# Chapter 3
# Systems of Linear Equations

## Introduction

Many physical systems can be modeled by a set of linear equations, which describe relationships between system variables. In simple cases, there are two or three variables; in complex systems (for example, in a linear model of the economy of a country) there may be several hundred variables. Linear systems also arise in connection with many problems of numerical analysis. Examples of these are the solution of partial differential equations by finite difference methods, statistical regression analysis, and the solution of eigenvalue problems.

Hence there arises a need for rapid and accurate methods for solving **systems of linear equations**. The two methods of solving system of equations that are commonly known are the *direct method* and *indirect* (or *iterative method*). The direct method is based on the elimination of variables to transform the set of equations to a triangular form, the completed in a finite number of steps resulting in the exact solution and thus the amount of computation involved can be specified in advance. The method is independent of the accuracy desired.

The indirect methods always begins with an approximate solution, and obtains an improved solution with each step of the iteration but would require an infinite number of steps to obtain an exact solution in the absence of round – off errors. The accuracy of the solution unlike the direct method depends on the number of iterations performed. Usually 'iterative methods' are used for *spars*[1] matrices whereas for *dense*[2] matrices we use direct methods.

## Notation and definitions

We first consider an example in three variables:

$$\begin{aligned} x + y - z &= 2 \\ x + 2y + z &= 6 \\ 2x - y + z &= 1 \end{aligned}$$

a set of three linear equations in the three variables or unknowns *x, y, z*. During solution of such a system, we determine a set of values for *x, y* and *z* which satisfies each of the equations. In other words, if values *(X, Y, Z)* satisfy al1 equations simultaneously, then they constitute a solution of the system.

Consider now the general system of *n* equations in *n* variables:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

Obviously, the dots indicate similar terms in the variables and the remaining (*n* - 3) equations.

In this notation, the variables are denoted by $x_1, x_2, \ldots, x_n$; they are sometimes referred to as $x_i$ , $i = 1, 2, \cdot\cdot, n$. The **coefficients** of the variables may be detached and written in a coefficient matrix:

---

[1] *Sparse* matrices have few non-zero elements. Such types of matrices arise in partial differential equations.

[2] *Dense* matrices have few zero elements. Such matrices occur in science and engineering problems.

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

The notation $a_{ij}$ will be used to denote the coefficient of $x_j$ in the $i$-th equation. Note that $a_{ij}$ occurs in the $i$-th row and $j$-th column of the matrix.

The numbers on the right-hand side of the equations are called **constants**; they may be written in a column vector:

$$\mathbf{b} = [b_1, b_2, b_3, ..., b_n]^T$$

The coefficient matrix may be combined with the constant vector to form the **augmented matrix**:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix}$$

As a rule, one works in the **elimination method** directly with the augmented matrix.

**The existence of solutions**

For any particular solution of $n$ linear equations, there may be a single solution ($X_1$, $X_2$, . . . $X_n$), or no solution, or infinitely money solutions. In the theory of **linear algebra**, theorems are given and conditions stated which allow to make a decision regarding the category into which a given system falls. We shall not treat the question of existence of solutions in this content, but for the benefit of students, familiar with matrices and determinants, we state the theorem:

**Theorem**: A linear system of n equations in n variables with coefficient matrix **A** and non-zero constants vector **b** has a unique solution, if and only if the determinant of **A** is not zero.

If **b = 0**, the system has the trivial solution **x = 0**. It has no other solution unless the determinant of **A** is zero, in which case it has an infinite number of solutions.

If the determinant of **A** is non-zero, there exists an $n \times n$ matrix, called the **inverse of A** (denoted by $\mathbf{A}^{-1}$) such that the matrix product of $\mathbf{A}^{-1}$ and **A** is equal to the ***n* × *n*-identity** or **unit matrix I**. The elements of the identity matrix are 1 on the main diagonal and 0 elsewhere. Its algebraic properties include **Ix = x** for any $n \times 1$ vector **x**, and **IM** = **MI** = **M** for any $n \times n$ matrix **M**. For example, the 3×3 identity matrix is

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplication of the equation **Ax** = **b** from the left by the inverse matrix $\mathbf{A}^{-1}$ yields $\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$, whence the unique solution is $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ (since $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ and $\mathbf{Ix} = \mathbf{x}$). Thus, in principle, a linear system with a unique solution may be solved by first evaluating $\mathbf{A}^{-1}$ and then $\mathbf{A}^{-1}\mathbf{b}$. This approach is discussed in more detail in the

next section. The Gauss elimination method that we want to consider next is a more general and efficient direct procedure for solving systems of linear equations.

### *3.1 Direct Methods*

#### 3.1.1 Upper-triangular Linear Systems

We will now develop the **back-substitution algorithm,** which is useful for solving a linear system of equations that has an upper-triangular coefficient matrix. This will be incorporated in the algorithm for solving a general linear system in Section 3.2.

**Definition 3.1**

An $n \times n$ matrix $\mathbf{A} = (a_{ij})$ is called **upper triangular** provided that the elements satisfy $a_{ij} = 0$ whenever $i > j$. The $n \times n$ matrix $\mathbf{A} = (a_{ij})$ is called lower triangular provided that $a_{ij} = 0$ whenever $i < j$.

We will develop a method for constructing the solution to upper-triangular linear system of equations and leave the investigation of lower-triangular systems to the reader. If **A** is an upper-triangular matrix, then **AX=B** is said to be an **upper-triangular system** of linear equations and has the form

$$\begin{array}{rrrrrrrrr} a_{11}x_1 + & a_{12}x_2 + & a_{13}x_3 + & \cdots & + a_{1,n-1}x_{n-1} + & a_{1,n}x_n & = & b_1 \\ & a_{22}x_2 + & a_{23}x_3 + & \cdots & + a_{2,n-1}x_{n-1} + & a_{2n}x_n & = & b_2 \\ & & a_{33}x_3 + & \cdots & + a_{3,n-1}x_{n-1} + & a_{3n}x_n & = & b_3 \\ & & & & \vdots & \vdots & & \\ & & & & + a_{n-1,n-1}x_{n-1} + & a_{n-1,n}x_n & = & b_{n-1} \\ & & & & & a_{nn}x_n & = & b_n . \end{array} \quad (1)$$

To solve this system of equations where $a_{kk} \neq 0 \;\; for\; k = 1,2,...,n,$ we start from the last equation since it involves only $x_n$ so solving it first we have:

$$x_n = \frac{b_n}{a_{nn}} \quad (2)$$

Now $x_n$ is known and it can be used in the next-to-last equation:

$$x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}} .$$

Now $x_n$ and $x_{n-1}$ are used to find $x_{n-2}$

$$x_{n-2} = \frac{b_{n-2} - a_{n-2,n-1}x_{n-1} - a_{n-2,n}x_n}{a_{n-2,n-2}} .$$

Once the values $x_n$, $x_{n-1}$, …, $x_{k+1}$ are known, the general step is

$$x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj}x_j}{a_{kk}} \quad for \;\; k = n-1, n-2, ...1. \quad (3)$$

The uniqueness of the solution is easy to see. So in general the **back substation algorithm** is stated as below:

Given the upper triangular linear system (1) the values $x_1, x_2, x_3, ..., x_n$ are calculated by applying (2) : for k=n, $x_n = \frac{b_n}{a_{nn}}$ and applying (3) $for\ \ k = n-1, n-2, ...1$i.e

$$x_k = \frac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{a_{kk}} \quad for\ \ k = n-1, n-2, ...1.$$

**Example 1**: Use back substitution to solve the linear system

$$\begin{aligned} 4x_1 - x_2 + 2x_3 + 3x_4 &= 20 \\ -2x_2 + 7x_3 + 4x_4 &= -7 \\ 6x_3 + 5x_4 &= 4 \\ 3x_4 &= 6 \end{aligned}$$

**Solution**: Solving for $x_4$ in the last equation yields $x_4 = 6/3 = 2$.

Using $x_4 = 2$ in the third equation, we obtain

$$x_3 = \frac{4 - 5(2)}{6} = -1.$$

Now $x_3 = -1$ and $x_4 = 2$ are used to find $x_2$ in the second equation:

$$x_2 = \frac{-7 - 7(-1) - 4(2)}{-2} = 4.$$

Finally, $x_1$ is obtained using the first equation:

$$x_1 = \frac{20 + 1(4) - 2(-1) - 3(2)}{4} = 5.$$

The condition that $a_{kk} \neq 0$ is essential because equation (2) involves division by $a_{kk}$. If this requirement is not fulfilled, either no solution exists or infinitely many solutions exist.

**Example 2**: Show that there is no solution to the linear system

$$\begin{aligned} 4x_1 - x_2 + 2x_3 + 3x_4 &= 20 \\ 0x_2 + 7x_3 + 4x_4 &= -7 \\ 6x_3 + 5x_4 &= 4 \\ 3x_4 &= 6 \end{aligned} \qquad (3)$$

**Solution**: - Using the last equation in (3) we must have $x_4 = 2$, which is substituted into the second and third equations to obtain

$$\begin{aligned} 7x_3 - 8 &= -7 \\ 6x_3 + 10 &= 4 \end{aligned} \qquad (4)$$

The first equation in (4) implies that $x_3 = 1/7$, and the second equation implies that $x_3 = -1$. This contradiction leads to the conclusion that there is no solution to the linear system (3).

