An Introduction To Numerical Methods and Analysis Using MATLAB

Dr. Rizwan Butt[1]
Department of Mathematics, College of Science
King Saud University, P. O. Box 2455
Riyadh, 11451, Saudi Arabia.

September 14, 2021

[1]E-mail: rizwanbu@ksu.edu.sa - http://faculty.ksu.edu.sa/rizwanbutt/default.aspx

Contents

Chapter 1

Introduction to Numerical Methods

1.1 Introduction

I have written this book as an introductory course in numerical methods and numerical analysis for mathematicians, computer scientists, engineers, and other scientists. Numerical analysis is the branch of mathematics concerned with the theoretical foundations of numerical algorithms for the solution of problems arising in scientific applications. The subject addresses a variety of questions ranging from the approximation of functions and integrals to the approximate solution of algebraic, transcendental, differential and integral equations, with particular emphasis on the stability, accuracy, efficiency and reliability of numerical algorithms.
The intention of this book is to provide a gentle and sympathic introduction to many of the problems of scientific computing, and the wide variety of methods used for their solutions. The presentation of each numerical method is based on the successful teaching methodology of providing examples and geometric motivation for a method, and a concise statement of the steps to carry out the computation, before giving a mathematical derivation of the process or a discussion after more theoretical issues that are relevant to the use and understanding of the topic. Each topic illustrated by examples that range in complexity from very simple to moderate. Geometrical or graphical illustrations are included whenever they appropriate.
This book is concerned with the practical solution of problems on computers. In the process of problem solving, it is possible to distinguish several more or less distinct phases. The first phase is *formulation*. In formulating a mathematical model of a physical situation, the scientist should take into account beforehand the fact that he expects to solve his problem on a computer. He will therefore provide for specific objectives, proper input data, adequate checks, and for the type and amount of output. Once a problem has been formulated, numerical methods, together with a preliminary error analysis, must be devised for solving the problem. A numerical method which can be used to solve a problem will be called an *algorithm*. An algorithm is a complete and unambiguous set of procedures leading to the solution of a mathematical problem. The selection or construction of appropriate algorithms properly falls within the scope of numerical analysis. Having decided on a specific algorithm or set of algorithms for solving the problem, the numerical analyst should consider all the sources of error that may affect the results. He must consider how much accuracy is required, estimate the magnitude of the round-off and discretization error, determine an appropriate step size or the number of iterations required, provide for adequate checks on the accuracy, and make allowance for corrective action in cases of nonconvergence.

The third phase of problem solving is *programming*. The programmer must transform the suggested algorithm into a set of unambiguous step-by-step instructions to the computer. the first step in this procedure is called *flow charting*. A flow chart is simply a set of procedures, usually in logical block form, which the computer will follow. It may be given in graphical or procedural statement form. The complexity of the problem and the amount of detail included. However, it should be possible for someone other than the programmer to follow the flow of information from the chart. The flow chart is an effactive aid to the programmer, who must translate its major functions into machine code, and, at the same time, it is an effactive means of communication to others who wish to understand what the program does. In this book we sometimes use flow charts in graphical form, but more often in procedural statement form. Having produced a flow chart, the programmer must transform the indicated procedures into a set of machine instructions. This may be done directly in machine language or procedure-oriented language (sometimes called an algorithmic language). In this book MATLAB language is used exclusively. The practical justification of the methods is presented through computer examples through the use of MATLAB. In recent years, the number of MATLAB users has dramatically increased and now includes professionals who were trained in other high-level languages, Fortran, C, etc. but are now switching to MATLAB as well as students who are learning MATLAB as their first programming language. The surge of popularity in MATLAB is related to the increasing popularity of UNIX and computer graphics. To what extend numerical computations in the future will be programmed in MATLAB is uncertain. Nonetheless, there is no question that a need exists for comprehensive text especially geared to the requirements of those who want to learn, or use, numerical methods in MATLAB. This book has been written in response to this need.
The objectives of using MATLAB in this book include: *(1)* to be easily understood by undergraduate students with minimal knowledge of MATLAB, *(2)* to enable students to practice the methods in MATLAB, *(3)* to provide the short programs that can be easily used for scientific applications with or without modifications, and *(4)* to provide software that are easy to understand.

To provide maximum teaching flexibility, each chapter and each section begins with the basic, elementary material and gradually builds up to the more advanced material. The level of mathematical justification is determined largely by the desire to keep the mathematical prerequisites to a minimum. Thus, for example, no knowledge of linear algebra is assumed beyond the basic matrix algebra, and analytical results are based on a sound knowledge of the calculus.
In elementary calculus we learn how to differentiate and integrate to get exact answers to remarkably diverse range of realistic problems that could not be solved by purely algebraic methods. Unfortunately, from a practical point of view, the techniques of elementary (or even advanced) calculus alone are not adequate for solving calculus type problems such as solving polynomial equations of degree greater than four or even a simple equation such as

$$x = \cos x,$$

also, to evaluate integrals of type

$$\int_a^b e^{x^2} dx \quad \text{and} \quad \int_a^b \frac{\sin x}{x} dx; \qquad \text{etc.},$$

it is impossible to get the exact solutions of these problems. Even when an analytical solution can be found it may be of more theoretical than practical. Fortunately, one rarely needs exact answers.

Indeed, in the *real world* the problems themselves are usually inexact because they are generally possessed in terms of parameters that are measured, hence only approximate. What we are likely to require in a realistic situation is not an exact answer but rather one having a prescribed accuracy. The basic approach used to solve problems in numerical analysis is the algorithm which is used to describe a step-by-step procedure and requires a finite number of steps. So a numerical method is an algorithm which consists of a sequence of arithmetic and logical operations and which produces an approximate solution to within any prescribed accuracy. There are different numerical methods for the solution of one problem but the particular method chosen depends on the context from which the problem is taken.

1.1.0.1 Types of Numerical Methods

There are two basic types of numerical methods, direct numerical and indirect (iterative) numerical methods.

Direct methods compute the solution to a problem in a finite number of steps. These methods would give the precise answer if they were performed in infinite precision arithmetic. Examples include Gaussian elimination, the LU factorization method for solving systems of linear equations. In practice, finite precision is used and the result is an approximation of the true solution (assuming stability). In the absence of rounding errors, direct methods would deliver an exact solution.

In contrast to direct methods, *iterative methods* are not expected to terminate in a finite number of steps. Starting from an initial guess, iterative methods form successive approximations that converge to the exact solution only in the limit. In computational mathematics, an iterative method is a mathematical procedure that generates a sequence of improving approximate solutions for a class of problems. A specific implementation of an iterative method, including the termination criteria, is an algorithm of the iterative method. An iterative method is called convergent if the corresponding sequence converges for given initial approximations. A mathematically rigorous convergence analysis of an iterative method is usually performed. A convergence test, often involving the residual, is specified in order to decide when a sufficiently accurate solution has (hopefully) been found. Even using infinite precision arithmetic these methods would not reach the solution within a finite number of steps (in general). Iterative methods are often the only choice for nonlinear equations. However, iterative methods are often useful even for linear problems involving a large number of variables (sometimes of the order of millions), where direct methods would be prohibitively expensive (and in some cases impossible) even with the best available computing power. Examples include Newton's method, bisection method, and Jacobi iteration. In computational matrix algebra, iterative methods are generally needed for large problems.

An iterative method for the given problem converges means:- approximate values should come in side the given interval **I**- difference between two successive approximations should be small. Otherwise diverges. An iterative process may converge or diverge. If the divergence occurs, the procedure should be terminated because there may be no solution. We can restart the procedure by changing the initial approximation if necessary. But in the case of convergence we have to apply some stopping procedures to end the computations. In the following there are some more stopping criterion that can be used, each of them can be apply to any iterative technique considered in this chapter. By selecting a tolerance $\epsilon > 0$ and generate approximate solutions $x_1, x_2, \ldots, x_n$ until one of the

following conditions is satisfied:

$$|x_n - x_{n-1}| < \epsilon \quad \text{or} \quad \frac{|x_n - x_{n-1}|}{|x_n|} < \epsilon, \qquad x_n \neq 0.$$

Sometimes difficulties can arise using any of these stopping criteria. For example, there exist sequence $\{x_n\}_0^\infty$ with the property that the differences $(x_n - x_{n-1})$ converge to zero while the sequence itself diverges. It is also possible for $f(x_n)$ to be close to zero while x_n differs significantly from α. Without additional knowledge about $f(x)$ or α, the above second inequality is the best stooping criterion to apply because it tests relative error. Also, one of the other stopping criteria is to use a fixed number of iterations, and then the final approximation x_n may be considered as the value of the required root. This type of stopping criteria is helpful when the convergence is very slow. It is important to note that in considering whether an iteration *converges* or not, it may be necessary to ignore the first few iterations since the procedure may appear diverge initially, even though it ultimately converges.
Iterative methods are more common than direct methods in numerical analysis. Some methods are direct in principle but are usually used as though they were not. For these methods the number of steps needed to obtain the exact solution is so large that an approximation is accepted in the same manner as for an iterative method. The numerical methods deal with numbers. We exam the sources of various types of computational errors.

1.2 Error Analysis

In generally numerical methods give an approximate solution (in number) of the given problem. How good is the approximate answer, we have to check by using the error analysis theory. There are basically two ways to know about the resulting approximation:- by using actual (exact) error and the error bound (upper bound) formulas of the using numerical methods.

1.2.0.1 What is Error

An approximate number p is a number that differs but slightly from an exact number α. We write

$$p \approx \alpha.$$

By error E of an approximate number p, we mean the difference between the exact number α and its computed approximation p. Thus we define

$$E = \alpha - p. \tag{1.1}$$

If $\alpha > p$, the error E is positive, and if $\alpha < p$, the error E is negative. In many situations, the sign of the error may not be known and might even be irrelevant. Therefore, we define *absolute error* as

$$|E| = |\alpha - p|. \tag{1.2}$$

The *relative error* RE of an approximate number p is the ratio of the absolute error of the number to the absolute value of the corresponding exact number α. Thus

$$RE = \frac{|\alpha - p|}{|\alpha|}, \qquad \alpha \neq 0. \tag{1.3}$$

If we approximate $\frac{1}{3}$ by 0.333, we have

$$E = \frac{1}{3} \times 10^{-3} \qquad \text{and} \qquad RE = 10^{-3}.$$

Note that relative error is generally a better measure of the extend of error than the actual error. But one should also note that relative error is undefined if the exact answer is equal to zero. Generally, we shall be interested in E (or sometimes $|E|$) rather than RE, but when the true value of a quantity is very small or very large, relative errors are more meaningful. For example, if the true value of a quantity is 10^{15}, and error of 10^6 is probably not serious, but this is more meaningfully expressed by saying that $RE = 10^{-9}$. In actual computation of the relative error, we shall often replace the unknown true value by the computed approximate value. Sometimes the quantity

$$\frac{|\alpha - p|}{|\alpha|} \times 100\%, \tag{1.4}$$

is defined as *percentage error*. From the above example, we have

$$PE = 0.001 \times 100 = 0.1\%.$$

In investigating the effect of the total error in various methods, we shall often mathematically derive an error, called, *error bound* and which is a limit on how large the error can be. We shall have the reason to compute error bounds in many situations. This applies to both absolute and relative errors. Note that the error bound can be much larger than the actual error and that this is often the case in practice. Any mathematically derived error bound must account for the worst possible case that can occur and is often based upon certain simplifying assumptions about the problem which in many practical cases cannot be actually tested. For the error bound to be used in any practical way, the user must have a good understanding of how the error bound was derived in order to know how crude it is, that is, how likely it is to over estimate the actual error. Of course, whenever possible, our goal is to eliminate or lesser the effects of errors, rather than trying to estimate them after they occur.

1.3 Sources of Errors

In analysing the accuracy of numerical result, one should be aware of the possible sources of error in each stage of the computational process and of the extend to which these errors can affect the final answer. We will consider that there are three types of errors which occur in a computation. We discuss them step by step as follows.

1.3.1 Human Error

These types of errors arise when the equations of the mathematical model are formed, due to sources such as the idealistic assumptions made to simplify the model, inaccurate measurements of data, miscopying of figures, the inaccurate representation of mathematical constants (for example, if the constant π occurs in an equation, we must replace π by 3.1416 or 3.141593, etc.).

1.3.2 Truncation Error

This type of error is caused when we are forced to use mathematical techniques which give approximate, rather than exact, answer. For example, suppose that we use the Maclaurin's series expansion to represent $\sin x$, so that

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

If we want a number that approximates $\sin(\frac{\pi}{2})$, we must terminate the expansion in order to obtain

$$\sin(\frac{\pi}{2}) = \frac{\pi}{2} - \frac{(\pi/2)^3}{3!} + \frac{(\pi/2)^5}{5!} - \frac{(\pi/2)^7}{7!} + E,$$

where E is the truncation error introduced in the calculation. Truncation errors in numerical analysis usually occur because many numerical methods are iterative in nature, with the approximations theoretically becoming more accurate as we take more iterations. As a practical matter, we must stop the iteration after a finite number of steps, thus introducing a truncation error. The Taylor series is the most important means used to derive numerical schemes and analysis truncation errors.

1.3.3 Round-off Error

This type of errors are associated with the limited number of digits numbers in the computer. For example, by rounding off 1.32463672 to six decimal places to give 1.324637. Any further calculation involving such a number will also contain an error. Round-off numbers according to following rules:

1. If first discarded digit is less than 5, leave the remaining digits of number unchanged, that is, $48.47263 \approx 48.4726$.

2. If the first discarded digit is exceeds 5, add 1 to the retained digit. For example, $48.4726 \approx 48.473$.

3. If the first discarded digit is exactly 5 and there are nonzero among those discarded, add, 1 to the last retained digit. For example, $3.0554 \approx 3.06$.

4. If the first discarded digit is exactly 5 and all other discarded digits are zero, the last retained digit is left unchanged if it is *even*, otherwise 1 is added to it. For example,

$$\begin{aligned} 3.05500 &\approx 3.06 \\ 3.04500 &\approx 3.04. \end{aligned}$$

with these rules, the error is never larger in magnitude than one-half unit of the place of the *nth* digit in the rounded number. To understand the nature of round-off errors, it is necessary to learn the ways numbers are stored and additions and subtractions are performed in a computer. •

A solution is *correct within k decimal places* if the error is less than 0.5×10^{-k}.

If x^* is an approximation to x, then we say that x^* approximates x to k *significant digits* if k is the largest nonnegative integer for which $\left|\frac{x - x^*}{x}\right| < 5 \times 10^{-k}$. •

Chapter 2

Solution of Nonlinear Equations

2.1 Introduction

In this chapter we study one of the fundamental problems of numerical analysis, namely the numerical solution of nonlinear equations. Most equations arising in practice are nonlinear and are rarely of a form which allows the roots to be determined exactly. Consequently, numerical methods are used to solve nonlinear algebraic equations when the equations prove intractable to ordinary mathematical techniques. These numerical methods are all iterative, and they may be used for equations that contain one or several variables. These techniques can be divided into two categories; one-point (need one initial approximation) and two- point (need two initial approximations) methods.
A nonlinear equation in this chapter may be considered any one of the following types:

1. An equation may be an *algebraic equation (a polynomial equation of degree n)* expressible in the form:
$$a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = 0, \quad a_n \neq 0, \quad n > 1,$$
where $a_n, a_{n-1}, \ldots, a_1$ and a_0 are constants. For example, the following equations are nonlinear.
$$x^2 + 5x + 6 = 0; \quad x^3 = 2x + 1; \quad x^{200} - 2x + 1 = 0.$$

2. The power of the unknown variable (not a positive integer number) involved in the equation must be difficult to manipulate. For example, the following non-polynomial equations are nonlinear
$$x^{-1} + 2x = 1; \quad \sqrt{x} + x = 2; \quad x^{2/3} + \frac{2}{x} + 4 = 0.$$

3. An equation may be a *transcendental equation*, the equation which involves the trigonometric functions, exponential functions and logarithmic functions. For example, all the following transcendental equations are nonlinear
$$x = cos(x); \qquad e^x + x - 10 = 0; \qquad x + \ln x = 10.$$

Definition 2.1 (Root of an Nonlinear Equation)

Consider a nonlinear equation

$$f(x) = 0, \tag{2.1}$$

and assume that $f(x)$ is a continuous function. An number α for which $f(\alpha) = 0$ is called a root of the equation, or a zero of the function $f(x)$. •

The process of finding the approximate values of the roots of the nonlinear equation $f(x) = 0$ can be divided into two stages: (a) location of the roots, and (b) computation of the values of the roots with the specified degree of accuracy.

2.1.0.1 Location of Roots

An interval $[a, b]$ be the location of a root α if $f(\alpha) = 0$ for $a < \alpha < b$. To locate roots of nonlinear equation, we use graphical method and an analytical method called method of tabulation.

2.1.0.2 Graphical Method

Here, we draw the graph of a function $y = f(x)$ in a rectangular coordinate system and the abscissas of the points where graph intersects the x-axis are the required roots of the nonlinear equation. Practically, this way to locate a root α of nonlinear equation is not so accurate, specially, if the exact value of x where the graph intersects is in the decimal form. Also, the sketching of the function $y = f(x)$ is itself a complicated problem.
Some times, we find the approximate of the root of $f(x) = 0$ by dividing all the terms of the equation into groups, one of them is written on the left-hand side of the equation and the other on the right-hand side, that is, the given equation can be represented as $f_1(x) = f_2(x)$. The abscissas of the points of intersection of the graphs of $f_1(x)$ and $f_2(x)$ are the roots of the nonlinear equation. For example, the Figure 2.1 shows the graph of $y = x^3 - 3x^2 + 2$ and the graph of $y = x^3$ and $y = 3x^2 - 2$. From the graph of the function $y = x^3 - 3x^2 + 2$, we noted that the curve cuts the x-axis at three points and, so, the equation has three real roots. The roots belong to the intervals $[-1, 0]$, $[0, 2]$, and $[2, 3]$. While the graph $y = x^3$ and $y = 3x^2 - 2$ shows that the abscissas of the points of intersection of the graphs of these functions are roots of the equation. Again we have the same intervals $[-1, 0]$, $[0, 2]$, and $[2, 3]$ of the roots.

2.1.0.3 Method of Tabulation

This method depends on the continuity of the function.

Theorem 2.1 *If the function f is continuous on the interval I = [a, b] and if f(a) and f(b) have opposite signs, i.e. if*

$$f(a).f(b) < 0,$$

then the equation $f(x) = 0$ has at least one solution in I. •

Note that if $f(a)$ and $f(b)$ have same signs then nonlinear equation $f(x) = 0$ has no real roots or has an even number of real roots. If the curve $y = f(x)$ touches the x-axis at the same point, say, $x = \alpha$, then α is root of $f(x) = 0$ though $f(a)$ and $f(b)$ have same signs. For example, $f(x) = (x - 3)^2$ touches the x-axis at $x = 3$, also, $f(2.5) > 0$ and $f(3.5) > 0$, but $x = 3$ is the root of the equation $f(x) = (x - 3)^2 = 0$.

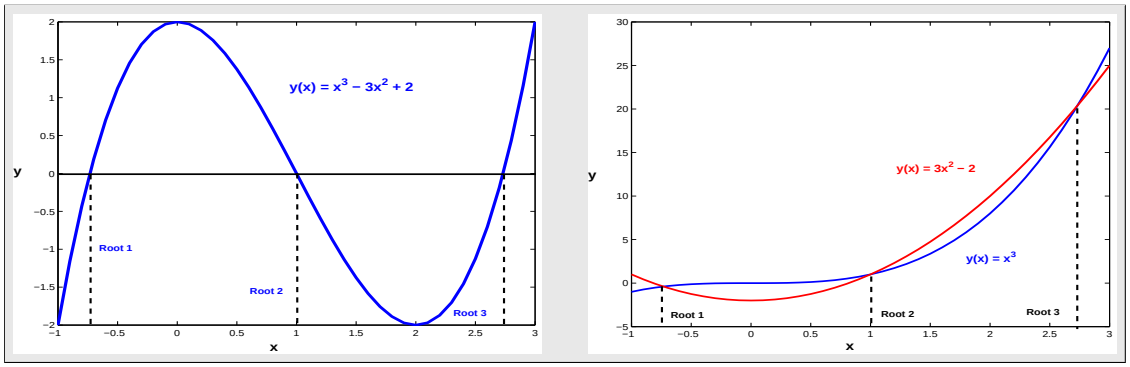


Figure 2.1: Graphical Solution of $x^3 - 3x^2 + 2 = 0$ in the intervals $[-1, 0]$, $[0, 2]$, and $[2, 3]$.

2.1.0.4 A Trial Method for Tabulation

Here, we form a table of signs of $f(x)$ setting $x = 0, \pm 1, \pm 2, \ldots$. If the function changes its signs from two consecutive values of x then atleast one real root lies between these two values, that if $f(a)$ and $f(b)$ have opposite signs then a real root lies between a and b. For example, finding the location of the roots of the equations $8x^3 - 20x^2 - 2x + 5 = 0$ by the method of tabulation, we form a table of sign of $f(x)$, taken $x = -2, -1, 0, 1, 2, 3, \ldots$, as follows: So the equation has three roots

x	-2	-1	0	1	2	3
Sign of f(x)	-	-	+	-	+	-

as its degree is 3 and their locations are $(-1, 0), (0, 1)$, and $(2, 3)$.

2.1.0.5 A Systematic Process for Tabulation

The following sequence of steps to be performed to located the root of $f(x) = 0$ by tabulation method is:

- Find $f(x)$.
- Prepare a table of signs of the function $f(x)$ by setting x equal to:

  The roots of $f(x) = 0$ or the roots close to them.

  The boundary values (preceding from the domain of permissible values of the variables).
- Determine the intervals at the endpoints of which the function assumes values of opposite signs. These intervals contain one and only one root each in its interior.

Example 2.1 *Find the number of real roots of the equation* $3^x = 3x + 2$ *and locate them.*

Solution. *Given function is* $f(x) = 3^x - 3x - 2$ *and the domain of definition of it is* $(-\infty, \infty)$. *Since the roots of* $f'(x) = 3^x log3 - 3 = 0$ *can be obtained as* $x = \dfrac{log3 - log(log3)}{log3} = 0.914$, *so a*

table of signs of $f(x)$ is then form by setting x equal to:
(a) the values close of the roots of $f'(x) = 0$, that is, $x = 0,\ x = 1$ and
(b) boundary values of domain, that is, $\pm\infty$. So the equation has two real roots since the function

x	-∞	0	1	∞
Sign of f(x)	+	-	-	+

twice changes sign, among them one is negative root and other is greater than 1. A new table with small intervals of the location of the root is constructed in the follow:
The locations of the roots of the given function are $(-1, 0)$ and $(1, 2)$. •

x	-1	0	1	2
Sign of f(x)	+	-	-	+

2.1.0.6 Types of Roots of a Nonlinear Equation

Root of a nonlinear equation $f(x) = 0$ may be **simple** (not repeating) or **multiple** (repeating). If $x = \alpha$ be a simple root of $f(x) = 0$ then

$$f(\alpha) = 0 \quad \text{but} \quad f'(\alpha) \neq 0.$$

For example, $\alpha_1 = -3$ and $\alpha_2 = -2$ are the simple roots of the nonlinear equation $x^2 + 5x + 6 = 0$.
If $x = \alpha$ be a multiple root of $f(x) = 0$ then

$$f(\alpha) = 0 \quad \text{and} \quad f'(\alpha) = 0.$$

For example, $\alpha = -2$ is the multiple root of the nonlinear equation $x^2 + 4x + 4 = 0$.

2.2 Approximation of Simple Root of a Nonlinear Equation

First, we shall discuss the numerical iterative methods for the approximation of a simple root of nonlinear equation in a *single variable* $f(x) = 0$. They are, bisection method, fixed-point method, Newton's method (also called, Newton-Raphson method), secant method. The best method among them is Newton's method.

2.2.1 Bisection Method

This is one of the simplest iterative technique for determining roots of (2.1) and it needs two initial approximations to start. It is based on the *Intermediate Value Theorem*. This method is also called the *interval-halving method* because the strategy is to bisect or halve the interval from one endpoint of the interval to the other endpoint and then retain the half interval whose end still bracket the root. It is also referred to a *bracketing method* or sometimes called the *Bolzano's method*. The fact that the function is required to change sign only once gives us a way to determine which half interval to retain; we keep the half on which $f(x)$ changes sign or became zero. The basis for this

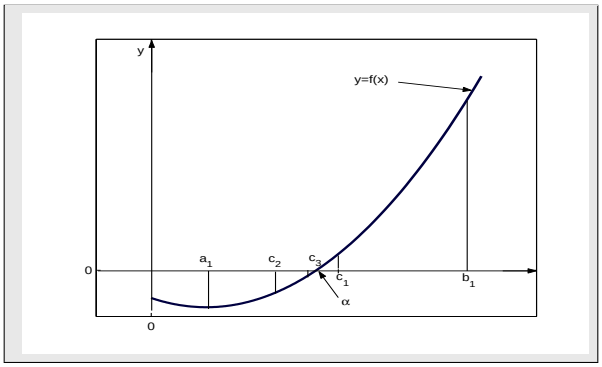


Figure 2.2: Graphical Solution of Bisection Method.

method can be easily illustrated by considering a function $y = f(x)$. Our object is to find an x value for which y is zero.
The implication is that one of the values is negative and the other is positive. These conditions can be easily satisfied by sketching the function, see Figure 2.2. Obviously, the function is negative at one endpoint a of the interval and positive at other endpoint b and is continuous on $a \leq x \leq b$. Therefore the root must lies between a and b (by Intermediate Value Theorem) and a new approximation to the root α be calculated as

$$c = \frac{a+b}{2},$$

and, in general

$$c_n = \frac{a_n + b_n}{2}, \qquad n \geq 1. \tag{2.2}$$

The iterative formula (2.2) is known as the *bisection method*.
If $f(c) \approx 0$, then $c \approx \alpha$ is the desired root, and, if not, then there are two possibilities. Firstly, if $f(a)f(c) < 0$, then $f(x)$ has a zero between point a and point c. The process can then be repeated on the new interval $[a, c]$. Secondly, if $f(a)f(c) > 0$ it follows that $f(b)f(c) < 0$ since it is known that $f(b)$ and $f(c)$ have opposite signs. Hence, $f(x)$ has zero between point c and point b and the process can be repeated with $[c, b]$. We see that after one step of the process, we have found either a zero or a new bracketing interval which is precisely half the length of the original one. The process continue until the desired accuracy is achieved. We use the bisection process in the following example.

Example 2.2 *Use the bisection method to find the approximation to the root of the equation*

$$x^3 = 2x + 1,$$

that is located in the interval $[1.5, 2.0]$ *accurate to within* 10^{-2}.

