Chapter 1

Linear Algebraic Equations

1.1 INTRODUCTION

Several physical operations (most notably, staged operations) can be described in terms of a set of N *coupled, linear algebraic* equations for N unknowns, x_i, i = 1,2, ..., N, having the following form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2N}x_N &= b_2 \\ \cdot & \quad \cdot \\ \cdot & \quad \cdot \\ a_{N1}x_1 + a_{N2}x_2 + ... + a_{NN}x_N &= b_N \end{aligned} \qquad ...(1.1)$$

Such equations are also obtained when numerical techniques such as finite difference, orthogonal collocation, finite elements, etc., are applied to sets of ordinary or partial differential equations. Hence, it is important to develop efficient techniques for solving these equations, particularly for large values of N (values of N larger than 100 are not uncommon).

Equation 1.1 can be written more compactly using vector-matrix notation as

$$\mathbf{Ax = b} \qquad ...(1.2)$$

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & ... & a_{1N} \\ a_{21} & a_{22} & a_{23} & & a_{2N} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ a_{N1} & a_{N2} & a_{N3} & ... & a_{NN} \end{bmatrix}$$

$$\mathbf{x}^T = \begin{bmatrix} x_1 & x_2 & ... & x_N \end{bmatrix}$$

$$\mathbf{b}^T = \begin{bmatrix} b_1 & b_2 & ... & b_N \end{bmatrix} \qquad ...(1.3)$$

T represents the transpose [1–3], and the (i, j) term (ith row and jth column) in $\mathbf{A}$ is a_{ij}.

The solution of eq. 1.2 can be written using Cramer's rule [1–3] as

$$x_j = |\mathbf{A}_j| / |\mathbf{A}|; \quad j = 1, 2, ..., N \qquad ...(1.4)$$

where $|\mathbf{A}|$ represents the determinant of matrix **A**. Matrix $\mathbf{A}_j$ is obtained by replacing the jth column of **A** by the column vector, **b**. Equation 1.2 has a unique solution only if $|\mathbf{A}| \neq 0$ (*i.e.* **A** is *non-singular*). If, in addition to this condition, $\mathbf{b} = \mathbf{0}$ (homogeneous equations), then $|\mathbf{A}_j| = 0$ and all the $x_j = 0$ (trivial solution).

Example 1.1: Solve by Cramer's rule:

$$\begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 11 \end{bmatrix}$$

$$|\mathbf{A}| = \begin{vmatrix} 1 & 1 \\ 2 & 3 \end{vmatrix} = 1$$

$$|\mathbf{A}_1| = \begin{vmatrix} 4 & 1 \\ 11 & 3 \end{vmatrix} = 1$$

$$|\mathbf{A}_2| = \begin{vmatrix} 1 & 4 \\ 2 & 11 \end{vmatrix} = 3$$

$$\therefore \quad x_1 = |\mathbf{A}_1| / |\mathbf{A}| = 1$$

$$x_2 = |\mathbf{A}_2| / |\mathbf{A}| = 3$$

$$\mathbf{x}^T = [1 \quad 3]$$

■

Two possibilities exist if **A** is singular, as illustrated by the following example:

Example 1.2: Solve

$$(a) \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 22 \end{bmatrix} \text{ and } (b) \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 20 \end{bmatrix}$$

A is singular, since $|\mathbf{A}| = 12 - 12 = 0$

(*a*) In this case, it is observed that the two equations are, in fact, identical:

$$2x_1 + 3x_2 = 11$$

This situation, where there are more variables than equations, leads to a set of *infinite solutions*. One can find a value of x_1 corresponding to each value assigned to x_2. Uniqueness is no longer present.

(*b*) In this case the two equations are incompatible:

$$2x_1 + 3x_2 = 11$$

$$2x_1 + 3x_2 = 10$$

and *no solutions* exist. ■

One can also study the more general case of M linear algebraic equations in N unknowns $\mathbf{x} = [x_1 \; x_2 \; ... \; x_N]^T$:

$$\mathbf{Ax} = \mathbf{b} \qquad ...(1.5)$$

where $\mathbf{b} = [b_1\ b_2\ ...\ b_M]^T$ and $\mathbf{A}$ is an $M \times N$ matrix. An *augmented matrix*, aug $\mathbf{A}$, is formed as given below:

$$\text{aug } \mathbf{A} \equiv [\mathbf{A} \mid \mathbf{b}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & ... & a_{1N} & b_1 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ a_{M1} & a_{M2} & ... & a_{MN} & b_M \end{array}\right] \qquad ...(1.6)$$

The *rank* (see Appendix 1 – *A*) [1–3] of $\mathbf{A}$ and aug $\mathbf{A}$ are first determined. The following general results give the conditions under which unique, infinite, or no solutions exist [1]:

(*a*) The necessary and sufficient condition that eq. 1.5 has solutions is that the rank of $\mathbf{A}$ be the same as the rank of aug $\mathbf{A}$ (If this condition is violated the equations are incompatible, and therefore, have no solution).