**Exercises**

In problems 1 through 3, solve the upper-triangular system

1. $$\begin{aligned} 3x_1 - 2x_2 + x_3 - x_4 &= 8 \\ 4x_2 - x_3 + 2x_4 &= -3 \\ 2x_3 + 3x_4 &= 11 \\ 5x_4 &= 15 \end{aligned}$$

2. $$\begin{aligned} 5x_1 - 3x_2 - 7x_3 + x_4 &= -14 \\ 11x_2 + 9x_3 + 5x_4 &= 22 \\ 3x_3 - 13x_4 &= -11 \\ 7x_4 &= 14 \end{aligned}$$

3. $$\begin{aligned} 4x_1 - x_2 + 2x_3 + 2x_4 - x_5 &= 4 \\ -2x_2 + 6x_3 + 2x_4 + 7x_5 &= 0 \\ x_3 - x_4 - 2x_5 &= 3 \\ -2x_4 - x_5 &= 10 \\ 3x_5 &= 6 \end{aligned}$$

In problems 4 and 5, solve the lower-triangular system.

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

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

6. Show that back substitution requires $(n^2 + n)/2$ multiplications or divisions, and $(n^2 - n)/2$ additions or subtractions. Hint: You can use the formula

$$\sum_{k=1}^{m} k = m(m+1)/2$$

7. Write a forward-substitution algorithm for solving a system of equations with a lower triangular coefficient matrix and draw a flow chart for the method.
8. Write an efficient computer program for:
   a. Back substitution method.
   b. Forward substitution method.

### 3.1.2 Gauss elimination method

In this section we develop a scheme for solving a general system **Ax = B** of $n \times n$ system of equations. The aim is to construct an equivalent upper-triangular system **Ux = Y** that can be solved by using back-substitution.

During transformation of a system to upper triangular form, one or more of the following elementary operations are used at every step:

1. **Interchanging** of two equations.
2. **Multiplication** of an equation by a non-zero constant;
3. **Subtraction** from one equation some nonzero multiple of another equation;

Mathematically speaking, it should be clear to the student that performing elementary operations on a system of linear equations leads to equivalent systems with the same solutions. This statement requires proof which may be found as a theorem in books on linear algebra. It forms the basis of all elimination methods for solving systems of linear equations.

**Example**: Find the parabola $y = a + bx + cx^2$ that passes through the three points (1,1), (2,– 1), and (3, 1).

**Solution**: First we obtain an equation relating the value of *x* to the value of *y*. It results the linear system

$$\begin{aligned} a + b + c &= 1 \quad at\ (1,1) \\ a + 2b + 4c &= -1 \quad at\ (2,-1) \\ a + 3b + 9c &= 1 \quad at\ (1,1) \end{aligned}$$

**Step 1**: The variable *a* is eliminated form the second and third equations by subtracting the first equation from them.

$$\begin{aligned} a + b + c &= 1 \\ b + 3c &= -2 \quad (R_2 - R_1) \\ 2b + 8c &= 0 \quad (R_3 - R_1) \end{aligned}$$

**Step 2**: The variable *b* is eliminated from the third equation in the last system by subtracting from it two times the second equation. We arrive at the equivalent upper-triangular system:

$$\begin{aligned} a + b + c &= 1 \\ b + 3c &= -2 \\ 2c &= 0 \quad (R_3 - 2R_2) \end{aligned}$$

The back substitution algorithm is now used to find the coefficients $c$=4/2=2, $b$ = – 2 – 3(2) = – 8, and $a$=1 – (– 8) – 2 = 7, and the equation of the parabola is $y = 7 - 8x + 2x^2$.

We can also solve the system **Ax = B** by performing elementary row operations on the augmented matrix **[A|B]**. In this case the number $a_{kk}$ in the coefficient matrix **A** that is used to eliminate $a_{ik}$, where k = *i* + 1, *i* + 2, …, n, is called the *k*th **pivotal element**, and the *k*th row is called **pivot row**.

**General treatment of the Gaussian elimination process**

We will now describe the application of the elimination process to a general $n \times n$ linear system, written in general notation, which is suitable for implementation on a computer (Pseudo-code).

The process transforming the general $n \times n$ linear system **Ax=B** into upper triangular system **Ux=b**, where the coefficient matrix A is non-singular can be described as below:

Recall that the original augmented matrix for the system can be written as:

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

We display the step by step transformation as follows:

**Step 1**: If necessary, switch rows so that $a_{11} \neq 0$; then eliminate the elements $a_{21}, a_{31}, \ldots, a_{n1}$ by calculating the multipliers $m_{i1}$ and subtracting $m_{i1}$ multiple of row 1 from row i.

$$
\begin{aligned}
&for\ \ i = 2\ to\ n \\
&\ m_{i1} = a_{i1} / a_{11}; \\
&a_{i1} = 0; \\
&\ for\ \ j = 2\ to\ n+1 \\
&\quad a_{ij} = a_{ij} - m_{i1} * a_{1j};
\end{aligned}
$$

This leads to the modified augmented system

$$
\left[\begin{array}{ccccc|c}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\
0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\
0 & a_{32} & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{n,n+1}
\end{array}\right]
$$

**Step 2**: If necessary, switch rows so that $a_{22} \neq 0$; then eliminate the elements $a_{32}, a_{42}, \ldots, a_{n2}$ by calculating the multipliers and performing row operation applying the algorithm

$$
\begin{aligned}
&for\ \ i = 3\ to\ n \\
&\ m_{i2} = a_{i2} / a_{22}; \\
&a_{i2} = 0; \\
&\ for\ \ j = 3\ to\ n+1 \\
&\quad a_{ij} = a_{ij} - m_{i2} * a_{2j};
\end{aligned}
$$

hence

$$
\left[\begin{array}{ccccc|c}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\
0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\
0 & 0 & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & a_{n3} & \cdots & a_{nn} & a_{n,n+1}
\end{array}\right]
$$

We continue to eliminate elements, going on to columns 3, 4, . . so that by the beginning of the *k*-th stage we have the augmented matrix

$$
\left[\begin{array}{cccccc|c}
a_{11} & a_{12} & \cdots & \cdots & \cdots & a_{1n} & a_{1,n+1} \\
0 & a_{22} & \cdots & \cdots & \cdots & a_{2n} & a_{2,n+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & a_{kk} & \cdots & a_{3n} & a_{3,n+1} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & a_{kn} & \cdots & a_{nn} & a_{n,n+1}
\end{array}\right]
$$

**Step *k***: If necessary, switch rows so that $a_{22} \neq 0$; then eliminate $a_{k+1,k}, a_{k+2,k}, \ldots, a_{k+n,kl}$ by calculating the multipliers and elementary row operations according the algorithm

$$
\begin{aligned}
&for\ \ i = k+1\ to\ n \\
&\quad m_{ik} = a_{ik} / a_{kk}; \\
&\quad a_{ik} = 0 \\
&\quad for\ \ j = k+1\ to\ n+1 \\
&\qquad a_{ij} = a_{ij} - m_{ik} * a_{kj};
\end{aligned}
$$

At the **end** of the *k*-th stage, we obtain the augmented system

$$
\left[\begin{array}{ccccccc|c}
a_{11} & a_{12} & \cdots & \cdots & \cdots & \cdots & a_{1n} & a_{1,n+1} \\
0 & a_{22} & \cdots & \cdots & \cdots & \cdots & a_{2n} & a_{2,n+1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & a_{kk} & a_{k,k+1} & \cdots & a_{3n} & a_{3,n+1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & a_{kn} & \cdots & a_{nn} & a_{n,n+1}
\end{array}\right]
$$

Continuing in this way, we obtain after $n$ -1 stages the **augmented matrix**

$$
\left[\begin{array}{ccccc|c}
a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\
0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\
0 & 0 & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & a_{nn} & a_{n,n+1}
\end{array}\right]
$$

Note that the original coefficient matrix has been transformed into the **upper triangular form**. So we now solve the last system by back substitution.

**Pivoting to avoid $a_{kk} = 0$**

If $a_{kk} = 0$, row k cannot be used to eliminate the elements in column k below the main diagonal. It is necessary to find row r, where $a_{rk} \neq 0$ and r > k, and then interchange row k and row r so that a nonzero pivot element is obtained. This process is called **pivoting**, and the criterion for deciding which row to choose is called a **pivoting strategy**. The trivial pivoting strategy is as follows. If $a_{kk} \neq 0$, do not switch rows. If $a_{kk} = 0$, locate the first row below k in which $a_{rk} \neq 0$ and switch rows k and r. This will result in a new element $a_{kk} \neq 0$, which is a nonzero pivot element.

**Counting arithmetic operations in the Gauss elimination method**

Numerical methods for solving systems of linear equations involve large numbers of arithmetic operations. For example, the **Gauss elimination** of section 3.1.2, according to Atkinson (1993), involves $(n^3 + 3n^2 - n)/3$ multiplications/divisions and $(2n^3 + 3n^2 - 5n)/6$ additions/subtractions in the case of a system with $n$ unknowns. Since round-off errors are propagated at each step of an algorithm, the growth of round-off errors can be such that, when $n$ is large, a solution differs greatly from the true one.

### 3.1.3 Gaussian Elimination with Partial Pivoting

In Gauss elimination, the buildup of round-off errors may be reduced by rearranging the equations so that the use of large multipliers in the elimination operations is avoided. The corresponding procedure is known as **partial pivoting** (or **pivotal condensation**). The general rule to follow involves: At each elimination

stage, rearrange the rows of the augmented matrix so that the new pivot element is larger in absolute value than (or equal to) any element beneath it in its column. i.e

- In the *k*th column, choose the *r*th row where
  $|a_{rk}| = \max\{|a_{kk}|, |a_{k+1,k}|, ..., |a_{n=1,k}|, |a_{nk}|\}$
- And interchange the *k*th row with the *r*th row. Now, each of the multipliers will be less than or equal to 1 in absolute value.
- Lastly Perform Gauss elimination.

The following example illustrates how the use of the trivial pivoting strategy in Gaussian elimination can lead to significant error in the solution of a linear system of equations.