Solution. *Since the given function* $f(x) = x^3 - 2x - 1$ *is a polynomial function and so is continuous on* $[1.5, 2.0]$, *starting with* $a_1 = 1.5$ *and* $b_1 = 2$, *we compute:*

$$a_1 = 1.5: \quad f(1.5) = -0.625 \quad \textit{and} \quad b_1 = 2.0: \quad f(2) = 3.0,$$

and since $f(1.5)f(2.0) < 0$*, so that a root of* $f(x) = 0$ *lies in the interval* $[1.5, 2.0]$*. Using formula (2.2) (when* $n = 1$*), we get:*

$$c_1 = \frac{a_1 + b_1}{2} = \frac{1.5 + 2.0}{2} = 1.75; \qquad f(c_1) = 0.859375.$$

Hence the function changes sign on $[a_1, c_1] = [1.5, 1.75]$*. To continue, we squeeze from right and set* $a_2 = a_1$ *and* $b_2 = c_1$*. Then the midpoint is:*

$$c_2 = \frac{a_2 + b_2}{2} = \frac{1.5 + 1.75}{2} = 1.625; \qquad f(c_2) = 0.041056.$$

Continue in this way we obtain a sequence $\{c_k\}$ *of approximation shown by Table 2.1.*

Table 2.1: Solution of $x^3 = 2x + 1$ by Bisection method

n	Left Endpoint a_n	Midpoint c_n	Right Endpoint b_n	Function Value $f(c_n)$
01	1.500000	1.750000	2.000000	0.8593750
02	1.500000	1.625000	1.750000	0.0410156
03	1.500000	1.562500	1.625000	-0.3103027
04	1.562500	1.593750	1.625000	-0.1393127
05	1.593750	1.609375	1.625000	-0.0503273
06	1.609375	1.617188	1.625000	-0.0049520

We see that the functional values are approaching zero as the number of iterations is increase. We got the desired approximation to the root of the given equation is $c_6 = 1.617188 \approx \alpha$ *after 6 iterations with the accuracy* $\epsilon = 10^{-2}$*.* •

To use MATLAB command for the bisection method, first we define a function m-file as fn.m for the equation as follows:

```
function y = fn(x)
y = x.^ 3 - 2 * x - 1;
```

then use the single commands:

```
>> s = bisect('fn', 1.5, 2, 1e - 2)
```

We can easily find the roots $(1.61803399, -1.00, -0.61803399)$ of the equation $x^3 = 2x + 1$ by defining the coefficients of the polynomial equation using MATLAB commands as:

```
>> CP = [1 0 - 2 - 1]; Sol = roots(CP);
```

Theorem 2.2 *If f(x) is continuous on I, then the bisection method always converges to the exact solution.* •

Example 2.3 *Find the approximation of the point of intersection of the graphs* $y = x^3 + 2x - 1$ *and* $y = \sin x$*, by using bisection method within accuracy* 10^{-3}*, when* $a = 0.5$ *and* $b = 1$.

Solution. *The point of intersection of the graphs means* $x^3 + 2x - 1 = \sin x$*, from this we get* $f(x) = x^3 + 2x - \sin x - 1 = 0$*. Given the interval* $[0.5, 1.0]$*, so we compute:*

$$a_1 = 0.5: \quad f(0.5) = -0.3544 \quad \textit{and} \quad b_1 = 1.0: \quad f(1.0) = 1.1585.$$

Since $f(x)$ *is continuous on* $[0.5, 1.0]$ *and* $f(0.5).f(1.0) < 0$*, so that a root of* $f(x) = 0$ *lies in the interval* $[0.5, 1.0]$*. Using formula (2.2) (when* $n = 1$*), we get:*

$$c_1 = \frac{a_1 + b_1}{2} = \frac{0.5 + 1.0}{2} = 0.75; \qquad f(c_1) = 0.240236.$$

Hence the function changes sign on $[a_1, c_1] = [0.5, 0.75]$*. To continue, we squeeze from right and set* $a_2 = a_1$ *and* $b_2 = c_1$*. Then the midpoint is:*

$$c_2 = \frac{a_2 + b_2}{2} = \frac{0.5 + 0.75}{2} = 0.625; \qquad f(c_2) = -0.090957.$$

Then continue in this manner we obtain a sequence $\{c_k\}$ *of approximation shown by Table 2.2.*

Table 2.2: Solution of $x^3 + 2x - \sin x - 1 = 0$ by Bisection method

n	Left Endpoint a_n	Right Endpoint b_n	Midpoint c_n	Function Value $f(c_n)$
01	0.5000	1.0000	0.750000	0.240236
02	0.5000	0.7500	0.625000	-0.090957
03	0.6250	0.7500	0.687500	0.065344
⋮	⋮	⋮	⋮	⋮
07	0.6563	0.6641	0.660156	-0.005228
08	0.6602	0.6641	0.662109	-0.000302

Program 2.1

```
MATLAB m-file for the Bisection Method
function sol=bisect(fn,a,b,tol)
fa = feval(fn, a); fb = feval(fn, b);
if fa * fb > 0; fprintf('Endpoints have same sign') return end
while abs (b − a) > tol c = (a + b)/2; fc = feval(fn, c);
if fa * fc < 0; b = c; else a = c; end; end; sol=(a + b)/2;
```

We see that the functional values are approaching zero as the number of iterations is increase. We got the desired approximation to the root of the given equation is $c_8 = 0.662109 \approx \alpha$ *after 8 iterations with accuracy* $\epsilon = 10^{-3}$ *and point of intersection is* $(0.662109, 0.614479)$*. The exact point of intersection is* $(0.66, 0.61)$. •

Theorem 2.3 (Bisection Convergence and Error Theorem)

Let $f(x)$ be continuous function defined on the given initial interval $[a_0, b_0] = [a, b]$ and suppose that $f(a)f(b) < 0$. Then bisection method (2.2) generates a sequence $\{c_n\}_{n=1}^{\infty}$ approximating $\alpha \in (a, b)$ with the property

$$|\alpha - c_n| \leq \frac{b-a}{2^n}, \quad n \geq 1. \tag{2.3}$$

Moreover, to obtain accuracy of

$$|\alpha - c_n| \leq \epsilon,$$

(for $\epsilon = 10^{-k}$, where k is nonnegative integer) it suffices to take

$$n \geq \frac{\ln\left\{10^k(b-a)\right\}}{\ln 2} = 1.443 \ln\left\{10^k(b-a)\right\}. \tag{2.4}$$

Proof.

Since both the root α and the midpoint c_1 lie in the interval $[a, b]$, the distance between them cannot be greater than of this width interval. Thus

$$|\alpha - c_n| \leq \frac{b_{n-1} - a_{n-1}}{2}, \qquad \text{for all } n.$$

Observe that

$$b_1 - a_1 = \frac{b_0 - a_0}{2},$$

then

$$b_2 - a_2 = \frac{b_1 - a_1}{2} = \frac{b_0 - a_0}{2^2}.$$

Finite mathematical induction is used to conclude that

$$b_{n-1} - a_{n-1} = \frac{b_0 - a_0}{2^{n-1}}.$$

Therefore, the error is bounded as follows

$$|\alpha - c_n| \leq \frac{b_{n-1} - a_{n-1}}{2} = \frac{b_0 - a_0}{2^n},$$

gives the estimate.
Now to establish the bound on the number of bisections n (or iterations), we simply observe that

$$\frac{b-a}{2^n} \leq 10^{-k},$$

together with (2.3) implies that

$$|\alpha - c_n| \leq 10^{-k},$$

that is, we wish to calculate a root to within 10^{-k}. Since $2^n \geq 10^k(b-a)$, so by taking logarithms, we get $n \ln 2 \geq \ln\left\{10^k(b-a)\right\}$, and solving for n, we get the inequality (2.4). •

The above Theorem 2.3 gives us information about bounds for errors in approximation and the number of bisections needed to obtain any given accuracy.

Example 2.4 *Show that number of iterations of bisection will require to attain an accuracy of* 10^{-4} *using the starting interval* $[a, b]$ *is*

$$n \geq \frac{\ln(b-a) + 4\ln 5}{\ln 2} + 4.$$

Determine the number of iterations needed to achieve the an approximation with same above given accuracy to the solution of $x^3 - 2x - 1 = 0$ *lying in the interval* $[1.5, 2]$.

Solution. *By using the inequality (2.4), we get*

$$n \geq \frac{\ln\left\{10^k(b-a)\right\}}{\ln 2} = \frac{\ln(b-a) + \ln[(5)(2)]^4}{\ln 2},$$

or

$$n \geq \frac{\ln(b-a) + 4[\ln 5 + \ln 2]}{\ln 2} = \frac{\ln(b-a) + 4\ln 5}{\ln 2} + 4.$$

Now by taking $a = 1.5$ *and* $b = 2$ *in the above inequality, we get*

$$n \geq \frac{\ln(2-1.5) + 4\ln 5}{\ln 2} + 4 = 8.2877 + 4 = 12.2877.$$

So no more than thirteen iterations are required to obtain an approximation accurate to within the given accuracy 10^{-4}. •

Example 2.5 *Find a bound for the number of iterations needed to achieve an approximation with accuracy* 10^{-1} *to the solution of* $xe^x = 1$ *lying in the interval* $[0.5, 1.0]$ *using the bisection method. Find an approximation to the root with this degree of accuracy.*

Solution. *Here* $a = 0.5$, $b = 1.0$ *and* $k = 1$, *then by using inequality (2.4), we get*

$$n \geq \frac{\ln[10^1(1.0-0.5)]}{\ln 2} = 2.3219 \approx 3.$$

So no more than three iterations are required to obtain an approximation accurate to within 10^{-1}. *The given function* $f(x) = xe^x - 1$ *is continuous on* $[0.5, 1.0]$, *so starting with* $a_1 = 0.5$ *and* $b_1 = 1.0$, *we compute:*

$$a_1 = 0.5: \quad f(0.5) = -0.1756 \quad \textit{and} \quad b_1 = 1.0: \quad f(1.0) = 1.7183,$$

since $f(0.5)f(1.0) < 0$, *so that a root of* $f(x) = 0$ *lies in the interval* $[0.5, 1]$. *Using formula (2.2) (when* $n = 1$*), we get:*

$$c_1 = \frac{a_1 + b_1}{2} = \frac{0.5 + 1.0}{2} = 0.75; \qquad f(c_1) = 0.5878.$$

Hence the function changes sign on $[a_1, c_1] = [0.5, 0.75]$. *To continue, we squeeze from right and set* $a_2 = a_1$ *and* $b_2 = c_1$. *Then the bisection formula gives*

$$c_2 = \frac{a_2 + b_2}{2} = \frac{0.5 + 0.75}{2} = 0.625; \qquad f(c_2) = 0.1677.$$

Finally, the function changes sign on $[a_1, c_2] = [0.5, 0.625]$, *gives*

$$c_3 = \frac{a_3 + b_3}{2} = \frac{0.5 + 0.625}{2} = 0.5625,$$

the value of the third approximation which is accurate to within 10^{-1}. •

It is important to keep in mind that the error analysis gives only a bound for the number of iterations necessary, and in many cases this bound is much larger than the actual number required.

Example 2.6 *Find a nonlinear equation which has the simple root* $\sqrt[4]{18}$ *in* $[2.0, 2.5]$. *Use bisection method to compute the second approximation to the root using* $a = 2$ *and* $b = 2.5$. *Also, compute an error bound and absolute error for your approximation.*

Solution. *Given*

$$\alpha = x = \sqrt[4]{18} = (18)^{1/4}, \quad \textit{which can be written as} \qquad x^4 = 18, \quad \textit{or} \qquad x^4 - 18 = 0,$$

which is the nonlinear equation and it gives $f(x) = x^4 - 18$. *Since*

$$f(x) = x^4 - 18, \quad \textit{gives} \quad f'(x) = 4x^3, \quad \textit{and} \quad f'((18)^{1/4}) = 4((18)^{1/4})^3 = 34.9554 \neq 0,$$

and it shows that $\sqrt[4]{18}$ *is a simple root of* $x^4 - 18 = 0$. *Given the interval* $[2.0, 2.5]$ *on which the function* $f(x)$ *is continuous (as it is a polynomial) and* $f(x)$ *satisfies the sign property, that is*

$$a_1 = 2.0 : \ f(2.0) = -2, \quad b_1 = 2.5 : \ f(2.5) = 21.0625, \quad \textit{so} \quad f(2.0)f(2.5) = (-2)(21.0625) = -42.125 < 0.$$

Now compute its first approximate value by using bisection formula, we have

$$c_1 = \frac{a_1 + b_1}{2} = \frac{2.0 + 2.5}{2} = 2.25 \quad \textit{and} \quad f(2.25) = 7.6289.$$

Since the function $f(x)$ *changes sign on* $[2.0, 2.25]$. *To continue, we squeeze from right and use formula (2.2) again to get the following second approximate value of the root* α *as:*

$$c_2 = \frac{a_1 + c_1}{2} = \frac{2.0 + 2.25}{2} = 2.125 \quad \textit{and} \quad f(2.125) = 2.3909.$$

Finally, $f(x)$ *changes sign on* $[2.0, 2.125]$, *so* $c_3 = \dfrac{2.0 + 2.125}{2} = 2.0625$ *with* $f(2.0625) = 0.0957$. *Note that the value of* $f(x)$ *at each new approximate value is decreasing which shows that the approximate values are coming closer to the root* $\sqrt[4]{18} = 2.0598$. *Now to compute the error bound for the approximation we use the error bound formula (2.3) which gives*

$$|\alpha - c_3| \leq \frac{2.5 - 2.0}{2^3} = \frac{1}{2^4} = 0.0625,$$

which is the maximum error in our approximation and $|E| = |2.0598 - 2.0625| = 0.0027$, *be the absolute error in the approximation.* •

One drawback of the bisection method is the convergence rate is raster slow. However, the rate of convergence is guaranteed. So for this reason it is often used as a started for the more efficient used to find roots of the nonlinear equations. The method may give a false root if $f(x)$ is discontinuous on the given interval $[a, b]$.

Procedure 2.1 (Bisection Method)

1. *Establish an interval* $a \leq x \leq b$ *such that* $f(a)$ *and* $f(b)$ *are of opposite sign, that is,* $f(a).f(b) < 0$.
2. *Choose an error tolerance* $(\epsilon > 0)$ *value for the function.*
3. *Compute a new approximation for the root:* $c_n = \dfrac{(a_n + b_n)}{2}; \qquad n = 1, 2, 3, \ldots.$
4. *Check tolerance. If* $|f(c_n)| \leq \epsilon$, *use* c_n, $n \geq 1$ *for desired root; otherwise continue.*
5. *Check, if* $f(a_n)f(c_n) < 0$, *then set* $b_n = c_n$; *otherwise set* $a_n = c_n$.
6. *Go back to step 3, and repeat the process.*

2.2.2 Fixed-Point Method

This is another iterative method to solve the nonlinear equation (2.1) and needs one initial approximation to start. This is a very general method for finding the root of (2.1) and it provides us with a theoretical framework within which the convergence properties of subsequent methods can be evaluated. The basic idea of this method which is also called successive approximation method or function iteration, is to rearrange the original equation

$$f(x) = 0, \tag{2.5}$$

into an equivalent expression of the form

$$x = g(x). \tag{2.6}$$

Any solution of (2.6) is called a fixed-point for the iteration function $g(x)$ and hence a root of (2.5).

Definition 2.2 (Fixed-Point of a Function)

A fixed-point of a function $g(x)$ *is a real number* α *such that* $\alpha = g(\alpha)$.
For example, $x = 2$ *is a fixed-point of the function* $g(x) = \dfrac{x^2 - 4x + 8}{2}$, *because* $g(2) = 2$. •

The task of solving (2.5) is therefore reduced to that of finding a point satisfying the fixed-point condition (2.6). The fixed-point method essentially solves two functions simultaneously; $y = x$ and $y = g(x)$. The point of intersection of these two functions is the solution to $x = g(x)$, and thus to $f(x) = 0$, see Figure 2.3.
This method is conceptually very simple. Since $g(x)$ is also nonlinear, the solution must be obtained iteratively. An initial approximation to the solution, say, x_0, must be determined. For choosing

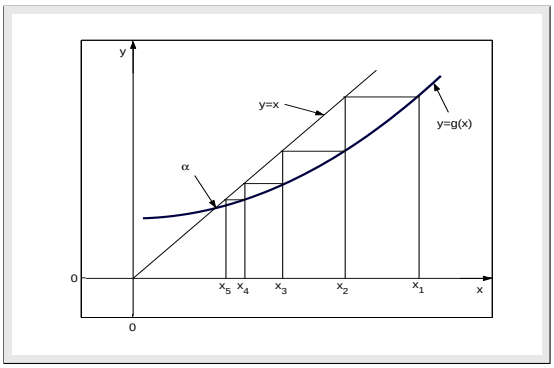


Figure 2.3: Graphical Solution of Fixed-Point Method.

the best initial value x_0 for using this iterative method, we have to find an interval $[a, b]$ on which the original function $f(x)$ satisfies the sign property and then use the midpoint $\dfrac{a+b}{2}$ as the initial approximation x_0. Then this initial value x_0 is substituted in the function $g(x)$ to determined the next approximation x_1 and so on.

Definition 2.3 (Fixed-Point Method)

The iteration defined in the following

$$x_{n+1} = g(x_n); \qquad n = 0, 1, 2, \ldots, \tag{2.7}$$

is called the fixed-point method or the fixed-point iteration. •

The value of the initial approximation x_0 is chosen arbitrarily and the hope is that the sequence $\{x_n\}_{n=0}^{\infty}$ converges to a number α which will automatically satisfies (2.5). Moreover, since (2.5) is a rearrangement of (2.6), α is guaranteed to be a zero of $f(x)$. In general, there are many different ways of rearranging of (2.6) in (2.5) form. However, only some of these are likely to give rise to successful iterations but sometime we don't have successful iterations. To describe such behaviour, we discuss the following example.

Example 2.7 *Consider the nonlinear equation $x^3 = 2x+1$ which has a root in the interval $[1.5, 2.0]$ using fixed-point method with $x_0 = 1.5$, take three different rearrangements for the equation.*

Solution. *Let us consider the three possible rearrangement of the given equation as follows:*

$$(i) \quad x_{n+1} = g_1(x_n) = \frac{(x_n^3 - 1)}{2}; \qquad n = 0, 1, 2, \ldots,$$

$$(ii) \quad x_{n+1} = g_2(x_n) = \frac{1}{(x_n^2 - 2)}; \qquad n = 0, 1, 2, \ldots,$$

$$(iii) \quad x_{n+1} = g_3(x_n) = \sqrt{\frac{(2x_n + 1)}{x_n}}; \qquad n = 0, 1, 2, \ldots,$$

then the numerical results for the corresponding iterations, starting with the initial approximation $x_0 = 1.5$ *with accuracy* 5×10^{-2}*, are given in Table 2.3. We note that the first two considered*

Table 2.3: Solution of $x^3 = 2x + 1$ by fixed-point method

n	x_n	$x_{n+1} = g_1(x_n)$ $= (x_n^3 - 1)/2$	$x_{n+1} = g_2(x_n)$ $= 1/(x_n^2 - 2)$	$x_{n+1} = g_3(x_n)$ $= \sqrt{(2x_n + 1)/x_n}$
00	x_0	1.500000	1.500000	1.500000
01	x_1	1.187500	4.000000	1.632993
02	x_2	0.337280	0.071429	1.616284
03	x_3	-0.480816	-0.501279	1.618001
04	x_4	-0.555579	-0.571847	1.618037
05	x_5	-0.585745	-0.597731	1.618034

sequences diverge and the last one converges. This example asks the need for a mathematical analysis of the method. The following theorem gives sufficient conditions for the convergence of the fixed-point iteration. •

Theorem 2.4 (Fixed-Point Theorem)

If g *is continuously differentiable on the interval* $[a, b]$ *and* $g(x) \in [a, b]$ *for all* $x \in [a, b]$*, then*

(a) g *has at-least one fixed-point in the given interval* $[a, b]$*.*

Moreover, if the derivative $g'(x)$ *of the function* $g(x)$ *exists on an interval* $[a, b]$ *which contains the starting value* x_0*, with*

$$k \equiv \max_{a \le x \le b} |g'(x)| < 1; \qquad \textit{for all} \qquad x \in [a, b]. \tag{2.8}$$

Then

(b) The sequence (2.7) will converge to the attractive (unique) fixed-point α *in* $[a, b]$*.*

(c) The iteration (2.7) will converge to α *for any initial approximation.*

(d) We have the error estimate

$$|\alpha - x_n| \le \frac{k^n}{1 - k} |x_1 - x_0|, \qquad \textit{for all} \quad n \ge 1. \tag{2.9}$$

(e) The limit holds:

$$\lim_{n \to \infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\alpha). \tag{2.10}$$

Proof

(a) Suppose g is continuous on $[a,b]$ and $g(x) \in [a,b]$. We need to show it has a fixed point. If $g(a) = a$ and $g(b) = b$, then the function g has a fixed-point at the endpoints. Suppose that it is not happening, that is, $g(a) \neq a$ and $g(b) \neq b$ and define a function $f(x) = x - g(x)$ which is continuous on $[a,b]$. Then $f(x)$ has a zero in $[a,b]$ if and only if $g(x)$ has a fixed point in $[a,b]$ but

$$f(a) = a - g(a) > 0,$$

since $g(a)$ is in $[a,b]$ and hence cannot be smaller than a, and we have assumed that $g(a)$ is not equal to a. Similarly,

$$f(b) = b - g(b) < 0,$$

and so by the Intermediate Value Theorem there is a α in the interval (a,b) such that $f(\alpha) = 0$, which implies that $\alpha = g(\alpha)$. Thus the function $g(x)$ has at least one fixed-point in $[a,b]$.

(b) Suppose now that (2.8) holds, and α and β are two fixed-points of the function g in $[a,b]$. Then we have

$$\alpha = g(\alpha) \qquad \text{and} \qquad \beta = g(\beta).$$

In addition, by the Mean Value Theorem, we have that for any two points α and β in $[a,b]$, there exits a number η such that

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| = |g'(\eta)||\alpha - \beta| \leq k|\alpha - \beta|,$$

where $\eta \in (a,b)$. Thus

$$|\alpha - \beta| - k|\alpha - \beta| \qquad or \qquad (1-k)|\alpha - \beta| \leq 0.$$

Since $k < 1$, we must have $\alpha = \beta$; and thus, the function g has a unique fixed-point α in the interval $[a,b]$.

(c) For the convergence, consider the iteration

$$x_n = g(x_{n-1}), \qquad for\ all \qquad n \geq 1, 2, \ldots,$$

and the definition of the fixed-point, that is

$$\alpha = g(\alpha).$$

If we subtract last two equations and take the absolute values, we get

$$|\alpha - x_n| = |g(\alpha) - g(x_{n-1})| \leq k|\alpha - x_{n-1}|.$$

The recursion can be solved readily to get

$$|\alpha - x_n| \leq k|\alpha - x_{n-1}| \leq k^2|\alpha - x - n - 2| \cdots \leq k^n|\alpha - x_0|, \tag{2.11}$$

from which it follows that

$$as \quad n \to \infty, \quad k^n \to 0, \quad (since \quad k < 1),$$

therefore, $x_n \to \alpha$. Hence the iteration converges.

(d) Since we note that

$$\begin{aligned} |\alpha - x_0| &= |\alpha - x_1 + x_1 - x_0| \leq |\alpha - x_1| + |x_1 - x_0| \\ &\leq |g(\alpha) - g(x_0)| + |x_1 - x_0| \leq k|\alpha - x_0| + |x_1 - x_0|, \end{aligned}$$

which gives

$$|\alpha - x_0| - k|\alpha - x_0| \leq |x_1 - x_0| \quad or \quad (1-k)|\alpha - x_0| \leq |x_1 - x_0|,$$

and from this it follows that

$$|\alpha - x_0| \leq \frac{1}{1-k}|x_1 - x_0|.$$

From (2.11), we can write above equation as follows

$$|\alpha - x_n| \leq \frac{k^n}{1-k}|x_1 - x_0|.$$

(e) *Finally, by subtracting iteration $x_{n+1} = g(x_n)$ and $\alpha = g(\alpha)$, we have*

$$\alpha - x_{n+1} = g(\alpha) - g(x_n) = g'(\eta(x))(\alpha - x_n),$$

which implies that

$$\frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\eta(x)),$$

and by taking limits, we have

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = \lim_{n\to\infty} g'(\eta(x)) = g'(\alpha),$$

since $\eta(x) \to \alpha$ is forced by the convergence of x_n to α. •

Now we come back to our previous Example 2.7 and discuss that why the first two rearrangements we considered, do not converge but on the other hand, last sequence has a fixed-point and converges. Since, we observe that $f(1.5)f(2) < 0$, then the solution we seek is in the interval $[1.5, 2]$.

(i) For $g_1(x) = \dfrac{x^3 - 1}{3}$, we have $g_1'(x) = x^2$, which is greater than unity throughout the interval $[1.5, 2]$. So by Fixed-Point Theorem 2.4 this iteration will fail to converge.

(ii) For $g_2(x) = \dfrac{1}{x^2 - 2}$, we have $g_2'(x) = \dfrac{-2x}{(x^2-2)^2}$, and $|g_2'(1.5)| > 1$, so from Fixed-Point Theorem 2.4 this iteration will fail to converge.

(iii) For $g_3(x) = \sqrt{\dfrac{2x+1}{x}}$, we have $|g_3'(x)| = |\dfrac{-x^{-3/2}}{2\sqrt{2x+1}}| < 1$, in the given interval $[1.5, 2]$. Also, g_3 is decreasing function of x, as $g_3(1.5) = 1.63299$ and $g_3(2) = 1.58114$ both lie in the interval $[1.5, 2]$. Thus $g_3(x) \in [1.5, 2]$, for all $x \in [1.5, 2]$, so from Fixed-Point Theorem 2.4 the iteration will converge. •

Example 2.8 *The expression* $x_{n+1} = \dfrac{2x_n^2 + 3}{5}$ *is an iterative scheme to find a root of the equation* $f(x) = 0$. *Find the function* $f(x)$.

Solution. *Consider a* α *be the root obtained by performing the iteration scheme* $x_{n+1} = \dfrac{2x_n^2 + 3}{5}$. *Therefore,* $\lim_{n\to\infty} x_n = \alpha$. *Thus*

$$\lim_{n\to\infty} x_{n+1} = \frac{1}{5}[2\lim_{n\to\infty}(x_n^2) + 3], \quad \textit{gives} \quad \alpha = \frac{1}{5}[2\alpha + 3], \textit{or} \quad 2\alpha^2 - 5\alpha + 3 = 0.$$

This is the required equation $2x^2 - 5x + 3 = 0$ *and so* $f(x) = 2x^2 - 5x + 3$. •

Example 2.9 *Find an interval* $[a, b]$ *on which fixed-point problem* $x = \dfrac{2 - e^x + x^2}{3}$ *will converges. Estimate the number of iterations* n *within accuracy* 10^{-5}.