(*b*) If condition (*a*) is satisfied, and if r is the rank of both $\mathbf{A}$ and aug $\mathbf{A}$, then one can assign arbitrary values to $N - r$ variables, and compute the values of the remaining r variables. A little reflection gives the following table for the number of solutions, when condition (*a*) is satisfied:

$M = N$ $(r \le N)$		$M < N$ $(r \le M)$	$M > N$ $(r \le N)$	
$r = N$	Unique	Infinite	$r = N$	Unique
$r < N$	Infinite		$r < N$	Infinite

(*c*) If $\mathbf{b} = \mathbf{0}$ (homogeneous equations), then *nontrivial* solutions are obtained if and only if the rank of $\mathbf{A}$ is less than N.

Example 1.3: Study examples 1.1 and 1.2 in terms of the general conditions given above.

For example 1.1:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix}, \text{ rank } (\mathbf{A}) = 2$$

$$\text{aug}\,\mathbf{A} = \begin{bmatrix} 1 & 1 & 4 \\ 2 & 3 & 11 \end{bmatrix}, \text{ rank } (\text{aug}\,\mathbf{A}) = 2$$

Hence, the equations are compatible (condition (*a*)). Also, $r = N = 2$, and a unique solution exists.

For example 1.2(a):

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}, \text{ rank } (\mathbf{A}) = 1$$

$$\text{aug}\,\mathbf{A} = \begin{bmatrix} 2 & 3 & 11 \\ 4 & 6 & 22 \end{bmatrix}, \text{ rank } (\text{aug}\,\mathbf{A}) = 1$$

since all three 2×2 determinants:

$$\begin{vmatrix} 2 & 3 \\ 4 & 6 \end{vmatrix}, \begin{vmatrix} 2 & 11 \\ 4 & 22 \end{vmatrix} \text{ and } \begin{vmatrix} 3 & 11 \\ 6 & 22 \end{vmatrix}$$

are zero. Hence, the equations are compatible. Since $r = 1$ and $N = 2$ ($r < N$), we have infinite solutions.

For example 1.2(*b*):

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}, \text{ rank}(\mathbf{A}) = 1$$

$$\text{aug}\,\mathbf{A} = \begin{bmatrix} 2 & 3 & 11 \\ 4 & 6 & 20 \end{bmatrix}, \text{ rank } (\text{aug}\,\mathbf{A}) = 2$$

since $\begin{vmatrix} 2 & 11 \\ 4 & 20 \end{vmatrix} \neq 0$

Hence condition (*a*) is violated. The equations are, therefore, incompatible, and have *no* solution. ■

Problems 1.1 and 1.2 consider two examples where there are more equations than unknowns, and the general rules described above can be used to find out the number of solutions possible.

1.2 GAUSS ELIMINATION AND LU DECOMPOSITION

In this and the following sections, methods for solving N linear, simultaneous, algebraic equations in N unknowns are discussed. $\mathbf{A}$ is then an $N \times N$ square matrix. Such systems of equations are quite commonly encountered in Engineering. Cramer's rule, unfortunately, requires excessive calculations. In fact, it can easily be estimated [1, 3, 4] that $(N - 1)\ (N + 1)\ (N!) + N$ or approximately $N^2 \times N!$ multiplications and divisions are required to obtain the solution (using determinant expansions in terms of minors [1–3]—see prob. 6.4.4 in Ref. 4). For a value of N equal to 100, this means approximately 10^{162} multiplications and divisions—a prohibitively large number (a third generation computer, e.g. DEC 1090, taking about 5×10^{-6} s per multiplication, would take about 10^{149} years to do these!).