**Example** The values $x_1 = x_2 = 1.000$ are the solutions to the system of equations

$$1.133x_1 + 5.281x_2 = 6.414$$
$$24.14x_1 - 1.210x_2 = 22.93$$

Use four-digit arithmetic and Gaussian elimination with trivial pivoting to find a computed approximate solution to the system.

**Solution** The multiplier $m_{21} = 24.14/1.133 = 21.31$, thus subtracting the $m_{21}$ multiple of row 1 from row 2 using four digits calculations i.e

$$1.133x_1 + 5.281x_2 = 6.414$$
$$24.14x_1 - 1.210x_2 = 22.93 \quad (R_2 - 21.31R_1)$$

We have the computed upper-triangular system is

$$1.133x_1 + 5.281x_2 = 6.414$$
$$-113.7x_2 = -113.8$$

Back substitution is used to compute $x_2 = -113.8/(-113.7) = 1.001$ *and* $x_1 = 0.9956$.

The error in the solution of this linear system is due to the magnitude of the multiplier $m_{21} = 21.31$. In the next example the magnitude of the multiplier $m_{21}$ is reduced by using partial pivoting as can be seen in the next example.

**Example** Use four-digit arithmetic and Gaussian elimination with partial pivoting to solve the linear system

$$24.14x_1 - 1.210x_2 = 22.93$$
$$1.133x_1 + 5.281x_2 = 6.414$$

**Solution**: This time $m_{21} = 1.133/24.14 = 0.04693$ and subtracting the $m_{21}$ multiple of row 1 from row 2 using four digits calculations i.e

$$24.14x_1 - 1.210x_2 = 22.93$$
$$1.133x_1 + 5.281x_2 = 6.414 \quad (R_2 - m_{21}R_1)$$

We have the computed upper-triangular system is

$$24.14x_1 - 1.210x_2 = 22.93$$
$$5.338x_2 = 5.338$$

Back substitution is used to compute $x_2 = 5.338/5.338 = 1.000$, *and* $x_1 = 1.000$.

**Ill-conditioning**

Certain systems of linear equations are such that their solutions are very sensitive to small changes (and therefore to errors) in their coefficients and constants. We give an example below in which 1 % changes in two coefficients change the solution by a factor of 10 or more. Such systems are said to be **ill-conditioned**. If a system is ill-conditioned, a solution obtained by a numerical method may differ greatly from the exact solution, even though great care is taken to keep round-off and other errors very small.

As an example, consider the system of equations:

$$2x + y = 4$$
$$2x + 1.01y = 4.02$$

which has the exact solution x =1, y = 2. Changing coefficients of the second equation by 1% and the constant of the first equation by 5% yields the system:

$$2x + y = 3.8$$
$$2.02x + y = 4.02$$

It is easily verified that the exact solution of this system is $x$ = 11, $y$ = -18.2. This solution differs greatly from the solution of the first system. Both these systems are said to be **ill-conditioned**.

Obtaining accurate solutions to ill-conditioned linear systems can be difficult, and many tests have been proposed for determining whether or not a system is ill-conditioned.

The question is why does such systems behave in such a manner and how do we identify them? A close analysis shows that most of the time the coefficient matrix of such systems is almost singular i.e |A| is very small. Indeed in order to determine whether the determinant of the coefficient matrix is "small," we need a reference against which the determinant can be measured. This reference is called the norm of the matrix, denoted by $\|\mathbf{A}\|$. We can then say that the determinant is small if

$$|\mathbf{A}| << \|\mathbf{A}\|.$$

At this junction given matrix $\mathbf{A} = (a_{ij})$, there are several ways of assigning norm to the matrix the two usual definitions of matrix norm i.e $\|\mathbf{A}\|$ are the following:

1. $\|\mathbf{A}\|_2 = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$ which is called **Euclidian norm.**
2. $\|\mathbf{A}\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$ which is called **maximum norm** or **infinite norm**.

**Note**: In this book unless otherwise mentioned, the norm of a matrix **A**, $\|\mathbf{A}\|$ is understood as the maximum norm. For instance given

$$\mathbf{A} = \begin{bmatrix} 8 & -6 & 2 \\ -4 & 11 & -7 \\ 4 & -7 & 6 \end{bmatrix}$$

Since

$$\sum_{j=1}^{3} |a_{1j}| = |8| + |-6| + |2| = 16$$

$$\sum_{j=1}^{3} |a_{2j}| = |-4| + |11| + |-7| = 22$$

$$\sum_{j=1}^{3} |a_{3j}| = |4| + |-7| + |6| = 17$$

And $\|\mathbf{A}\|_{\infty} = \max_{1 \le i \le 3} \sum_{j=1}^{3} |a_{ij}| = \max_{1 \le i \le 3} \{16, 22, 17\} = 22$

Back to the discussion of ill-conditioning, the important quantity that has relation with it is the **condition number**, which is defined as

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$$

It can be shown that for any nonsingular matrix **A,**

$$1 = \|1\| = \|\mathbf{A} \cdot \mathbf{A}^{-1}\| \le \|\mathbf{A}\| \|\mathbf{A}^{-1}\| = \kappa(\mathbf{A})$$

Hence a matrix A is well- conditioned if $\kappa(\mathbf{A})$ is close to 1, and is ill-conditioned when $\kappa(\mathbf{A})$ is significantly greater than 1. Also, it can be shown that the condition number $\kappa(\mathbf{A})$ measures the transfer of error from the matrix **A** and the vector **b** to the solution ***x***. The rule of thumb is that if $\kappa(\mathbf{A}) = 10^k$, then one can expect to lose at least k digits of precision in solving the system **A*x* = *b***.

In our example in the first equation the norm of the coefficient matrix

$$A = \begin{pmatrix} 2 & 1 \\ 2 & 1.01 \end{pmatrix} \text{ is } \|\mathbf{A}\| = 3.01 \text{ and that of it inverse}$$

$$\mathbf{A}^{-1} = \begin{pmatrix} 50.5 & -50 \\ -100 & 100 \end{pmatrix} \text{is } \|\mathbf{A}^{-1}\| = 200$$

Hence the condition number is

$\kappa(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| = (3.01)(200) = 600.02$, which is significantly larger than one indicating the system could be ill conditioned. So this result keeps us from making hasty accuracy decisions based on the residual of an approximation.

Consequently, if a system is ill-conditioned, then the usual procedure of checking a numerical solution by calculation of the residuals may not be valid. In order to see why this is so, suppose we have an approximation **X** to the true solution **x**. The **vector of residuals r** is then given by **r = b - AX = A(x - X)**. Thus **e = x - X** satisfies the linear system **Ae = r**.

In general, **r** will be a vector with small components. However, in an **ill-conditioned system**, even if the components of **r** are small so that it is `close' to 0, the solution of the linear system **Ae = r** could differ greatly from the solution of the system **Ae = 0**, namely **0**. It then follows that **X** may be a poor approximation to **x** despite the residuals in **r** being small.

**Exercise 3.1**

In Problems 1 through 4 solve the system **Ax = B** using Gaussian Elimination.

1. $\begin{aligned} 2x_1 + 4x_2 - 6x_3 &= -4 \\ x_1 + 5x_2 + 3x_3 &= 10 \\ x_1 + 3x_2 + 2x_3 &= 5 \end{aligned}$

2. $\begin{aligned} 2x_1 - 2x_2 + 5x_3 &= 6 \\ 2x_1 + 3x_2 + x_3 &= 13 \\ -x_1 + 4x_2 - 4x_3 &= 3 \end{aligned}$

3. $\begin{aligned} 2x_1 + 4x_2 - 4x_3 &= 12 \\ x_1 + 5x_2 - 5x_3 - 3x_4 &= 18 \\ 2x_1 + 3x_2 + x_3 + 3x_4 &= 8 \\ x_1 + 4x_2 - 2x_3 + 2x_4 &= 8 \end{aligned}$

4. $\begin{aligned} x_1 + 2x_2 - x_4 &= 9 \\ 2x_1 + 3x_2 - x_3 &= 9 \\ 3x_2 + x_3 + 3x_4 &= 26 \\ 5x_1 + 5x_2 + 2x_3 - 4x_4 &= 32 \end{aligned}$

5. Find the parabola $y = a + bx + cx^2$ that passes through (1, 4), (2,7), and (3, 14).

6. Find the parabola $y = a + bx + cx^2$ that passes through (1, 6), (2,5), and (3, 2).

7. Find the solution to the following linear system

$$\begin{aligned} x_1 + 2x_2 &= 7 \\ 2x_1 + 3x_2 - x_3 &= 9 \\ 4x_2 + 2x_3 + 3x_4 &= 10 \\ 2x_3 - 4x_4 &= 12 \end{aligned}$$

8. Find the solution to the following linear system

$$\begin{aligned} x_1 + x_2 &= 5 \\ 2x_1 - x_2 + 5x_3 &= -9 \\ 3x_2 - 4x_3 + 2x_4 &= 19 \\ 2x_3 + 6x_4 &= 2 \end{aligned}$$

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

a. $\begin{aligned} 0.03x_1 + 58.9x_2 &= 59.2 \\ 5.31x_1 - 6.10x_2 &= 47.0 \end{aligned}$

Actual solution (10, 1)'.

b. $\begin{aligned} 58x_1 + 0.03x_2 &= 59.2 \\ -6.10x_1 + 5.31x_2 &= 47.0 \end{aligned}$

Actual solution (I, 10)'.