Solution. *Since* $x = g(x) = \dfrac{2 - e^x + x^2}{3}$ *can be written as,* $3x = 2 - e^x + x^2$, *gives*

$$f(x) = e^x - x^2 + 3x - 2 = 0,$$

and we observe that $f(0)f(1) = (-1)(e^1) < 0$, *then the solution we seek is in the interval* $[0, 1]$. *For* $g(x) = \dfrac{2 - e^x + x^2}{3}$, *we have* $|g'(x)| = |\dfrac{2x - e^x}{3}| < 1$, *for all* x *in the given interval* $[0, 1]$. *Also,* g *is decreasing function of* x *and* $g(0) = 0.3333$ *and* $g(1) = \dfrac{3 - e}{3} = 0.0939$ *both lie in the interval* $[0, 1]$. *Thus* $g(x) \in [0, 1]$, *for all* $x \in [0, 1]$, *so from Fixed-Point Theorem 2.4 the g(x) has a unique fixed-point in* $[0, 1]$. *Taking* $x_0 = 0.5$, *we have*

$$x_1 = g(x_0) = \frac{2 - e^{x_0} + x_0^2}{3} = 0.2004.$$

Also, we have

$$k_1 = |g'(0)| = 0.3333 \quad \textit{and} \quad k_2 = |g'(1)| = 0.2394,$$

which give $k = \max\{k_1, k_2\} = 0.3333$. *Thus the error estimate (2.9) within the accuracy* 10^{-5} *is*

$$|\alpha - x_n| \le 10^{-5}, \qquad \textit{gives} \qquad \frac{(0.3333)^n}{1 - 0.3333}(0.2996) \le 10^{-5}, \quad \textit{and} \quad n \ge 9.7507.$$

So we need ten (10) approximations to get the desired accuracy for the given problem. •

Note 2.1 *From (2.9) Note that the rate of convergence of the fixed-point method depends on the factor* $\dfrac{k^n}{(1-k)}$; *the smaller the value of* k, *then faster the convergence. The convergence may be very slow if the value of* k *is very close to 1.* •

Note 2.2 *Assume that* $g(x)$ *and* $g'(x)$ *are continuous functions of* x *for some open interval* I, *with the fixed-point* α *contained in this interval. Moreover assume that*

$$|g'(\alpha)| < 1, \qquad \textit{for} \quad \alpha \in I,$$

then, there exists an interval $[a, b]$, around the solution α for which all the conditions of Theorem 2.4 are satisfied, that is, the fixed-point method converges. But if

$$|g'(\alpha)| > 1, \qquad for \quad \alpha \in I,$$

then the sequence (2.7) will not converge to α. In this case α is called a repulsive fixed-point. If

$$|g'(\alpha)| = 0, \qquad for \quad \alpha \in I,$$

then the sequence (2.7) converges very fast to the fixed-point (or root) α while if

$$|g'(\alpha)| = 1, \qquad for \quad \alpha \in I,$$

then the convergence the sequence (2.7) is not guaranteed and if convergence happened, it would be very slow. Thus to get the faster convergence, the value of $|g'(\alpha)|$ should be equal to zero or very close to zero. In other words,
1. *If $|g'(\alpha)| \neq 0$ but $|g'(\alpha)| < 1$, then the speed of convergence of method is slow (linear).*
2. *If $|g'(\alpha)| = 0$, then the speed of convergence method will be very fast (quadratic).* •

Example 2.10 *Consider the two different rearrangements for the nonlinear equation $\sin x = 5x - 2$ as follows:*

$$\begin{array}{llllll} (1) & x_{n+1} & = & g_1(x_n) & = & \sin^{-1}(5x-2); \qquad n = 0, 1, 2, \ldots, \\ (2) & x_{n+1} & = & g_2(x_n) & = & \dfrac{1}{5}(\sin x + 2); \qquad n = 0, 1, 2, \ldots. \end{array}$$

Find out which one is converges near to 0.5 and then use the convergent one to find the second approximation x_2 when $x_0 = 0.5$.

Solution. *Since the derivative of the first rearrangement $g_1(x) = \sin^{-1}(5x - 2)$ is $g_1'(x) = \dfrac{5}{\sqrt{1-(5x-2)^2}}$ and one can easily check that $|g_1'(x)| > 0$ for all x for which $(5x - 2)^2 < 1$ or $x < 0.6$. So this form of the rearrangement of the given equation is fail to converge. Let us check the other form and the derivative for this is, $g_2'(x) = \dfrac{1}{5}\cos x$ and note that $|g_2'(x)| \leq \dfrac{1}{5}$ for all x because $|cosx| \leq 1$. Thus second form for the nonlinear equation will converge.*
Now using $x_0 = 0.5$ and the second rearrangement, we get

$$x_1 = g_2(x_0) = \frac{1}{5}(\sin x_0 + 2) = 0.4017, \quad and \quad x_2 = g_2(x_1) = \frac{1}{5}(\sin x_1 + 2) = 0.4014,$$

the required approximations. •

Example 2.11 *Convert the equation $x^2 - 5 = 0$ to the fixed-point problem $x = x + k(x^2 - 5)$ with k a nonzero constant. Find a value of k to ensure rapid convergence of the scheme $x_{n+1} = x_n + k(x_n^2 - 5)$, for $n \geq 0$ to $\alpha = \sqrt{5}$. If $x_0 = 2$, compute $|\sqrt{5} - x_2|$.*

Solution. *Given $x^2 - 5 = 0$, and it can be written as for $k \neq 0$*

$$k(x^2 - 5) = 0 \quad or \quad -x + x + k(x^2 - 5) = 0.$$

From this we have

$$x = x + k(x^2 - 5) = g(x),$$

and it gives the iterative scheme

$$x_{n+1} = x_n + k(x_n^2 - 5) = g(x_n), \qquad n \geq 0.$$

For guaranteed convergence of this scheme, we mean that

$$|g'(x)| < 1 \qquad or \qquad |1 + 2kx| < 1 \qquad or \qquad -1 < 1 + 2kx < 1.$$

Moreover, the convergence will be rapid if

$$g'(\alpha) = 1 + 2\alpha k = 0.$$

Since $\alpha = \sqrt{5}$*, therefore,* $1 + 2\sqrt{5}k = 0$*. Thus, we have* $k = -\dfrac{1}{2\sqrt{5}} = -0.2236$*. Using* $x_0 = 2$*, we get*

$x_1 = x_0 + k(x_0^2 - 5) = 2 - 0.2236(2^2 - 5) = 2.2236$, $x_2 = x_1 + k(x_1^2 - 5) = 2.2236 - 0.2236(2.2236^2 - 5) = 2.2360$,

and so $|\sqrt{5} - x_2| = |2.2361 - 2.2360| = 0.0001$. •

Example 2.12 *Convert the equation* $2^x - 5x + 1 = 0$ *to the fixed-point problem* $x = \dfrac{1}{1+c}\left(cx + \dfrac{2^x + 1}{5}\right)$ *with* c *a constant. Find a value of* c *to ensure rapid convergence of the following scheme near* $x = 0.1$

$$x_{n+1} = \frac{1}{1+c}\left(cx_n + \frac{2^{x_n} + 1}{5}\right), \qquad n \geq 0.$$

Compute the third iterates, starting with $x_0 = 0.1$.

Solution. *Given* $2^x - 5x + 1 = 0$ *and it can be written as for any* c

$$x(c - c + 1) = \frac{2^x + 1}{5} \quad or \quad x(c+1) - xc = \frac{2^x + 1}{5} \quad or \quad x(c+1) = xc + \frac{2^x + 1}{5}.$$

From this we have

$$x = \frac{1}{1+c}\left(cx + \frac{2^x + 1}{5}\right) = g(x),$$

and it gives the iterative scheme

$$x_{n+1} = \frac{1}{1+c}\left(cx_n + \frac{2^{x_n} + 1}{5}\right) = g(x_n), \qquad n \geq 0.$$

For guaranteed the convergence will be rapid if

$$g'(x) = 0, \quad which\ gives \quad c = -\frac{2^x \ln 2}{5}.$$

Thus, $c = g'(0.1) = -\dfrac{2^x \ln 2}{5} = -0.1486$. *Now to find third iterates when* $x_0 = 0.5$

$$\begin{aligned} x_1 &= \frac{1}{1+c}\left(cx_0 + \frac{2^{x_0}+1}{5}\right) = 0.4692, \\ x_2 &= \frac{1}{1+c}\left(cx_1 + \frac{2^{x_1}+1}{5}\right) = 0.4782, \\ x_3 &= \frac{1}{1+c}\left(cx_2 + \frac{2^{x_2}+1}{5}\right) = 0.4787, \end{aligned}$$

the required approximations at the value of $c = -0.1486$. •

Example 2.13 *Show that the function* $g(x) = 3^{-x}$ *on the interval* $[0, 1]$ *has at least one fixed-point but it is not unique.*

Solution. *Given* $x = g(x) = 3^{-x}$, *and it can be written as*

$$x - 3^{-x} = f(x) = 0.$$

So $f(0)(1) = (-1)(2/3) < 0$, *so* $f(x)$ *has a root in the interval* $[0, 1]$. *Note that* g *is decreasing function of* x *and* $g(0) = 1$ *and* $g(1) = 0.3333$ *both lie in the interval* $[0, 1]$. *Thus* $g(x) \in [0, 1]$, *for all* $x \in [0, 1]$, *so from Fixed-Point Theorem 2.4 the function* $g(x)$ *has at least one fixed-point in* $[0, 1]$. *Since the derivative of the function* $g(x)$ *is*

$$g'(x) = -3^{-x} \ln 3,$$

which is less than zero on $[0, 1]$, *therefore, the function* g *is decreasing on* $[0, 1]$. *But* $g'(0) = -\ln 3 = -1.0986$, *so*

$$|g'(x)| > 1 \quad on \qquad (0, 1).$$

Thus from Fixed-Point Theorem 2.4 the function $g(x)$ *has no unique fixed-point in* $[0, 1]$. •

Example 2.14 *Show that the function* $g(x) = \sqrt{2x - 1}$ *on the interval* $[0, 1]$ *that satisfies none of the hypothesis of Theorem 2.4 but still has a unique fixed-point on* $[0, 1]$.

Solution. *Since* $x = g(x) = \sqrt{2x - 1}$, *it gives*

$$x^2 - 2x + 1 = (x - 1)^2 = f(x) = 0.$$

Then $x = \alpha = 1 \in [0, 1]$ *is the root of the nonlinear equation* $f(x) = 0$ *and the fixed-point of the function* $g(x)$ *as* $g(1) = 1$. *But notice that the function* $g(x)$ *is not continuous on the interval* $[0, 1]$ *and the derivative of the function* $g(x)$

$$g'(x) = \frac{1}{\sqrt{2x - 1}},$$

does not exist on the interval $(0, 1)$. *So all the conditions of Fixed-Point Theorem 2.4 fail.* •

Example 2.15 *Show that the fixed point form of the equation $x = N^{1/3}$ can be written as $x = Nx^{-2}$ and the associated iterative scheme*

$$x_{n+1} = Nx_n^{-2}, \qquad n \geq 0,$$

will not successful (diverge) in finding the approximation of cubic root of the positive number N.

Solution. *Given $x = N^{1/3}$ and it can be written as*

$$x^3 - N = 0 \quad or \quad x = \frac{N}{x^2} = Nx^{-2}.$$

It gives the iterative scheme

$$x_{n+1} = Nx_n^{-2} = g(x_n), \qquad n \geq 0.$$

From this, we have

$$g(x) = Nx^{-2} \qquad and \qquad g'(x) = -2Nx^{-3}.$$

Since $\alpha = x = N^{1/3}$, therefore

$$g'(\alpha) = -2N\alpha^{-3} \quad and \quad g'(N^{1/3}) = -2N(N^{1/3})^{-3} = -2NN^{-1} = -2.$$

Thus, $|g'(N^{1/3})| = |-2| = 2 > 1$, which shows the divergence. •

Example 2.16 *Which of the following sequences will converge faster to $\sqrt{5}$*

$$(a) \quad x_{n+1} = x_n + 1 - \frac{x_n^2}{5}, \qquad (b) \quad x_{n+1} = \frac{1}{3}\Big[3x_n + 1 - \frac{x_n^2}{5}\Big].$$

Use faster convergence sequence to find x_2 using $x_0 = 2$ and compute absolute error.

Solution. *It can be easily verify by using the Note 2.2. From the first sequence, we have*

$$g_1(x) = x + 1 - \frac{x^2}{5} \quad and \quad g_1'(x) = 1 - \frac{2x}{5},$$

which implies that

$$|g_1'(\sqrt{5})| = \left|1 - \frac{2\sqrt{5}}{5}\right| = 0.1056 < 1.$$

Similarly, from the second sequence, we have

$$g_2(x) = \frac{1}{3}\left[3x + 1 - \frac{x^2}{5}\right] \quad and \quad g_2'(x) = \frac{1}{3}\left[3 - \frac{2x}{5}\right], \quad gives, \quad |g_2'(\sqrt{5})| = 0.701186 < 1.$$

We note that both sequences are converging to $\sqrt{5}$ but the sequence (a) will converges faster than the sequence (b) because the value of $|g_1'(\sqrt{5})|$ is closer to zero than by $|g_2'(\sqrt{5})|$. Using $x_0 = 2$,

$x_1 = x_0 + k(x_0^2 - 5) = 2 - 0.2236(2^2 - 5) = 2.2236$, $x_2 = x_1 + k(x_1^2 - 5) = 2.2236 - 0.2236(2.2236^2 - 5) = 2.2360$,

and so $|\sqrt{5} - x_2| = |2.2361 - 2.2360| = 0.0001$.

$$x_1 = x_0 + 1 - \frac{x_0^2}{5} = 2.2320 \qquad and \quad x_2 = x_1 + 1 - \frac{x_1^2}{5} = 2.2356,$$

and so $|\sqrt{5} - x_2| = |2.2361 - 2.2356| = 0.0005$. •

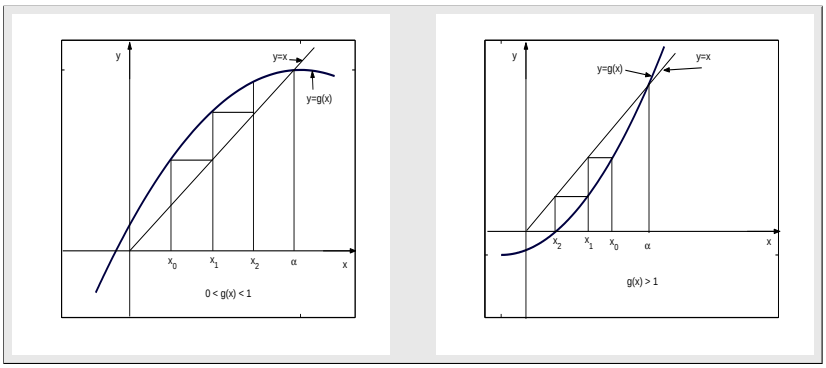


Figure 2.3.1. Convergent function iterations.

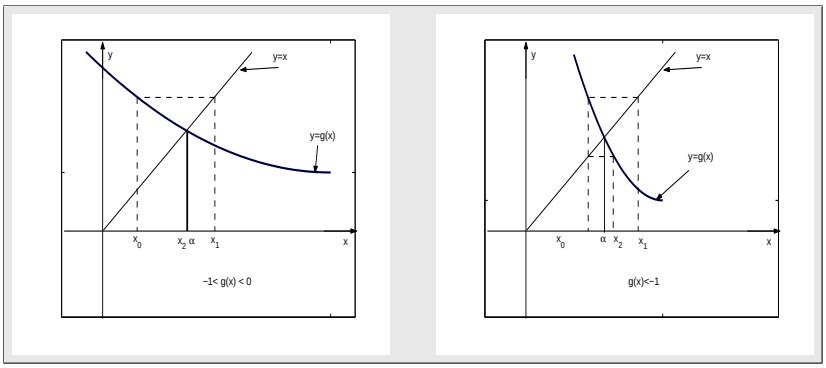


Figure 2.3.2. Divergent function iterations.

The graphical interpretation of the fixed-point method is illustrated in Figure 2.3.1(a) and in Figure 2.3.1(b). The fixed point α is the abscissa of the intersection of the graph of the function $g(x)$ with the line $y = x$. The ordinate of the function $g(x)$ at x_0 is the value of x_1. To turn this ordinate into an abscissa, reflect it in the line $y = x$. We may repeat this process to get x_2, x_3 and so on. It is seen that the iterates see Figure 2.3.1(a) moving (zigzag) towards the fixed-point, while in Figure 2.3.2(b) they going away: the iterations in Figure 2.3.2(a) converge if you start near enough to the fixed point, whereas the other diverge no matter how close you start. The fixed-point in Figure 2.3.2(a) is said to be *attractive*, and the one in Figure 2.3.2(b) is said to be *repulsive*.

Example 2.17 *Assuming the following convergent iterative schemes generated by*

$$\begin{aligned} x_{n+1} &= g_1(x_n) = \tfrac{1}{3}(x_n^2 + 2); \qquad n = 0, 1, \ldots, \\ x_{n+1} &= g_2(x_n) = 3 - \frac{2}{x_n}; \qquad n = 0, 1, \ldots. \end{aligned}$$

Show that they converge to different roots of the same equation $x^2 - 3x + 2 = 0$. *Use the iterative scheme which converges to* 1 *and find the third approximation* x_3 , *using* $x_0 = 0.5$.

Solution. *If* x_n *converges to* α, *then the first iterative scheme gives*

$$\alpha = g_1(\alpha) = \frac{1}{3}(\alpha^2 + 2), \quad \textit{that is}, \quad \alpha^2 - 3\alpha + 2 = 0,$$

which has the roots $\alpha = 1$ *and* $\alpha = 2$. *Since the first derivative of the function* $g_1(x)$ *is*

$$g_1'(x) = \frac{2}{3}x, \quad |g_1'(1)| < 1 \qquad \textit{and} \qquad |g_1'(2)| > 1.$$

Hence convergence to $\alpha = 1$. *Now if* x_n *converges to* α, *then other scheme gives*

$$\alpha = g_2(\alpha) = 3 - \frac{2}{\alpha}, \qquad \textit{that is,} \qquad \alpha^2 - 3\alpha + 2 = 0.$$

Since the first derivative of the function $g_2(x)$ *is*

$$g_2'(x) = \frac{2}{x^2}, \quad \textit{so} \quad |g_2'(2)| < 1 \quad \textit{and} \quad |g_2'(1)| > 1.$$

Hence convergence to $\alpha = 2$. *Thus first iterative scheme is the most suitable one. Using this suitable iterative scheme with* $x_0 = 0.5$, *we get*

$$x_1 = g_1(x_0) = 0.7500, \quad x_2 = g_1(x_1) = 0.8542, \quad x_3 = g_1(x_2) = 0.9099,$$

and $|\alpha - x_3| = |1 - 0.9099| = 0.0901$, *the required absolute error.* •

Example 2.18 *Consider the two following schemes:*

$$(i) \quad x_{n+1} = \sqrt{\frac{5}{x_n}}, \qquad (ii) \quad x_{n+1} = \frac{1}{3}\left[2x_n + \frac{5}{x_n^2}\right], \ n = 1, 2, \ldots$$

Which scheme will converge faster to $\sqrt[3]{5}$ *? Explain your answer. Use this faster scheme to find the third approximation, starting with* $x_0 = 1.5$.

Solution. *It can be easily verify as follows. From the first sequence, we have*

$$g_1(x) = 5^{1/2}x^{-1/2} \quad \textit{and} \quad g_1'(x) = 5^{1/2}\left[-\frac{1}{2}x^{-3/2}\right],$$

which implies that

$$|g_1'(5^{1/3})| = \left|-\frac{5^{1/2}}{2}[(5^{1/3})^{-3/2}]\right| = \frac{1}{2} = 0.5 < 1.$$

Similarly, from the second sequence, we have

$$g_2(x) = \frac{1}{3}\left[2x + \frac{5}{x^2}\right] \quad \textit{and} \quad g_2'(x) = \frac{1}{3}\left[2 - \frac{10}{x^3}\right],$$

gives

$$|g_2'(5^{1/3})| = \frac{1}{3}\left[2 - \frac{10}{(5^{1/3})^3}\right] = \frac{1}{3}[2 - 2] = 0.0.$$

We note that both sequences are converging to $\sqrt[3]{5}$ *but the second sequence (ii) will converges faster than the first sequence* **(i)** *because the value of* $|g_2'(\sqrt[3]{5})|$ *is smaller than by* $|g_1'(\sqrt[3]{5})|$.

Given $x_0 = 1.5$, using faster convergent sequence (first one), we have

$$\begin{aligned} x_1 &= \frac{1}{3}\left[2x_0 + \frac{5}{x_0^2}\right] = \frac{1}{3}\left[2(1.5) + \frac{5}{(1.5)^2}\right] = 1.7407, \\ x_2 &= \frac{1}{3}\left[2x_1 + \frac{5}{x_1^2}\right] = \frac{1}{3}\left[2(1.7407) + \frac{5}{(1.7407)^2}\right] = 1.7105, \\ x_3 &= \frac{1}{3}\left[2x_2 + \frac{5}{x_2^2}\right] = \frac{1}{3}\left[2(1.7105) + \frac{5}{(1.7105)^2}\right] = 1.7100, \end{aligned}$$

the required first three approximations and $|\alpha - x_3| = |5^{1/3} - x_3| = |1.7100 - 1.7100| = 0.0$, the required absolute error. •

Example 2.19 *Find two fixed points α_1 and α_2 of the function $g(x) = 0.5x^2 - 1.5x + 2$ and check for which fixed point the fixed-point method will converge. Compute x_2 using $x_0 = 1.5$ and compute absolute error.*

Solution. *Since $f(x) = x - g(x) = x - (0.5x^2 - 1.5x + 2) = 0$, so we have*

$$f(x) = 0.5x^2 - 2.5x + 2 = 0.$$

Solving this quadratic equation, we get, $x_1 = 1 \quad and \quad x_2 = 4$. Then

$$g(1) = 0.5(1^2) - 1.5(1) + 2 = 1 \quad and \quad g(4) = 0.5(4^2) - 1.5(4) + 2 = 4,$$

showing that $\alpha_1 = 1$ and $\alpha_2 = 4$ are the two fixed points of the given function.
Since the first derivative of the given function $g(x)$ is

$$g'(x) = x - 1.5,$$

and its absolute value at the both fixed points are

$$|g'(1)| = |1 - 1.5| = 0.5 < 1 \quad and \quad |g'(4)| = |4 - 1.5| = 2.5 > 1.$$

Therefore, from Fixed-Point Theorem 2.4, we conclude that the fixed-point method will converge for the fixed-point $\alpha_1 = 1$ only. Now using $x_0 = 1.5$, we get

$$x_1 = 0.5x_0^2 - 1.5x_0 + 2 = 0.8750 \quad and \quad x_2 = 0.5x_1^2 - 1.5x_1 + 2 = 1.0703,$$

the required first two approximations and $|\alpha - x_2| = |1 - 1.0703| = 0.0703$, the absolute error. •

Example 2.20 *One of the possible rearrangement of the nonlinear equation $e^x = x + 2$, which has root in $[1, 2]$ is*

$$x_{n+1} = g(x_n) = \ln(x_n + 2); \qquad n = 0, 1, \ldots.$$

(a) Show that $g(x)$ has a unique fixed-point in $[1, 2]$.

(b) Use fixed-point iteration formula (2.7) to compute approximation x_3, using $x_0 = 1.5$.

(c) *Compute an error estimate* $|\alpha - x_3|$ *for your approximation.*

(d) *Determine the number of iterations needed to achieve an approximation with accuracy* 10^{-2} *to the solution of* $g(x) = \ln(x+2)$ *lying in the interval* $[1,2]$ *by using the fixed-point iteration method.*

Solution. *Since, we observe that* $f(1)f(2) < 0$*, then the solution we seek is in the interval* $[1,2]$*.*

(a) *For* $g(x) = \ln(x+2)$*, we have* $g'(x) = 1/(x+2) < 1$*, for all* x *in the given interval* $[1,2]$*. Also,* g *is increasing function of* x*, and* $g(1) = \ln(3) = 1.0986123$ *and* $g(2) = \ln(4) = 1.3862944$ *both lie in the interval* $[1,2]$*. Thus* $g(x) \in [1,2]$*, for all* $x \in [1,2]$*, so from fixed-point theorem the g(x) has a unique fixed-point.*

(b) *using the given initial approximation* $x_0 = 1.5$*, we have the other approximations as*

$$x_1 = g(x_0) = 1.252763, \quad x_2 = g(x_1) = 1.179505, \quad x_3 = g(x_2) = 1.156725.$$

(c) *Since* $a = 1$ *and* $b = 2$*, then the value of* k *can be found as follows*

$$k_1 = |g'(1)| = |1/3| = 0.333 \quad \textit{and} \quad k_2 = |g'(2)| = |1/4| = 0.25,$$

which give $k = \max\{k_1, k_2\} = 0.333$*. Thus using the error formula (2.9), we have*

$$|\alpha - x_3| \leq \frac{(0.333)^3}{1 - 0.333}|1.252763 - 1.5| = 0.013687.$$

(d) *From the error bound formula (2.9), we have*

$$\frac{k^n}{1-k}|x_1 - x_0| \leq 10^{-2}.$$

By using above parts (b) and (c), we have

$$\frac{(0.333)^n}{1 - 0.333}|1.252763 - 1.5| \leq 10^{-2}, \quad n\ln(0.333) \leq \ln(0.02698), \quad \textit{gives}, \quad n \geq 3.28539.$$

So we need four (4) approximations to get the desired accuracy for the given problem. •

MATLAB command for the above given rearrangement $x = g(x)$ of $f(x) = e^x - x - 2$ by using the initial approximation $x_0 = 1.5$, can be written as follows:

function $y = fn(x)$
$y = log(x+2)$;
$>> x0 = 1.5; tol = 0.01;\ sol = fixpt('fn', x0, tol);$

Program 2.2
MATLAB m-file for the Fixed-Point Method
function sol=fixpt(fn,x0,tol)
old= x0+1; while abs(x0-old) > *tol*; old=x0;
$x0 = feval(fn, old)$; end; sol=$x0$;

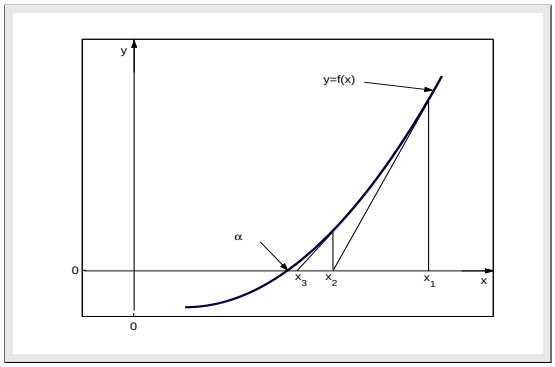


Figure 2.4: Graphical Solution of Newton's Method.

Procedure 2.2 (Fixed-Point Method)

1. *Choose an initial approximation* x_0 *such that* $x_0 \in [a, b]$.
2. *Choose a convergence parameter* $\epsilon > 0$.
3. *Compute new approximation* x_{new} *by using the iterative formula (2.7).*
4. *Check, if* $|x_{new} - x_0| < \epsilon$ *then* x_{new} *is the desire approximate root; otherwise set* $x_0 = x_{new}$ *and go to step 3.*

2.2.3 Newton's Method

This is one of the most popular and powerful iterative method for finding roots of the nonlinear equation. It is also known as the method of tangents because after estimated the actual root, the zero of the tangent to the function at that point is determined. It always converges if the initial approximation is sufficiently close to the exact solution. This method is distinguished from the methods of previous sections by the fact that it requires the evaluation of both the function $f(x)$ and the derivative of the function $f'(x)$, at arbitrary point x. The Newton's method consists geometrically of expanding the tangent line at a current point x_i until it crosses zero, then setting the next guess x_{i+1} to the abscissa of that zero crossing, see Figure 2.4. This method is also called the *Newton-Raphson method.*

There are many description of the Newton's method. We shall derive the method from the familiar Taylor's series expansion of a function in the neighborhood of a point.

