems with a high percentage of 0 entries, however, these techniques are efficient in terms of both computer storage and computation. Systems of this type arise frequently in circuit analysis and in the numerical solution of boundary-value problems and partial-differential equations.

An iterative technique to solve 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 $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converges to $\mathbf{x}$. 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 fixed 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 $k = 1, 2, 3, \ldots$. This result should be reminiscent of the fixed-point iteration studied in Chapter 2.

EXAMPLE 1 The linear system $A\mathbf{x} = \mathbf{b}$ given by

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

has the unique 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, for each $i = 1, 2, 3, 4$, to obtain

$$\begin{aligned}
x_1 &= \frac{1}{10}x_2 - \frac{1}{5}x_3 + \frac{3}{5},\\
x_2 &= \frac{1}{11}x_1 + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11},\\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 + \frac{1}{10}x_4 - \frac{11}{10},\\
x_4 &= -\frac{3}{8}x_2 + \frac{1}{8}x_3 + \frac{15}{8}.
\end{aligned}$$

Then $A\mathbf{x} = \mathbf{b}$ can be rewritten in the form $\mathbf{x} = T\mathbf{x} + \mathbf{c}$, with

$$T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}.$$

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

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

Additional iterates, $\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, x_4^{(k)})^t$, are generated in a similar manner and are presented in Table 7.1.

Table 7.1

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

The decision to stop after ten iterations was based on the criterion

$$\frac{\|\mathbf{x}^{(10)} - \mathbf{x}^{(9)}\|_\infty}{\|\mathbf{x}^{(10)}\|_\infty} = \frac{8.0 \times 10^{-4}}{1.9998} < 10^{-3}.$$

In fact, $\|\mathbf{x}^{(10)} - \mathbf{x}\|_\infty = 0.0002$. ■

The method of Example 1 is called the **Jacobi iterative method**. It consists of solving the ith equation in $A\mathbf{x} = \mathbf{b}$ for x_i to obtain (provided $a_{ii} \neq 0$)

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-\frac{a_{ij} x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}}, \qquad \text{for } i = 1, 2, \ldots, n$$

and generating each $x_i^{(k)}$ from components of $\mathbf{x}^{(k-1)}$ for $k \geq 1$ by

$$x_i^{(k)} = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij} x_j^{(k-1)} \right) + b_i}{a_{ii}}, \qquad \text{for } i = 1, 2, \ldots, n. \tag{7.4}$$

The method is written in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$ 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,

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

is split into

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ -a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

$$= D - L - U.$$

The equation $A\mathbf{x} = \mathbf{b}$, or $(D - L - U)\mathbf{x} = \mathbf{b}$, is then transformed into

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b},$$

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

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

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

$$\mathbf{x}^{(k)} = D^{-1}(L+U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b}, \quad k = 1, 2, \ldots. \tag{7.5}$$

Introducing the notation $T_j = D^{-1}(L+U)$ and $\mathbf{c}_j = D^{-1}\mathbf{b}$, the Jacobi technique has the form

$$\mathbf{x}^{(k)} = T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j. \tag{7.6}$$

In practice, Eq. (7.4) is used in computation and Eq. (7.6) for theoretical purposes.
Algorithm 7.1 implements the Jacobi iterative technique.

ALGORITHM 7.1

Jacobi Iterative

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \leq i, j \leq n$ of the matrix A; the entries b_i, $1 \leq i \leq n$ of $\mathbf{b}$; the entries XO_i, $1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \leq N$) do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{-\sum_{\substack{j=1\\ j\neq i}}^{n} (a_{ij}XO_j) + b_i}{a_{ii}}.$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT ($x_1, \ldots, x_n$);
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP. ■

Step 3 of the algorithm requires that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$. If one of the a_{ii} entries is 0 and the system is nonsingular, a reordering of the equations can be performed so that no $a_{ii} = 0$. To speed convergence, the equations should be arranged so that a_{ii} is as large as possible. This subject is discussed in more detail later in this chapter.

Another possible stopping criterion in Step 4 is to iterate until

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|}{\|\mathbf{x}^{(k)}\|}$$

is smaller than some prescribed tolerance. For this purpose, any convenient norm can be used, the usual being the l_∞ norm.