An alternate scheme to solve eq. 1.2 is by using the inverse of $\mathbf{A}$, *i.e.*

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \qquad ...(1.7)$$

This, too, is not a suitable method of solution for large N. In fact, the number of operations involved in computing $\mathbf{A}^{-1}$ and then $\mathbf{A}^{-1}\mathbf{b}$ is identical to that using Cramer's rule to obtain $\mathbf{x}$, if $\mathbf{A}^{-1}$ is computed by its definition [1–3], $\mathbf{A}^{-1} = (\text{adj}\ \mathbf{A})/|\ \mathbf{A}\ |$. Here adj $\mathbf{A}$ is the adjoint of matrix $\mathbf{A}$, with $(\text{adj}\ \mathbf{A})_{ji} = \mathbf{A}_{ij}$, where $\mathbf{A}_{ij}$ is the cofactor of the element a_{ij} in the original matrix $\mathbf{A}$. However, faster numerical algorithms could be used to obtain $\mathbf{A}^{-1}$. Besides the extensive computational time required to obtain $\mathbf{A}^{-1}$, this method suffers from the additional problem of *sensitivity* (or *ill-conditioning*), *i.e.* small changes in $\mathbf{A}$ (due to round off errors) may lead to large changes in $\mathbf{A}^{-1}$ and hence in $\mathbf{x}$ (see prob. 1.3). III-conditioning of $\mathbf{A}$ occurs when $|\ \mathbf{A}\ |$ is close to zero.

Several efficient techniques have been developed to obtain solutions of large sets of equations of the type given in eq. 1.2. These can be classified into *direct* (e.g., Gauss elimination, Gauss-Jordan elimination, Cholesky's or Crout's method, etc.) and *indirect* or *iterative* (e.g., Jacobi, Gauss-Seidel and relaxation) methods. Some of these methods are described in this chapter.

Gauss elimination [5] is a relatively simple technique for solving a system of N linear algebraic equations in N unknowns:

$$\mathbf{A}^{(1)}\mathbf{x} \equiv \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \ldots & a_{1N}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & a_{23}^{(1)} & \ldots & a_{2N}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & a_{33}^{(1)} & \ldots & a_{3N}^{(1)} \\ . & . & . & & . \\ . & . & . & & . \\ a_{N1}^{(1)} & a_{N2}^{(1)} & a_{N3}^{(1)} & \ldots & a_{NN}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} \\ . \\ . \\ b_N^{(1)} \end{bmatrix} \equiv \mathbf{b}^{(1)} \qquad \ldots(1.8)$$

The superscripts (1) on the elements of the non-singular $N \times N$ matrix $\mathbf{A}$ and on $\mathbf{b}$ indicate that these are the starting (or first iterate) values. A set of simple operations (e.g., multiplying an entire equation by a constant, adding two equations, etc., which do not alter the solutions) are now performed on the N equations of eq. 1.8. The first row of $\mathbf{A}^{(1)}$ and $\mathbf{b}^{(1)}$ (corresponding to the first equation of the set) is multiplied by $-a_{21}^{(1)}/a_{11}^{(1)}$ and added to the second row, to eliminate the (2, 1) term. Similarly, the first row is multiplied by $-a_{31}^{(1)}/a_{11}^{(1)}$ and added to the third row to eliminate the (3, 1) term, etc. This leads to

$$\mathbf{A}^{(2)}\mathbf{x} \equiv \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \ldots \\ 0 & \left[a_{22}^{(1)} - \dfrac{a_{21}^{(1)}}{a_{11}^{(1)}} a_{12}^{(1)}\right] & \left[a_{23}^{(1)} - \dfrac{a_{21}^{(1)}}{a_{11}^{(1)}} a_{13}^{(1)}\right] & \ldots \\ 0 & \left[a_{32}^{(1)} - \dfrac{a_{31}^{(1)}}{a_{11}^{(1)}} a_{12}^{(1)}\right] & \left[a_{33}^{(1)} - \dfrac{a_{31}^{(1)}}{a_{11}^{(1)}} a_{13}^{(1)}\right] & \ldots \\ . & . & . & \ldots \\ . & . & . & \ldots \\ . & . & . & \ldots \\ 0 & \left[a_{N2}^{(1)} - \dfrac{a_{N1}^{(1)}}{a_{11}^{(1)}} a_{12}^{(1)}\right] & \left[a_{N3}^{(1)} - \dfrac{a_{N1}^{(1)}}{a_{11}^{(1)}} a_{13}^{(1)}\right] & \ldots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} - \dfrac{a_{21}^{(1)}}{a_{11}^{(1)}} b_1^{(1)} \\ b_3^{(1)} - \dfrac{a_{31}^{(1)}}{a_{11}^{(1)}} b_1^{(1)} \\ . \\ . \\ . \\ b_N^{(1)} - \dfrac{a_{N1}^{(1)}}{a_{11}^{(1)}} b_1^{(1)} \end{bmatrix} \equiv \mathbf{b}^{(2)} \qquad \ldots(1.9)$$