*c*. $\begin{aligned} 3.03x_1 - 12.1x_2 + 14x_3 &= -119 \\ -3.03x_1 + 12.1x_2 - 7x_3 &= 120 \\ 6.11x_1 - 14.2x_2 + 21x_3 &= -139 \end{aligned}$

*Actual solution* (0,10,1/7)

*d*. $\begin{aligned} 3.3330x_1 + 15920x_2 + 10.333x_3 &= 7958 \\ 2.220x_1 + 16.710x_2 + 9.6120x_3 &= 0.965 \\ -1.5611x_1 + 51792x_2 - 16855x_3 &= 2.714 \end{aligned}$

*Actual solution* (1,0.5,−1)

10. Solve the following linear systems using Gaussian elimination with partial pivoting.

(*a*) $\begin{aligned} 2x_1 - 3x_2 + 100x_3 &= 1 \\ x_1 + 10x_2 - 0.001x_3 &= 0 \\ 3x_1 - 100x_2 + 0.01x_3 &= 0 \end{aligned}$

(*b*) $\begin{aligned} x_1 + 20x_2 - x_3 + 0.001x_4 &= 0 \\ 2x_1 - 5x_2 + 30x_3 - 0.1x_4 &= 1 \\ 5x_1 + x_2 - 100x_3 - 10x_4 &= 0 \\ 2x_1 - 100x_2 - x_3 + x_4 &= 0 \end{aligned}$

11. The Hilbert matrix is a classical ill conditioned matrix and small changes in its coefficients will produce a large change in the solution to the perturbed system.

    (a) Find the exact solution of **Ax=B** (leave all numbers as fractions and do exact arithmetic) using the Hilbert matrix of dimension 4 × 4:

$$\mathbf{A} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

    (b) Now solve **Ax** = **B** using four-digit rounding arithmetic.

12. Construct a program for a Gaussian elimination method without partial pivoting.

13. Many applications involve matrices with many zeros. Of practical importance are tri-diagonal systems(see problem 7 and 8) of the form

$$\begin{aligned} d_1x_1 + c_1x_2 &= b_1 \\ a_1x_1 + d_2x_2 + c_2x_3 &= b_2 \\ a_2x_2 + d_3x_3 + c_3x_3 &= b_3 \\ \cdots\quad \cdots\quad \cdots &\quad \cdots \\ a_{n-2}x_{n-2} + d_{n-1}x_{n-1} + c_{n-1}x_n &= b_{n-1} \\ a_{n-1}x_{n-1} + d_nx_n &= b_n \end{aligned}$$

    Construct a program that will solve a tri-diagonal system. You may assume that row interchanges are not needed and that row k can be used to eliminate $x_k$ in row k + 1.

### 3.1.4 Gauss-Jordan Method

The Gauss-Jordan method consists of transforming the linear system **A*x*** = **b** into an equivalent system **I*x*** = **b'**, where **I** is the identity matrix of order n so that **x** = **b'** is the solution of the original linear system.

**Example** Using Gauss-Jordan method sol1ve the system of equations

$$\begin{aligned} x + y + z &= 2 \\ 2x + 3y + z &= 3 \\ x - y - 2y &= -6 \end{aligned}$$

**Solution**: We start with the augmented matrix and use the first row to create zeros in the first column. (This corresponds to using the first equation to eliminate x from the second and third equations.)

$$\begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 3 & 1 & 3 \\ 1 & -1 & -2 & -6 \end{bmatrix} \overset{\begin{matrix} R_2 - 2R_1, \\ R_3 - R_1 \end{matrix}}{\approx} \begin{bmatrix} 1 & 1 & 1 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & -2 & -3 & -8 \end{bmatrix}$$

Create appropriate zeros in column 2:

$$\begin{matrix} R_1 - R_2 \\ R_3 + 2R_2 \end{matrix} \overset{\approx}{\begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & -5 & -10 \end{bmatrix}} (-1/5)R_3 \overset{\approx}{\begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 2 \end{bmatrix}}$$

Creating zeros in column 3:

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

The last matrix corresponds to the system 1

$$\begin{aligned} x &= -1 \\ y &= 1 \\ z &= 2 \end{aligned}$$

Consequently the solution is given by $x = -1, y = 1, z = 2.$

**Description of the Method**

Let us consider the following linear system defined by

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} a_{1,n+1} \\ a_{2,n+1} \\ \vdots \\ a_{n,n+1} \end{bmatrix}$$

**Step 1**: Assume that $a_{11} \neq 0.$ The normalization operation replaces $a_{11}$ by 1 in the augmented matrix [**A**, **b**] and this is possible by pre-multiplying row 1 of [**A**, **b**] by $1/a_{11}$.

So, after normalization the first row of the matrix [**A**, **b**] becomes

$$a_{1j}^1 = a_{1j} / a_{11} \quad for \;\; j = 2,...,(n+1)$$

Now we make the non-diagonal elements of the first column of A to become zero. This possible by multiplying $\mathbf{R_1}$ by $a_{i1}$ and adding it to the *ith row* $i \geq 2$. Then we get the new coefficients defined by

$$a_{ij}^1 = a_{ij} - a_{i1}a_{1j}^1 \qquad for\ i = 2,...,n;\ \ j = 2,...,(n+1)$$

Thus, the new system $[\mathbf{A}, \mathbf{b}]^1$ is written as

$$\left[\begin{array}{ccccc|c} 1 & a_{12}^1 & a_{13}^1 & \cdots & a_{1n}^1 & a_{1,n+1}^1 \\ 0 & a_{22}^1 & a_{23}^1 & \cdots & a_{2n}^1 & a_{2,n+1}^1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & a_{n2}^1 & a_{n3}^1 & \cdots & a_{nn}^1 & a_{n,n+1}^1 \end{array}\right]$$

**Step 2**: Assume, that $a_{22}^1 \neq 0$. The normalization operation changes the pivot element $a_{22}^1$ by 1. This is resulted by pre-multiplying the second row of the augmented matrix by $1/a_{22}^1$. Thus, the new coefficients of the second row will become

$$a_{2j}^2 = a_{2j}^1 / a_{22}^1 \qquad for\ \ j = 3,4,...,(n+1)$$

The coefficient above the diagonal is made zero by pre-multiplying $R_2$ by $a_{12}^1$ and subtracting the multiple from $R_1$. In fact, the row coefficients of the first row are given by

$$a_{1j}^2 = a_{1j}^1 - a_{12}^1 a_{2j}^2 \qquad for\ \ j = 3,4,...,(n+1)$$

But to make the coefficients below the diagonal element of the second column of **A** zero we multiply the second row by $a_{i2}^1$ *while* $i \geq 3,...,n$.

Therefore, the general formula for the new coefficients can be written as

$$a_{ij}^2 = a_{ij}^1 - a_{i2}a_{2j}^2 \qquad \forall j = 3,...,(n+1);\ i = 1,...,n\ \ with\ \ i \neq 2,$$

and the new system becomes

$$\left[\begin{array}{ccccc|c} 1 & 0 & a_{13}^2 & \cdots & a_{1n}^2 & a_{1,n+1}^2 \\ 0 & 1 & a_{23}^2 & \cdots & a_{2n}^2 & a_{2,n+1}^2 \\ 0 & 0 & a_{33}^2 & \cdots & \vdots & a_{3,n+1}^2 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & a_{n3}^2 & \cdots & a_{nn}^2 & a_{n,n+1}^2 \end{array}\right]$$

Step k. Assume $a_{kk}^{k-1} \neq 0$. Then pre-multiplying the kth row of $[\mathbf{A}, \mathbf{b}]^{k-1}$ by $1/a_{kk}^{k-1}$, we change $a_{kk}^{k-1}\ to 1$, but the other coefficients of the k-th row are changed. In fact, the new coefficients will be

$$a_{kj}^k = a_{kj}^{k-1} / a_{kk}^{k-1} \qquad for\ j = k,...,(n+1)$$

The non-diagonal coefficients of the k-th column are made zero by pre-multiplying the k-th row by $a_{ik}^{k-1}$ for i=1,…,n and $i \neq k$. This implies that the new coefficients will be defined by

$$a_{ij}^k = a_{ij}^{k-1} - a_{ik}^{k-1}.a_{kj}^k \qquad for\ j = (k+1),...,(n+1);\ i = 1,2,...n; i \neq k$$

Now, we can write Gauss-Jordan algorithm for a linear system.

**Gauss-Jordan Algorithm for solving a Linear System** [**A**, **b**]

| Gauss-Jordan Algorithm without pivoting |
|---|
| (i) Transforming [A, b] into [I, b']. For k = 1, 2, …,n,<br>$a_{kj}^{k} = a_{kj}^{k-1} / a_{kk}^{k-1} \quad \forall j = (k+1),...,(n+1)$<br>$a_{ij}^{k} = a_{ij}^{k-1} - a_{ik}^{k-1} a_{kj}^{k}, \quad \forall j = (k+1),...,(n+1)$<br>$i = 1,...,n; (i \neq k)$<br>(ii) Solution of the system<br>$x_i = a_{i,n+1}^{n} \quad for \quad i = 1, 2, ..., n.$ |

The number of operations necessary to transform the system [**A**, **b**]$^{k}$ is: (n-1)(n-k+1) additions, (n-1)(n-k+1) multiplications (n-k+1)division. Therefore, the total number of operations necessary to transform the original system [**A**, **b**] into [**I**, **b**'] is

$$\text{No. of additions} = \sum_{k=1}^{n}(n-1)(n-k+1) = n(n^2-1)/2.$$

$$\text{No. of additions} = \text{No of multiplication} = n(n^2-1)/2.$$

$$\text{No. of division} = \sum_{k=1}^{n}(n-k+1) = n(n+1)/2.$$

Note for n>20, the number of operations will be of the order of $n^3/2$.

**Exercise**: Find the number of operations necessary to solve a linear system of order n by Gauss elimination and Gauss-Jordan methods, for n = 5 and n = 10?