A possible improvement in Algorithm 7.1 can be seen by reconsidering Eq. (7.4). The components of $\mathbf{x}^{(k-1)}$ are used to compute $x_i^{(k)}$. Since, for $i > 1$, $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ have already been computed and are probably better approximations to the actual solutions $x_1, \ldots, x_{i-1}$ than $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$, it seems more reasonable to compute $x_i^{(k)}$ using these most recently calculated values. That is, we can use

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1}(a_{ij}x_j^{(k)}) - \sum_{j=i+1}^{n}(a_{ij}x_j^{(k-1)}) + b_i}{a_{ii}}, \tag{7.7}$$

for each $i = 1, 2, \ldots, n$, instead of Eq. (7.4). This modification is called the **Gauss-Seidel iterative technique** and is illustrated in the following example.

EXAMPLE 2 The linear system given by

$$\begin{aligned} 10x_1 - x_2 + 2x_3 \quad &= 6, \\ -x_1 + 11x_2 - x_3 + 3x_4 &= 25, \\ 2x_1 - x_2 + 10x_3 - x_4 &= -11, \\ 3x_2 - x_3 + 8x_4 &= 15 \end{aligned}$$

was solved in Example 1 by the Jacobi iterative method. Incorporating Eq. (7.7) into Algorithm 7.1 gives the equations to be used for each $k = 1, 2, \ldots,$

$$\begin{aligned} x_1^{(k)} &= \frac{1}{10}x_2^{(k-1)} - \frac{1}{5}x_3^{(k-1)} + \frac{3}{5}, \\ x_2^{(k)} &= \frac{1}{11}x_1^{(k)} + \frac{1}{11}x_3^{(k-1)} - \frac{3}{11}x_4^{(k-1)} + \frac{25}{11}, \\ x_3^{(k)} &= -\frac{1}{5}x_1^{(k)} + \frac{1}{10}x_2^{(k)} + \frac{1}{10}x_4^{(k-1)} - \frac{11}{10}, \\ x_4^{(k)} &= -\frac{3}{8}x_2^{(k)} + \frac{1}{8}x_3^{(k)} + \frac{15}{8}. \end{aligned}$$

Letting $\mathbf{x}^{(0)} = (0, 0, 0, 0)^t$, we generate the iterates in Table 7.2.

Table 7.2

k	0	1	2	3	4	5
$x_1^{(k)}$	0.0000	0.6000	1.030	1.0065	1.0009	1.0001
$x_2^{(k)}$	0.0000	2.3272	2.037	2.0036	2.0003	2.0000
$x_3^{(k)}$	0.0000	−0.9873	−1.014	−1.0025	−1.0003	−1.0000
$x_4^{(k)}$	0.0000	0.8789	0.9844	0.9983	0.9999	1.0000

Since

$$\frac{\|\mathbf{x}^{(5)} - \mathbf{x}^{(4)}\|_\infty}{\|\mathbf{x}^{(5)}\|_\infty} = \frac{0.0008}{2.000} = 4 \times 10^{-4},$$

$\mathbf{x}^{(5)}$ 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. ■

To write the Gauss-Seidel method in matrix form, multiply both sides of Eq. (7.7) by a_{ii} and collect all kth iterate terms, to give

$$a_{i1}x_1^{(k)} + a_{i2}x_2^{(k)} + \cdots + a_{ii}x_i^{(k)} = -a_{i,i+1}x_{i+1}^{(k-1)} - \cdots - a_{in}x_n^{(k-1)} + b_i,$$

for each $i = 1, 2, \ldots, n$. Writing all n equations gives

$$\begin{aligned}
a_{11}x_1^{(k)} \qquad\qquad\qquad\qquad &= -a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1,\\
a_{21}x_1^{(k)} + a_{22}x_2^{(k)} \qquad\qquad &= \qquad\qquad -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2,\\
&\vdots\\
a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \cdots + a_{nn}x_n^{(k)} &= \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad b_n;
\end{aligned}$$

with the definitions of D, L, and U given previously, we have the Gauss-Seidel method represented by

$$(D - L)\mathbf{x}^{(k)} = U\mathbf{x}^{(k-1)} + \mathbf{b}$$

or

$$\mathbf{x}^{(k)} = (D - L)^{-1}U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b}, \quad \text{for each } k = 1, 2, \ldots. \tag{7.8}$$

Letting $T_g = (D - L)^{-1}U$ and $\mathbf{c}_g = (D - L)^{-1}\mathbf{b}$, the Gauss-Seidel technique has the form

$$\mathbf{x}^{(k)} = T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g. \tag{7.9}$$

For the lower-triangular matrix $D - L$ to be nonsingular, it is necessary and sufficient that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$.