Let $f \in C^2[a, b]$ and let x_n be the *nth* approximation to the root α such that $f'(x_n) \neq 0$ and $|\alpha - x_n|$ is small. Consider the first Taylor polynomial for $f(x)$ expanded about x_n, so we have

$$f(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{(x - x_n)^2}{2} f''(\eta(x)), \tag{2.12}$$

where $\eta(x)$ lies between x and x_n. Since $f(\alpha) = 0$, then (2.12), with $x = \alpha$, gives

$$f(\alpha) = 0 = f(x_n) + (\alpha - x_n)f'(x_n) + \frac{(\alpha - x_n)^2}{2} f''(\eta(\alpha)).$$

Since $|\alpha - x_n|$ is small, then we neglect the term involving $(\alpha - x_n)^2$ and so

$$0 \approx f(x_n) + (\alpha - x_n)f'(x_n) \quad \text{or} \quad \alpha \approx x_n - \frac{f(x_n)}{f'(x_n)}, \tag{2.13}$$

which should be better approximation to α than is x_n. We call this approximation as x_{n+1}, then we get

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0, \qquad \text{for all} \quad n \geq 0. \tag{2.14}$$

The iterative method (2.14) is called the Newton's method. After each iteration we should check to see if the convergence condition, namely, $|f(x_{i+1})| < \epsilon$, is satisfied.
Usually the Newton's method converges well and quickly but its convergence cannot, however guaranteed and it may sometime converge to a different root from the one expected. In particular, there may be difficulties if initial approximation is not sufficiently close to the actual root. The most serious problem of the Newton's method is that some functions are difficult to differentiate analytically, and some functions cannot be differentiated analytically at all. The Newton's method is not restricted to one-dimension only. The method readily generalizes to multiple dimensions. It should be noted that this method is suitable for finding real as well as imaginary roots of the polynomials.

Example 2.21 *Use the Newton's method to find the third approximation to the root of* $x^3 = 2x+1$ *that is located in the interval* $[1.5, 2.0]$*, take an initial approximation* $x_0 = 1.5$*.*

Solution. *Given* $f(x) = x^3 - 2x - 1$ *and so* $f'(x) = 3x^2 - 2$*. Now evaluating* $f(x)$ *and* $f'(x)$ *at the give approximation* $x_0 = 1.5$*, gives,* $f(1.5) = -0.625$ *and* $f'(1.5) = 4.750$*. Using the Newton's iterative formula (2.14), we get*

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1.5 - \frac{(-0.625)}{4.75} = 1.631579.$$

Now evaluating $f(x)$ *and* $f'(x)$ *at the new approximation* x_1*, gives*

$$x_1 = 1.631579, \qquad f(1.631579) = 0.0801869, \qquad f'(1.631579) = 5.9861501.$$

Using the iterative formula (2.14) again to get other new approximation. The successive iterates were shown in the Table 2.4. Just after the third iterations the required root is approximated to

Table 2.4: Solution of $x^3 = 2x + 1$ using Newton's method

n	x_n	$f(x_n)$	$f'(x_n)$	Error $= x - x_n$
00	1.500000	-0.625000	4.750000	0.1180339
01	1.631579	0.0801869	5.9861501	-0.0135451
02	1.618184	0.000878	5.855558	-0.0001501
03	1.618034	0.00000007	5.854102	-0.0000001

be $x_3 = 1.618034$ *and the functional value is reduced to* 7.0×10^{-8}*. Since the exact solution is* 1.6180339*, so the actual error is* 1×10^{-7}*. We see that the convergence is quite faster than the methods considered previously.* •

To get the above results using MATLAB command, firstly the function x^3-2x-1 and its derivative $3x^2-2$ were saved in m-files called fn.m and dfn.m, respectively written as follows:

$function\ y = fn(x)$	$function\ dy = dfn(x)$
$y = x.\hat{}\ 3 - 2 * x - 1;$	$dy = 3 * x.\hat{}\ 2 - 2;$

after which we do the following:

$$>> x0 = 1.5; tol = 0.01;\ sol = newton('fn','dfn', x0, tol);$$

Theorem 2.5 (Newton's Error Formula)

If $f(\alpha) = 0$ *and*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad for\ all \quad n \geq 0,$$

then there is a point η_n *between* α *and* x_n *such that*

$$E_{n+1} = \alpha - x_{n+1} = -\frac{(\alpha - x_n)^2}{2}\frac{f''(\eta_n)}{f'(x_n)}, \quad or \quad E_{n+1} = \alpha - x_{n+1} \approx -\frac{(\alpha - x_n)^2}{2}\frac{f''(x_n)}{f'(x_n)}, \quad (2.15)$$

is called the error formula of Newton's method. •

Example 2.22 *Find the root of* $x^2 - 7 = 0$ *using an initial estimate* $x_0 = 4$*, using Newton' method and compute error using formula (2.15) and exact error.*

Solution. *Let* $f(x) = x^2 - 7$ *and* $f'(x) = 2x$*, then using Newton's formula and above its error term, the successive iterates were shown in the Table 2.5. Exact root is* $\alpha = 2.64575131$. •

Table 2.5: Solution of the equation $x^2 - 7 = 0$ by using Newton's iterative method

n	x_n	$f(x_n)$	$f'(x_n)$	$f''(x_n)$	x_{n+1}	$\lvert E_{n+1}\rvert$	$\lvert E_{exact}\rvert$
0	4	9	8	2	2.87500	0.1582	0.22924
1	2.87500	1.2656	5.7500	2	2.654891	0.00843	0.00913
2	2.654891	0.04844	5.30978	2	2.645767	0.000015	0.0000157
3	2.645767	0.0000832	5.291534	2	2.64575131	4.6×10^{-11}	-

Theorem 2.6 *If simultaneously*

- $f(x)$ *has continuous first derivative* $f'(x)$ *and continuous second derivative* $f''(x)$ *on I,*
- $f'(x) \neq 0$ *for all* $x \in I$*, (*α *is simple root),*
- *the sign of* $f''(x)$ *does not change on I,*
- *choose initial approximation* $x_0 \in I$ *(which is close enough to* α*) such that* $f(x_0).f''(x_0) > 0$*, then the Newton's method will generate a sequence of numbers* $\{x_k\}_{k\geq 0}$ *that converges to zero,* α*, of* $f(x)$. •

Note that:

- The Newton's method will fail (the sequence $\{x_k\}$ does not converge) on solving function $f(x) = x^2 - 4x + 5 = 0$, since $f(x)$ does not have any real root.
- When Newton's method applied to $f(x) = \cos(x)$ with starting point $x_0 = 3$, which is close to the root $\frac{\pi}{2}$, it produces

  $$x_1 = -4.01525,\ x_2 = -4.8526, ...,$$

  which converges to another root $-\frac{3\pi}{2}$.
- When Newton's method applied to $f(x) = xe^{-x}$ with starting point $x_0 = 2.0$, it produces

  $$x_1 = 4.0,\ x_2 = 5.333, ...,$$

  the sequence $\{x_k\}$ diverges to ∞ slowly, however, $f(x_k)$ goes to zero rapidly as x_k gets larger, and could be mistaken as a zero of $f(x)$.
- When Newton's method applied to $f(x) = x^3 - x - 3 = 0$ with starting value $x_0 = 0$, it produces

  $$x_1 = -3,\ x_2 = -1.961538,\ x_3 = -1.147176,\ x_4 = -0.006078,\ x_5 = -3.000389,\ x_6 = -1.96818,$$

  so on, the sequence will not converge. But if the method starts with $x = 2$, then it produces

  $$x_1 = 1.727273,\ x_2 = 1.67369,\ x_3 = 1.6717025, ...,$$

  the sequence converges to the root $\alpha = 1,671699881$. This shows that the starting point x_0 must be close enough to the zero of $f(x)$.
- When Newton's method applied to linear function $f(x) = mx + c,\ m \neq 0$, it will converge to a root in the first iteration. Since

  $$x_{n+1} = x_n - \frac{(mx_n + c)}{m} = \frac{mx_n - mx_n - c}{m} = -\frac{c}{m}.$$

  So given any initial approximation x_0, $x_= -\frac{c}{m}$ the root.

Example 2.23 *Use Newton's method to approximate, to within 10^{-4}, the value of x that produces the point on the graph of $y = x^2$ that is closest to $(1, 0)$, using initial approximation $x_0 = 1$.*

Solution. *The distance between an arbitrary point (x, x^2) on the graph of $y = x^2$ and the point $(1, 0)$ is*

$$d(x) = \sqrt{(x-1)^2 + (x^2 - 0)^2} = \sqrt{x^4 + x^2 - 2x + 1}.$$

Because a derivative is needed to find the critical point of d, it is easier to work with the square of this function

$$F(x) = [d(x)]^2 = x^4 + x^2 - 2x + 1,$$

whose minimum will occur at the same value of x as the minimum of $d(x)$. To minimize $F(x)$, we need x so that

$$F'(x) = 4x^3 + 2x - 2 = 0, \quad \textit{gives} \quad f(x) = 4x^3 + 2x - 2 \quad \textit{and} \quad f'(x) = 12x^2 + 2.$$

Applying Newton's iterative formula (2.14) to find the approximation of this equation, we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{4x_n^3 + 2x_n - 2}{12x_n^2 + 2}.$$

Finding the approximation to the x within 10^{-4} using the initial approximation $x_0 = 1$, we get

$$x_1 = x_0 - \frac{4x_0^3 + 2x_0 - 2}{12x_0^2 + 2} = 0.7143.$$

Continue in the same manner, we get, $x_2 = 0.6052$, $x_3 = 0.5900$, $x_4 = 0.5898$, $x_5 = 0.5898$. So the point on the graph that is closest to $(1, 0)$ has the approximate coordinates $(0.5898, 0.3479)$. •

Program 2.3
MATLAB m-file for the Newton's Method
function sol=newton(fn,dfn,x0,tol)
old = x0+1; while abs $(x0 - old) > tol$; $old = x0$;
$x0 = old - feval(fn, old)/feval(dfn, old)$; end; sol=x0;

Example 2.24 *If the difference of two numbers x and y is 6 and the square root of their product is 4, then use Newton's method to approximate, to within 10^{-4}, the largest value of the number x and the corresponding number y using initial approximation $x_0 = 7.5$.*

Solution. *Given*

$$x - y = 6 \quad \textit{and} \quad \sqrt{xy} = 4.$$

Solving the above equations for x, we have

$$x(x - 6) = 16 \quad \textit{or} \quad x^2 - 6x - 16 = 0 = f(x), \quad \textit{and} \quad f'(x) = 2x - 6.$$

Applying Newton's iterative formula (2.14) to find the approximation of this equation, we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - 6x_n - 16}{2x_n - 6}.$$

Finding the approximation to within 10^{-4} using the initial approximation $x_0 = 7.5$, we get

$$x_1 = x_0 - \frac{x_0^2 - 6x_0 - 16}{2x_0 - 6} = 8.0278,$$

and continue in the same manner, we get the approximations within accuracy 10^{-4} as follows

$$x_2 = 8.0001, \quad x_3 = 8.0000, \quad x_4 = 8.0000.$$

Thus the largest value of number x is 8 and its corresponding y value is 2. •

Example 2.25 *The graphs of* $y = 2\sin x$ *and* $y = \ln(x)+k$ *touch each other in the neighborhood of point* $x = 8$. *Find the value of the constant* k *and the second approximation of the point of contact* (x, y), *use* $x_0 = 8$, *by Newton's method.*

Solution. *Since we know that the graphs will touch each other if the values of derivatives at their point of contact is same. So for*

$$y = 2\sin x, \quad \textit{gives,} \quad y' = 2\cos x, \quad \textit{and} \quad y = \ln(x) + k, \quad \textit{gives,} \quad y' = \frac{1}{x}.$$

Thus

$$2\cos x = \frac{1}{x}, \quad \textit{or} \quad x\cos x = \frac{1}{2}, \quad \textit{gives,} \quad x\cos x - 0.5 = 0,$$

and from this we have the function and its derivative as follows

$$f(x) = x\cos x - 0.5 \quad \textit{and} \quad f'(x) = \cos x - x\sin x.$$

Using Newton's iterative formula (2.14), we get

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n\cos x_n - 0.5}{\cos x_n - x_n\sin x_n},$$

and for finding the approximations, starting $x_0 = 8$, *we obtain,* $x_1 = 7.7936$ *and* $x_2 = 7.7897$. *Taking* $x = 7.79$, *we have* $y = 2\sin 7.79 = 1.996$. *Therefore, the point of contact is* $(7.79, 1.996)$. *To find the value of* k, *we solve the equation,* $1.996 = \ln(7.79) + k$, *and it gives,* $k = -0.0568$. •

Example 2.26 *Successive approximations* x_n *to the desired root are generated by the scheme*

$$x_{n+1} = \frac{1 + 3x_n^2}{4 + x_n^3}, \qquad n \geq 0.$$

Find $f(x_n)$ *and* $f'(x_n)$ *and then use the Newton's method to find the approximation of the root accurate to* 10^{-2}, *starting with* $x_0 = 0.5$.

Solution. *Given*

$$x = \frac{1 + 3x^2}{4 + x^3} = g(x),$$

and

$$x - g(x) = x - \frac{1 + 3x^2}{4 + x^3} = \frac{x^4 - 3x^2 + 4x - 1}{4 + x^3}.$$

Since, $f(x) = x - g(x) = 0$, *therefore, we have,* $f(x_n) = x_n^4 - 3x_n^2 + 4x_n - 1$ *and* $f'(x_n) = 4x_n^3 - 6x_n + 4$. *Using these functions values in the Newton's iterative formula (2.14), we have,*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^4 - 3x_n^2 + 4x_n - 1}{4x_n^3 - 6x_n + 4}.$$

Finding the first approximation of the root using the initial approximation $x_0 = 0.5$, *we get*

$$x_1 = x_0 - \frac{x_0^4 - 3x_0^2 + 4x_0 - 1}{4x_0^3 - 6x_0 + 4} = 0.5 - \frac{0.3125}{1.5} = 0.2917.$$

Similarly, the other approximations can be obtained as

$$x_2 = 0.2917 - \frac{(-0.0813)}{2.3491} = 0.3263; \quad \textit{and} \quad x_3 = 0.3263 - \frac{(-0.0029)}{2.1812} = 0.3276.$$

Notice that $|x_3 - x_2| = |0.3276 - 0.3263| = 0.0013 < 10^{-2}$. •

Example 2.27 *Successive approximations x_n to the desired root $\sqrt{3}$ are generated by the scheme*

$$x_{n+1} = \frac{1}{2}x_n + \frac{3}{2x_n}, \qquad n \geq 0.$$

Use Newton's method to find the second approximation x_2 of the root, starting with $x_0 = 2$. Show that Newton's method converges very fast ($g'(\sqrt{3}) = 0$).

Solution. *Given*

$$x_{n+1} = \frac{1}{2}x_n + \frac{3}{2x_n} = g(x_n), \qquad n \geq 0,$$

so

$$x = \frac{1}{2}x + \frac{3}{2x} = g(x), \quad f(x) = x - g(x) = \frac{1}{2}x - \frac{3}{2x}.$$

Thus

$$f(x_n) = \frac{1}{2}x_n - \frac{3}{2x_n} \quad \textit{and} \quad f'(x_n) = \frac{1}{2} + \frac{3}{2x_n^2}.$$

Using these functions values in the Newton's iterative formula (2.14), we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{\left(\frac{x_n}{2} - \frac{3}{2x_n}\right)}{\left(\frac{1}{2} + \frac{3}{2x_n^2}\right)}.$$

Finding the first approximation of the root using the initial approximation $x_0 = 2$, we get

$$x_1 = x_0 - \frac{\left(\frac{x_0}{2} - \frac{3}{2x_0}\right)}{\left(\frac{1}{2} + \frac{3}{2x_0^2}\right)} = 1.7143, \quad \textit{and} \quad x_2 = x_1 - \frac{\left(\frac{x_1}{2} - \frac{3}{2x_1}\right)}{\left(\frac{1}{2} + \frac{3}{2x_1^2}\right)} = 1.7319.$$

Since

$$g(x) = x - \frac{\left(\frac{x}{2} - \frac{3}{2x}\right)}{\left(\frac{1}{2} + \frac{3}{2x^2}\right)} = \frac{6x}{x^2+3}, \quad \textit{and} \quad g'(x) = \frac{18 - 6x^2}{(x^2+3)^2}, \quad g'(\sqrt{3}) = \frac{18 - 6(3)}{(3+3)^2} = 0,$$

which shows that the convergence of Newton's method is fast. •

Example 2.28 *Find a quadratic equation $ax^2 + bx + c = 0$ from the following iterative scheme to get successive approximations x_n to the desired root*

$$x_{n+1} = \frac{bx_n^2 + 2cx_n}{c - ax_n^2}, \qquad n \geq 0.$$

Show that if $c \neq 0$ and $ax^2 \neq c$ at a root then this iterative scheme is exactly second order. Use the scheme to find the third approximation of the positive root of the equation $2x^2 - 5x = 3$, starting with $x_0 = 2.5$. What happen if $c = 0$.

Solution. *Since $f(x) = x - g(x) = 0$, therefore, we have*

$$ax^3 + bx^2 + cx = 0 \quad or \quad x(ax^2 + bx + c) = 0.$$

For $x \neq 0$, we have the quadratic equation, $ax^2 + bx + c = 0$. Since

$$g(x) = \frac{bx^2 + 2cx}{c - ax^2} \quad and \quad g'(x) = \frac{2c(ax^2 + bx + c)}{(c - ax^2)^2}.$$

At root α, $ax^2 + bx + c$ is identically zero and $g'(\alpha) = 0$ if $c - a\alpha^2 \neq 0$. Also,

$$g''(x) = \frac{2c[2ax(ax^2 + bx + c) + ax(bx + 4c) + bc]}{(c - ax^2)^3} \quad and \quad g''(\alpha) \neq 0.$$

Finding the first three approximations of the positive root of $2x^2 - 5x = 3$ using the initial approximation $x_0 = 2.5$ and $a = 2, b = -5, c = -3$, we use the above formula by taking $n = 0, 1, 2$ as follows

$$x_1 = \frac{bx_0^2 + 2cx_0}{c - ax_0^2} = 2.9839, \quad x_2 = \frac{bx_1^2 + 2cx_1}{c - ax_1^2} = 3.0961, \quad x_3 = \frac{bx_2^2 + 2cx_2}{c - ax_2^2} = 2.9996,$$

are the possible three approximations. If $c = 0$, then the iterative scheme

$$x_{n+1} = \frac{bx_n^2}{-ax_n^2} = -\frac{b}{a},$$

gives root immediately. Note that the positive root of $2x^2 - 5x - 3 = 0$ is $\alpha = 3$, so we have, $|\alpha - x_3| = |3 - 2.9996| = 0.0004$, the possible absolute error. •

Example 2.29 *Show that the following iterative formula*

$$x_{n+1} = \frac{x_n^2 - b}{2x_n - a}, \qquad n \geq 0,$$

is Newton's formula for the approximate roots of the quadratic equation $x^2 - ax + b = 0$. Then use this formula to find the third approximation of the positive root of the equation $x^2 - 3x = 4$, starting with $x_0 = 3.5$.

Solution. *Given the nonlinear equation*

$$x^2 - ax + b = 0,$$

therefore, we have

$$f(x_n) = x_n^2 - ax_n + b \quad and \quad f'(x_n) = 2x_n - a.$$

Using these functions values in the Newton's iterative formula (2.14), we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - ax_n + b}{2x_n - a} = \frac{x_n^2 - b}{2x_n - a}, \qquad n \geq 0.$$

Finding the first three approximations of the positive root of $x^2 - 3x = 4$ *using the initial approximation* $x_0 = 3.5$ *and* $a = 3, b = -4$, *we use the above formula by taking* $n = 0, 1, 2$ *as follows*

$$x_1 = \frac{x_0^2 - b}{2x_0 - a} = 4.0625, \quad x_2 = \frac{x_1^2 - b}{2x_1 - a} = 4.0008, \quad x_3 = \frac{x_2^2 - b}{2x_2 - a} = 4.0000,$$

are the possible three approximations. Note that the positive root of $x^2 - 3x - 4 = 0$ *is* $\alpha = 4$, *so we have,* $|\alpha - x_3| = |4 - 4| = 0.0000$, *the possible absolute error.* •

The Newton's method is widely used in computers as a basis for the square root and the reciprocal evaluation.

Example 2.30 *Develop an iterative procedure for evaluating pth root of a positive number* N *by using Newton's method. Use the developed formula to find second approximation to the square root of* 19, *taking an initial approximation* $x_0 = 5$. *Compute absolute error.*

Solution. *We shall compute* $x = N^{1/p}$ *by finding a positive root for the nonlinear equation*

$$x^p - N = 0,$$

where p *is any positive integer and* $N > 0$ *is the number whose root is to be found. Therefore, if* $f(x) = 0$, *then* $x = N^{1/p}$ *is the exact root. Let*

$$f(x) = x^p - N \quad \textit{and} \quad f'(x) = px^{p-1}.$$

Hence, assuming an initial estimate to the root, say, $x = x_n$ *and by using iterative formula (2.14), we get*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{(x_n^p - N)}{px_n^{p-1}} = x_n - \frac{x_n^p}{px_n^{p-1}} + \frac{N}{px_n^{p-1}} = (1 - \frac{1}{p})x_n + \frac{N}{p}x_n^{1-p},$$

which gives Newton's formula for approximating pth root of a positive number N

$$x_{n+1} = (1 - \frac{1}{p})x_n + \frac{N}{p}x_n^{1-p}, \quad n = 0, 1, \ldots, \quad p = 2, 3, \ldots. \tag{2.16}$$

Since we want the approximations of the square root of number 19, *so we take* $N = 19$ *and* $p = 2$. *Given the initial approximation* $x_0 = 5$, *then by using the iterative formula (2.16), we get*

$$x_1 = (1 - \frac{1}{2})5 + \frac{19}{2}5^{1-2} = 4.4 \quad \textit{and} \quad x_2 = (1 - \frac{1}{2})4.4 + \frac{19}{2}4.4^{1-2} = 4.3590909.$$

After just two iterations the estimated value compares rather favorably with the exact value of $\sqrt{19} \approx 4.3588989$. *Thus the absolute error is*

$$|E| = |\sqrt{19} - x_2| = |4.3588989 - 4.3590909| = 0.000192.$$

We can calculate higher roots of a number by using the general iterative formula (2.16). •

Example 2.31 *Develop an iterative procedure for evaluating the reciprocal of a positive number N by using Newton's method. Use the developed formula to find third approximation to the reciprocal of 3, taking an initial approximation $x_0 = 0.4$. Compute absolute error.*

Solution. *Consider $x = 1/N$. This problem can be easily solved by noting that we seek to find a root to the nonlinear equation*

$$1/x - N = 0,$$

where $N > 0$ is the number whose reciprocal is to be found. Therefore, if $f(x) = 0$, then $x = 1/N$ is the exact root. Let

$$f(x) = 1/x - N \quad \text{and} \quad f'(x) = -1/x^2.$$

Hence, assuming an initial estimate to the root, say, $x = x_n$ and by using iterative formula (2.14), we get

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{(1/x_n - N)}{(-1/x_n^2)} = x_n + (1/x_n - N)x_n^2 = x_n + x_n - Nx_n^2 = x_n(2 - Nx_n).$$

which gives Newton's formula for approximating reciprocal of a positive number N

$$x_{n+1} = x_n(2 - Nx_n), \qquad n = 0, 1, \ldots. \tag{2.17}$$

Note that this sequence converges to $\alpha = \frac{1}{N}$ if and only if the initial guess x_0 satisfies $0 < x_0 < \frac{2}{N}$. If the condition on the initial guess is violated, the calculated value of x_1 and all further iterates would be negative. We have to find the approximation of the reciprocal of number $N = 3$. Given the initial gauss of say $x_0 = 0.4$, then by using the iterative formula (2.17), we get

$$x_1 = 0.4(2-3(0.4)) = 0.32, \qquad x_2 = 0.32(2-3(0.32)) = 0.3328, \qquad x_3 = 0.3328(2-3(0.3328)) = 0.3333.$$

After just three iterations the estimated value compares rather favorably with the exact value of $1/3 \approx 0.3333$. Thus the absolute error using 4 decimal places, is

$$|E| = \left|\frac{1}{3} - x_3\right| = |0.3333 - 0.3333| = 0.0000.$$

We can calculate the other reciprocal of the number in the same way by using the general iterative formula (2.17). •

Lemma 2.1 *Assume that $f \in C^2[a, b]$ and there exists a number $\alpha \in [a, b]$, where $f(\alpha) = 0$. If $f'(\alpha) \neq 0$, then there exists a number $\delta > 0$ such that the sequence $\{x_n\}_{n=0}^{\infty}$ defined by the iteration*

$$x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}, \quad \text{for} \quad n = 0, 1, \ldots, \tag{2.18}$$

will converges to α for any initial approximation $x_0 \in [\alpha - \delta, \alpha - \delta]$. •

The Newton's method uses the iteration function

$$g(x) = x - \frac{f(x)}{f'(x)}, \tag{2.19}$$

is called the *Newton's iteration function.* Since $f(\alpha) = 0$, it is easy to see that $g(\alpha) = \alpha$. Thus the Newton's iteration for finding the root of the equation $f(x) = 0$ is accomplished by finding a fixed-point of the equation $g(x) = x$.

Procedure 2.3 (Newton's Method)

1. *Find the initial approximation x_0 for the root by sketching the graph of the function.*
2. *Evaluate function $f(x)$ and the derivative $f'(x)$ at initial approximation.*

   Check: if $f(x_0) = 0$ then x_0 is the desire approximation to a root. But if $f'(x_0) = 0$, then go back to step 1 to choose new approximation.
3. *Establish Tolerance $(\epsilon > 0)$ value for the function.*
4. *Compute new approximation for the root by using the iterative formula (2.14).*
5. *Check Tolerance. If $|f(x_n)| \leq \epsilon$, for $n \geq 0$, then end; otherwise, go back to step 4, and repeat the process.*

2.2.4 Secant Method

Since we known the main obstacle to using the Newton's method is that it may be difficult or impossible to differentiate the function $f(x)$. The calculation of $f'(x_n)$ may be avoided by approximating the slope of the tangent at $x = x_n$ by that of the chord joining the two points $(x_{n-1}, f(x_{n-1}))$ and $(x_n, f(x_n))$, see Figure 2.5.

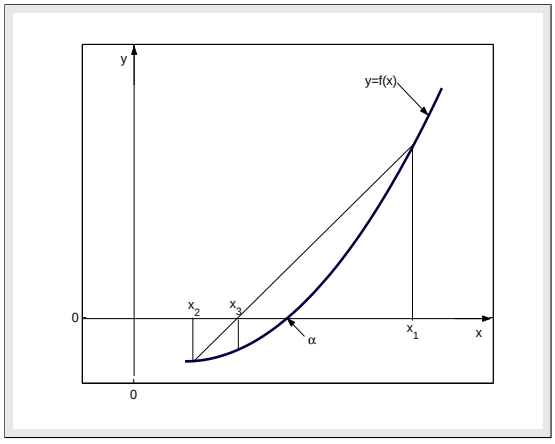


Figure 2.5: Graphical Solution of Secant Method.

The slope of the chord (or secant) is

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \tag{2.20}$$

Then by using this approximation of the derivative of the function in the Newton's iterative formula (2.14), we get

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}, \qquad n \geq 1. \tag{2.21}$$

Note that when $f(x_n) = f(x_{n-1})$, the calculation of x_{n+1} fails. This is because the chord is horizontal. The iterative formula (2.21) known as the *secant method*. It needs two initial approximations to start. This method is very similar to false position method described in Section 2.3. However, for the secant method it is not necessary for the interval to contain a root and no account is taken of signs of the numbers $f(x_n)$.
This method suffers from the same disadvantages as the Newton's method, that is, convergence to a particular root cannot guaranteed but nevertheless it is a powerful general purpose method. The order of convergence of the secant method is $(1 + \sqrt{5})/2 \approx 1.618$, so its ultimate convergence is not quite as fast as the Newton's method (order of convergence is quadratically) but the order of convergence of this method is somewhat better than the bisection method, the false position method, and the fixed-point method (all these methods have linear convergence). This is sometimes called *superlinear*. We will discuss the order of convergence of all these methods in some details later in the chapter.