It can easily be verified that these operations are described by the following equations:

$$a_{ij}^{(k)} \equiv a_{ij}^{(k-1)} - \frac{a_{i,k-1}^{(k-1)}}{a_{k-1,\,k-1}^{(k-1)}} a_{k-1,\,j}^{(k-1)} \qquad \ldots(a)$$

$$b_i^{(k)} \equiv b_i^{(k-1)} - \frac{a_{i,k-1}^{(k-1)}}{a_{k-1,\,k-1}^{(k-1)}} b_{k-1}^{(k-1)} \qquad \ldots(b) \qquad \ldots(1.10)$$

$i = 2, 3, \ldots, N$

$j = 2, \ldots, N$

with $k = 2$. The unnecessary operations for $j = 1$ need not be performed since it is clear that the end result is zero. We therefore, have

$$\mathbf{A}^{(2)}\mathbf{x} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & .. & a_{1N}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & .. & a_{2N}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} & .. & a_{3N}^{(2)} \\ . & . & . & & . \\ . & . & . & & . \\ . & . & . & & . \\ 0 & a_{N2}^{(2)} & a_{N3}^{(2)} & & a_{NN}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(2)} \\ . \\ . \\ . \\ b_N^{(2)} \end{bmatrix} \qquad ...(1.11)$$

The next iteration attempts to eliminate the (3, 2), (4, 2), ..., (N, 2) terms from $\mathbf{A}^{(2)}$. This is done by successively multiplying the second row with $-a_{32}^{(2)}/a_{22}^{(2)}, -a_{42}^{(2)}/a_{22}^{(2)}, \ldots,$ and $-a_{N2}^{(2)}/a_{22}^{(2)}$, and adding it to the 3rd, 4th, ..., Nth rows. This leads to

$$\mathbf{A}^{(3)}\mathbf{x} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \ldots \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \ldots \\ 0 & 0 & \left[a_{33}^{(2)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}} a_{23}^{(2)}\right] & \ldots \\ 0 & 0 & \left[a_{43}^{(2)} - \frac{a_{42}^{(2)}}{a_{22}^{(2)}} a_{23}^{(2)}\right] & \ldots \\ . & . & . & \ldots \\ . & . & . & \ldots \\ . & . & . & \ldots \\ 0 & 0 & \left[a_{N3}^{(2)} - \frac{a_{N2}^{(2)}}{a_{22}^{(2)}} a_{23}^{(2)}\right] & \ldots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ . \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(2)} - \frac{a_{32}^{(2)}}{a_{22}^{(2)}} b_2^{(2)} \\ b_4^{(2)} - \frac{a_{42}^{(2)}}{a_{22}^{(2)}} b_2^{(2)} \\ . \\ . \\ . \\ b_N^{(2)} - \frac{a_{N2}^{(2)}}{a_{22}^{(2)}} b_2^{(2)} \end{bmatrix} \equiv \mathbf{b}^{(3)} \qquad ...(1.12)$$

It can again be verified that these operations are described by eq. 1.10 using

$$\begin{aligned} i &= 3, 4, \ldots, N \\ j &= 3, 4, \ldots, N \\ k &= 3 \end{aligned} \qquad ...(1.13)$$

This sequence is continued until all the lower triangular elements are eliminated, to give finally

$$\mathbf{A}^{(N)}\mathbf{x} \equiv \mathbf{U}\mathbf{x} \equiv \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & a_{14}^{(1)} & \ldots & a_{1N}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & a_{24}^{(2)} & \ldots & a_{2N}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & a_{34}^{(3)} & \ldots & a_{3N}^{(3)} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & \ldots & a_{NN}^{(N)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_N \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(3)} \\ . \\ . \\ . \\ b_N^{(N)} \end{bmatrix} \equiv \mathbf{b}^{(N)} \qquad \ldots(1.14)$$

where **U** is the upper triangular matrix $\mathbf{A}^{(N)}$.