Algorithm 7.2 implements the Gauss-Seidel method.

ALGORITHM 7.2

Gauss-Seidel Iterative

To solve $A\mathbf{x} = \mathbf{b}$ given an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$ of the matrix A; the entries b_i, $1 \le i \le n$ of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

$$\text{set } x_i = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i}{a_{ii}}.$$

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$;
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP. ■

The comments following Algorithm 7.1 regarding reordering and stopping criteria also apply to the Gauss-Seidel Algorithm 7.2.

The results of Examples 1 and 2 appear 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 (see Exercises 9 and 10).

To study the convergence of general iteration techniques, we consider the formula

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k = 1, 2, \ldots,$$

where $\mathbf{x}^{(0)}$ is arbitrary.

Lemma 7.18 If the spectral radius $\rho(T)$ satisfies $\rho(T) < 1$, then $(I - T)^{-1}$ exists, and

$$(I - T)^{-1} = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j.$$ ■

Proof Since $T\mathbf{x} = \lambda\mathbf{x}$ is true precisely when $(I - T)\mathbf{x} = (1 - \lambda)\mathbf{x}$, we have λ as an eigenvalue of T precisely when $1 - \lambda$ is an eigenvalue of $I - T$. But $|\lambda| \le \rho(T) < 1$, so $\lambda = 1$ is not an eigenvalue of T, and 0 cannot be an eigenvalue of $I - T$. Hence, $(I - T)^{-1}$ exists.

Let $S_m = I + T + T^2 + \cdots + T^m$. Then

$$(I - T)S_m = (1 + T + T^2 + \cdots + T^m) - (T + T^2 + \cdots + T^{m+1}) = I - T^{m+1},$$

and, since T is convergent, the result at the end of Section 7.2 implies that

$$\lim_{m\to\infty} (I - T)S_m = \lim_{m\to\infty} (I - T^{m+1}) = I.$$

Thus, $(I - T)^{-1} = \lim_{m\to\infty} S_m = I + T + T^2 + \cdots = \sum_{j=0}^{\infty} T^j$. ■ ■ ■

Theorem 7.19 For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k \geq 1, \tag{7.10}$$

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ if and only if $\rho(T) < 1$. ■

Proof First assume that $\rho(T) < 1$. Then,

$$\begin{aligned} \mathbf{x}^{(k)} &= T\mathbf{x}^{(k-1)} + \mathbf{c} \\ &= T(T\mathbf{x}^{(k-2)} + \mathbf{c}) + \mathbf{c} \\ &= T^2\mathbf{x}^{(k-2)} + (T + I)\mathbf{c} \\ &\vdots \\ &= T^k\mathbf{x}^{(0)} + (T^{k-1} + \cdots + T + I)\mathbf{c}. \end{aligned}$$

Since $\rho(T) < 1$, the matrix T is convergent and

$$\lim_{k\to\infty} T^k\mathbf{x}^{(0)} = \mathbf{0}.$$

Lemma 7.18 implies that

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \lim_{k\to\infty} T^k\mathbf{x}^{(0)} + \left(\sum_{j=0}^{\infty} T^j\right)\mathbf{c} = \mathbf{0} + (I - T)^{-1}\mathbf{c} = (I - T)^{-1}\mathbf{c}.$$

Hence, the sequence $\{\mathbf{x}^{(k)}\}$ converges to the vector $\mathbf{x} \equiv (I - T)^{-1}\mathbf{c}$ and $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

To prove the converse, we show that for any $\mathbf{z} \in \mathbb{R}^n$, we have $\lim_{k\to\infty} T^k\mathbf{z} = \mathbf{0}$. By Theorem 7.17, this is equivalent to $\rho(T) < 1$.