Example 2.32 *Show that the iterative procedure for evaluating the reciprocal of a number N by using the secant method is*

$$x_{n+1} = x_n + (1 - Nx_n)x_{n-1}, \qquad n \geq 1. \tag{2.22}$$

Use this iterative formula to find the third approximation of the reciprocal of number $N = 5$ by using the initial approximations $x_0 = 0$ and $x_1 = 0.1$. Compute absolute error.

Solution. *Let N be a positive number and $x = 1/N$ or $1/x = N$. If $f(x) = 0$, then $x = \alpha = 1/N$ is the exact zero of the function*

$$f(x) = 1/x - N \qquad \textit{whicih can be written as} \qquad f(x_n) = 1/x_n - N.$$

Assuming the initial estimates to the root, say, $x = x_{n-1}, x = x_n$ and by using the secant iterative formula, we have

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})} = x_n - \frac{(x_n - x_{n-1})(1/x_n - N)}{(1/x_n - N) - (1/x_{n-1} - N)}, \quad n \geq 1,$$

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})(1/x_n - N)}{-(x_n - x_{n-1})/x_n x_{n-1}} = x_n + (1/x_n - N)x_n x_{n-1}.$$

It gives

$$x_{n+1} = x_n + (1 - Nx_n)x_{n-1}, \qquad n = 1, 2, \ldots.$$

Starting with $x_0 = 0$ and $x_1 = 0.1$, and using the above iterative formula, we get the first three approximations as follows:

$$x_2 = 0.1+(1-5(0.1))(0) = 0.1, \qquad x_3 = 0.1+(1-5(0.1))(0.1) = 0.15, \qquad x_4 = 0.15+(1-5(0.15))(0.1) = 0.175,$$

with the absolute error $|\alpha - x_4| = |1/5 - 0.175| = 0.025$. •

Example 2.33 *Show that the secant method for finding approximation of the square root of a positive number N is*

$$x_{n+1} = \frac{x_n x_{n-1} + N}{x_n + x_{n-1}}, \quad n \geq 1. \tag{2.23}$$

Carry out the first three approximations for the square root of 9, *using* $x_0 = 2, x_1 = 2.5$ *and also compute absolute error.*

Solution. *We shall compute* $x = N^{1/2}$ *by finding a positive root for the nonlinear equation* $x^2 - N = 0$, *where* $N > 0$ *is the number whose root is to be found. If* $f(x) = 0$, *then* $x = \alpha = N^{1/2}$ *is the exact zero of the function* $f(x) = x^2 - N$. *Assuming* $x = x_n$, $f(x_n) = x_n^2 - N$ *and* $x = x_{n-1}$, $f(x_{n-1}) = x_{n-1}^2 - N$, *then the secant formula becomes*

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})f(x_n)}{f(x_n) - f(x_{n-1})} == x_n - \frac{(x_n - x_{n-1})f(x_n)}{(x_n^2 - N) - (x_{n-1}^2 - N)}, \quad n \geq 1.$$

Hence, assuming the initial estimates to the root, say, $x = x_{n-1}, x = x_n$, *and by using the secant iterative formula, we have*

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})(x_n^2 - N)}{(x_n^2 - N) - (x_{n-1}^2 - N)} = x_n - \frac{(x_n - x_{n-1})(x_n^2 - N)}{(x_n - x_{n-1})(x_n + x_{n-1})},$$

which gives the secant formula for approximating the square root of a number N

$$x_{n+1} = \frac{x_n x_{n-1} + N}{x_n + x_{n-1}}, \qquad n = 1, 2, \ldots,$$

Now using this formula for approximation of the square root of $N = 9$, *taking* $x_0 = 2$ *and* $x_1 = 2.5$, *we have*

$$x_2 = \frac{(2.5)(2) + 9}{2.5 + 2} = 3.1111, \qquad x_3 = \frac{(3.1111)(2.5) + 9}{3.1111 + 2.5} = 2.9901, \qquad x_4 = \frac{(2.9901)(3.1111) + 9}{2.9901 + 3.1111} = 2.9998.$$

Hence, the absolute error is, $|9^{1/2} - x_4| = |3 - 2.9998| = 0.0002$. •

Example 2.34 *Use the secant method to find the approximate root of the equation* $x^3 = 2x + 1$ *within the accuracy* 10^{-2} *using* $x_0 = 1.5$ *and* $x_1 = 2.0$.

Solution. *Since* $f(x) = x^3 - 2x - 1$ *and*

$$x_0 = 1.5, \quad f(1.5) = -0.625, \quad x_1 = 2.0, \quad f(2.0) = 3.0,$$

therefore, we see that $f(x_0) \neq f(x_1)$. *Hence, one can use the iterative formula (2.21), to get new approximation:*

$$x_2 = \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)} = \frac{(1.5)(3.0) - (2.0)(-0.625)}{3.0 - (-0.625)} = 1.586207,$$

and $f(1.586207) = -0.18434$. *Similar way, we can find the other possible approximation of the root. A summary of the calculations is given in Table 2.6.* •

To use MATLAB command for the secant method, the function has been used in the m-file as fn.m, then the first few iterations are easily performed by the following sequence of MATLAB commands:

```
>> x0 = 1.5; x1 = 2; x2 = x1 - (x1 - x0)/(fn(x1) - fn(x0)) * fn(x1)
>> x0 = x1; x1 = x2; x3 = x2 - (x2 - x1)/(fn(x2) - fn(x1)) * fn(x2)
```

The last two commands can be repeated to generate the subsequent iterates shown in Table 2.6.

Table 2.6: Solution of $x^3 = 2x + 1$ by secant method

n	x_{n-1}	x_n	x_{n+1}	$f(x_{n+1})$
01	1.500000	2.000000	1.586207	-0.1814342
02	2.000000	1.586207	1.609805	-0.0478446
03	1.586207	1.609805	1.618257	0.0013040

Program 2.4
MATLAB m-file for the Secant Method
function sol=secant(fn,a,b,tol)
$x0 = a; x1 = b; fa = feval(fn, x0); fb = feval(fn, x1);$
while abs(x1-old)> tol $new = x1 - fb * (x1 - x0)/(fb - fa);$
$x0 = x1; fa = fb; x1 == new; fb = feval(fn, new);$ end; sol=new;

We see that the functional values are approaching zero as the number of iterations is increased a and note that this method converges *faster than* the methods discussed in Sections 2.1 and 2.2 because we got the desired approximation $c_3 = 1.618032$ after 4 iterations with accuracy $\epsilon = 10^{-2}$.

One can use MATLAB single command to get the same above results by the secant method as

$>> a = 1.5; b = 2.0; tol = 1e-2;\ sol = secant('fn', a, b, tol);$

Example 2.35 *Use secant method to find the second approximation, using $x_0 = 1$ and $x_1 = 2$, of the value of x that produces the point on the graph of $y = \dfrac{1}{x}$ that is closest to the point $(2, 1)$.*

Solution. *The distance between an arbitrary point $(x, 1/x)$ on the graph of $y = 1/x$ and the point $(2, 1)$ is*

$$d(x) = \sqrt{(x-2)^2 + (1/x - 1)^2} = \sqrt{x^2 - 4x + 4 + 1/x^2 - 2/x + 1}.$$

Because a derivative is needed to find the critical point of d, it is easier to work with the square of this function

$$F(x) = [d(x)]^2 = x^2 - 4x + 4 + 1/x^2 - 2/x + 1,$$

whose minimum will occur at the same value of x as the minimum of $d(x)$. To minimize $F(x)$, we need x so that

$$F'(x) = 2x - 4 - 2/x^3 + 2/x^2 = 0, \quad \textit{from this,} \quad f(x) = 2x - 4 - 2/x^3 + 2/x^2.$$

Applying secant iterative formula to find the approximation of this equation, we have

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})(2x_n - 4 - 2/x_n^3 + 2/x_n^2)}{(2x_n - 4 - 2/x_n^3 + 2/x_n^2) - (2x_{n-1} - 4 - 2/x_{n-1}^3 + 2/x_{n-1}^2)}, \qquad n \geq 1.$$

Finding the second approximation using the initial approximations $x_0 = 1$ and $x_1 = 2$, we get

$$x_2 = x_1 - \frac{(x_1 - x_0)(2x_1 - 4 - 2/x_1^3 + 2/x_1^2)}{(2x_1 - 4 - 2/x_1^3 + 2/x_1^2) - (2x_0 - 4 - 2/x_0^3 + 2/x_0^2)} = 2 - 1/9 = 1.8889,$$

and

$$x_3 = x_2 - \frac{(x_2 - x_1)(2x_2 - 4 - 2/x_2^3 + 2/x_2^2)}{(2x_2 - 4 - 2/x_2^3 + 2/x_2^2) - (2x_1 - 4 - 2/x_1^3 + 2/x_1^2)} = 1.8667.$$

The point on the graph that is closest to $(2, 1)$ *has the approximate coordinates* $(1.8667, 0.5356)$. •

Example 2.36 *Show that the x-value of the intersection point* (x, y) *of the graphs* $y = x^3 + 2x - 1$ *and* $y = \sin x$ *is lying in the interval* $[0.5, 1]$. *Then use Secant method to find its second approximation, when* $x_0 = 0.5$ *and* $x_1 = 0.55$. *Also, find the intersection point.*

Solution. *For the intersection of the graphs, we mean that* $x^3 + 2x - 1 = \sin x$ *and it gives,* $x^3 + 2x - 1 - \sin x = 0$. *Thus,* $f(x) = x^3 + 2x - \sin x - 1$. *Since* $f(x)$ *is continuous on* $[0.5, 1.0]$ *and* $f(0.5) = -0.3544$, $f(1.0) = 1.1585$, *which shows that* $f(0.5)f(1.0) < 0$. *Hence the x-value (or root of* $f(x) = 0$*) lies in the interval* $[0.5, 1.0]$. *Applying Secant iterative formula to find the approximation of this root of the equation, we have*

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})(x_n^3 + 2x_n - \sin x_n - 1)}{(x_n^3 + 2x_n - \sin x_n - 1) - (x_{n-1}^3 + 2x_{n-1} - \sin x_{n-1} - 1)}, \qquad n \geq 1.$$

Finding the first approximation using the initial approximations $x_0 = 0.5$ *and* $x_1 = 0.55$, *we get*

$$x_2 = 0.55 - \frac{(0.55 - 0.5)((0.55)^3 - 2(0.55) - \sin(0.55) - 1)}{((0.55)^3 - 2(0.55) - \sin(0.55) - 1) - ((0.5)^3 - 2(0.5) - \sin(0.5) - 1)} = 0.6806,$$

and the second approximation using the initial approximations $x_1 = 0.55$ *and* $x_2 = 0.6806$, *we get*

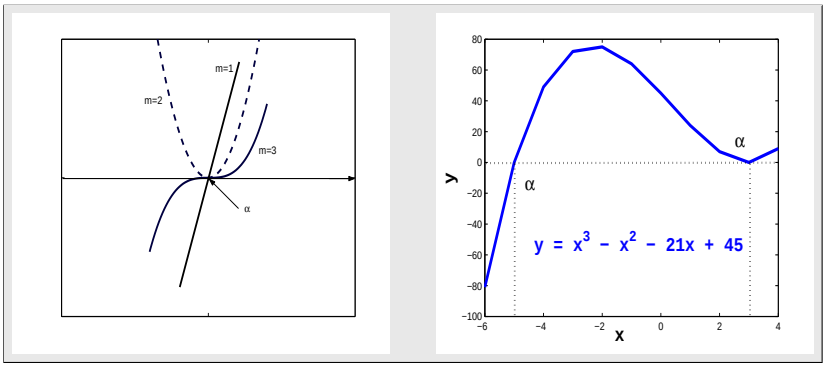
$$x_3 = 0.6806 - \frac{(0.6806 - 0.55)((0.0.6806)^3 - 2(0.6806) - \sin(0.6806) - 1)}{((0.0.6806)^3 - 2(0.6806) - \sin(0.6806) - 1) - (0.55 - 0.5)((0.55)^3 - 2(0.55) - \sin(0.55) - 1)},$$

So $x_3 = 0.6603$ *is the second approximation of the x-value of the intersection point* $(0.6603, 0.61)$. •

Procedure 2.4 (Secant Method)

1. *Choose the two initial approximation* x_0 *and* x_1.
2. *Check, if* $f(x_0) = f(x_1)$, *go to step 1 otherwise, continue.*
3. *Establish Tolerance* $(\epsilon > 0)$ *value for the function.*
4. *Compute new approximation for the root by using the iterative formula (2.21).*
5. *Check tolerance. If* $|x_n - x_{n-1}| \leq \epsilon$, *for* $n \geq 1$, *then end; otherwise, go back to step 4, and repeat the process.*

In the preceding discussion the restriction was made that $f'(\alpha) \neq 0$, where α is the solution to $f(x) = 0$. The rapid convergence of both the Newton's method and the secant method depend because of this restriction. From the definition of the Newton's method, it is clear that difficulties might occur if $f'(x_n)$ goes to zero simultaneously with $f(x_n)$. In particular, the Newton's method and the secant method will generally give problems if $f'(\alpha) = 0$ when $f(\alpha) = 0$. In the following section we investigate this situation and will uncover an interesting fact, namely, how fast the iteration converges. We will see that both the Newton's method and the secant method will continue to converge, but not as rapidly as we expect.



Multiple roots of $f(x) = 0$.

Figure 2.6: Graphical Solution of Multiple root of $f(x) = 0$.

2.3 Approximation of Multiple Root of a Nonlinear Equation

Here, we shall discuss only one numerical iterative method called, modified Newton's method. For the approximation of a multiple root of nonlinear equation we can also use bisection method, fixed-point method, Newton,s method, and secant method. But the best method is modified Newton,s method.

2.3.0.1 Multiplicity of a Multiple Root

So far we discussed about the function which has simple root. Now we will discuss about the function which has multiple roots. A root is called a *simple root* if it is distinct, otherwise roots that are of the same order of magnitude are called *multiple*.

Definition 2.4 *(**Order of Multiplicity of a Multiple Root**)*

The equation $f(x) = 0$ has a multiple root α of order m, if there exists a continuous function $h(x)$, and $f(x)$ can be expressed as the product

$$f(x) = (x - \alpha)^m h(x), \qquad \text{where} \quad h(\alpha) \neq 0. \tag{2.24}$$

So $h(x)$ can be used to obtain the remaining roots of $f(x) = 0$. It is called polynomial deflation. •

A root of order $m = 1$ is called a *simple root* and if $m > 1$ it is called *multiple root*. In particular, a root of order $m = 2$ is sometimes called a *double root*, and so on.
The behavior of the graph of $f(x)$ near a root of multiplicity m ($m = 1, 2, 3$) is shown in Figure 2.6. It can be seen that when α is a root of odd multiplicity, the graph of $f(x)$ will cross the x-axis at $(\alpha, 0)$; and when α has even multiplicity the graph will be tangent to but will not cross the x-axis at $(\alpha, 0)$. Moreover, the higher the value of m the flatter the graph will be near the point $(\alpha, 0)$.
Sometime it is more difficult to deal with the Definition 2.4 concerning about the order of the multiple root. We will use the following Lemma which will illuminate these concepts.

Lemma 2.2 *Assume that function $f(x)$ and its derivatives $f'(x), f''(x), \cdots, f^{(m)}(x)$ are defined and continuous on an interval about $x = \alpha$. Then $f(x) = 0$ has a multiple root α of order m if and only if*

$$f(\alpha) = f'(\alpha) = f''(\alpha) = \cdots = f^{(m-1)}(\alpha) = 0, \qquad f^{(m)}(\alpha) \neq 0. \tag{2.25}$$

For example, consider the equation $f(x) = x^3 - x^2 - 21x + 45 = 0$, which has three roots; a simple root at $\alpha = -5$ and a double root at $\alpha = 3$. This can be verified by considering the derivatives of the function as follows

$$f'(x) = 3x^2 - 2x - 21, \qquad f''(x) = 6x - 2.$$

At the value $\alpha = -5$, we have $f(5) = 0$ and $f'(5) = 64 \neq 0$, so by (2.25), we see that $m = 1$. Hence $\alpha = -5$ is a simple root of the equation. For the value $\alpha = 3$, we have

$$f(3) = 0, \qquad f'(3) = 0, \qquad f''(3) = 16 \neq 0,$$

so that $m = 2$ by (2.25), hence $\alpha = 3$ is a double root of the equation. Note that this function $f(x)$ has the factorization and can be written in the form of (2.24) as (see Figure **??**),

$$f(x) = (x - 3)^2(x + 5).$$

Note that for a *simple root* α of the nonlinear equation $f(x) = 0$, means that

$$f(\alpha) = 0 \quad \text{but} \quad f'(\alpha) \neq 0.$$

But for *multiple root* α of the nonlinear equation, we must have

$$f(\alpha) = 0 \quad \text{and} \quad f'(\alpha) = 0.$$

The order of multiplicity of the multiple root can be easily find out by taking the higher derivatives of the function at α unless the higher derivative becomes nonzero at α. Then the order of nonzero higher derivative will be the order of multiplicity of the multiple root. •

Example 2.37 *Find the multiplicity of the multiple root $\alpha = 1$ of the equation $x \ln x = \ln x$.*

Solution. *From the given equation, we have*

$$\begin{aligned} f(x) &= x \ln x - \ln x && \text{and} && f(1) = 0, \\ f'(x) &= \ln x + 1 - \frac{1}{x} && \text{and} && f'(1) = 0, \\ f''(x) &= \frac{1}{x} + \frac{1}{x^2} && \text{and} && f''(1) = 2 \neq 0. \end{aligned}$$

Thus the multiplicity of the multiple root $\alpha = 1$ of the given equation is 2. •

Usually we don't know in advance that an equation has multiple roots, although we might suspect it from sketching the graph. Many problems which leads to multiple roots, are in fact ill-posed. The methods we discussed so far cannot be guaranteed to converge efficiently for all problems. In particular, when a given function has a multiple root which we require, the methods we have described will either not converge at all or converge more slowly. For example, the Newton's method converges very fast to simple root but converges more slowly when used for functions involving multiple roots.

Example 2.38 *Consider the following two nonlinear equations*

$$(1) \qquad xe^x = 0 \qquad\qquad (2) \qquad x^2e^x = 0.$$

(a) Find the Newton's formulas for the solutions of the both equations. Explain why one of the sequences converges much faster than the other to the root $\alpha = 0$.

Solution. *For the first equation, we have, $f(x) = xe^x$ and $f'(x) = (1+x)e^x$. Then the Newton's method for the solution of the first equation is*

$$x_{n+1} = g_1(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} = \frac{x_n^2}{(1+x_n)}, \qquad n \geq 0,$$

which is the first sequence. Similarly, we can find the Newton's method for the solution of the second equation as follows:

$$x_{n+1} = g_2(x_n) = x_n - \frac{x_n^2e^{x_n}}{(2x_n + x_n^2)e^{x_n}} = \frac{x_n + x_n^2}{(2+x_n)}, \qquad n \geq 0,$$

and it is the second sequence. From the first sequence, we have, $g_1(x) = \dfrac{x^2}{(1+x)}$ and $g_1'(x) = \dfrac{x^2+2x}{(1+x)^2}$. Then, $|g_1'(\alpha)| = |g_1'(0)| = \left|\dfrac{0}{1}\right| = 0$, which shows that the first sequence converges to zero. Similarly, from the second sequence, we have

$$g_2(x) = \frac{x+x^2}{(2+x)} \qquad g_2'(x) = \frac{x^2+4x+2}{(2+x)^2}, \qquad |g_2'(0)| = \left|\frac{2}{4}\right| = \frac{1}{2} < 1,$$

which shows that the second sequence is also converges to zero. Since the value of $|g_1'(0)|$ is smaller than $|g_2'(0)|$, therefore, the first sequence converges faster than the second one. •

Note that in the above Example 2.38, the root $\alpha = 0$ is the simple root for the first equation because

$$f(0) = 0 \qquad \text{but} \qquad f'(0) = 1 \neq 0,$$

and for the second equation it is a multiple root because

$$f(0) = 0 \qquad \text{and} \qquad f'(0) = 0.$$

Therefore, the Newton's method converges very fast for the first equation and converges very slow for the second equation. However, in some cases simple modifications can be made to the methods to maintain the rate of convergence. Two such modified methods are considered here, called the Newton modified methods.

2.3.1 First Modified Newton's Method

If we wish to determine a root of known multiplicity m for the equation $f(x) = 0$, then the *first Newton's modified method* (also called the *Schroeder's method*) may be used. It has the form

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}, \qquad m > 1 \; f'(x_n) \neq 0, \quad n = 0, 1, 2, \ldots. \tag{2.26}$$

It is assumed that we have an initial approximation x_0. The similarity to the Newton's method is obvious and like the Newton's method it converges very fast for the multiple roots. The major disadvantage of this method is that the multiplicity of the root must be known in advance and this is generally not the case in practice.

Example 2.39 *Show that $\alpha = 1$ is root of the equation $1 = xe^{1-x}$. Use the best numerical method to find second the approximation to this root using $x_0 = 0.75$. Compute absolute error.*

Solution. *To check $\alpha = 1$ is a root of $1 - xe^{1-x} = 0$, we do*

$$f(x) = 1 - xe^{1-x}, \quad f(0) = 1 - 1e^{1-1} = 1 - 1 = 0,$$

which shows that $\alpha = 1$ is root of the given equation. Now we check the type of the root as

$$f'(x) = -e^{1-x} + xe^{1-x} = (x-1)e^{1-x}, \quad f'(0) = 0,$$

which shows that the $\alpha = 1$ is a multiple root of the given equation. To find its order of multiplicity, we do as

$$f''(x) = e^{1-x} - (x-1)e^{1-x}, \quad f''(0) = 1 \neq 0,$$

which shows that the order of multiplicity of the multiple root is 2. So using modified Newton's method by taking $x_0 = 0.75$, we get first two approximation

$$x_1 = x_0 - 2\frac{1 - x_0e^{1-x_0}}{(x_0-1)e^{1-x_0}} = 0.9804 \quad \textit{and} \quad x_2 = x_1 - 2\frac{1 - x_0e^{1-x_0}}{(x_0-1)e^{1-x_0}} = 0.9999,$$

and the absolute error, $|\alpha - x_2| = |1 - 0.9999| = 0.0001$. •

Example 2.40 *Show that Modified Newton's formula for the triple root $\alpha = 2$ of the equation $x^5 - 8x^4 + 25x^3 - 38x^2 + 28x - 8 = 0$, is*

$$x_{n+1} = \frac{(2x_n^5 - 8x_n^4 - 38x_n^2 - 56x_n + 24)}{(5x_n^4 - 32x_n^3 + 75x_n^2 - 76x_n + 28)}, \qquad n \geq 0.$$

Using $x_0 = 2.3$, find the third approximation x_3 of the multiple root $\alpha = 2$.

Solution. *Let $f(x) = x^5 - 8x^4 + 25x^3 - 38x^2 + 28x - 8$, then $f'(x) = 5x^4 - 32x^3 + 75x^2 - 76x + 28$. Then*

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)} = x_n - 3\frac{(x_n^5 - 8x_n^4 + 25x_n^3 - 38x_n^2 + 28x_n - 8)}{(5x_n^4 - 32x_n^3 + 75x_n^2 - 76x_n + 28)},$$

which can be written as

$$x_{n+1} = \frac{(2x_n^5 - 8x_n^4 - 38x_n^2 - 56x_n + 24)}{(5x_n^4 - 32x_n^3 + 75x_n^2 - 76x_n + 28)}, \quad n \geq 0.$$

$$\begin{aligned}
x_1 &= \frac{(2x_0^5 - 8x_0^4 - 38x_0^2 - 56x_0 + 24)}{(5x_0^4 - 32x_0^3 + 75x_0^2 - 76x_0 + 28)} = 2.04, \\
x_2 &= \frac{(2x_1^5 - 8x_1^4 - 38x_1^2 - 56x_1 + 24)}{(5x_1^4 - 32x_1^3 + 75x_1^2 - 76x_1 + 28)} = 2.0010000 \\
x_3 &= \frac{(2x_2^5 - 8x_2^4 - 38x_2^2 - 56x_2 + 24)}{(5x_2^4 - 32x_2^3 + 75x_2^2 - 76x_2 + 28)} = 2.00000066555741.
\end{aligned}$$

Note that $f(x_3) = f(2.00000066555741) = -7.105427357601002 \times 10^{-15}$. •

2.3.2 Second Modified Newton's Method

An alternative approach to this problem that does not require any knowledge of the multiplicity of the root is to replace the function $f(x)$ in the equation by $q(x)$, where

$$q(x) = \frac{f(x)}{f'(x)}.$$

One can show that $q(x)$ has only a simple root at $x = \alpha$. Thus the Newton's method applied to find a root of $q(x)$ will avoid any problems of multiple roots. If

$$f(x) = (x-\alpha)^m h(x), \quad \text{and} \quad f'(x) = m(x-\alpha)^{m-1}h(x) + (x-\alpha)^m h'(x).$$

Thus

$$q(x) = \frac{(x-\alpha)h(x)}{[mh(x) + (x-\alpha)h'(x)]}.$$

Obviously we find that $q(x)$ has the root α to multiplicity one. So with this modification, the Newton's method becomes

$$x_{n+1} = x_n - \frac{q(x_n)}{q'(x_n)} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - [f(x_n)][f''(x_n)]}, \quad n = 0, 1, 2, \ldots \tag{2.27}$$

This iterative formula (2.27) is known as the *second modified Newton's method*. The disadvantage of this method is that we must calculate a further higher derivative.

Example 2.41 *Use the second modified Newton's method to find the first approximation x_1 to the multiple root of the nonlinear equation $1 - \cos(x) = 0$, using $x_0 = 0.1$.*

Solution. *Since $f(x) = 1 - \cos x$, we have $f'(x) = \sin x$ and $f''(x) = \cos x$. Now using the second modified Newton's formula (2.27)*

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - [f(x_n)][f''(x_n)]} = x_n - \frac{(1-\cos x_n)(\sin x_n)}{[\sin x_n]^2 - (1-\cos x_n)(\cos x_n)}, \quad n \geq 0.$$

Using $x_0 = 0.1$, we obtain

$$x_1 = x_0 - \frac{(1-\cos x_0)(\sin x_0)}{[\sin x_0]^2 - (1-\cos x_0)(\cos x_0)} = 0.1 - \frac{(1-\cos 0.1)(\sin 0.1)}{[\sin 0.1]^2 - (1-\cos 0.1)(\cos 0.1)} = 0.098,$$

which is the required first approximation to the multiple $\alpha = 0$. •

Example 2.42 *Show that the function $f(x) = e^x - \dfrac{x^2}{2} - x - 1$ has zero of multiplicity 3 at $\alpha = 0$ and then, find the approximate solution of the zero of the function with the help of the Newton's method, first and second modified Newton's methods, by taking initial approximation $x_0 = 1.5$ within an accuracy of 10^{-4}.*

Solution. *Since $\alpha = 0$ is a root of $f(x) = 0$, so*

$$\begin{array}{lllllll} f(x) & = & e^x - \dfrac{x^2}{2} - x - 1, & f(0) & = & 0, \\ f'(x) & = & e^x - x - 1, & f'(0) & = & 0, \\ f''(x) & = & e^x - 1, & f''(0) & = & 0, \\ f'''(x) & = & e^x, & f'''(0) & = & 1 \neq 0, \end{array}$$

the function has zero of multiplicity 3. In Table 2.7 we showed the comparison of three methods. •

Table 2.7: Comparison results of three methods for the Example 2.42

n	Newton's Method x_n	1st. M.N. Method x_n	2nd. M.N. Method x_n
00	1.500000	1.500000	1.500000
01	1.067698	0.2030926	-0.297704
02	0.745468	3.482923e-03	-6.757677e-03
03	0.513126	1.010951e-06	-3.798399e-06
..	.....		
25	7.331582e-05		

To use MATLAB command for the first modified Newton's method (2.26) and the second modified Newton's method (2.27), we define a function m-file as fn1.m and its derivatives m-files as dfn1.m and ddfn1.m for the equation as follows:

$function\ y = fn1(x)$	$function\ dy = dfn1(x)$	$function\ ddy = ddfn1(x)$
$y = exp(x) - x.\hat{}\ 2/2 - x - 1;$	$dy = exp(x) - x - 1;$	$ddy = exp(x) - 1;$

then use the following commands:

$$>> x0 = 1.5; m = 3; tol = 1e-4;\ sol = mnewton1('fn1', 'dfn1', x0, m, tol)$$

and

$$\begin{array}{l} >> x0 = 1.5; tol = 1e-4; \\ >> sol = mnewton2('fn1', 'dfn1', 'ddfn1', x0, tol) \end{array}$$

We note that for the multiple root the both modified Newton's methods converge very fast as they took 3 iterations to converge while the Newton's method converges very slow and took 25 iterations to converge for the same accuracy.