The algorithm which can be readily used on a computer is described by eq. 1.10, with the following values of the indices i, j and k (in the sequence given)

$$\begin{aligned} k &= 2, 3, \quad 4, \quad \ldots, N \\ i &= k, k+1, \quad k+2, \quad \ldots, N \\ j &= k, k+1, \quad k+2, \quad \ldots, N \end{aligned} \qquad \ldots(1.15)$$

It is assumed that $a_{k-1,k-1}^{(k-1)}$ does not become zero at any stage (see Example 1.5 and the discussion thereafter for what is to be done if it is so). It may be added that at each iteration, the computed values of $\mathbf{A}^{(k)}$ can be stored at the same locations as $\mathbf{A}^{(k-1)}$, since the older elements are not used later.

The solution **x** can now be easily obtained by solving eq. 1.14 sequentially, in the reverse direction (backward sweep), since the last equation involves only x_N, the one before this x_N and x_{N-1}, etc. Thus we have

$$x_N = b_N^{(N)} \Big/ a_{NN}^{(N)} \qquad \ldots(1.16)$$

and

$$x_i = \left[b_i^{(i)} - \sum_{j=i+1}^{N} a_{ij}^{(i)} x_j \right] \Bigg/ a_{ii}^{(i)}, \quad i = N-1, N-2, \ldots, 1 \qquad \ldots(1.17)$$

Example 1.4: Solve using Gauss elimination with backward sweep

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$$

2nd Iterate Values: Multiply 1st row by −1/2 and add to 2nd row; no operation is required on third row since $a_{31}^{(1)} = 0$ already:

$$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 3/2 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3/2 \\ 4 \end{bmatrix}$$

3rd Iterate Values: Multiply 2nd row by −2/3 and add to third row;

$$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 3/2 & 1 \\ 0 & 0 & 1/3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 3/2 \\ 3 \end{bmatrix}$$

Backward substitution:

$$\frac{1}{3}x_3 = 3 \qquad \therefore \quad x_3 = 9$$

$$\frac{3}{2}x_2 + x_3 = \frac{3}{2} \qquad \therefore \quad x_2 = -5$$

$$2x_1 + x_2 = 1 \qquad \therefore \quad x_1 = 3$$

Hence, $\mathbf{x}^T = [3 \quad -5 \quad 9]$. ■

The total number of operations required in Gauss elimination and backward sweep are given as follows:

Multiplications and divisions: $\dfrac{(N^3 + 3N^2 - N)}{3} \cong N^3/3$ ...(*a*)

Additions and subtractions: $\dfrac{(2N^3 + 3N^2 - 5N)}{6} \cong N^3/3$ ...(*b*)

...(1.18)

The number of multiplications and divisions (which take relatively more time on a computer than additions and subtractions) is found to be far less than in Cramer's rule and fewer than the number required to compute $\mathbf{A}^{-1}$ (used in $\mathbf{x} = \mathbf{A}^{-1}\,\mathbf{b}$) using one of the more efficient algorithms. For $N = 100$, Gauss elimination and backward sweep requires only 3.3×10^5 of these operations, as compared to about 10^{160} for Cramer's rule. Even though the number required is lower than for Cramer's rule, it is still very high, and more efficient relaxation methods are, in fact, available, which are preferred for values of N above about twenty five. The number of storage locations required for Gauss elimination is approximately $2N^3 + N$.

The Gauss elimination technique would lead to problems when the diagonal or *pivot* element, $a_{k-1,\,k-1}^{(k-1)}$ (in eq. 1.10), becomes zero, or close to zero at any stage. This could occur even when **A** is nonsingular, as described in the following example. In such cases, the order of the equations is changed by exchanging the rows, and the elimination procedure is continued.

Example 1.5: Solve by Gauss elimination and backward sweep $\begin{bmatrix} 1 & -1 & 2 \\ 2 & -2 & 3 \\ 1 & 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} -8 \\ -20 \\ -2 \end{bmatrix}$

The second iterate is seen to be $\begin{bmatrix} 1 & -1 & 2 \\ 0 & 0 & -1 \\ 0 & 2 & -1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} -8 \\ -4 \\ 6 \end{bmatrix}$ ...(*a*)

We cannot proceed further since $a_{22}^{(2)} = 0$. However, we can reorder the set of three equations represented by the matrix equation to circumvent this problem. It is obvious that if we interchange the second and third equations, we can continue Gauss elimination without changing the solution values. Reordering leads to

$$\begin{bmatrix} 1 & -1 & 2 \\ 0 & 2 & -1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -8 \\ 6 \\ -4 \end{bmatrix}$$

The third iteration need not be performed since $a_{32}^{(2)} = 0$. In the general case of more equations, one would continue the elimination process after exchanging the rows. The solution is

$$\mathbf{x}^T = [-11 \quad 5 \quad 4]$$ ■

The above technique is referred to as maximal *column* pivoting (or partial pivoting). In case it turns out at any stage, that an exchange of rows is not possible (*i.e.* all elements below the pivot are also zero), the procedure stops. This occurs when the set of equations does not have a unique solution.