Let $\mathbf{z}$ be an arbitrary vector, and $\mathbf{x}$ be the unique solution to $\mathbf{x} = T\mathbf{x} + \mathbf{c}$. Define $\mathbf{x}^{(0)} = \mathbf{x} - \mathbf{z}$, and, for $k \geq 1$, $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$. Then $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}$. Also,

$$\mathbf{x} - \mathbf{x}^{(k)} = (T\mathbf{x} + \mathbf{c}) - \left(T\mathbf{x}^{(k-1)} + \mathbf{c}\right) = T\left(\mathbf{x} - \mathbf{x}^{(k-1)}\right),$$

so

$$\mathbf{x} - \mathbf{x}^{(k)} = T\left(\mathbf{x} - \mathbf{x}^{(k-1)}\right) = T^2\left(\mathbf{x} - \mathbf{x}^{(k-2)}\right) = \cdots = T^k\left(\mathbf{x} - \mathbf{x}^{(0)}\right) = T^k\mathbf{z}.$$

Hence $\lim_{k\to\infty} T^k\mathbf{z} = \lim_{k\to\infty} T^k\left(\mathbf{x} - \mathbf{x}^{(0)}\right) = \lim_{k\to\infty}\left(\mathbf{x} - \mathbf{x}^{(k)}\right) = \mathbf{0}$.

Since $\mathbf{z} \in \mathbb{R}^n$ was arbitrary, this implies that T is a convergent matrix and that $\rho(T) < 1$. ■ ■ ■

The proof of the following corollary is similar to the proofs in Corollary 2.4. It is considered in Exercise 11.

Corollary 7.20 If $\|T\| < 1$ for any natural matrix norm and $\mathbf{c}$ is a given vector, then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$ converges, for any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to a vector $\mathbf{x} \in \mathbb{R}^n$, and the following error bounds hold:

(i) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \le \|T\|^k \|\mathbf{x}^{(0)} - \mathbf{x}\|$;

(ii) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \le \frac{\|T\|^k}{1-\|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$. ■

We have seen that the Jacobi and Gauss-Seidel iterative techniques can be written

$$\mathbf{x}^{(k)} = T_j \mathbf{x}^{(k-1)} + \mathbf{c}_j \quad \text{and} \quad \mathbf{x}^{(k)} = T_g \mathbf{x}^{(k-1)} + \mathbf{c}_g,$$

using the matrices

$$T_j = D^{-1}(L + U) \quad \text{and} \quad T_g = (D - L)^{-1} U.$$

If $\rho(T_j)$ or $\rho(T_g)$ is less than 1, then the corresponding sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ will converge to the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$. For example, the Jacobi scheme has

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

and, if $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ converges to $\mathbf{x}$, then

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

This implies that

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b} \quad \text{and} \quad (D - L - U)\mathbf{x} = \mathbf{b}.$$

Since $D - L - U = A$, the solution $\mathbf{x}$ satisfies $A\mathbf{x} = \mathbf{b}$.

We can now give easily verified sufficiency conditions for convergence of the Jacobi and Gauss-Seidel methods. (To prove convergence for the Jacobi scheme, see Exercise 12, and for the Gauss-Seidel scheme, see [Or2, p. 120].)

Theorem 7.21 If A is strictly diagonally dominant, then for any choice of $\mathbf{x}^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge to the unique solution of $A\mathbf{x} = \mathbf{b}$. ■

The relationship of the rapidity of convergence to the spectral radius of the iteration matrix T can be seen from Corollary 7.20. Since the inequalities hold for any natural matrix norm, it follows from the statement after Theorem 7.15 that

$$\|\mathbf{x}^{(k)} - \mathbf{x}\| \approx \rho(T)^k \|\mathbf{x}^{(0)} - \mathbf{x}\|. \tag{7.11}$$

Thus, it is desirable to select the iterative technique with minimal $\rho(T) < 1$ for a particular system $A\mathbf{x} = \mathbf{b}$. No general results exist to tell which of the two techniques, Jacobi or Gauss-Seidel, will be most successful for an arbitrary linear system. In special cases, however, the answer is known, as is demonstrated in the following theorem. The proof of this result can be found in [Y, pp. 120–127].