Example 2.43 *The equation $1 - 2\cos(x) + \cos^2(x) = 0$ has multiple root $\alpha = 0$. Develop the Modified Newton's formula for computing this root, then use it to find the x_2 using $x_0 = 0.5$. Show that the developed formula converges faster by showing $g'(0) = 0$.*

Solution. *Since $\alpha = 0$ is a root of $f(x)$, so*

$$\begin{array}{lcllcl} f(x) & = & 1-2\cos(x)+\cos^2(x)=(1-\cos(x))^2, & f(0) & = & 0, \\ f'(x) & = & 2\sin(x)(1-\cos(x)), & f'(0) & = & 0, \\ f''(x) & = & 2\sin^2(x)-2\cos^2(x)+2\cos(x), & f''(0) & = & 0, \\ f'''(x) & = & 4\sin(2x)-2\sin(x), & f'''(0) & = & 0, \\ f^{(4)}(x) & = & 8\cos(2x)-2\cos(x), & f^{(4)}(0) & = & 6\neq 0, \end{array}$$

the function has zero of multiplicity 4. Using modified Newton's iterative formula (2.26), we get

$$x_{n+1}=x_n-m\frac{f(x_n)}{f'(x_n)}=x_n-4\frac{(1-\cos(x_n))^2}{2\sin(x_n)(1-\cos(x_n))}=x_n-\frac{2(1-\cos(x_n))}{\sin(x_n)},\quad n\geq 0.$$

Now evaluating this at the give approximation $x_0 = 0.5$, gives

$$x_1=x_0-\frac{2(1-\cos(x_0))}{\sin(x_0)}=-0.0107\quad \textit{and}\quad x_2=x_1-\frac{2(1-\cos(x_1))}{\sin(x_1)}=1.0209\times 10^{-7}.$$

The fixed point form of the developed modified Newton's formula is

$$x_{n+1}=g(x_n)=x_n-\frac{2(1-\cos(x_n))^2}{\sin(x_n)(1-\cos(x_n))}=x_n-\frac{2(1-\cos(x_n))}{\sin(x_n)},$$

where

$$g(x)=x-\frac{2(1-\cos(x))}{\sin(x)}.$$

By taking derivative, we have

$$g'(x)=1-\left(\frac{2\sin^2(x)-2(1-\cos(x))\cos(x)}{\sin^2(x)}\right)=1-\left(\frac{2(1-\cos(x))}{\sin^2(x)}\right).$$

$$g'(0)=1-\left(\frac{2(1-\cos(x))}{\sin^2(x)}\right)\qquad \left(\frac{0}{0}\right).$$

Using L'Hopital's rule, gives

$$g'(0)=1-\left(\frac{2\sin(x))}{2\sin(x)\cos(x)}\right)\qquad \left(\frac{0}{0}\right).$$

Again using L'Hopital's rule, we have

$$g'(0)=1-\left(\lim_{x\to 0}\frac{2\cos(x)}{-2\sin^2(x)\cos(x)+2\cos^2(x)}\right)=1-\frac{2}{2}=1-1=0.$$

Thus the method converges fast to the given root. •

Example 2.44 *Find a root of the equation $x^3-5x^2+7x-3=0$ by using Newton's and second modified Newton's method, taking $x_0 = 4$.*

Solution. *Since* $f(x) = x^3 - 5x^2 + 7x - 3$, *we have* $f'(x) = 3x^2 - 10x + 7$ *and* $f''(x) = 6x - 10$. *Now using the Newton's formula,*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^3 - 5x_n^2 + 7x_n - 3}{3x_n^2 - 10x_n + 7}, \quad for \quad n \geq 0,$$

gives the following approximations

$$x_1 = 3.4, \quad x_2 = 3.1, \quad x_3 = 3.008696, \quad x_4 = 3.000075, \quad x_5 = 3.000000,$$

and the second modified Newton's formula, for $n \geq 0$

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - [f(x_n)][f''(x_n)]} = x_n - \frac{(x_n^3 - 5x_n^2 + 7x_n - 3)(3x_n^2 - 10x_n + 7)}{(3x_n^2 - 10x_n + 7)^2 - (x_n^3 - 5x_n^2 + 7x_n - 3)(6x - 10)}.$$

Using above formula for $n = 0, 1, 2, 3, 4$, *we get*

$$x_1 = 2.636364, \quad x_2 = 2.820225, \quad x_3 = 2.961728, \quad x_4 = 2.998479, \quad x_5 = 2.999998.$$

Therefore, one root of the given equation is $\alpha = 3$*(simple root). Both methods converge quickly, with Newton's method being somewhat more efficient.*
To find other root of the equation with Newton's method, using $x_0 = 0$, *gives*

$$x_1 = 0.4285714, x_2 = 0.6857143, x_3 = 0.8328654, x_4 = 0.9133290, x_5 = 0.9557832, x_6 = 0.977655,$$

and by using second modified Newton's method, using $x_0 = 0$, *gives*

$$x_1 = 1.105263, \qquad x_2 = 1.003082, \qquad and \qquad x_3 = 1.000002.$$

Thus the other root of the given equation is $\alpha = 1$*(multiple root), therefore, second modified Newton's method converges more quickly than Newton's method.* •

Note 2.3 *If* α *is a root of equation* $f(x) = 0$ *with multiplicity* m, *then* α *is a root of equation* $f'(x) = 0$ *with multiplicity* $m - 1$, *of equation* $f''(x) = 0$ *with multiplicity* $m - 2$ *and so on. Hence the expression*

$$x_0 - m\frac{f(x_0)}{f'(x_0)}, \; x_0 - (m-1)\frac{f'(x_0)}{f''(x_0)}, \; x_0 - (m-2)\frac{f''(x_0)}{f'''(x_0)}, \ldots,$$

should have the same value if there is a root with multiplicity m, *when the initial guess is very close to the exact root* α. •

Example 2.45 *Find the double root of the equation* $x^3 - 3x^2 + 4 = 0$, *using* $x_0 = 1.5$.

Solution. *Let* $f(x) = x^3 - 3x^2 + 4$, *then* $f'(x) = 3x^2 - 6x$, $f''(x) = 6x - 6$. *Then*

$$x_1 = x_0 - 2\frac{f(x_0)}{f'(x_0)} = 1.5 - 2\frac{0.625}{-2.25} = 2.05556,$$

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 1.5 - \frac{-2.25}{3} = 2.25000.$$

The close values of x_1 *indicates that there is a double root near* 2. *Let* $x_1 = 2.05556$, *then*

$$x_2 = x_1 - 2\frac{f(x_1)}{f'(x_1)} = 2.05556 - 2\frac{0.00943}{0.34262} = 2.00051,$$

$$x_2 = x_1 - \frac{f'(x_1)}{f''(x_1)} = 2.05556 - \frac{0.34262}{6.33336} = 2.00146.$$

Thus there is a double root at $x = 2.00051$ *which is sufficiently close to the actual root* 2. •

2.3.2.1 Problems with Multiple Roots

Multiple roots pose difficulties for the methods discussed so far.

- The bisection method has difficulties with multiple roots because the function does not change sign at even multiple roots.
- The Newton's and secant methods have difficulties because the derivative at a multiple root is zero.
- Since $f(x)$ reaches zero at a faster rate than $f'(x)$ as x approaches the multiple root, it is impossible to check for the condition $f(x) = 0$ and terminate the computations before $f'(x) = 0$.

Program 2.5
MATLAB m-file for First Modified Newton's Method
function sol=mnewton1(fn1,dfn1,x0,m,tol)
old = x0+1; while abs $(x0 - old) > tol$; $old = x0$;
fa=feval(fn,old); fb=feval(dfn,old);
$x0 = old - (m * fa)/fb$; end; sol=x0;

Note that when the order of multiplicity of a root of the equation $f(x) = 0$ is not known, then the second modified Newton's formula (2.27) can be used. MATLAB m-file can be written as follows

Program 2.6
MATLAB m-file for Second Modified Newton's Method
function sol=mnewton2(fn1,dfn1,ddfn1,x0,tol)
old = x0+1; while abs $(x0 - old) > tol$; $old = x0$;
fa=feval(fn,old); fb=feval(dfn,old); fc=feval(ddfn,old);
$x0 = old - (fa * fb)/((fb).\hat{}\ 2 - (fa * fc))$;end; sol=x0;

2.4 Convergence of Iterative Methods

Now we define the order of the convergence of functional iteration schemes discussed in the previous sections. This is a measure of how rapidly a sequence converges.

Definition 2.5 (Order of Convergence)

Suppose that the sequence $\{x_n\}_{n=0}^{\infty}$ *converges to* α*, and let* $e_n = \alpha - x_n$ *define the error of the nth iterate. If two positive constants* $\beta \neq 0$ *and* $R > 0$ *exist, and*

$$\lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|^R} = \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^R} = \beta, \tag{2.28}$$

then the sequence is said to converge to α *with order of convergence* R*. The number* β *is called the asymptotic error constant. The cases* $R = 1, 2$ *are given special consideration.*
If $R = 1$*, the convergence of the sequence* $\{x_n\}_{n=0}^{\infty}$ *is called linear.*
If $R = 2$*, the convergence of the sequence* $\{x_n\}_{n=0}^{\infty}$ *is called quadratic.*

If R is large, the sequence $\{x_n\}$ converges rapidly to α; that is, (2.28) implies that for large values of n we have the approximation $|e_{n+1}| \approx \beta |e_n|^R$. For example, suppose that $R = 2$ and $|e_n| \approx 10^{-3}$; then we could expect that $|e_{n+1}| \approx \beta \times 10^{-6}$. •

Example 2.46 *Show that the following sequence*

$$x_{n+1} = \frac{1}{2} x_n \left(1 + \frac{N}{x_n^2}\right), \qquad n \geq 0,$$

will converge quadratically to $\sqrt{N}$*.*

Solution. *Since the sequence is given as*

$$x_{n+1} = \frac{1}{2} x_n \left(1 + \frac{N}{x_n^2}\right),$$

and $\alpha = \sqrt{N}$*, then we have*

$$\begin{aligned} x_{n+1} - \sqrt{N} &= \frac{1}{2} x_n \left(1 + \frac{N}{x_n^2}\right) - \sqrt{N} = \frac{1}{2}\left(x_n + \frac{N}{x_n} - 2\sqrt{N}\right) \\ &= \frac{1}{2}\left(\sqrt{x_n} - \frac{\sqrt{N}}{\sqrt{x_n}}\right)^2 = \frac{1}{2x_n}(x_n - \sqrt{N})^2. \end{aligned}$$

Thus

$$e_{n+1} = \frac{1}{2x_n} e_n^2 \qquad or \qquad e_{n+1} \propto e_n^2,$$

which shows the quadratic convergence. •

Example 2.47 *Find the value of* a *and* b *so that the rate of convergence of the iterative scheme*

$$x_{n+1} = a x_n + b \left(\frac{N}{x_n^2}\right), \quad for \quad n \geq 0,$$

for computing $N^{1/3}$ *becomes quadratically or higher.*

Solution. *Since we have*

$$x = N^{1/3} \qquad or \qquad x^3 = N, \quad gives \quad f(x) = x^3 - N.$$

Let α be the exact root and $e_n = \alpha - x_n$, be the error in nth step, then by substituting

$$\alpha^3 = N, \quad x_n = \alpha + e_n, \quad x_{n+1} = \alpha + e_{n+1},$$

in the given iterative scheme, we get

$$\begin{aligned}
\alpha + e_{n+1} = a(\alpha + e_n) + b\left(\frac{\alpha^3}{(\alpha+e_n)^2}\right) &= a(\alpha + e_n) + b\left(\frac{\alpha^3}{\alpha^2\left(1 + e_n/\alpha\right)^2}\right) \\
&= a(\alpha + e_n) + b\alpha\left(1 + \frac{e_n}{\alpha}\right)^{-2} \\
&= a(\alpha + e_n) + b\alpha\left\{1 - 2\frac{e_n}{\alpha} + 3\left(\frac{e_n}{\alpha}\right)^2 - \ldots\right\} \\
&= a(\alpha + e_n) + b\alpha - 2be_n + 3b\frac{e_n^2}{\alpha} - \ldots.
\end{aligned}$$

Thus

$$e_{n+1} = (a + b - 1)\alpha + (a - 2b)e_n + O(e_n^2) + \ldots.$$

Now for the method to become of order 2, we must have

$$a + b - 1 = 0 \quad and \quad a - 2b = 0,$$

and by solving this system for a and b, we have, $a = \dfrac{2}{3}$ and $b = \dfrac{1}{3}$, the required values. •

Quadratically convergent sequences generally converge much faster than those that converge only linearly, but many techniques that generate convergent sequences do so only linearly. The following two lemmas tell us about the conditions of the linear convergence and the quadratic convergence of the sequences.

Lemma 2.3 (Linear Convergence (R = 1))

Let g is continuously differentiable on the interval $[a, b]$ and suppose that $g(x) \in [a, b]$ for all $x \in [a, b]$. Suppose that $g'(x)$ is continuous on (a, b) with

$$|g'(x)| \leq k < 1; \qquad for\ all \quad x \in (a, b).$$

If $g'(\alpha) \neq 0$, then for any $x_0 \in [a, b]$, the sequence $x_{n+1} = g(x_n),\ for\ n \geq 0$, converges only linearly to the unique fixed-point α in $[a, b]$. •

Example 2.48 *Consider an iterative scheme*

$$x_{n+1} = 0.4 + x_n - 0.1x_n^2, \qquad n \geq 0.$$

Will this scheme converge to the fixed-point $\alpha = 2$? If yes, find its rate of convergence.

Solution. *Since $g(x) = 0.4 + x - 0.1x^2$ and $g'(x) = 1 - 0.2x$, so g(2) = 2, and $|g'(2)| < 1$ which shows that the scheme converges to $\alpha = 2$. As $g'(2) = 1 - 0.4 = 0.6 \neq 0$, therefore, the scheme converges linearly ($R = 1$).* •

Lemma 2.4 (Quadratic Convergence (R = 2))

Let α be a solution of the equation $x = g(x)$. Suppose that $g'(\alpha) = 0$ and g'' is continuous on an open interval (a, b) containing α. Then there exists a $\delta > 0$ such that, for $x_0 \in [\alpha - \delta, \alpha + \delta]$, the sequence $\{x_n\}_{n=0}^{\infty}$ defined by the iteration $x_{n+1} = g(x_n)$, for $n \geq 0$, converges at least quadratically to α. •

Example 2.49 *The iterative scheme $x_{n+1} = 2 - (1 + k)x_n + ax_n^2, \quad n \geq 0$, converges to $\alpha = 1$ for some values of a. Find the value of k for which the convergence is at least quadratic.*

Solution. *Since $g(x) = 2 - (1 + k)x + ax^2$ and $g(1) = 2 - (1 + k) + k = 1$, so, the given iterative scheme converges to 1. Also*

$$g'(x) = -(1 + k) + 2kx, \quad g'(1) = 0 = -(1 + k) + 2k, \quad \textit{gives}, \quad k = 1.$$

Thus, the convergence of the given iterative scheme is at least quadratic for the value of $k = 1$. •

Note 2.4 *The sequence $\{x_n\}_{n=0}^{\infty}$ defined by the iteration*

$$x_{n+1} = g(x_n), \quad \textit{for} \qquad n \geq 0,$$

converges only quadratically to α if

$$g'(\alpha) = 0 \quad \textit{but} \quad g''(\alpha) \neq 0,$$

and cubically (order three) to α if

$$g'(\alpha) = 0, \quad g''(\alpha) = 0 \quad \textit{but} \quad g'''(\alpha) \neq 0.$$

In general, it converges to α with order of convergence equal R, if

$$g(\alpha) = \alpha, \quad g'(\alpha) = g''(\alpha) = \cdots = g^{(R-1)}(\alpha) = 0, \quad \textit{but} \quad g^{(R)}(\alpha) \neq 0.$$

Example 2.50 *(a) Find the values of k_1 and k_2 such that the iterative scheme*

$$x_{n+1} = k_1 x_n^2 + \frac{k_2}{x_n} - 5, \qquad n \geq 0,$$

converges quadratically to $\alpha = 1$.
(b) What is the order of convergence of the iteration as it converges to the fixed-point $\alpha = \sqrt{k}$.

$$x_{n+1} = \frac{x_n(x_n^2 + 3k)}{3x_n^2 + k}, \qquad k > 0,$$

Solution. *(a) Given*

$$g(x) = k_1x^2 + \frac{k_2}{x} - 5, \quad g(1) = 1 = k_1 + k_2 - 5, \quad \textit{gives}, \quad k_1 + k_2 = 6.$$

Also

$$g'(x) = 2k_1x - \frac{k_2}{x^2}, \quad g'(1) = 0 = 2k_1 - k_2, \quad \textit{gives}, \quad 2k_1 - k_2 = 0.$$

Solving these two equations for unknowns k_1 and k_2, we obtain $k_1 = 2$ and $k_2 = 4$. Note that

$$g''(x) = 2k_1 + \frac{2k_2}{x^3} \quad \textit{and} \quad g''(1) = 12 \neq 0.$$

(b) Since the given iteration is

$$x_{n+1} = \frac{x_n(x_n^2 + 3k)}{3x_n^2 + k} = g(x_n), \quad \textit{which gives}, \quad g(x) = \frac{x(x^2 + 3k)}{3x^2 + k}.$$

The first derivative of $g(x)$ can be found as

$$g'(x) = \frac{3(x^2 - k)^2}{(3x^2 + k)^2}, \quad g'(\sqrt{k}) = \frac{3[(\sqrt{k})^2 - k]^2}{[3(\sqrt{k})^2 + k]^2} = 0.$$

Therefore, the order of convergence for the given iteration is at least quadratic. One can find the second derivative of $g(x)$ as

$$g''(x) = \frac{48xk(x^2 - k)}{(3x^2 + k)^3}, \quad g''(\sqrt{k}) = 0, \ \textit{but} \ g'''(x) = \frac{-48k(9x^4 - 18kx^2 + b^2)}{(3x^2 + k)^4}, \quad g'''(\sqrt{k}) = \frac{3}{2k} \neq 0.$$

Hence, the order of convergence for the given iteration is exactly cubic. •

Now we discuss the rate of the convergence of all the iterative methods for the nonlinear equations which we discussed in the previous sections.

2.4.1 Convergence of Bisection Method

The convergence of the *bisection method* is very slow. At each step we gain one binary digit in accuracy. Since $10^{-1} \approx 2^{-3.3}$, we gain on the average one decimal digit per 3.3 steps. Note that the rate of convergence is completely independent of the function $f(x)$. This is because we only make use of the sign of the computed function values.

To investigate the rate of convergence of the bisection method, suppose that f is a continuous function on the interval [a, b] and the equation $f(x) = 0$ has a root α in [a, b].

Let $[a_n, b_n]$, $n = 1, 2, ...,$ be the successive subintervals generated by applying the bisection method, where $a_1 = a$ and $b_1 = b$. Then the bisection method generates a sequence of approximations $\{x_n := \frac{a_n + b_n}{2}\}$, $n \geq 1$, satisfies

$$|e_n| = |x_n - \alpha| \leq \frac{b - a}{2^n}.$$

Thus, the rate of convergence of $\{x_n\}_{n=1}^{\infty}$ to α is as the same as the convergence of the sequence $\{2^{-n}\}$ to 0. But the later one converges linearly to 0, therefore the bisection method converges linearly, that is $e_{n+1} \approx e_n$, $n \geq 1$, which explains the slow rate of convergence of this method.

Example 2.51 *Let α be the root of the equation $f(x) = 0$ in the interval $[x, \alpha]$, where f is a continuous function. Show that the sequence generated by the bisection method for approximating α converges only linearly.*

Solution. Since α is an end point of the given interval, by Equation (2.2), the bisection iterative formula for approximating α can be written as

$$x_{n+1} = \frac{x_n + \alpha}{2}, \ n \geq 0.$$

Thus, setting $g(x) = \dfrac{x + \alpha}{2}$, implies $g'(x) = \dfrac{1}{2}$. Hence we have $g'(\alpha) = \dfrac{1}{2} \neq 0$, therefore Lemma 2.3 implies that the convergence is linear. •

2.4.2 Convergence of Fixed-point Method

The convergence rate of the *fixed-point iteration* can be analyzed as follows. The general procedure is given by

$$x_{n+1} = g(x_n), \qquad n = 0, 1, 2, \ldots \tag{2.29}$$

Let $x = \alpha$ denote the solution to $f(x) = 0$, so $f(\alpha) = 0$ and $\alpha = g(\alpha)$. Then

$$x_{n+1} - \alpha = e_{n+1} = g(x_n) - g(\alpha), \tag{2.30}$$

where e_{n+1} denote the error of the *(n+1)th* iterate. Expressing $g(\alpha)$ in the Taylor series about x_n gives:

$$g(\alpha) = g(x_n) + g'(\eta)(\alpha - x_n), \qquad x_n \leq \eta \leq \alpha. \tag{2.31}$$

Solving (2.31) for $g(x_n) - g(\alpha)$ and substituting into (2.30), we get

$$e_{n+1} = g'(\eta)e_n, \qquad \text{or} \qquad |e_{n+1}| = |g'(\eta)||e_n|. \tag{2.32}$$

Now suppose that $|g'(x)| \leq k < 1$ for all values of x in an interval. If x_1 is choose in this interval, x_2 will also be in the interval and the fixed-point iteration method will converge, since

$$\left|\frac{e_{n+1}}{e_n}\right| = |g'(\eta)| < 1. \tag{2.33}$$

Convergence is linear since e_{n+1} is linearly dependent on e_n. If $|g'(\eta)| > 1$, the procedure diverges. If $|g'(\eta)| < 1$, but close to one, convergence is quite slow. •

Example 2.52 *If α and β are roots of the nonlinear equation $x^2 + ax + b = 0$, then show that the following iterative schemes*

$$x_{n+1} = -\left(\frac{ax_n + b}{x_n}\right) = g_1(x_n), \qquad n \geq 0,$$

will converge near $x = \alpha$ if $|\alpha| > |\beta|$ and the other scheme

$$x_{n+1} = -\left(\frac{b}{x_n + a}\right) = g_2(x_n), \qquad n \geq 0,$$

will converge near $x = \alpha$ if $|\alpha| < |\beta|$.

Solution. *Since α and β are roots of the nonlinear equation $x^2 + ax + b = 0$, then we have*

$$\alpha + \beta = -a \quad \textit{and} \quad \alpha\beta = b.$$

The first iterative scheme will converge to $x = \alpha$ if

$$|g_1'(\alpha)| < 1, \quad \textit{that is,} \quad \left|\frac{b}{\alpha^2}\right| < 1,$$

$$|\alpha|^2 > |b| = |\alpha||\beta|, \quad \textit{gives} \quad |\alpha| > |\beta|.$$

The second iterative scheme will converge to $x = \alpha$ if

$$|g_2'(\alpha)| < 1, \quad \textit{that is,} \quad \left|\frac{b}{(\alpha + a)^2}\right| < 1,$$

$$(\alpha + a)^2 > |b| \quad \textit{or} \quad |\beta|^2 > |b|, \qquad (\textit{since} \quad \alpha + a = -\beta).$$

Hence, $|\beta|^2 > |b| = |\alpha||\beta|$, gives $|\beta| > |\alpha|$, or, $|\alpha| < |\beta|$. •

2.4.3 Convergence of Newton's Method

The convergence rate of the *Newton's method* can be analyzed as follows. The general procedure

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 0, 1, 2, \ldots,$$

is of the form $x_{n+1} = g(x_n)$. Consequently, if the method converges, then the absolute value of the derivative of the function $g(x)$ with respect to x must be less than one, that is, $|g'(x)| < 1$. Since

$$g(x) = x - \frac{f(x)}{f'(x)}, \quad g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

Hence if

$$\left|\frac{f(x)f''(x)}{[f'(x)]^2}\right| < 1, \quad \text{or} \quad |f(x)f''(x)| < [f'(x)]^2, \tag{2.34}$$

on an interval about the root α, the method will converge for any initial approximation in the interval. The (2.34) represents a sufficient condition for convergence. It is evident that $f'(x)$ must not be zero. This is an important factor to consider when choosing the initial x value.
Now we show that the Newton's method is quadratically convergent for the simple root. Let $x = \alpha$ denote the solution to $f(x) = 0$, so $f(\alpha) = 0$ and $\alpha = g(\alpha)$. Since $x_{n+1} = g(x_n)$, we can write

$$x_{n+1} - \alpha = e_{n+1} = g(x_n) - g(\alpha), \tag{2.35}$$

where e_n denote the error of the *nth* iterate. Let us expand $g(x_n)$ as a Taylor series in terms of $(x_n - \alpha)$ with the second derivative term as the remainder:

$$g(x_n) = g(\alpha) + g'(\alpha)(x_n - \alpha) + \frac{g''(\eta)}{2}(x_n - \alpha)^2, \quad x_n \le \eta \le \alpha.$$

Since

$$g'(\alpha) = \frac{f(\alpha)f''(\alpha)}{[f'(\alpha)]^2} = 0, \quad \text{because} f(\alpha) = 0.$$

$$g(x_n) = g(\alpha) + \frac{g''(\eta)}{2}(x_n - \alpha)^2.$$

Solving above equation for $[g(x_n) - g(\alpha)]$ and substituting into (2.35), we get

$$e_{n+1} = g(\alpha) - g(x_n) = -\frac{g''(\eta)}{2}(e_n)^2. \tag{2.36}$$

This implies that each error is (in the limit) proportional to the square of the previous error, that is, the Newton's method is quadratically convergent.