Instead of interchanging rows as in example 1.5, one could also interchange columns (being careful to redefine **x** so that the equations remain the same). Thus, eq. (*a*) in example 1.5 could be rewritten by exchanging the second and third columns, as

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & -1 & 0 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_2 \end{bmatrix} = \begin{bmatrix} -8 \\ -4 \\ 6 \end{bmatrix} \qquad \text{...(1.19)}$$

It can easily be confirmed that eq. 1.19 is the same as eq. (*a*) of example 1.5. Gauss elimination can now be continued to give

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_2 \end{bmatrix} = \begin{bmatrix} -8 \\ -4 \\ 10 \end{bmatrix} \qquad \text{...(1.20)}$$

giving the same solution as before, $\mathbf{x}^T = [-11 \quad 5 \quad 4]$.

Example 1.6 shows that at times, particularly when **A** is non-singular, it helps to interchange rows and/or columns, so that one brings the largest (in the absolute value sense) of the *remaining* elements (below and to the right) in $\mathbf{A}^{(k)}$ to the pivot position. This strategy is known as Gauss elimination using the *maximum pivoting* strategy, [6, 7] and it reduces round-off errors. Several computer libraries have programs which use this technique, e.g., IMSL [6, 8] has LEQT1F, the NAG (Numerical Algorithms Group, Oxford University) library has F01BTF; special packages like LINPAK and ITPACK etc., are also available [6, 9, 10], etc. Since search operations involved in maximum pivoting take somewhat more computer time than additions or subtractions, this increases the computer time. For this reason, maximum pivoting is used only when the larger CPU times required (because of comparisons) can be justified. Partial pivoting, on the other hand, is quite commonly used.

Example 1.6: Solve

$$\begin{bmatrix} 0.00010\overline{0} & 1.\overline{0} \\ 1.\overline{0} & 1.\overline{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.\overline{0} \\ 2.\overline{0} \end{bmatrix}$$

by Gauss elimination and backward sweep, with and without maximum pivoting, using 4 decimal place accuracy.

With pivoting: Since the largest element in $\mathbf{A}^{(1)}$ is 1, we can exchange rows to give

$$\begin{bmatrix} 1.0000 & 1.0000 \\ 0.0001 & 1.0000 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.0000 \\ 1.0000 \end{bmatrix}$$

Use Gauss elimination to give

$$\begin{bmatrix} 1.0000 & 1.0000 \\ 0.0000 & 0.9999 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.0000 \\ 0.9998 \end{bmatrix}$$

and so

$$\mathbf{x}^T = [1.0000 \quad 1.0000]$$

More accuracy would lead to $\mathbf{x}^T = [1.0001 \quad 0.9999]$.

Without pivoting: the second iterate is (correct to four decimal places)

$$\begin{bmatrix} 0.0001 & 1.0000 \\ 0.0000 & -9999 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.0000 \\ -9998 \end{bmatrix}$$

giving

$$\mathbf{x}^T = \begin{bmatrix} 0.0000 & 1.0000 \end{bmatrix}$$

which is quite different from the solution with pivoting, and does not satisfy the original equations. Problems with such ill-conditioned **A** should be solved with single as well as double precision arithmetic on the computer, to ascertain how good the results are. Scaling of equations so that the maximum $| a_{ij} |$ in any row is unity can also be helpful at times. ■

Example 1.7: Thomas Algorithm [11, 12]

Using Gauss elimination (with normalization of the diagonal elements) and backward sweep, obtain the algorithm required to solve

$$\mathbf{A}^{(1)}\mathbf{x} \equiv \begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & \dots & 0 & 0 & 0 \\ \cdot & & & & \cdot & & \cdot & & \cdot \\ \cdot & & & & \cdot & & \cdot & & \cdot \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_N & b_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \cdot \\ \cdot \\ d_{N-1} \\ d_N \end{bmatrix} \qquad \dots(a)$$