Theorem 7.22 **(Stein-Rosenberg)**

If $a_{ij} \le 0$, for each $i \ne j$ and $a_{ii} > 0$, for each $i = 1, 2, \ldots, n$, then one and only one of the following statements holds:

a. $0 \leq \rho(T_g) < \rho(T_j) < 1$;

b. $1 < \rho(T_j) < \rho(T_g)$;

c. $\rho(T_j) = \rho(T_g) = 0$;

d. $\rho(T_j) = \rho(T_g) = 1$. ∎

For the special case described in Theorem 7.22, we see from part (a) that when one method gives convergence, then both give convergence, and the Gauss-Seidel method converges faster than the Jacobi method. Part (b) indicates that when one method diverges then both diverge, and the divergence is more pronounced for the Gauss-Seidel method.

Since the rate of convergence of a procedure depends on the spectral radius of the matrix associated with the method, one way to select a procedure to accelerate convergence is to choose a method whose associated matrix has minimal spectral radius. Before describing a procedure for selecting such a method, we need to introduce a new means of measuring the amount by which an approximation to the solution to a linear system differs from the true solution to the system. The method makes use of the vector described in the following definition.

Definition 7.23 Suppose $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is an approximation to the solution of the linear system defined by $A\mathbf{x} = \mathbf{b}$. The **residual vector** for $\tilde{\mathbf{x}}$ with respect to this system is $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$. ∎

In procedures such as the Jacobi or Gauss-Seidel methods, a residual vector is associated with each calculation of an approximation component to the solution vector. The object is to generate a sequence of approximations that will cause the residual vectors to converge rapidly to zero. Suppose we let

$$\mathbf{r}_i^{(k)} = (r_{1i}^{(k)}, r_{2i}^{(k)}, \ldots, r_{ni}^{(k)})^t$$

denote the residual vector for the Gauss-Seidel method corresponding to the approximate solution vector $\mathbf{x}_i^{(k)}$ defined by

$$\mathbf{x}_i^{(k)} = (x_1^{(k)}, x_2^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)})^t.$$

The mth component of $\mathbf{r}_i^{(k)}$ is

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i}^{n} a_{mj} x_j^{(k-1)}, \tag{7.12}$$

or, equivalently,

$$r_{mi}^{(k)} = b_m - \sum_{j=1}^{i-1} a_{mj} x_j^{(k)} - \sum_{j=i+1}^{n} a_{mj} x_j^{(k-1)} - a_{mi} x_i^{(k-1)},$$

for each $m = 1, 2, \ldots, n$.

In particular, the ith component of $\mathbf{r}_i^{(k)}$ is

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

so

$$a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}. \tag{7.13}$$

Recall, however, that in the Gauss-Seidel method, $x_i^{(k)}$ is chosen to be

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)}\right], \tag{7.14}$$

so Eq. (7.13) can be rewritten as

$$a_{ii}x_i^{(k-1)} + r_{ii}^{(k)} = a_{ii}x_i^{(k)}.$$

Consequently, the Gauss-Seidel method can be characterized as choosing $x_i^{(k)}$ to satisfy

$$x_i^{(k)} = x_i^{(k-1)} + \frac{r_{ii}^{(k)}}{a_{ii}}. \tag{7.15}$$

We can derive another connection between the residual vectors and the Gauss-Seidel technique. Consider the residual vector $\mathbf{r}_{i+1}^{(k)}$, associated with the vector $\mathbf{x}_{i+1}^{(k)} = (x_1^{(k)}, \ldots, x_i^{(k)}, x_{i+1}^{(k-1)}, \ldots, x_n^{(k-1)})^t$. By (7.12), the ith component of $\mathbf{r}_{i+1}^{(k)}$ is

$$\begin{aligned} r_{i,i+1}^{(k)} &= b_i - \sum_{j=1}^{i} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} \\ &= b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} - a_{ii}x_i^{(k)}. \end{aligned}$$

Equation (7.14) implies that $r_{i,i+1}^{(k)} = 0$. In a sense, then, the Gauss-Seidel technique is also characterized choosing $x_{i+1}^{(k)}$ in such a way that the ith component of $\mathbf{r}_{i+1}^{(k)}$ is zero.

Choosing $x_{i+1}^{(k)}$ so that one coordinate of the residual vector is zero, however, is not the most efficient way to reduce the norm of the vector $\mathbf{r}_{i+1}^{(k)}$. If we modify the Gauss-Seidel procedure, as given by Eq. (7.15), to

$$x_i^{(k)} = x_i^{(k-1)} + \omega\frac{r_{ii}^{(k)}}{a_{ii}}, \tag{7.16}$$

for certain choices of positive ω, we can reduce the norm of the residual vector and obtain significantly faster convergence.