**Comparison of Gauss-Jordan and Gaussian Elimination**

The method of Gaussian elimination is in general **more efficient** than Gauss-Jordan elimination in that it involves fewer operations of addition and multiplication. It is during the back substitution that Gaussian elimination picks up this advantage. Particularly in larger systems of equations, many more operations are saved in Gaussian elimination during back substitution. The reduction in the number of operations not only saves time on a computer but also increases the accuracy of the final answer. With large systems, the method of Gauss-Jordan elimination involves approximately 50% more arithmetic operations than does Gaussian elimination.

Gauss-Jordan elimination, on the other hand, has the advantage of being more straightforward for hand computations. It is superior for solving small systems.

### 3.1.5 Matrix Inversion Using Jordan Elimination

Let A be an $n \times n$ matrix.

1. Adjoin the identity $n \times n$ matrix $\mathbf{I_n}$ to **A** to form the matrix [**A**: $\mathbf{I_n}$]
2. Compute the reduced echelon form of [**A**: $\mathbf{I_n}$]. If the reduced echelon form is of the type [$\mathbf{I_n}$: **B**], then **B** is the inverse of **A**. If the reduced echelon form is not of the type [$\mathbf{I_n}$: **B**], in that the first $n \times n$ submatrix is not $\mathbf{I_n}$, then **A** has no inverse.

The following examples illustrate the method.

**Example:** Determine the inverse of the matrix

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

**Solution**: Applying the method of Gauss-Jordan elimination, we get

$$[A:I_3]=\begin{bmatrix} 1 & -1 & -2 & 1 & 0 & 0 \\ 2 & -3 & -5 & 0 & 1 & 0 \\ -1 & 3 & 5 & 0 & 0 & 1 \end{bmatrix} \overset{\begin{matrix} R_2+(-2)R_1 \\ R_3+R_1 \end{matrix}}{\approx} \begin{bmatrix} 1 & -1 & -2 & 1 & 0 & 0 \\ 0 & -1 & -1 & -2 & 1 & 0 \\ 0 & 2 & 3 & 1 & 0 & 1 \end{bmatrix}$$

$$\overset{(-1)R_2}{\approx} \begin{bmatrix} 1 & -1 & -2 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 & -1 & 0 \\ 0 & 2 & 3 & 1 & 0 & 1 \end{bmatrix}$$

$$\overset{\begin{matrix} R_1+R_2 \\ R_3+(-2)R_2 \end{matrix}}{\approx} \begin{bmatrix} 1 & 0 & -1 & 3 & -1 & 0 \\ 0 & 1 & 1 & 2 & -1 & 0 \\ 0 & 0 & 1 & -3 & 2 & 1 \end{bmatrix}$$

$$\overset{\begin{matrix} R_1+R_3 \\ R_2+(-1)R_3 \end{matrix}}{\approx} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 5 & -3 & -1 \\ 0 & 0 & 1 & -3 & 2 & 1 \end{bmatrix}=[I_3:A^{-1}]$$

Thus

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

Observe that we can solve the system of equations

$$\begin{aligned} x_1-x_2-2x_3 &= 1 \\ 2x_1-3x_2-5x_3 &= 3 \\ -x_1+3x_2+5x_3 &= -2 \end{aligned}$$

using the inverse of the coefficient which is the result of the problem above using the result $\mathbf{x = A^{-1}B}$ i.e

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}=\begin{bmatrix} 0 & 1 & 1 \\ 5 & -3 & -1 \\ -3 & 2 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}=\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

So the solution of the system is $x_1=1, x_2=-2,$ *and* $x_3=1.$

**Exercise**

1. Solve (if possible) each of the following system using the method of Gauss-Jordan elimination.

*a*)
$$\begin{aligned} x_1+4x_2+3x_3 &= 1 \\ 2x_1+8x_2+11x_3 &= 7 \\ x_1+6x_2+7x_3 &= 3 \end{aligned}$$

*b*)
$$\begin{aligned} x_1+4x_2+x_3 &= 2 \\ x_1+2x_2-x_3 &= 0 \\ 2x_1+6x_2 &= 3 \end{aligned}$$

*c*) $\begin{aligned} x_1 + 2x_2 + 3x_3 &= 8 \\ 3x_1 + 7x_2 + 9x_3 &= 26 \\ 2x_1 + 6x_3 &= 11 \end{aligned}$

*d*) $\begin{aligned} x_1 + 2x_2 + 8x_3 &= 7 \\ 2x_1 + 4x_2 + 16x_3 &= 14 \\ x_2 + 3x_2 &= 4 \end{aligned}$

*e*) $\begin{aligned} x_1 + x_2 + x_3 - x_4 &= -3 \\ 2x_1 + 3x_2 + x_3 - 5x_4 &= -9 \\ x_1 + 3x_2 - x_3 - 6x_4 &= -7 \\ -x_1 - x_2 - x_3 &= 1 \end{aligned}$

*e*) $\begin{aligned} x_1 - x_2 + 2x_3 &= 7 \\ 2x_1 - 2x_2 + 2x_3 - 4x_4 &= 12 \\ -x_1 + x_2 - x_3 + 2x_4 &= -4 \\ -3x_1 + x_2 - 8x_3 - 10x_4 &= -29 \end{aligned}$

2. Solve the following system of linear equations by applying the method of Gauss-Jordan elimination to a large augmented matrix that represents two systems wih the same matrix of coefficients.

$$\begin{aligned} x_1 + x_2 + 5x_3 &= b_1 \\ x_1 + 2x_2 + 8x_3 &= b_2 \\ 2x_1 + 4x_2 + 16x_3 &= b_3 \end{aligned} \quad \text{for} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 10 \end{bmatrix} \textit{and} \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} \text{ in turn}$$

3. Solve the following systems of equations by determining the inverse of the matrix of coefficients and then using matrix multiplication.

*a*) $\begin{aligned} x_1 + 2x_2 - x_3 &= 2 \\ x_1 + x_2 + 2x_3 &= 0 \\ x_1 - x_2 - x_3 &= 1 \end{aligned}$

*b*) $\begin{aligned} x_1 - x_2 &= 1 \\ x_1 + x_2 + 2x_3 &= 2 \\ x_1 + 2x_2 + x_3 &= 0 \end{aligned}$

4. Draw a flow chart for Gauss-Jordan elimination method

### 3.1.6 LU Decomposition

A third method for the solution of general systems of linear algebraic equations is the **LU decomposition** method. The objective of this method is to find a lower triangular factor **L** and an upper triangular factor **U** such that the system of equations can be transformed according to

$$\mathbf{A \cdot x = b \rightarrow (L \cdot U) \cdot x = A^* \cdot x = b^*}$$

The matrix **A*** in the above equation is the matrix **A** after row exchanges have been made to allow the factors **L** and **U** to be computed accurately; the vector **b*** is the vector **b** after an identical set of row exchanges.

A decomposition in which each diagonal element $l_{ii}$ of **L** has a unit value is known as the **Doolittle method**; one in which each diagonal element $u_{ii}$ of **U** has a unit value is known as the **Crout method**. Another method in which corresponding diagonal elements $l_{ii}$ and $u_{ii}$ are equal to each other is known as the **Cholesky method**. Regardless of which method is used to obtain the factors **L** and **U** of **A***, the methods described in sections 3.1.1 and 3.1.2 for triangular matrices are used to obtain **x** by solving

$$\mathbf{L \cdot y = b^*; \quad U \cdot x = y}.$$

Here we consider the Doolittle and Crout method and leave the discussion of the Cholesky method as reading assignment.

**Doolittle Method**

From the description of the Doolittle method we can infer that a given 4 × 4 matrix can be decompose into the form

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix}$$

The condition that A is nonsingular implies that $u_{kk} \neq 0$ for all *k*. The notation for the entries in L is $m_{ij}$, and the reason for the choice of $m_{ij}$ instead of $l_{ij}$, will be pointed out in the next example.

**Example** Use Gaussian elimination to construct the triangular factorization of the matrix

$$\mathbf{A} = \begin{pmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{pmatrix} \begin{bmatrix} 2 & 1 & 3 \\ 4 & 1 & 7 \\ -6 & -2 & -12 \end{bmatrix}$$

The matrix L will be constructed from an identity matrix placed at the left. For each row operation used to construct the upper-triangular matrix, the multipliers $m_{ij}$ will be put in their proper places at the left. Start with

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 2 & 1 & 3 \\ 4 & 1 & 7 \\ -6 & -2 & -12 \end{bmatrix}$$

Row 1 is used to eliminate the elements of A in column 1 below $a_{11}$. The multiples $m_{21} = 2$ *and* $m_{31} = -3$ of row 1 are subtracted from rows 2 and 3, respectively. These multipliers are put in the matrix at the left and the result is

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 \\ 0 & -1 & 1 \\ 0 & 1 & -3 \end{pmatrix}$$

Row 2 is used to eliminate the elements of **A** in column 2 below $a_{22}$. The multiple $m_{32} = -1$ of the second row is subtracted from row 3, and the multiplier is entered in the matrix at the left and we have desired triangular factorization of **A** as:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 \\ 0 & -1 & 1 \\ 0 & 0 & -2 \end{pmatrix}$$

$$\mathbf{L} \qquad\qquad \mathbf{U}$$

**Theorem (Direct Factorization A=LU. No Row Interchange).** Suppose that Gaussian elimination, without row interchanges, can be successfully performed to solve the general linear system **AX = B.** Then the matrix **A** can be factored as the product of a lower-triangular matrix **L** and an upper-triangular matrix **U**:

**A = LU**.

Furthermore, **L** can be constructed to have 1's on its diagonal and **U** will have nonzero diagonal elements. After finding **L** and **U**, the solution **X** is computed in two steps

1. Solve **LY= B** for **Y** using forward substitution.
2. Solve **UX=Y** for **X** using back substitution.

**Proof**: We will show that, when the Gaussian elimination process is followed and B is stored in column N+1 of the augmented matrix, the result after the upper triangularization step is the equivalent upper-triangular system **UX = Y**. The matrices L, U, B, and Y will have the form

$$\mathbf{L}=\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ m_{31} & m_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \cdots & 1 \end{pmatrix} \mathbf{U}=\begin{pmatrix} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{pmatrix} \mathbf{B}=\begin{pmatrix} a_{1n+1}^{(0)} \\ a_{2n+1}^{(0)} \\ a_{3n+1}^{(0)} \\ \vdots \\ a_{nn+1}^{(0)} \end{pmatrix} \mathbf{Y}=\begin{pmatrix} a_{1n+1}^{(0)} \\ a_{2n+1}^{(1)} \\ a_{3n+1}^{(2)} \\ \vdots \\ a_{nn+1}^{(n-1)} \end{pmatrix}$$

*Remark*: To find just **L** and **U**, the(n + 1)st column is not needed.