Example 2.53 *The nonlinear equation $f(x) = \tan x = 0$ has a root $\alpha = \pi$. Develop the best numerical method for computing this root, then use it to find the x_2 using $x_0 = 0.5$. Find the rate of convergence of the developed formula.*

Solution. *As $f(x) = \tan x$ and so $f'(x) = \sec^2 x$, and*

$$f(\pi) = \tan(\pi) = 0, \qquad f'(\pi) = \sec^2(\pi) \neq 0,$$

therefore, the root is the simple root of the given nonlinear equation and the best numerical method is Newton's method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{\tan(x_n)}{\sec^2(x_n)} = x_n - \sin(x_n)\cos(x_n), \ n \geq 0.$$

To find the second approximation to the root by using above scheme using $x_0 = 0.5$, we obtain

$$x_1 = x_0 - \sin(x_0)\cos(x_0) = 0.0793, \quad x_2 = x_1 - \sin(x_1)\cos(x_1) = 0.00033.$$

Since the fixed-point form of the Newton's method is

$$g(x) = x - \frac{\tan x}{\sec^2 x} = x - \sin x \cos x, \qquad g(\pi) = \pi - \sin(\pi)\cos(\pi) = \pi,$$

therefore,

$$\begin{aligned}
g(x) &= x - \sin x \cos x, \quad g(\pi) = \pi - \sin(\pi)\cos(\pi) = \pi, \\
g'(x) &= 1 + \sin^2(x) - \cos^2(x) = 0, \quad g'(\pi) = 1 + \sin^2(\pi) - \cos^2(\pi) = 0, \\
g''(x) &= 2\sin(x)\cos^2(x) + 2\sin^2(x)\cos(x), \quad g''(x) = 2\sin(\pi)\cos^2(\pi) + 2\sin^2(\pi)\cos(\pi) = 0, \\
g'''(x) &= 2\cos^3(x) - 4\sin^2(x)\cos(x) - 2\sin^3(x) + 4\sin(x)\cos^2(x), \\
g'''(\pi) &= 2\cos^3(\pi) - 4\sin^2(\pi)\cos(\pi) - 2\sin^3(\pi) + 4\sin(\pi)\cos^2(\pi) = -2 \neq 0.
\end{aligned}$$

Hence the convergence of the Newton's method is cubic. •

Since we know that rate of convergence of the Newton's method is linear if the function has multiple root. In the following example we discuss the rate of the convergence of the Newton's method for the multiple root. •

Example 2.54 *Show that the rate of convergence of Newton's method at the root $\alpha = 0$ of the equation $x^2e^x = 0$ is linear. Use quadratic convergence method to find x_2 using $x_0 = 0.1$. Compute the absolute error.*

Solution. *Since $f(x) = x^2e^x$ and $f'(x) = (x^2 + 2x)e^x$, and $f'(0) = 0$, gives that $\alpha = 0$ is the multiple root. Using Newton's iterative formula, we get*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{(x_n^2 e^{x_n})}{(x_n^2 + 2x_n)e_n^x} = \frac{(x_n + x_n^2)}{(2 + x_n)}, \quad n \geq 0.$$

The fixed point form of the developed Newton's formula is

$$x_{n+1} = g(x_n) = \frac{(x_n + x_n^2)}{(2 + x_n)}.$$

Then

$$g(x) = \frac{(x + x^2)}{(2 + x)}, \quad g'(x) = \frac{(x^2 + 4x + 2)}{(2 + x)^2}, \quad g'(0) = \frac{1}{2} \neq 0.$$

Thus Newton's method converges linearly. The quadratic convergent method for multiple root is modified Newton's method

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}, \qquad n \geq 0,$$

where m is the order of multiplicity of the zero of the function. To find m, we check that $f''(x) = (x^2 + 4x + 2)e^x$, and $f''(0) = 2 \neq 0$, so $m = 2$. Thus

$$x_{n+1} = x_n - 2\frac{f(x_n)}{f'(x_n)} = x_n - 2\frac{x_n^2 e^{x_n}}{(x_n^2 + 2x_n)e_n^x} = x_n - 2\frac{x_n^2}{(x_n^2 + 2x_n)}, \qquad n \geq 0.$$

Now using initial approximation $x_0 = 0.1$, we have the following two approximations

$$x_1 = x_0 - 2\frac{x_0^2}{(x_0^2 + 2x_0)} = 0.00476, \quad x_2 = x_1 - 2\frac{x_1^2}{(x_1^2 + 2x_1)} = 0.0000311,$$

with the absolute error, $|\alpha - x_2)| = |0.0 - 0.0000311| = 0.0000311$. •

2.4.4 Convergence of Secant Method

The convergence rate of the *secant method* can be analyzed as follows. The general procedure is

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \tag{2.37}$$

As before, let $x_{n-1} = \alpha - e_{n-1}, x_n = \alpha - e_n$, and $x_{n+1} = \alpha - e_{n+1}$. Then

$$e_{n+1} = e_n - \frac{f(\alpha - e_n)(e_n - e_{n-1})}{f(\alpha - e_n) - f(\alpha - e_{n-1})}$$

Since by using the Taylor's theorem

$$y_n = f(\alpha - e_n) = f(\alpha) - e_n f'(\alpha) + \frac{e_n^2}{2!}f''(\alpha) - \cdots$$

$$y_{n-1} = f(\alpha - e_{n-1}) = f(\alpha) - e_{n-1}f'(\alpha) + \frac{e_{n-1}^2}{2!}f''(\alpha) - \cdots$$

therefore, we have

$$\begin{aligned}
e_{n+1} &= e_n - \frac{(e_n - e_{n-1})[f(\alpha) - e_n f'(\alpha) + 1/2e_n^2 f''(\alpha) - \cdots]}{[-(e_n - e_{n-1})f'(\alpha) + 1/2(e_n^2 - e_{n-1}^2)f''(\alpha) - \cdots]} \\
&= e_n - \left[\frac{-e_n f'(\alpha) + 1/2e_n^2 f''(\alpha) - \cdots}{-f'(\alpha) + 1/2(e_n + e_{n-1})f''(\alpha) - \cdots}\right] \quad (\text{because} f(\alpha) = 0) \\
&= e_n - \frac{1}{f'(\alpha)}[-e_n f'(\alpha) + 1/2e_n^2 f''(\alpha) - \cdots] \\
&\times \quad [-1 + 1/2(e_n + e_{n-1})\tfrac{f''(\alpha)}{f'(\alpha)} - \cdots]^{-1} \\
&= e_n - \frac{1}{f'(\alpha)}[e_n f'(\alpha) + 1/2e_n^2 f''(\alpha) + \cdots] \\
&\times \quad [1 - 1/2(e_n + e_{n-1})\frac{f''(\alpha)}{f'(\alpha)} + \cdots]^{-1} \\
&= e_n - \frac{1}{f'(\alpha)}[e_n f'(\alpha) - 1/2e_n^2 f''(\alpha) + 1/2e_n(e_n + e_{n-1})f''(\alpha) - \cdots] \\
&= e_n - \frac{1}{f'(\alpha)}[-e_n f'(\alpha) + 1/2e_n e_{n-1} f''(\alpha) - \cdots] = -\frac{f''(\alpha)}{2f'(\alpha)}e_n e_{n-1} + \cdots
\end{aligned}$$

Hence

$$e_{n+1} \approx K e_n e_{n-1}, \quad \text{where} \quad K = -\frac{f''(\alpha)}{2f'(\alpha)}, \tag{2.38}$$

so that the each error is proportional to the product of the previous two errors. By comparison with the Newton's method convergence we expect the rate of the convergence of the secant method is inferior to that of the Newton's method. If we put, $e_n = \beta e_{n-1}^R$, where R is the order of convergence and the constant β is called the *asymptotic error constant*, then we obtain

$$e_{n+1} = \beta e_n^R = \beta(\beta e_{n-1}^R)^R, \quad \text{and} \quad \beta(\beta e_{n-1}^R)^R \approx K\beta e_{n-1}^R e_{n-1}.$$

Thus

$$e_{n-1}^{R^2} \approx \lambda e_{n-1}^{R+1},$$

for some constant λ, it follows that $R^2 = R + 1$, gives $R = \dfrac{1 \pm \sqrt{5}}{2} = 1.61803$, neglecting the negative value. This formula for R tells us that

$$|e_n| \approx \beta |e_{n-1}|^{1.61803}.$$

Thus the error of the secant method is of order 1.61803, which is in-between 1 and 2. This shows that the order of the secant method is better than the bisection method and fixed-point method

but less than the Newton's method. Remember, however, that the secant method does not require the derivative of $f(x)$ to be evaluated at each step, so that in many ways the secant method is a very attractive alternative to the standard Newton's method. •

Example 2.55 *Find the values of a, b and c such that the iterative scheme*

$$x_{n+1} = ax_n + \frac{bN}{x_n^2} + \frac{cN^2}{x_n^5}, \qquad n \geq 0,$$

converges at least cubically to $\alpha = N^{\frac{1}{3}}$. Use this scheme to find second approximation of $(27)^{\frac{1}{3}}$ when $x_0 = 2.8$.

Solution. *Given the iterative scheme converges at least cubically means $g' = g'' = 0$ at $\alpha = N^{\frac{1}{3}}$. Let*

$$g(x) = ax + \frac{bN}{x^2} + \frac{cN^2}{x^5}, \quad g(N^{\frac{1}{3}}) = 1 = a + b + c,$$

$$g'(x) = a - \frac{2bN}{x^3} - \frac{5cN^2}{x^6}, \quad g'(N^{\frac{1}{3}}) = 0 = a - 2b - 5c,$$

$$g''(x) = 0 + \frac{6bN}{x^4} + \frac{30cN^2}{x^7}, \quad g''(N^{\frac{1}{3}}) = 0 = 3b + 15c,$$

Solving these three equations for unknowns a, b and c, we obtain $a = b = \frac{5}{9}$ and $c = -\frac{1}{9}$. Thus

$$x_{n+1} = \frac{5x_n}{9} + \frac{5N}{9x_n^2} - \frac{N^2}{9x_n^5}, \qquad n \geq 0.$$

Using $x_0 = 2.8,\ N = 27$, we obtain

$$x_1 = \frac{5x_0}{9} + \frac{5N}{9x_0^2} - \frac{N^2}{9x_0^5} = 2.9982 \quad \textit{and} \quad x_2 = \frac{5x_1}{9} + \frac{5N}{9x_1^2} - \frac{N^2}{9x_1^5} = 3.0000,$$

the required approximations with absolute error $|(27)^{1/3} - 3.0000| = 0.00000$. •

Example 2.56 *Choose a constant λ to ensure the rapid convergence of the iterative scheme*

$$x_{n+1} = \frac{\lambda x_n + x_n^{-2} + 1}{\lambda + 1}, \qquad n \geq 0,$$

to the root lying in the interval $[1.4, 1.5]$, using Newton's method by computing x_2, if $x_0 = 1.46$.

Solution. *Since $\lim_{n\to\infty} x_n = \lim_{n\to\infty} x_{n+1} = \alpha$, we have, $\alpha = \left[\frac{\lambda\alpha + \alpha^{-2} + 1}{\lambda + 1}\right]$, and it gives*

$$(\lambda + 1)\alpha^3 = \lambda\alpha^3 + \alpha^2 + 1, \quad \textit{or} \quad \alpha^3 - \alpha^2 - 1 = 0,$$

and α can be obtained by finding a root of the equation $x^3 - x^2 - 1 = 0$. Therefore, we have

$$f(x_n) = x_n^3 - x_n^2 - 1 \quad \textit{and} \quad f'(x_n) = 3x_n^2 - 2x_n.$$

Using these functions values in the Newton's iterative formula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^3 - x_n^2 - 1}{3x_n^2 - 2x_n}.$$

Finding the first approximation of the root using the initial approximation $x_0 = 1.46$*, we get*

$$x_1 = x_0 - \frac{x_0^3 - x_0^2 - 1}{3x_0^2 - 2x_0} = 1.465601, \quad x_2 = x_1 - \frac{x_1^3 - x_1^2 - 1}{3x_1^2 - 2x_1} = 1.46557.$$

Hence $\alpha = 1.465$ *correct to three decimal places. Now we have*

$$g(x) = \frac{\lambda x + x^{-2} + 1}{\lambda + 1}, \quad g'(x) = \frac{\lambda - 2x^{-3}}{\lambda + 1}.$$

Hence for rapid convergence, $g'(\alpha) = g'(1.465) = \dfrac{\lambda - 2(1.465)^{-3}}{\lambda + 1} = 0$*, gives,* $\lambda = 0.636$. •

Example 2.57 *If* $x = \alpha$ *is a simple root of* $f(x) = 0$*, then show that the rate of convergence of the Newton's method is at least quadratic.*

Solution. *Consider the Newton's iteration function which is define as follows:*

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = g(x_n), \quad \text{or} \quad g(x) = x - \frac{f(x)}{f'(x)}.$$

Since α *is a simple root of nonlinear equation* $f(x) = 0$*, so,* $f(\alpha) = 0$ *but* $f'(\alpha) \neq 0$*. Thus taking derivative of* $g(x)$*, we get*

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

At $x = \alpha$*, we know that* $f(\alpha) = 0$ *and* $f'(\alpha) \neq 0$*, so we have*

$$g'(\alpha) = 1 - \frac{f'(\alpha)f'(\alpha) - f(\alpha)f''(\alpha)}{[f'(\alpha)]^2} = \frac{f(\alpha)f''(\alpha)}{[f'(\alpha)]^2} = 0.$$

Thus from Lemma 2.4, the rate of convergence of Newton's method is at least quadratic. •

Example 2.58 *If* $x = \alpha$ *is a double root of* $f(x) = 0$*, then show that the rate of convergence of the Newton's method is linear.*

Solution. *Consider the Newton's iteration function which is define as follows:*

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Since α *is a double root of nonlinear equation* $f(x) = 0$*, so,* $f(\alpha) = 0,\ f'(\alpha) = 0,\ f''(\alpha) \neq 0$*. Thus taking derivative of* $g(x)$*, we get*

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

At $x = \alpha$, we know that $f(\alpha) = 0$ and $f'(\alpha) = 0$, so we have indeterminate form $\left(\frac{0}{0}\right)$. Using L'Hôpital's rule we have

$$g'(x) = \frac{f'(x)f''(x) + f(x)f'''(x)}{2[f'(x)f''(x)]},$$

once again we get the indeterminate form. So again applying the L'Hôpital's rule, we obtain

$$g'(x) = \frac{(f''(x))^2 + 2f'(x)f'''(x) + f'(x)f^{(4)}}{2[(f''(x))^2 + f'(x)f'''(x)]}, \quad g'(\alpha) = \frac{(f''(\alpha))^2 + 0 + 0}{2[(f''(x))^2 + 0]} = \frac{1}{2} \neq 0.$$

Thus from Lemma 2.3, the rate of convergence of Newton's method is linear. •

In the following example we discuss the rate of convergence of Newton's method for the root of nonlinear equation with multiplicity m without using the L'Hôpital's rule.

Example 2.59 *If $x = \alpha$ is a root of multiplicity m of $f(x) = 0$, then show that the rate of convergence of the Newton's method is linear.*

Solution. *Consider the Newton's iteration function which is define as follows:*

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Since the function $f(x)$ has multiple root, so

$$f(x) = (x-\alpha)^m h(x) \quad \text{and} \quad f'(x) = m(x-\alpha)^{m-1}h(x) + (x-\alpha)^m h'(x).$$

Substituting the values of the $f(x)$ and $f'(x)$ in the above equation, we get

$$g(x) = x - \frac{(x-\alpha)^m h(x)}{(m(x-\alpha)^{m-1}h(x) + (x-\alpha)^m h'(x))} = x - \frac{(x-\alpha)h(x)}{(mh(x) + (x-\alpha)h'(x))}.$$

Then

$$\begin{aligned} g'(x) = 1 \quad - \quad & \{([mh(x) + (x-\alpha)][h(x) + (x-\alpha)h'(x)] - [(x-\alpha)h(x)] \\ & [mh'(x) + h'(x) + (x-\alpha)h''(x)])\}/([mh(x) + (x-\alpha)h'(x)]^2). \end{aligned}$$

At $x = \alpha$, and since $f(\alpha) = 0$, we have

$$g'(\alpha) = 1 - \frac{[mh(\alpha)][h(\alpha)]}{[mh(\alpha)]^2} = 1 - \frac{1}{m} \neq 0, \qquad \text{because} \quad m > 1.$$

Therefore, the Newton's method converges to a multiple zero from any sufficiently close approximation and the convergence is linear (by the Lemma 2.3), with ration $(1 - \frac{1}{m})$. In particular for a double root, the ration is $\frac{1}{2}$, which is comparable with the convergence of the bisection method. •

Example 2.60 *If $x = \alpha$ is a root of multiplicity m of nonlinear equation $f(x) = 0$, then show that the rate of convergence of the first modified Newton's method is at least quadratic.*

Solution. *The first modified Newton's iteration function is define as follows:*

$$g(x) = x - m\frac{f(x)}{f'(x)}. \tag{2.39}$$

Since the function $f(x)$ has multiple root, so

$$f(x) = (x-\alpha)^m h(x) \quad \text{and} \quad f'(x) = m(x-\alpha)^{m-1}h(x) + (x-\alpha)^m h'(x).$$

Substituting the values of the $f(x)$ and $f'(x)$ in (2.40), we get

$$g(x) = x - \frac{m(x-\alpha)^m h(x)}{(m(x-\alpha)^{m-1}h(x) + (x-\alpha)^m h'(x))} = x - \frac{m(x-\alpha)h(x)}{(mh(x) + (x-\alpha)h'(x))}.$$

Then

$$\begin{aligned} g'(x) = 1 \quad - \quad & m\{([mh(x) + (x-\alpha)][h(x) + (x-\alpha)h'(x)] - [(x-\alpha)h(x)] \\ & [mh'(x) + h'(x) + (x-\alpha)h''(x)])\}/([mh(x) + (x-\alpha)h'(x)]^2). \end{aligned}$$

At $x = \alpha$, and since $f(\alpha) = f'(\alpha) = 0$, we have $g'(\alpha) = 1 - \dfrac{[m^2h^2(\alpha)]}{[mh(\alpha)]^2} = 0$. Therefore, the modified Newton's method converges to a multiple root α and the rate of convergence is at least quadratically (by Lemma 2.4). •

Example 2.61 *If $x = \alpha$ is a root of multiplicity 5 of $f(x) = 0$, then show that the rate of convergence of modified Newton's method is at least quadratic.*

Solution. *The first modified Newton's iteration function is define as follows:*

$$g(x) = x - m\frac{f(x)}{f'(x)}. \tag{2.40}$$

Since the function $f(x)$ has multiple root of multiplicity 5, that is, $m = 5$, so

$$f(x) = (x-\alpha)^5 h(x) \quad \text{and} \quad f'(x) = 5(x-\alpha)^4 h(x) + (x-\alpha)^5 h'(x).$$

Substituting the values of the $f(x)$ and $f'(x)$ in (2.40), we get

$$g(x) = x - \frac{5(x-\alpha)^5 h(x)}{(5(x-\alpha)^4 h(x) + (x-\alpha)^5 h'(x))} = x - \frac{5(x-\alpha)h(x)}{(5h(x) + (x-\alpha)h'(x))}.$$

Then

$$\begin{aligned} g'(x) = 1 \quad - \quad & 5\{([5h(x) + (x-\alpha)][h(x) + (x-\alpha)h'(x)] - [(x-\alpha)h(x)] \\ & [5h'(x) + h'(x) + (x-\alpha)h''(x)])\}/([5h(x) + (x-\alpha)h'(x)]^2). \end{aligned}$$

At $x = \alpha$, we have $g'(\alpha) = 1 - \dfrac{[5^2h^2(\alpha)]}{[5h(\alpha)]^2} = 0$. Therefore, the modified Newton's method converges to a multiple root α of multiplicity 5 and the rate of convergence is at least quadratically (by Lemma 2.4). •

2.5 Systems of Nonlinear Equations

A system of nonlinear algebraic equations may arise when one is dealing with problems involving optimization and numerical integration (Gauss quadratures). Generally, the system of equations may not be of the polynomial variety. Therefore a system of n equations in n unknowns is called nonlinear if one or more of the equations in the systems is/are nonlinear.
The numerical methods we discussed so far have been concerned with finding a root of a nonlinear algebraic equation with one independent variable. We now consider methods for solving systems of nonlinear algebraic equations in which each equation is a function of a specified number of variables. Consider the system of two nonlinear equations with two variables

$$f_1(x, y) = 0 \qquad \text{and} \qquad f_2(x, y) = 0. \tag{2.41}$$

The problem can be stated as follows:
Given the continuous functions $f_1(x, y)$ and $f_2(x, y)$, find the values $x = \alpha$ and $y = \beta$ such that

$$f_1(\alpha, \beta) = 0 \quad \text{and} \quad f_2(\alpha, \beta) = 0. \tag{2.42}$$

The function $f_1(x, y)$ and $f_2(x, y)$ may be algebraic equations, transcendental or any nonlinear relationship between the input x and y and the output $f_1(x, y)$ and $f_2(x, y)$. The solutions to the equations (2.41) and (2.42) are the intersections of the $f_1(x, y) = f_2(x, y) = 0$. This problem is considerably more complicated then solution of a single nonlinear equation. The one-point iterative method discussed in the previous Section 2.5 for the solution of a single equation may be extended to system. So to solve the system of nonlinear equations we have many methods but we will use the Newton's method.

2.5.1 Newton's Method for Solving Nonlinear System

Consider the two nonlinear equations specified by the equation (2.41). Suppose that (x_n, y_n) is an approximation to a root (α, β), then by using the Taylor's theorem for functions of two variables for $f_1(x, y)$ and $f_2(x, y)$ expanding about (x_n, y_n), we have

$$\begin{aligned} f_1(x, y) &= f_1(x_n + (x - x_n), y_n + (y - y_n)) \\ &= f_1(x_n, y_n) + (x - x_n)\frac{\partial f_1(x_n, y_n)}{\partial x} + (y - y_n)\frac{\partial f_1(x_n, y_n)}{\partial y} + \cdots \end{aligned}$$

$$\begin{aligned} f_2(x, y) &= f_2(x_n + (x - x_n), y_n + (y - y_n)) \\ &= f_2(x_n, y_n) + (x - x_n)\frac{\partial f_2(x_n, y_n)}{\partial x} + (y - y_n)\frac{\partial f_2(x_n, y_n)}{\partial y} + \cdots \end{aligned}$$

Since $f_1(\alpha, \beta) = 0$ and $f_2(\alpha, \beta) = 0$, these equations, with $x = \alpha$ and $y = \beta$, give

$$\begin{aligned} 0 &= f_1(x_n, y_n) + (\alpha - x_n)\frac{\partial f_1(x_n, y_n)}{\partial x} + (\beta - y_n)\frac{\partial f_1(x_n, y_n)}{\partial y} + \cdots \\ 0 &= f_2(x_n, y_n) + (\alpha - x_n)\frac{\partial f_2(x_n, y_n)}{\partial x} + (\beta - y_n)\frac{\partial f_2(x_n, y_n)}{\partial y} + \cdots \end{aligned}$$

The Newton's method has a condition that initial approximation (x_n, y_n) should sufficiently close to exact root (α, β), therefore, the higher order terms may be neglected to obtain

$$\begin{aligned} 0 &\approx f_1(x_n, y_n) + (\alpha - x_n)\frac{\partial f_1(x_n, y_n)}{\partial x} + (\beta - y_n)\frac{\partial f_1(x_n, y_n)}{\partial y} \\ 0 &\approx f_2(x_n, y_n) + (\alpha - x_n)\frac{\partial f_2(x_n, y_n)}{\partial x} + (\beta - y_n)\frac{\partial f_2(x_n, y_n)}{\partial y} \end{aligned} \tag{2.43}$$

We see that this represents a system of two linear algebraic equations for α and β. Of course, since the higher order terms are omitted in the derivation of these equations, their solution (α, β) is no longer an exact root of (2.42) and (2.43). However, it will usually be a better approximation than (x_n, y_n), so replacing (α, β) by (x_{n+1}, y_{n+1}) in (2.42) and (2.43), gives the iterative scheme

$$\begin{aligned} 0 &= f_1(x_n, y_n) + (x_{n+1} - x_n)\frac{\partial f_1(x_n, y_n)}{\partial x} + (y_{n+1} - y_n)\frac{\partial f_1(x_n, y_n)}{\partial y} \\ 0 &= f_2(x_n, y_n) + (x_{n+1} - x_n)\frac{\partial f_2(x_n, y_n)}{\partial x} + (y_{n+1} - y_n)\frac{\partial f_2(x_n, y_n)}{\partial y} \end{aligned}$$

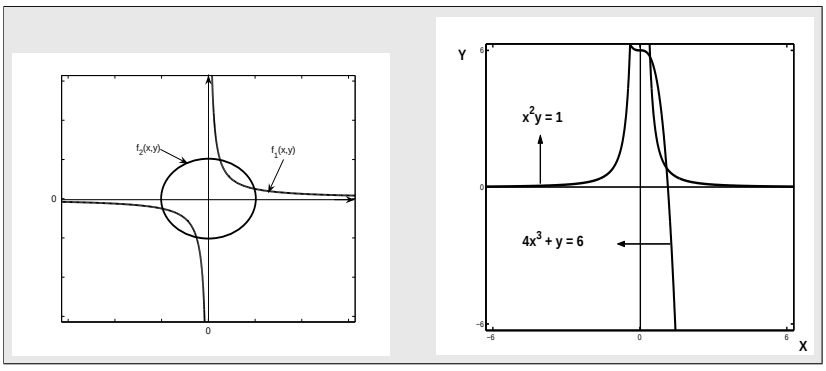
Then writing in the matrix form, we have

$$\begin{pmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{pmatrix} \begin{pmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \end{pmatrix} = - \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \tag{2.44}$$

where f_1, f_2 and their partial derivatives f_{1x}, f_{1y} are evaluated at (x_n, y_n). Hence

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{pmatrix}^{-1} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}. \tag{2.45}$$

We call the following matrix J a *Jacobian matrix*



Nonlinear system of equations in two variables.

$$J = \begin{pmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{pmatrix}. \tag{2.46}$$

Note that (2.44) can be written in the simplified form as follows

$$\begin{pmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{pmatrix} \begin{pmatrix} h \\ \\ k \end{pmatrix} = - \begin{pmatrix} f_1 \\ \\ f_2 \end{pmatrix},$$

where h and k can be evaluated as

$$h = \frac{\Big(- f_1 \dfrac{\partial f_2}{\partial y} + f_2 \dfrac{\partial f_1}{\partial y} \Big)}{\Big(\dfrac{\partial f_1}{\partial x} \dfrac{\partial f_2}{\partial y} - \dfrac{\partial f_1}{\partial y} \dfrac{\partial f_2}{\partial x} \Big)} \quad \text{and} \quad k = \frac{\Big(f_1 \dfrac{\partial f_2}{\partial x} - f_2 \dfrac{\partial f_1}{\partial x} \Big)}{\Big(\dfrac{\partial f_1}{\partial x} \dfrac{\partial f_2}{\partial y} - \dfrac{\partial f_1}{\partial y} \dfrac{\partial f_2}{\partial x} \Big)}, \tag{2.47}$$

where all functions are to be evaluated at (x, y). The Newton's method for a pair of equations in two unknowns is therefore

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} h \\ k \end{pmatrix}, \quad n = 0, 1, 2, \ldots \tag{2.48}$$

where (h, k) are given by (2.47) evaluated at (x_n, y_n).

At a starting approximation (x_0, y_0), the functions $f_1, f_{1x}, f_{1y}, f_2, f_{2x}$ and f_{2y} are evaluated. The linear equations are then solved for (x_1, y_1) and whole process is repeated until convergence is obtained. By comparison of the (2.14) and (2.45) shows that the above procedure is indeed an extension of the Newton's method in one variable, where division by f' generalized to pre-multiplication by J^{-1}.