Methods involving Eq. (7.16) are called **relaxation methods**. For choices of ω with $0 < \omega < 1$, 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. For choices of ω with $1 < \omega$, 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 particularly useful for solving the linear systems that occur in the numerical solution of certain partial-differential equations.

Before illustrating the advantages of the SOR method, we note that by using Eq. (7.13), Eq. (7.16) can be reformulated for calculation purposes to

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

To determine the matrix of the SOR method, we rewrite this 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

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

or

$$\mathbf{x}^{(k)} = (D-\omega L)^{-1}[(1-\omega)D + \omega U]\mathbf{x}^{(k-1)} + \omega(D-\omega L)^{-1}\mathbf{b}. \tag{7.17}$$

If we let $T_\omega = (D-\omega L)^{-1}[(1-\omega)D+\omega U]$ and $\mathbf{c}_\omega = \omega(D-\omega L)^{-1}\mathbf{b}$, the SOR technique has the form

$$\mathbf{x}^{(k)} = T_\omega \mathbf{x}^{(k-1)} + \mathbf{c}_\omega. \tag{7.18}$$

EXAMPLE 3 The linear system $A\mathbf{x} = \mathbf{b}$ given by

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

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 $\mathbf{x}^{(0)} = (1, 1, 1)^t$ for both methods. For each $k = 1, 2, \ldots$, the equations for the Gauss-Seidel method are

$$\begin{aligned} x_1^{(k)} &= -0.75x_2^{(k-1)} + 6, \\ x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5, \\ x_3^{(k)} &= 0.25x_2^{(k)} - 6, \end{aligned}$$

and the equations for the SOR method with $\omega = 1.25$ are

$$\begin{aligned} x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5, \\ x_2^{(k)} &= -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375, \\ x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5. \end{aligned}$$

The first seven iterates for each method are listed in Tables 7.3 and 7.4. For the iterates to be accurate to seven decimal places, the Gauss-Seidel method requires 34 iterations, as opposed to 14 iterations for the over-relaxation method with $\omega = 1.25$. ■

Table 7.3 Gauss-Seidel

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906	3.0549316	3.0343323	3.0214577	3.0134110
$x_2^{(k)}$	1	3.812500	3.8828125	3.9267578	3.9542236	3.9713898	3.9821186	3.9888241
$x_3^{(k)}$	1	−5.046875	−5.0292969	−5.0183105	−5.0114441	−5.0071526	−5.0044703	−5.0027940

Table 7.4 SOR with $\omega = 1.25$

k	0	1	2	3	4	5	6	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027	2.9570512	3.0037211	2.9963276	3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646	4.0074838	4.0029250	4.0009262	4.0002586
$x_3^{(k)}$	1	−6.6501465	−4.6004238	−5.0966863	−4.9734897	−5.0057135	−4.9982822	−5.0003486

The obvious question to ask is how the appropriate value of ω is chosen. Although no complete answer to this question is known for the general $n \times n$ linear system, the following results can be used in certain situations.

Theorem 7.24 **(Kahan)**

If $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$. ■

The proof of this theorem is considered in Exercise 13. The proof of the next two results can be found in [Or2, pp. 123–133]. These results will be used in Chapter 12.

Theorem 7.25 **(Ostrowski-Reich)**

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

Theorem 7.26 If A is positive definite and tridiagonal, then $\rho(T_g) = [\rho(T_j)]^2 < 1$, and the optimal choice of ω for the SOR method is

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

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

EXAMPLE 4 The matrix

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

given in Example 3, is positive definite and tridiagonal, so Theorem 7.26 applies. 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

$$T_j - \lambda I = \begin{bmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{bmatrix},$$

so

$$\det(T_j - \lambda I) = -\lambda(\lambda^2 - 0.625).$$

Thus,

$$\rho(T_j) = \sqrt{0.625}$$

and

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

This explains the rapid convergence obtained in Example 1 when using $\omega = 1.25$. ■

We close this section with Algorithm 7.3 for the SOR method.