Firs store the coefficients in the augmented matrix. The superscript on $a_{ij}^{0}$ means that this is the first time that a number is stored in location (*i*, *j*).

$$\left(\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & a_{1\,n+1}^{(0)} \\ a_{21}^{(0)} & a_{22}^{(0)} & a_{23}^{(0)} & \cdots & a_{2n}^{(0)} & a_{2\,n+1}^{(0)} \\ a_{31}^{(0)} & a_{32}^{(0)} & a_{33}^{(0)} & \cdots & a_{3n}^{(0)} & a_{3\,n+1}^{(0)} \\ \vdots & \vdots & \vdots & & \vdots & \\ a_{n1}^{(0)} & a_{n1}^{(0)} & a_{n1}^{(0)} & \cdots & a_{nn}^{(0)} & a_{n\,n+1}^{(0)} \end{array}\right)$$

**Step 1**: Eliminate $a_{i1}$ for i from 2 to n in rows 2 through n and store the multiplier $m_{i1}$, used to eliminate $a_{i1}$ in row i, in the matrix at location (i, 1)

for *i* = 2 to n

$m_{i1} = a_{i1}^{(0)} / a_{11}^{(0)}$

$a_{i1} = m_{i1};$

for *j* = 2 to n +1

$a_{ij}^{(1)} = a_{ij}^{(0)} - m_{i1} a_{1j}^{(0)}$

End for

End for

So at the end of this elimination we have the augmented matrix

$$\left(\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & a_{1\,n+1}^{(0)} \\ m_{21} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & a_{2\,n+1}^{(1)} \\ m_{31} & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} & a_{3\,n+1}^{(1)} \\ \vdots & \vdots & \vdots & & \vdots & \\ m_{n1} & a_{n1}^{(1)} & a_{n1}^{(1)} & \cdots & a_{nn}^{(1)} & a_{n\,n+1}^{(1)} \end{array}\right)$$

**Step 2**: Eliminate $a_{i2}$ for i from 3 to n in rows 3 through n and store the multiplier $m_{i1}$, used to eliminate $a_{i2}$ in row i, in the matrix at location (i, 2)

for $i$ = 3 to n

$m_{i2} = a_{i2}^{(1)} / a_{12}^{(1)}$

$a_{i2} = m_{i2}$;

for $j$ = 3 to n +1

$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i2} a_{2j}^{(1)}$

End for

End for

At the end of this elimination the augmented has the form

$$\left(\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n}^{(0)} & a_{1\ n+1}^{(0)} \\ m_{21} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & a_{2\ n+1}^{(1)} \\ m_{31} & m_{32} & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & a_{3\ n+1}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots & \\ m_{n1} & m_{n2} & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} & a_{n\ n+1}^{(2)} \end{array}\right)$$

***Step k***: This is the general step. Eliminate $a_{ik}$ in row k + 1 through n-1 and store the multipliers at the location ($i$, $k$).

for $i$ = k+1 to n

$m_{ik} = a_{ik}^{(k)} / a_{1k}^{(k)}$

$a_{ik} = m_{ik}$;

for $j$ = k+1 to n +1

$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$

End for

End for

The final matrix after n-1 steps of elimination is of the form

$$\left(\begin{array}{ccccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & \cdots & a_{1n-1}^{(0)} & a_{1n}^{(0)} & a_{1\ n+1}^{(0)} \\ m_{21} & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n-1}^{(1)} & a_{1n}^{(0)} & a_{2\ n+1}^{(1)} \\ m_{31} & m_{32} & a_{33}^{(2)} & \cdots & a_{3n-1}^{(2)} & a_{1n}^{(0)} & a_{3\ n+1}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & m_{n3} & \cdots & m_{n\,n-1} & a_{nn}^{(0)} & a_{n\ n+1}^{(2)} \end{array}\right)$$

The upper-triangular process is now complete. Notice that once array is used to store the elements of both **L** and **U**. the 1's of **L** are not stored, nor are the 0's of **L** and **U** that lies above and below the diagonal, respectively. Only the essential coefficients needed to reconstruct **L** and **U** are stored!

The last part of the proof is to verify the product **LU = A** and it is left as exercise.

**Example**: - Solve the following system of equations using Doolittle decomposition and Crout decomposition methods.

$$\begin{aligned} 2x_1 + x_2 + 3x_3 &= -1 \\ 4x_1 + x_2 + 7x_3 &= 5 \\ -6x_1 - 2x_2 - 12x_3 &= -2 \end{aligned}$$

**Solution**: We can write above equation in the matrix form of **A*x* = B** i.e

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 1 & 7 \\ -6 & -2 & -12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 5 \\ -2 \end{bmatrix}$$

But using the Doolittle decomposition method we have demonstrated that matrix **A** has the following **LU** decomposition

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 3 \\ 4 & 1 & 7 \\ -6 & -2 & -12 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 3 \\ 0 & -1 & 1 \\ 0 & 0 & -2 \end{bmatrix}$$

So first we solve the system **L*y* =B** i.e

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 5 \\ 2 \end{bmatrix}$$

using forward substitution and find that $y_1 = -1,\ y_2 = 7,\ and\ \ y_3 = 2$

Lastly we solve the system **U*x* = *y*** i.e

$$\begin{bmatrix} 2 & 1 & 3 \\ 0 & -1 & 1 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 7 \\ 2 \end{bmatrix}$$

By back ward substitution method for ***x*** which leads us to the solution $x_1 = 5, x_2 = -8, x_3 = -1.$

**Crout Method**

Let **A*x*=B** be a system of n equations in n variables, where A is a non-singular matrix that has **LU** decomposition. An alternative approach that involves a **U** matrix with 1's on the diagonal which is called ***Crout decomposition***. Like the *Doolittle method* the system can thus be written as

**LU*x* = B**

and the method involves writing this system as two subsystems, one of which is lower triangular and the other upper triangular of the form **U*x* = y** and **L*y* = B**.

In practice, we first solve **L*y* = B** for ***y*** and then solve **U*x* = *y*** to get the solution ***x***. Let us now consider the system of equations below

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{23}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

The above equation can be written as

$$\mathbf{A}x = \mathbf{B}$$

Where $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ *and* $\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

Let $\mathbf{A} = \mathbf{LU}$

Where $\mathbf{L} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{31} & l_{33} \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$ So that

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

Then $l_{11} = a_{11}, l_{21} = a_{21}, l_{31} = a_{31},$

$l_{11}u_{12} = a_{12}$ *then* $u_{12} = \dfrac{a_{12}}{a_{11}}$

$l_{11}u_{13} = a_{13}$ *then* $u_{13} = \dfrac{a_{13}}{a_{11}}$

$l_{21}u_{12} + l_{22} = a_{22}$ *then* $l_{22} = a_{22} - a_{21}.u_{12}$

$l_{31}u_{12} + l_{32} = a_{32}$ *then* $l_{32} = a_{32} - a_{31}.u_{12}$

$l_{21}.u_{13} + l_{22}.u_{23} = a_{23}$ *then* $u_{23} = \dfrac{1}{l_{22}}\left(a_{23} - a_{21}.u_{13}\right)$

$l_{31}u_{13} + l_{32}u_{23} + l_{33} = a_{33}$ *then* $l_{33} = \left[a_{33} - a_{31}.u_{13} - l_{32}.u_{23}\right]$

We may generalize the above relations by the following concise series of formulas:

*for* $i = 1,2,...,n$

$l_{i1} = a_{i1}$

*for* $j = 2,3,...,n$

$u_{1j} = a_{1j} / l_{11}$

*for* $j = 2,3,...,n-1$

*for* $i = j, j+1,...,n$

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}$$

*for* $k = j+1, j+2,...,n$

$$u_{jk} = \frac{a_{jk} - \sum_{i=1}^{j-1} l_{ji}u_{ik}}{l_{jj}}$$

and

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

Now $\mathbf{A}x = \mathbf{B}$ becomes $\mathbf{LU}x = \mathbf{B}$.

Let **Ux = y** then **Ly=B**

And from **Ly = B** we get

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{31} & l_{33} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_2 \end{bmatrix}=\begin{bmatrix} b_1 \\ b_2 \\ b_2 \end{bmatrix}$$

Thus

$$\begin{aligned} l_{11}.y_1 &= b_1 \\ l_{21}y_1 + l_{22}.y_2 &= b_2 \\ l_{31}y_1 + l_{32}.y_2 + l_{33}y_3 &= b_3 \end{aligned}$$

Solve for $y_1, y_2, y_3$ by forward substitution.

We know that **Ux = y**

Thus

$$\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

Consequently

$$\begin{aligned} x_1 + u_{12}x_2 + u_{13}x_3 &= y_1 \\ x_2 + u_{23}x_3 &= y_2 \\ x_3 &= y_3 \end{aligned}$$

By the backward substitution, we get the values of $x_1, x_2, x_3$.

