assuming that the diagonal elements are all nonzero. We start the solution process by using guesses for the $x$'s, say $x_1 = x_2 = \cdots = x_n = 0$. The first equation can be solved for $x_1$, the second for $x_2$, and so on. If we denote the estimates after the $k$th iteration as $x_1^k, x_2^k, \ldots, x_n^k$, the estimates after $(k+1)$th iteration can be obtained from Equation (C.16) as

$$x_i^{k+1} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1, j\neq 1}^{n} a_{ij}x_j^k\right], \qquad i = 1, 2, \ldots, n \tag{C.17}$$

The iteration process is continued until values of $x_i$ at two successive iterations are within an allowable prescribed deviation.

Convergence is measured in terms of the change in $x_i$ from the $k$th iteration to the next. If we compute

$$d_i = \left|\frac{x_i^{k+1} - x_i^k}{x_i^{k+1}}\right| \cdot 100\% \tag{C.18}$$

for each $x_i$, convergence can be checked using the criterion

$$d_i < \epsilon_s \tag{C.19}$$

where $\epsilon_s$ is a specified small quantity. A better test would be to compute

$$d = \frac{\sum_{i=1}^{n} \left|x_i^{k+1} - x_i^k\right|}{\sum_{i=1}^{n} \left|x_i^{k+1}\right|} \cdot 100\% \tag{C.20}$$

and require that $d < \epsilon_s$.

### C.2.2 Gauss–Seidel Method

This is the most commonly used iterative method. In Jacobi's method the entire set of $x_i$ from the $k$th iteration is used in calculating the new set during the $(k+1)$th iteration, whereas the most recently calculated value of each variable is used at each step in the Gauss–Seidel method. This makes the Gauss–Seidel method converge more rapidly than (about twice as) Jacobi's method and is always used in preference to it. Instead of Equation (C.17), we use

$$x_i^{k+1} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^k\right], \qquad i = 1, 2, \ldots, n \tag{C.21}$$

A computer program based on this method is displayed in Figure C.3.



```
% This program employs Gauss-
% simultaneous equations [A][
% backslash operator "\". Th
% diagonally dominant (i.e. t
% greater than the sum of abs

clear;

%Define A and B
A = [10 -7 0;-3 6 1;2 -1 5];
B = [7;4;6];

%MATLAB backslash operation
x = A\B;

%Begin Gauss-Seidel iterative
N = size(A,1);
X = zeros(N,1);
K = 0; % iteration count
TOL = 1e-3; % tolerance for ze
converged = 0; %while loop co
Xnew = zeros(N,1);

while converged == 0
    for ii = 1:N
        Xnew(ii) = 1/A(ii,ii)*
            - A(ii,(ii+1):N)*
    end
    %Convergent condition
    d = sum(abs(Xnew-X))/sum(a
    %Replace old value with ne
    X = Xnew;

    K = K + 1;
    disp([num2str(K),'
    if d<=TOL %convergence te
        converged = 1;
    end
end

thedifference = x-X(:)
```

**Figure C.3**
**Gauss–Seidel iterative meth**

### C.2.3 Relaxation Metho

This is a slight modification convergence. If $x_i^k$ is added to is subtracted from it, we obtai

$$x_i^{k+1} = x_i^k + \frac{1}{a_{ii}}\left[b_i - \sum \right.$$

The second term on the right correction term tends to zero by $\omega$, Equation (C.22) become

$$x_i^{k+1} = x_i^k + \frac{\omega}{a_{ii}}\left[b_i - \sum \right.$$

```
% This program employs Gauss-Seidel iterative method to solve a set of
% simultaneous equations [A][X] = [B] and compares the result to the MATLAB
% backslash operator "\".  This method will converge if the matrix A is
% diagonally dominant (i.e. the absolute value of the diagonal term is
% greater than the sum of absolute values of the other terms in each row).

clear;

%Define A and B
A = [10 -7 0;-3 6 1;2 -1 5];
B = [7;4;6];

%MATLAB backslash operation
x = A\B;

%Begin Gauss-Seidel iterative method
N = size(A,1);
X = zeros(N,1);
K = 0; % iteration count
TOL = 1e-3; % tolerance for zero
converged = 0; %while loop condition
Xnew = zeros(N,1);

while converged == 0
    for ii = 1:N
        Xnew(ii) = 1/A(ii,ii)*( B(ii) - A(ii,1:(ii-1))*Xnew(1:(ii-1))...
            - A(ii,(ii+1):N)*X((ii+1):N));
    end
    %Convergent condition
    d = sum(abs(Xnew-X))/sum(abs(Xnew));
    %Replace old value with newly computed value
    X = Xnew;

    K = K + 1;
    disp([num2str(K),'          ',num2str(X(:)'),'            ',num2str(d)])
    if d<=TOL %convergence test
        converged = 1;
    end
end

thedifference = x-X(:)
```

**Figure C.3**
**Gauss–Seidel iterative method of solving $[A][X] = [B]$.**

### C.2.3 Relaxation Method

This is a slight modification of the Gauss–Seidel method and is designed to enhance convergence. If $x_i^k$ is added to the right-hand side of Equation (C.21) and $(a_{ii}x_i^k)/a_{ii}$ is subtracted from it, we obtain

$$x_i^{k+1} = x_i^k + \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i}^{n} a_{ij}x_j^k\right], \qquad i = 1, 2, \ldots, n \qquad \text{(C.22)}$$

The second term on the right-hand side can be regarded as a correction term. The correction term tends to zero as convergence is approached. If this term is multiplied by $\omega$, Equation (C.22) becomes

$$x_i^{k+1} = x_i^k + \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i}^{n} a_{ij}x_j^k\right], \qquad i = 1, 2, \ldots, n \qquad \text{(C.23)}$$