ALGORITHM 7.3

SOR

To solve $A\mathbf{x} = \mathbf{b}$ given the parameter ω and an initial approximation $\mathbf{x}^{(0)}$:

INPUT the number of equations and unknowns n; the entries a_{ij}, $1 \le i, j \le n$, of the matrix A; the entries b_i, $1 \le i \le n$, of $\mathbf{b}$; the entries XO_i, $1 \le i \le n$, of $\mathbf{XO} = \mathbf{x}^{(0)}$; the parameter ω; tolerance TOL; maximum number of iterations N.

OUTPUT the approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1 Set $k = 1$.

Step 2 While ($k \le N$) do Steps 3–6.

Step 3 For $i = 1, \ldots, n$

set $x_i = (1-\omega)XO_i + \dfrac{\omega\left(-\sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^{n} a_{ij}XO_j + b_i\right)}{a_{ii}}$.

Step 4 If $||\mathbf{x} - \mathbf{XO}|| < TOL$ then OUTPUT $(x_1, \ldots, x_n)$;
(*The procedure was successful.*)
STOP.

Step 5 Set $k = k + 1$.

Step 6 For $i = 1, \ldots, n$ set $XO_i = x_i$.

Step 7 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was successful.*)
STOP. ■

EXERCISE SET 7.3

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

a.
$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1, \\ 3x_1 + 6x_2 + 2x_3 &= 0, \\ 3x_1 + 3x_2 + 7x_3 &= 4. \end{aligned}$$

b.
$$\begin{aligned} 10x_1 - x_2 \phantom{{}-2x_3} &= 9, \\ -x_1 + 10x_2 - 2x_3 &= 7, \\ - 2x_2 + 10x_3 &= 6. \end{aligned}$$

c.
$$\begin{aligned} 10x_1 + 5x_2 &= 6, \\ 5x_1 + 10x_2 - 4x_3 &= 25, \\ - 4x_2 + 8x_3 - x_4 &= -11, \\ - x_3 + 5x_4 &= -11. \end{aligned}$$

d.
$$\begin{aligned} 4x_1 + x_2 - x_3 + x_4 &= -2, \\ x_1 + 4x_2 - x_3 - x_4 &= -1, \\ -x_1 - x_2 + 5x_3 + x_4 &= 0, \\ x_1 - x_2 + x_3 + 3x_4 &= 1. \end{aligned}$$

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

f.
$$\begin{aligned} 4x_1 - x_2 - x_4 &= 0, \\ -x_1 + 4x_2 - x_3 - x_5 &= 5, \\ - x_2 + 4x_3 - x_6 &= 0, \\ -x_1 + 4x_4 - x_5 &= 6, \\ - x_2 - x_4 + 4x_5 - x_6 &= -2, \\ - x_3 - x_5 + 4x_6 &= 6. \end{aligned}$$

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_∞ norm.
4. Repeat Exercise 3 using the Gauss-Seidel Algorithm.
5. Find the first two iterations of the SOR method with $\omega = 1.1$ for the following linear systems, using $\mathbf{x}^{(0)} = \mathbf{0}$:

a.
$$\begin{aligned} 3x_1 - x_2 + x_3 &= 1, \\ 3x_1 + 6x_2 + 2x_3 &= 0, \\ 3x_1 + 3x_2 + 7x_3 &= 4. \end{aligned}$$

b.
$$\begin{aligned} 10x_1 - x_2 \phantom{{}-2x_3} &= 9, \\ -x_1 + 10x_2 - 2x_3 &= 7, \\ - 2x_2 + 10x_3 &= 6. \end{aligned}$$

c.
$$\begin{aligned}
10x_1 + 5x_2 \qquad\qquad &= 6,\\
5x_1 + 10x_2 - 4x_3 \qquad &= 25,\\
- 4x_2 + 8x_3 - x_4 &= -11,\\
- x_3 + 5x_4 &= -11.
\end{aligned}$$

d.
$$\begin{aligned}
4x_1 + x_2 - x_3 + x_4 &= -2,\\
x_1 + 4x_2 - x_3 - x_4 &= -1,\\
-x_1 - x_2 + 5x_3 + x_4 &= 0,\\
x_1 - x_2 + x_3 + 3x_4 &= 1.
\end{aligned}$$

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

f.
$$\begin{aligned}
4x_1 - x_2 - x_4 &= 0,\\
-x_1 + 4x_2 - x_3 - x_5 &= 5,\\
- x_2 + 4x_3 - x_6 &= 0,\\
-x_1 + 4x_4 - x_5 &= 6,\\
- x_2 - x_4 + 4x_5 - x_6 &= -2,\\
- x_3 - x_5 + 4x_6 &= 6.
\end{aligned}$$