**Example**: - Solve the following system of equations using Crout decomposition methods.

$$\begin{aligned} 2x_1 + x_2 + 3x_3 &= -1 \\ 4x_1 + x_2 + 7x_3 &= 5 \\ -6x_1 - 2x_2 - 12x_3 &= -2 \end{aligned}$$

**Solution**: We can write above equation in the matrix form of **A*x* = B** i.e

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 1 & 7 \\ -6 & -2 & -12 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} -1 \\ 5 \\ -2 \end{bmatrix}$$

Using Crout decomposition method, let **A = LU** then we must have

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & 1 & 7 \\ -6 & -2 & -12 \end{bmatrix}=\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{31} & l_{33} \end{bmatrix}\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Now solving the above matrix equation we have

$$l_{11} = 2, l_{21} = 4, l_{31} = -6, u_{12} = 1/2, u_{13} = 3/2$$

$$l_{22} = -1, l_{32} = 1, u_{23} = -1, l_{33} = -2.$$

Then we have

$$L = \begin{bmatrix} 2 & 0 & 0 \\ 4 & -1 & 0 \\ -6 & 1 & -2 \end{bmatrix}, \text{ and } U = \begin{bmatrix} 1 & 1/2 & 3/2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

writing Ly =B we have

$$\begin{bmatrix} 2 & 0 & 0 \\ 4 & -1 & 0 \\ -6 & 1 & -2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 5 \\ -2 \end{bmatrix}$$

Solving the above system by for ward substitution i.e

$$\begin{aligned} 2y_1 &= -1 \quad \textit{then } y_1 = -1/2 \\ 4y_1 - y_2 &= 5 \quad \textit{then } y_2 = -7 \\ -6y_1 + y_2 - 2y_3 &= -2 \quad \textit{then } y_3 = -1 \end{aligned}$$

Now again writing Ux =y we have

$$\begin{bmatrix} 1 & 1/2 & 3/2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -7 \\ -1 \end{bmatrix}$$

Solving the above equation by substitution we have

$$\begin{aligned} x_1 + 1/2x_2 + 3/2x_3 &= -1/2 \\ x_2 - x_3 &= -7 \\ x_3 &= -1 \end{aligned}$$

The solution to the given system is therefore $x_1 = 5, x_2 = -8, x_3 = -1.$

The method of **LU** decomposition can be applied to any system of n equations in n variables, **Ax = B**, that can be transformed into an upper triangular form **U** using row operations that involve adding multiples of rows to rows. In general, if in transforming say the coefficient matrix **A** row interchanges are required to arrive at the upper triangular form, then matrix **A** does not have an **LU** decomposition and the method cannot be used to solve the system **Ax = B** as it is.

The total number of arithmetic operations needed to solve a system of equations using **LU** decomposition is exactly the same as that needed in Gaussian elimination. However, if the linear system is to be solved many times, with the same coefficient matrix **A** but with different column matrix **B**, it is not necessary to decompose the matrix each time if the factors are served. This is the reason the **LU** decomposition method is usually chosen over the elimination method.

The other fact that **LU** decomposition is used to solve systems of equations on computers where applicable, rather than Gaussian elimination, is a result of the usefulness of **LU** decomposition of a matrix for many types of computations. The inverse of a triangular matrix can be computed very efficiently, and the determinant of a triangular matrix is the product of its diagonal elements.

**Exercise**:

1. Solve **LY = B**, and verify that **B = AX** for (a) $\mathbf{B} = [-4 \;\; 10 \;\; 5]^T$ and (b) $\mathbf{B} = [20 \;\; 49 \;\; 32]^T$, where **A = LU** is

$$\begin{bmatrix} 2 & 4 & -6 \\ 1 & 5 & 3 \\ 1 & 3 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/2 & 1/3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -6 \\ 0 & 3 & 6 \\ 0 & 0 & 3 \end{bmatrix}$$

2. Solve **LY = B**, and verify that **B = AX** for (a) $\mathbf{B} = [7 \;\; 2 \;\; 10]^T$ and (b) $\mathbf{B} = [23 \;\; 35 \;\; 7]^T$, where **A = LU** is

$$\begin{bmatrix} 1 & 1 & 6 \\ -1 & 2 & 9 \\ 1 & -2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 6 \\ 0 & 3 & 15 \\ 0 & 0 & 12 \end{bmatrix}$$

3. Find the Doolittle and Crout decomposition A = LU for the matrices

a) $\begin{bmatrix} -5 & 2 & -1 \\ 1 & 0 & 3 \\ 3 & 1 & 6 \end{bmatrix}$ b) $\begin{bmatrix} 1 & 0 & 3 \\ 3 & 1 & 6 \\ -5 & 2 & -1 \end{bmatrix}$

c) $\begin{bmatrix} 4 & 2 & 1 \\ 2 & 5 & -2 \\ 1 & -2 & 7 \end{bmatrix}$ d) $\begin{bmatrix} 1 & -2 & 7 \\ 4 & 2 & 1 \\ 2 & 5 & -2 \end{bmatrix}$

e) $\begin{bmatrix} 4 & 8 & 4 & 0 \\ 1 & 5 & 4 & -3 \\ 1 & 4 & 7 & 2 \\ 1 & 3 & 0 & -2 \end{bmatrix}$ f) $\begin{bmatrix} 1 & 2 & 4 & 1 \\ 2 & 8 & 6 & 4 \\ 3 & 10 & 8 & 8 \\ 4 & 12 & 10 & 6 \end{bmatrix}$

4. Consider the matrix

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

a) Show that **A** has no LU decomposition

b) Interchange the rows of **A** so that this can be done.

5. Solve the follow system of equations using

i) Doolittle method

a)
$$\begin{aligned} x_1 - 3x_2 + 4x_3 &= 12 \\ -x_1 + 5x_2 - 3x_3 &= -12 \\ 4x_1 - 8x_2 + 23x_3 &= 58 \end{aligned}$$

ii) Crout method

b)
$$\begin{aligned} 5x_1 - x_2 - 2x_3 &= 142 \\ x_1 - 3x_2 - x_3 &= -30 \\ 2x_1 - x_2 - 3x_3 &= 5 \end{aligned}$$

6. Write a C++ program for Crout method

### *3.2 Iterative methods*

We have discussed a number of elimination methods for solving systems of linear equations. We now introduce two-called **iterative methods** for solving systems of n equations in n variables that have a unique solution.

Consider the system

$$Ax = b$$

Let $x^{(0)}$ be the initial approximation vector and $x_T$ the vector of the true solution. We generate a sequence of vectors $x^{(0)}, x^{(1)}, ..., x^{(n)}$ which converge to the true solution $x_T$. In doing so, we must consider two things. That is, the convergence of the sequence and the stop criteria since we have to stop the iteration after n steps.

#### 3.2.1 Gauss Jacobi Iterative Method

Consider the system

$$\mathbf{Ax} = \mathbf{b},$$

which is an n × n system of equations of the form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n &= b_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n &= b_n \end{aligned}$$

and let $\mathbf{x}^{(0)} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix}$ be the vector of the initial approximation.

Step I: Obtain $x_1^{(1)}$ from the first equation as a function of $x_i^{(0)}, i = 2,3,...,n$ as follows:

$$x_1^{(1)} = \frac{b_1 - (a_{12}x_2^{(0)} + a_{13}x_3^{(0)} + ... + a_{1n}x_n^{(0)})}{a_{11}}$$

Similarly

$$x_2^{(1)} = \frac{b_2 - (a_{21}x_1^{(0)} + a_{23}x_3^{(0)} + ... + a_{2n}x_n^{(0)})}{a_{22}} \qquad \text{(from the second equation)}$$

$$x_k^{(1)} = \frac{b_k - (a_{k2}x_1^{(0)} + ... + a_{k,k-1}x_{k-1}^{(0)} + a_{k,k+1}x_{k+1} + ... + a_{1n}x_n^{(0)})}{a_{kk}} \quad \text{(from the kth equation)}$$

and

$$x_n^{(1)} = \frac{b_n - (a_{n1}x_1^{(0)} + a_{n2}x_2^{(0)} + ... + a_{nn-1}x_{n-1}^{(0)})}{a_{nn}} \qquad \text{(from the last equation)}$$

After similar steps we can see that at the (k+1)th step $x_i^{(k+1)}$, $i = 1,2,....,n$ from the previous

$x_i^{(k)}$, $i = 1,2,....,n$ by the formula

$$x_i^{(k+1)} = \frac{b_i - \sum_{\substack{i=1 \\ j \neq i}}^{n} a_{ij} x_j^k}{a_{ii}}, \qquad i = 1,2,...,n; k = 0,1,...$$

**Termination of Iterations**

Suppose we denote the residue vector by $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$, *i.e.* $r_i^{(k)} = \left( b_i - \sum_{j=1}^{n} a_{ij} x_j^{(k)} \right) \ \forall i = 1,2,...,n.$ Then the standard criterion for the termination of the iteration is

$$\frac{\left\|\mathbf{r}^{(k)}\right\|}{\left\|\mathbf{b}\right\|} < \varepsilon \qquad \text{where } \varepsilon \text{ is arbitrarily small.}$$

Another standard termination condition for the relative improvement for **x** is stated as

$$\frac{\left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}\right\|}{\left\|\mathbf{x}^{(k+1)}\right\|} < \varepsilon$$

This condition is 'practically' equivalent to the previous condition for the termination of the iteration.

Sometimes, we also stop the iteration procedure when $\left\|\mathbf{x}^{k} - \mathbf{x}^{(k+1)}\right\| < \varepsilon$.

**Convergence condition**

The next theorem, which we present without proof, gives one of conditions under which the Jacobi iterative method can be used.