6. Repeat Exercise 1 using $\omega = 1.3$.
7. Use the SOR method with $\omega = 1.2$ to solve the linear systems in Exercise 5 with a tolerance $TOL = 10^{-3}$ in the l_∞ norm.
8. Determine which matrices in Exercise 5 are tridiagonal and positive definite. Repeat Exercise 7 for these matrices using the optimal choice of ω.
9. The linear system

$$\begin{aligned}
2x_1 - x_2 + x_3 &= -1,\\
2x_1 + 2x_2 + 2x_3 &= 4,\\
-x_1 - x_2 + 2x_3 &= -5
\end{aligned}$$

has the solution $(1, 2, -1)^t$.

a. Show that $\rho(T_j) = \frac{\sqrt{5}}{2} > 1$.

b. Show that the Jacobi method with $\mathbf{x}^{(0)} = \mathbf{0}$ fails to give a good approximation after 25 iterations.

c. Show that $\rho(T_g) = \frac{1}{2}$.

d. Use the Gauss-Seidel method with $\mathbf{x}^{(0)} = \mathbf{0}$ to approximate the solution to the linear system to within 10^{-5} in the l_∞ norm.

10. The linear system

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

has the solution $(1, 2, -1)^t$.

a. Show that $\rho(T_j) = 0$.

b. Use the Jacobi method with $\mathbf{x}^{(0)} = \mathbf{0}$ to approximate the solution to the linear system to within 10^{-5} in the l_∞ norm.

c. Show that $\rho(T_g) = 2$.

d. Show that the Gauss-Seidel method applied as in part (b) fails to give a good approximation in 25 iterations.

11. **a.** Prove that

$$\|\mathbf{x}^{(k)} - \mathbf{x}\| \le \|T\|^k \, \|\mathbf{x}^{(0)} - \mathbf{x}\| \quad \text{and} \quad \|\mathbf{x}^{(k)} - \mathbf{x}\| \le \frac{\|T\|^k}{1 - \|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|,$$

where T is an $n \times n$ matrix with $\|T\| < 1$ and

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad k = 1, 2, \ldots,$$

with $\mathbf{x}^{(0)}$ arbitrary, $\mathbf{c} \in \mathbb{R}^n$, and $\mathbf{x} = T\mathbf{x} + \mathbf{c}$.

b. Apply the bounds to Exercise 1, when possible, using the l_∞ norm.

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

13. Prove Theorem 7.24. [*Hint:* If $\lambda_1, \ldots, \lambda_n$ are eigenvalues of T_ω, then $\det T_\omega = \prod_{i=1}^n \lambda_i$. Since $\det D^{-1} = \det(D - \omega L)^{-1}$ and the determinant of a product of matrices is the product of the determinants of the factors, the result follows from Eq. (7.17).]

14. 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,

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

a. Show that

$$\begin{bmatrix} 1 & -\frac{1}{2} & 0 & \cdots & \cdots & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & \ddots & & \vdots \\ 0 & -\frac{1}{2} & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & \cdots & \cdots & 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

b. Solve this system using $n = 10$, 50, and 100.

c. Change the probabilities to α 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 $\alpha = \frac{1}{3}$.

15. Use all the applicable methods in this section to solve the linear system $A\mathbf{x} = \mathbf{b}$ to within 10^{-5} in the l_∞ norm, where the entries of A are

$$a_{i,j} = \begin{cases} 2i, & \text{when } j = i \text{ and } i = 1, 2, \ldots, 80, \\ 0.5i, & \text{when } \begin{cases} j = i + 2 \text{ and } i = 1, 2, \ldots, 78, \\ j = i - 2 \text{ and } i = 3, 4, \ldots, 80, \end{cases} \\ 0.25i, & \text{when } \begin{cases} j = i + 4 \text{ and } i = 1, 2, \ldots, 76, \\ j = i - 4 \text{ and } i = 5, 6, \ldots, 80, \end{cases} \\ 0, & \text{otherwise,} \end{cases}$$

and those of $\mathbf{b}$ are $b_i = \pi$, for each $i = 1, 2, \ldots, 80$.