**Definition**: A matrix **A** of dimension n × n is said to be **diagonally dominated** provided that

$$|a_{kk}| > \sum_{\substack{j=1 \\ j \neq k}}^{n} |a_{kj}| \text{ for k = 1, 2, …,n.}$$

**Theorem 2:** Suppose that A is a diagonally dominated matrix. Then the system **A*x* = B**, has a unique solution and the Jacobi method will converge to this solution, no matter what the initial values.

**Example:** Solve the following system equations by Jacobi method.

$$\begin{aligned} 6x + 2y - z &= 4 \\ x + 5y + z &= 3 \\ 2x + y + 4z &= 27 \end{aligned}$$

**Solution**: Firs let us investigate the convergence condition for the system. The matrix coefficients is

$$A = \begin{bmatrix} 6 & 2 & -1 \\ 1 & 5 & 1 \\ 2 & 1 & 4 \end{bmatrix}$$

Comparing the magnitude of the diagonal elements with the sums of the magnitudes of the other elements in each row, we get the results show in table below. Observe that the diagonal elements dominate the rows; thus the Jacobi method can be used.

| Row | Absolute value of diagonal elements | Sum of absolute values of other elements in the row |
|---|---|---|
| 1 | 6 | \|2\| + \|-1\| =3 |
| 2 | 5 | \|1\| + \| 1\| =2 |
| 3 | 4 | \|2\| + \| 1\| = 3 |

Now to solve the system rewrite the equations as follows, isolating x in the first equation, y in the second equation, and z in the third equation.

$$x = \frac{4 - 2y + z}{6}$$
$$y = \frac{3 - x - z}{5} \qquad (2)$$
$$z = \frac{27 - 2x - y}{4}$$

Now make an estimate of the solution, say $x = 1, y = 1, z = 1.$ The accuracy of the estimate affects only the speed with which we get a good approximation to the solution. Let us label these values $x^{(0)}, y^{(0)},$ *and* $z^{(0)}$. They are called the **initial values** of the iterative process.

$$x^{(0)} = 1, y^{(0)} = 1, z^{(0)} = 1$$

Substitute these values into the right-hand side system (2) to get the next set of values in the iterative process.

$$x^{(1)} = 0.5, y^{(1)} = 0.2, z^{(1)} = 6\,1$$

These values of x,y, and z are now substituted into system (2) again to get

$$x^{(2)} = 1.6, y^{(2)} = -0.7, z^{(2)} = 6.45$$

This process can be repeated to get $x^{(3)}, y^{(3)}, z^{(3)},$ and so on. Repeating the iteration will give a better approximation to the exact solution at each step. For this straightforward system of equations, the solution can easily be seen to be

$$x = 2, y = -1, z = 6.$$

### 3.2.2 Gauss-Seidel Method

The Gauss-Seidel method is a refinement of the Jacobi method that usually (but not always) gives more rapid convergence. The latest value of each variable is substituted at each step in the iterative process. This method like the Jacobi method converges if the coefficient matrix A is diagonally dominant.

So given an nxn system of equations the general iteration process can be written as

$$x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \ldots - a_{1n}x_n^{(k)}) / a_{11}$$
$$x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \ldots - a_{2n}x_n^{(k)}) / a_{22}$$
$$\vdots \qquad\qquad \vdots$$
$$x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \ldots - a_{n,n-1}x_{n-1}^{(k+1)}) / a_{nn}$$

Here, as in the case of the Jacobi method, we assume that the pivots $a_{ii}$ are non-zero.

**Gauss-Seidel Algorithm**

| Gauss-Seidel Algorithm for the solution of Ax = b |
|---|
| Let A, b, $x^{(0)}$,$\varepsilon_1$, $\varepsilon_2$ and $k_{max}$ be given k=1,2,…, $k_{max}$, |
| 1. $x_i^{(k+1)} = \left[ b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right] / a_{ii}$ , $i = 1,2,...,n$ |
| 2. Termination condition: $\left\| x_i^{(k+1)} - x_i^{(k)} \right\| < \varepsilon_1$ or $\frac{\left\| x_i^{(k+1)} - x_i^{(k)} \right\|}{\left\| x_i^{(k+1)} \right\|} < \varepsilon_2$, i = 1,2, …,n. |

**Example**: Let us consider the previous system of equations. As before, let us take our initial guess to be

$$x^{(0)} = 1, y^{(0)} = 1, z^{(0)} = 1$$

Substituting the latest values of each variable into (2) system gives

$$x^{(1)} = \frac{4 - 2y^{(0)} + z^{(0)}}{6} = 0.5$$

$$y^{(1)} = \frac{3 - x^{(1)} - z^{(0)}}{5} = 0.3$$

$$z^{(1)} = \frac{27 - 2x^{(1)} - y^{(1)}}{4} = 6.4250$$

Observe that we have used $x^{(1)}$, the most up-to-date value of x, to get $y^{(1)}$. We have used $x^{(1)}$ and $y^{(1)}$ to get $z^{(1)}$. Continuing, we get

$$x^{(2)} = \frac{4 - 2y^{(1)} + z^{(1)}}{6} = 1.6375$$

$$y^{(2)} = \frac{3 - x^{(2)} - z^{(1)}}{5} = -1.0125$$

$$z^{(2)} = \frac{27 - 2x^{(2)} - y^{(2)}}{4} = 6.4250$$

The next two tables below give the results obtained for this system of equations using both methods. They illustrate the more rapid convergence of the Gauss-Seidel method to the exact solution $x = 2, y = -1, z = 6.$ And the last table gives the difference between the solution $x^{(6)}, y^{(6)}, z^{(6)}$ obtained in the two methods after six iterations and the actual solution $x = 2, y = -1, z = 6.$ The Gauss-Seidel method converges much more rapidly than the Jacobi method.

**Jacobi Method**

| **Index** | X | y | Z |
|---|---|---|---|
| Initial Guess | 1 | 1 | 1 |
| 1 | 0.5 | 0.2 | 6 |

| | | | |
|---|---|---|---|
| 2 | 1.6 | -0.7 | 6.45 |
| 3 | 1.975 | -1.01 | 6.125 |
| 4 | 2.024167 | -1.02 | 6.015 |
| 5 | 2.009167 | -1.007833 | 5.992917 |
| 6 | 2.001431 | -1.000417 | 5.997375 |

**Gauss-Seidel Method**

| **Index** | X | y | Z |
|---|---|---|---|
| Initial Guess | 1 | 1 | 1 |
| 1 | 0.5 | 0.3 | 6.425 |
| 2 | 1.6375 | -1.0125 | 6.184375 |
| 3 | 2.034896 | -1.043854 | 5.993516 |
| 4 | 2.013537 | -1.001411 | 5.993594 |
| 5 | 1.998597 | -0.998597 | 5.999949 |
| 6 | 1.999524 | -0.9998945 | 6.000212 |

**The comparison table**

| | $\left\|\mathbf{x}^{(6)} - 2\right\|$ | $\left\|\mathbf{y}^{(6)} - (-1)\right\|$ | $\left\|\mathbf{z}^{(6)} - 6\right\|$ |
|---|---|---|---|
| Jacobi Method | 0.001431 | 0.000417 | 0.002625 |
| Gauss-Seidel Method | 0.000476 | 0.0001055 | 0.000212 |

**Comparison of Gaussian Elimination and Gauss-Seidel**

***Limitations*** Gaussian elimination is a finite method and can be used to solve any system of linear equations. The Gauss-Seidel method converges only for special systems of equations; thus it can be used only for those systems.

***Efficiency*** The efficiency of a method is a function of the number of arithmetic operations (addition, subtraction, multiplication, and division) involved in each method. For a system of *n* equation in *n* variables, where the solution is unique and the value of *n* is large, Gaussian elimination involves $2n^3/3$ arithmetic operations to solve the problem, while Gauss-Seidel requires approximately $2n^2$ arithmetic operations per iteration. Therefore if the number of iterations is less than or equal to n/3, the iterative method requires fewer arithmetic operations.

***Accuracy*** In general, when Gauss-Seidel is applicable, it is more accurate than the Gaussian elimination.

***Storage*** Iterative methods are, in general, more economical in core-storage requirements of a computer.

**Exercise**

In problems 1 to 6:

a) Start with $\mathbf{x}^0 = \mathbf{0}$ and use Jacobi iteration to find $\mathbf{P}_k$ for k = 1, 2, 3. Will Jacobi iteration converge to the solution?

b) Start with $\mathbf{x}^0 = \mathbf{0}$ and use Gauss-Seidel iteration to find $\mathbf{P}_k$ for k = 1, 2, 3. Will Gauss-Seidel iteration converge to the solution?

1. $4x - y = 15$
   $x + 5y = 9$

2. $2x + 3y = 1$
   $7x - 2y = 1$

3. $5x - y + z = 10$
   $2x + 8y - z = 11$
   $-x + y + 4z = 3$

4. $2x + 8y - z = 11$
   $5x - y + z = 10$
   $-x + y + 4z = 3$

5. $x - 5y - z = -8$
   $4x + y - z = 13$
   $2x - y - 6z = -2$

6. $4x + y + 4z = 13$
   $x - 5y - z = -8$
   $2x - y - 6z = -2$

7. Write a program for Jacobi and Gauss-Seidel methods and solve the problems 1-6 using a tolerance $10^{-8}$.
8. In Theorem 2 the condition that **A** be diagonally dominated is a sufficient but not necessary condition. Use both of your programs that u developed in problem 7 and several different initial guesses on the following system of equations. Note the Jacobi iteration appears to converge, while the Gauss-Seidel iteration diverges.

$$\begin{aligned} x \qquad\quad &+ z = 2 \\ -x + y \quad &= 0 \\ x + 2y &- 3z = 0 \end{aligned}$$