Example 2.62 *Solve the following system of two equations using the Newton's method with given accuracy $\epsilon = 10^{-5}$ and assume $x_0 = 1.0$ and $y_0 = 0.5$ as starting values.*

$$\begin{aligned} 4x^3 + y &= 6 \\ x^2 y &= 1 \end{aligned}$$

Solution. *Obviously this system of nonlinear equations has an exact solution of $x = 1.088282$ and $y = 0.844340$, (see the Figure). Let us look how the Newton's method is used to approximate these roots. The first partial derivatives are as follows:*

$$\begin{aligned} f_1(x, y) &= 4x^3 + y - 6, \quad f_{1x} = 12x^2, \quad f_{1y} = 1, \\ f_2(x, y) &= x^2 y - 1, \qquad f_{2x} = 2xy, \quad f_{2y} = x^2. \end{aligned}$$

At the given initial approximation $x_0 = 1.0$ *and* $y_0 = 0.5$*, we get*

$$f_1(1.0, 0.5) = -1.5, \quad \frac{\partial f_1}{\partial x} = f_{1_x} = 12, \quad \frac{\partial f_1}{\partial y} = f_{1_y} = 1.0,$$

$$f_2(1.0, 0.5) = -0.5, \quad \frac{\partial f_1}{\partial x} = f_{2_x} = 1.0, \quad \frac{\partial f_2}{\partial y} = f_{2_y} = 1.0.$$

The Jacobian matrix J *and its inverse* J^{-1} *at the given initial approximation can be calculated as follows:*

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 12.0 & 1.0 \\ 1.0 & 1.0 \end{pmatrix} \quad \textit{and} \quad J^{-1} = \frac{1}{11.0} \begin{pmatrix} 1.0 & -1.0 \\ -1.0 & 12.0 \end{pmatrix}.$$

The Jacobian matrix can be find out by using MATLAB commands as follows:

```
>> syms x y
>> fun = [4 * x ^ 3+y - 6, x ^ 2*y - 1];
>> var = [x, y]; R = jacobian(f, var);
```

Substituting all these values in (2.46), we get the first approximation as follows:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.5 \end{pmatrix} - \frac{1}{11.0} \begin{pmatrix} 1.0 & -1.0 \\ -1.0 & 12.0 \end{pmatrix} \begin{pmatrix} -1.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 1.090909 \\ 0.909091 \end{pmatrix}.$$

The first and the further steps of the method are listed in Table 2.8. •

Table 2.8: Solution of a system of two nonlinear equations

n	x-approx. x_n	y-approx. y_n	1st. function $f_1(x_n, y_n)$	2nd. function $f_2(x_n, y_n)$
00	1.000000	0.500000	-1.50000	-0.500000
01	1.090909	0.909091	0.102178	0.081893
02	1.088264	0.844686	0.000091	0.000377
03	1.088282	0.844340	0.000001	0.000001

A typical iteration of this method for this pair of equations can be implemented in MATLAB command window using:

```
>> f1 = 4 * x0^ 3 + y0 - 6; f2 = x0^ 2 * y0 - 1;
>> f1x = 12 * x0^ 2; f1y = 1; f2x = 2 * x0 * y0; f2y = x0^ 2;
>> D = f1x * f2y - f1y * f2x;
>> h = (f2 * f1y - f1 * f2y)/D; k = (f1 * f2x - f2 * f1x)/D;
>> x0 = x0 + h; y0 = y0 + k;
```

Using the starting value $(1.0, 0.5)$, we found the possible approximations as shown in Table 2.8. •

We see that the values of both the functional are approaching zero as the number of iterations is increased. We got the desired approximations to the roots after 3 iterations with accuracy $\epsilon = 10^{-5}$. The Newton's method is fairly easy to implement for the case of two equations in two unknowns. We first need the function m-files for the equations and the partial derivatives. For the equations in the Example 2.62, we do the following:

```
function f = fn2(v)                    function J = dfn2(v)
%f and v are vector quantities          %Jacobian matrix forfn2.m
x = v(1); y = v(2);                    x = v(1); y = v(2);
f(1) = 4 * x.^ 2 + y - 6;              J(1,1) = 12 * x.^ 2; J(1,2) = 1;
f(2) = x.^ 2 * y - 1;                  J(2,1) = 2 * x * y; J(2,2) = x.^ 2;
```

Then the following MATLAB commands can be used to generate the solution of the Example 2.62:

```
>> s = newton2('fn2','dfn2',[1.0,0.5],1e-5)
```

The m-file Newton2.m will need both the function and its partial derivatives as well as starting vector and a tolerance. The following code can be used.

Program 2.7

MATLAB m-file for Newton's Method for a Nonlinear System

```
function sol=newton2(fn2,dfn2,x0,tol)
old=x0+1; while max(abs(x0-old))>tol; old=x0;
f = feval(fn2, old); f1 = f(1); f2 = f(2); J = feval(dfn2, old);
f1x = J(1,1); f1y = J(1,2); f2x = J(2,1); f2y = J(2,2);
D = f1x * f2y - f1y * f2x; h = (f2 * f1y - f1 * f2y)/D;
k = (f1 * f2x - f2 * f1x)/D; x0 = old + [h, k]; end; sol=x0;
```

Example 2.63 *Find the second approximation of the point of intersection of the equations* $x^3 + y = 1$ *and the ellipse* $-x + y^3 = -1$ *using Newton's method, starting with initial approximation* $(x_0, y_0)^T = (0.5, 0.5)^T$.

Solution. *We are given the nonlinear system*

$$\begin{array}{rcrcl} x^3 & + & y & = & 1 \\ -x & + & y^3 & = & -1 \end{array}$$

and it gives the functions and the first partial derivatives as follows:

$$\begin{array}{lcl} f_1(x,y) = x^3 + y - 1, & f_{1x} = 3x^2, & f_{1y} = 1, \\ f_1(x,y) = -x + y^3 + 1, & f_{2x} = -1, & f_{2y} = 3y^2. \end{array}$$

At the given initial approximation $x_0 = 0.5$ *and* $y_0 = 0.5$*, we get*

$$f_1(0.5, 0.5) = -0.3750, \quad \frac{\partial f_1}{\partial x} = f_{1_x} = 0.75, \quad \frac{\partial f_1}{\partial y} = f_{1_y} = 1,$$

$$f_2(0.5, 0.5) = 0.6250, \quad \frac{\partial f_1}{\partial x} = f_{2_x} = -1, \quad \frac{\partial f_2}{\partial y} = f_{2_y} = 0.75.$$

The Jacobian matrix J *and its inverse* J^{-1} *at the given initial approximation can be calculated as follows:*

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 0.75 & 1 \\ -1 & 0.75 \end{pmatrix} \quad \textit{and} \quad J^{-1} = \begin{pmatrix} 0.48 & -0.64 \\ 0.64 & 0.48 \end{pmatrix}.$$

Substituting all these values in (2.46), we get the first approximation as follows:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 0.48 & -0.64 \\ 0.64 & 0.48 \end{pmatrix} \begin{pmatrix} -0.3750 \\ 0.6250 \end{pmatrix} = \begin{pmatrix} 1.0800 \\ 0.4400 \end{pmatrix}.$$

Similarly, the second approximation can be obtained at first approximation $x_1 = 1.0800$ *and* $y_1 = 0.4400$ *as follows:*

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1.0800 \\ 0.4400 \end{pmatrix} - \begin{pmatrix} 3.4992 & 1 \\ -1 & 0.5805 \end{pmatrix} \begin{pmatrix} 0.6997 \\ 0.0052 \end{pmatrix} = \begin{pmatrix} 0.9477 \\ 0.2033 \end{pmatrix},$$

which is the required second approximation. Note that the exact point of intersection of these equations are $(1, 0)$. •

Using MATLAB built-in function **fsolve** to generate the solution of the Example 2.63:

```
>> f = @(x) [x(1)^3 + x(2) - 1; -x(1) + x(2)^3 + 1];
>> x = [0.5, 0.5];
>> x = fsolve(f, x)
```

and then reproduced above results using Newton's method, we do as:

```
>> f = @(x) [x(1)^3 + x(2) - 1; -x(1) + x(2)^3 + 1];
>> Df = @(x) [3 * x(1)^2, 1; -1, 3 * x(2)^2];
>> n = 2; x = [0.5, 0.5];
for i = 1 : n
Dx = -Df(x) f(x);
x = x + Dx;
f(x)
end
```

Example 2.64 *Consider the nonlinear system*

$$\begin{aligned} x^3 + 3y^2 &= a \\ x^2 + 2y &= -b \end{aligned}$$

Suppose that applying Newton's method to this system starting with the initial approximation $(x_0, y_0)^T = (1, -1)^T$ *gives the first approximation* $(x_1, y_1)^T = (2.5556, -3.0556)^T$. *Find the values of* a *and* b.

Solution. *Given*

$$\begin{aligned} f_1(x,y) &= x^3 + 3y^2 - a, \quad f_{1_x} = 3x^2, \quad f_{1_y} = 6y, \\ f_2(x,y) &= x^2 + 2y + b, \quad f_{2_x} = 2x, \quad f_{2_y} = 2. \end{aligned}$$

At the given initial approximation $x_0 = 1$ *and* $y_0 = -1$, *we have*

$$f_1(1,-1) = -17, \quad \frac{\partial f_1}{\partial x} = f_{1_x} = 3, \quad \frac{\partial f_1}{\partial y} = f_{1_y} = -6,$$

$$f_2(1,-1) = 1, \quad \frac{\partial f_1}{\partial x} = f_{2_x} = 2, \quad \frac{\partial f_2}{\partial y} = f_{2_y} = 2.$$

The Jacobian matrix J *at the given initial approximation can be calculated as*

$$J = \begin{pmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} 3 & -6 \\ 2 & 2 \end{pmatrix} \quad \textit{and} \quad J^{-1} = \frac{1}{18}\begin{pmatrix} 2 & 6 \\ -2 & 3 \end{pmatrix},$$

is the inverse of the Jacobian matrix. Since Newton's formula for the first approximation is

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} - J^{-1}\begin{pmatrix} f_1(x_0, y_0) \\ f_2(x_0, y_0) \end{pmatrix},$$

so using the given information, we get

$$\begin{pmatrix} 2.5556 \\ -3.0556 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} - \frac{1}{18}\begin{pmatrix} 2 & 6 \\ -2 & 3 \end{pmatrix}\begin{pmatrix} 4-a \\ -1+b \end{pmatrix}.$$

This gives us

$$28.0008 = -2(4-a) - 6(b-1) \quad \textit{and} \quad -37.0008 = 2(4-a) - 3(b-1).$$

Now solving for a *and* b, *we get* $a = 21$ *and* $b = 2$ *the required values.* •

For a large system of equations it is convenient to use vector notation. Consider the system

$$\mathbf{f}(\mathbf{x}) = \mathbf{0},$$

where $\mathbf{f} = (f_1, f_2, \ldots, f_n)^T$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. Denoting the *nth* iterate by $\mathbf{x}^{[n]} = (x_1^{[n]}, x_2^{[n]}, x_3^{[n]}, \ldots, x_n^{[n]})^T$, then the Newton's method is defined by

$$\mathbf{x}^{[n+1]} = \mathbf{x}^{[n]} - \left[J(\mathbf{x}^{[n]})\right]^{-1}\mathbf{f}(\mathbf{x}^{[n]}), \tag{2.49}$$

where the Jacobian matrix J is defined as

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}.$$

Since the iterative formula (2.49) involves the inverse of Jacobian J, in practice we do not attempt to find this explicitly. In stead of using the form of (2.49) we use the following form

$$J(\mathbf{x^{[n]}})\mathbf{Z^{[n]}} = -\mathbf{f}(\mathbf{x}^{[n]}), \quad \text{where} \quad \mathbf{Z}^{[n]} = \mathbf{x}^{[n+1]} - \mathbf{x}^{[n]}. \tag{2.50}$$

This represents a system of linear equations for $\mathbf{Z}^{[n]}$ and can be solved by any methods described in the next Chapter 3. Once $\mathbf{Z}^{[n]}$ has been found, the next iterate is calculated from

$$\mathbf{x}^{[n+1]} = \mathbf{Z}^{[n]} + \mathbf{x}^{[n]}. \tag{2.51}$$

There are two major disadvantages with method:

1. The method may not converges unless the initial approximation is a good one. Unfortunately, there are no general means by which an initial solution can be obtained. One can assume such values for which $\det(J) \neq 0$. This does not guarantee convergence but it does provide some guidance as to the appropriateness of one's initial approximation.

2. The method requires the user to provide the derivatives of each function with respect to each variable. Therefore one must evaluate the n functions and the n^2 derivatives at each iteration. So solving systems of nonlinear equations is a difficult task. For systems of nonlinear equations that have analytical partial derivatives, Newton's method can be used; otherwise, multi-dimensional minimization techniques should be used.

Procedure 2.5 (Newton's Method for Two Nonlinear Equations)

1. *Choose the initial guess for the roots of the system, so that the determinant of the Jacobian matrix is not zero.*

2. *Establish Tolerance* $\epsilon(> 0)$.

3. *Evaluate the Jacobian at initial approximations and then find inverse of Jacobian.*

4. *Compute new approximation to the roots by using iterative formula (2.51).*

5. *Check tolerance limit. If* $\|(x_n, y_n) - (x_{n-1}, y_{n-1})\| \leq \epsilon$, *for* $n \geq 0$, *then end; otherwise, go back to step 3, and repeat the process.*

2.6 Exercises

1. Use bisection method to find solutions accurate to within 10^{-4} on the interval $[-5, 5]$ of the following functions:
   (a) $f(x) = x^5 - 10x^3 - 4$, *(b)* $f(x) = 2x^2 + \ln(x) - 3$, *(c)* $f(x) = \ln(x) + 30e^{-x} - 3$.

2. The following equations have a root in the interval $[0, 1.6]$. Determine these with an error less than 10^{-4} using bisection method. (a) $2x - e^{-x} = 0$; (b) $e^{-3x} + 2x - 2 = 0$.

3. Estimate the number of iterations needed to achieve an approximation with accuracy 10^{-4} to the solution of $f(x) = x^3 + 4x^2 + 4x - 4$ lying in the interval $[0, 1]$ using bisection method.

4. Use the bisection method for $f(x) = x^3 - 3x + 1$ in $[1, 3]$ to find the first eight approximation to the root of the given equation. Compute an error estimate $|\alpha - x_8|$.

5. The cubic equation $x^3 - 3x - 20 = 0$ can be written as

   (a) $x = \dfrac{(x^3 - 20)}{3}$, *(b)* $x = \dfrac{3}{(x^3 - 3)}$, *(c)* $x = (3x + 20)^{1/3}$.
   Choose the form which satisfies the condition $|g'(x)| < 1$ on $[3, 4]$ and then find third approximation x_3 when $x_0 = 3.5$.

6. Find value of k such that the iterative scheme $x_{n+1} = \dfrac{x_n^2 - 4kx_n + 7}{4}$, $n \geq 0$ converges to 1. Also, find the rate of convergence of the iterative scheme.

7. Write the equation $x^2 - 6x + 5 = 0$ in the form $x = g(x)$, where $x \in [0, 2]$, so that the iteration $x_{n+1} = g(x_n)$ will converge to the root of the given equation for $x_0 \in [0, 2]$.

8. Which of the following iterations (a) $x_{n+1} = \dfrac{1}{4}\left(x_n^2 + \dfrac{6}{x_n}\right)$, *(b)* $x_{n+1} = \left(4 - \dfrac{6}{x_n^2}\right)$ is suitable to find a root of the equation $x^3 = 4x^2 - 6$ in the interval $[3, 4]$? Estimate the number of iterations required to achieve 10^{-3} accuracy, starting from $x_0 = 3$.

9. Let $f(x) = e^x + 3x^2$. Find Newton's formula $g(x_k)$. Start with $x_0 = 4, x_0 = -0.5$, find x_4.

10. Use Newton's formula for the reciprocal of square root of a number 15 and then find the 3rd approximation of number, with $x_0 = 0.05$.

11. Find Newton's formula for $f(x) = x^3 - 3x + 1$ in $[1, 3]$ to calculate x_3, if $x_0 = 1.5$. Also, find the rate of convergence of the method.

12. Find x_4 for $x^3 - 2x - 5 = 0$ by secant method using $x_0 = 2$ and $x_1 = 3$.

13. Find the root of multiplicity of the function $f(x) = (x - 1)^2 \ln(x)$ at $\alpha = 1$.

14. Show that iterative scheme $x_{n+1} = 1 + x_n - \dfrac{x_n^2}{2}$, $n \geq 0$ converges to $\sqrt{2}$. Find the rate of convergence of the sequence.

15. Solve the system $x + e^y = 68.1$ and $\sin x - y = -3.6$ using the Newton's method. Start with initial approximation $x_0 = 2.5$, $y_0 = 4$, compute the first three approximations.

Chapter 3

Systems of Linear Algebraic Equations

3.1 Introduction

When engineering systems are modeled, the mathematical description is frequently developed in terms of set of algebraic simultaneous equations. Sometimes these equations are non-linear and sometimes linear. In this chapter we discuss systems of simultaneous linear equations and describe the numerical methods for the approximate solutions of such systems. The solution of a system of simultaneous linear algebraic equations is probably one of the most important topics in the engineering computation. Problems involving simultaneous linear equations arise in the areas of elasticity, electric-circuit analysis, heat transfer, vibrations and so on. Also, the numerical integration of some types of ordinary and partial differential equations may be reduced to the solution of such a system of equations. It has been estimated, for example, that about 75% of all scientific problems require the solution of a system of linear equations at one stage or another. It is therefore important to be able to solve linear problems efficiently and accurately.

Definition 3.1 (Linear equation)

It is an equation in which the highest exponent in a variable term is no more than one. The graph of such equation is a straight line. •

A linear equation in two variables x_1 and x_2 is an equation that can be written in the form

$$a_1x_1 + a_2x_2 = b,$$

where a_1, a_2 and b are real numbers. Note that this is the equation of a straight line in the plane. For example, the equations

$$5x_1 + 2x_2 = 2, \qquad \frac{4}{5}x_1 + 2x_2 = 1, \qquad 2x_1 - 4x_2 = \pi,$$

are all linear equations in two variables.
A linear equation in n variables $x_1, x_2, \ldots, x_n$ is an equation that can be written as

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b,$$

where $a_1, a_2, \ldots, a_n$ are real numbers and called the *coefficients* of unknown variables $x_1, x_2, \ldots, x_n$ and the real number b, the right-hand side of equation, is called the *constant term* of the equation.

Definition 3.2 (System of Linear Equations)

A system of linear equations (or linear system) is simply a finite set of linear equations.

For example,

$$\begin{array}{rcrcl} 4x_1 & - & 2x_2 & = & 5 \\ 3x_1 & + & 2x_2 & = & 4 \end{array}$$

is the system of two equations in two variables x_1 and x_2, while

$$\begin{array}{rcrcrcrcl} 2x_1 & + & x_2 & - & 5x_3 & + & 2x_4 & = & 9 \\ 4x_1 & + & 3x_2 & + & 2x_3 & + & 4x_4 & = & 3 \\ x_1 & + & 2x_2 & + & 3x_3 & + & 2x_4 & = & 11 \end{array}$$

is the system of three equations in the four variables x_1, x_2, x_3 and x_4.
In order to write a general system of m linear equations in the n variables $x_1, \ldots, x_n$, we have

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \cdots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m \end{array} \tag{3.1}$$

or, in compact form the system (3.1) can be written

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \qquad i = 1, 2, \ldots, m. \tag{3.2}$$

For such system we seek all possible ordered sets of numbers $c_1, \ldots, c_n$ which satisfies all m equations when they are substituted for the variables $x_1, x_2, \ldots, x_n$. Any such set $\{c_1, c_2, \ldots, c_n\}$, is called a *solution* of the system of linear equations (3.1) or (3.2).

There are three possible types of linear systems arise in engineering problems and they are:

1. If there are more equations than unknown variables ($m > n$), then the system is usually called *overdetermined*. Typically, an overdetermined system has no solution. For example, the following system has no solution.

$$\begin{array}{rcrcl} 4x_1 & & & = & 8 \\ 3x_1 & + & 9x_2 & = & 13 \\ & & 3x_2 & = & 9 \end{array}$$

2. If there are more unknown variables than the number of the equations ($n > m$), then the system is usually called *underdetermined*. Typically, an underdetermined system has infinite number of solutions. For example, the following system has infinitely many solutions.

$$\begin{array}{rcrcrcl} x_1 & + & 5x_2 & & & = & 45 \\ & & 3x_2 & + & 4x_3 & = & 21 \end{array}$$

3. If there are same number of equations as the unknown variables ($m = n$), then the system is usually called *simultaneous* system. It has unique solution if the system satisfies the certain conditions (which we will discuss below). For example, the system

$$\begin{array}{rcrcrcr} 2x_1 & + & 4x_2 & + & x_3 & = & -11 \\ -x_1 & + & 3x_2 & - & 2x_3 & = & -16 \\ 2x_1 & - & 3x_2 & + & 5x_3 & = & 21 \end{array}$$

has unique solution $x_1 = 2, x_2 = -4$ and $x_3 = 1$.

Most engineering problems fall into this category. In this chapter we will solve the simultaneous linear systems using many numerical methods.

A simultaneous system of linear equations is said to be *linear independent* if no equation in the system can be expressed as a linear combination of the others. Under these circumstances a unique solution exists. For example, the following system of linear equations

$$\begin{array}{rcrcrcr} 2x_1 & + & x_2 & - & x_3 & = & 1 \\ x_1 & - & 2x_2 & + & 3x_3 & = & 4 \\ x_1 & + & x_2 & & & = & 1 \end{array}$$

are linear independent and therefore, has unique solution $x_1 = 1, x_2 = 0$ and $x_3 = 1$. However, the system

$$\begin{array}{rcrcrcr} 5x_1 & + & x_2 & + & x_3 & = & 4 \\ 3x_1 & - & x_2 & + & x_3 & = & -2 \\ x_1 & + & x_2 & & & = & 3 \end{array}$$

does not have a unique solution since the equations are not linear independent; the first equation is equal to the second equation plus twice the third equation.

Theorem 3.1 (Solution of a Linear System)

Every system of linear equations has either no solution, exactly one solution, or infinitely many solutions. •

For example, in the case of a system of two equations in two variables, we can have these three possibilities for the solutions of the linear system. Firstly, the two lines (since the graph of linear equation is straight line) may be parallel and distinct, in this case there is no solution to the system because the two lines do not intersect each other at any point. For example, consider the following system

$$\begin{array}{rcrcr} x_1 & + & x_2 & = & 1 \\ 2x_1 & + & 2x_2 & = & 3 \end{array}$$

From the graphs (see Figure 3.1(a)) of the given two equations show that lines are parallel so, the given system has no solution. It can be proved algebraically simply by multiplying the first equation of the system by 2, to get a system of the form

$$\begin{array}{rcl} 2x_1 + 2x_2 & = & 2 \\ 2x_1 + 2x_2 & = & 3 \end{array}$$

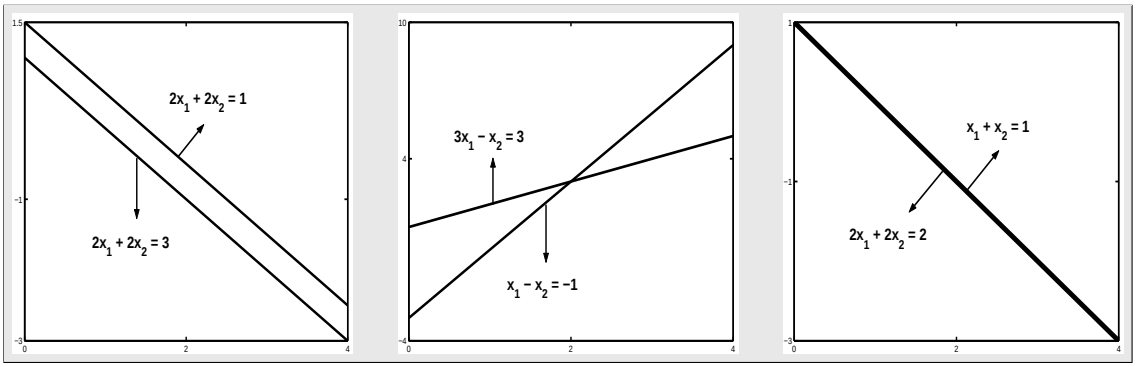


Figure 3.1: Three possible solutions of simultaneous linear systems.

which is not possible.
Secondly, the two lines may not be parallel, and they meet exactly one point, so in this case the system has exactly one solution. For example, consider the following system

$$\begin{array}{rcrcr} x_1 & - & x_2 & = & -1 \\ 3x_1 & - & x_2 & = & 3 \end{array}$$

From the graphs (see Figure 3.1(b)) of these two equations, we can see that the lines intersect in exactly one point, namely, the (2,3), and so the system has exactly one solution, $x_1 = 2, \quad x_2 = 3$. To show algebraically, if we substitute $x_2 = x_1 + 1$ in the second equation, we have $3x_1 - x_1 - 1 = 3$, or $x_1 = 2$ and using this value of x_1 in $x_2 = x_1 + 1$, gives $x_2 = 3$.
Finally, the two lines may actually be the same line, and so in this case, every point on the lines gives a solution to the system, and therefore, there are infinitely many solutions. For example, consider the following system

$$\begin{array}{rcrcr} x_1 & + & x_2 & = & 1 \\ 2x_1 & + & 2x_2 & = & 2 \end{array}$$

Here, both equations have same line for their graph, see Figure 3.1(c). So this system has infinitely many solutions because any point on this line gives a solution to this system. Since any solution of first equation is also solution of the second equation. For example, if we set $x_2 = x_1 - 1$ and choose $x_1 = 0, x_2 = 1$ and $x_1 = 1, x_2 = 0$, and so on. •
Note that a system of equations with no solution is said to be *inconsistent system* and if it has at least one solution, it is said to be *consistent system.*

3.1.1 Linear System in Matrix Notation

To write the general simultaneous system of n linear equations in the n unknown variables $x_1, x_2, \ldots, x_n$, is

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array} \tag{3.3}$$

The system of linear equations (3.3) can be written as the single matrix equation

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \tag{3.4}$$

If we compute the product of the two matrices on the left-hand side of (3.4), we have

$$\begin{pmatrix} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \tag{3.5}$$

But two matrices are equal if and only if their corresponding elements are equal. Hence the single matrix equation (3.4) is equivalent to the system of the linear (3.3). If we define

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix},$$

the coefficient matrix, the column matrix of unknowns, and the column matrix of the constants, respectively, then the system (3.3) can be written very compactly as

$$A\mathbf{x} = \mathbf{b}, \tag{3.6}$$

which is called the *matrix form* of the system of linear equations (3.3). The column matrices $\mathbf{x}$ and $\mathbf{b}$ are called *vectors*. If right-hand sides of the equal signs of (3.6) are not zero, then the linear system (3.6) is called a *nonhomogeneous system*, and we will find that all the equations must be independent to obtain unique solution.

3.1.1.1 Augmented Matrix

One can think of an augmented matrix as being a way to organize the important parts of a system of linear equations. These "important parts" would be the coefficients (numbers in front of the variables) and the constants (numbers not associated with variables).
If the constants $\mathbf{b}$ of (3.6) are added to the coefficient matrix A as a column of elements in the position shown below

$$[A|\mathbf{b}] = \left(\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right), \tag{3.7}$$

then the matrix $[A|\mathbf{b}]$ is called *augmented matrix* of the system (3.6). In many instances, it may be found convenient to operate on the augmented matrix instead of manipulating the equations. It is

customary to put a bar between the last two columns of the augmented matrix remind us where the last column come from. However, the bar is not absolutely necessary. The augmented matrix of a linear system will play key roles in our methods of solving linear systems.
Let's take a look at an example. Here is the system of equations

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & + & 3x_3 & = & 10 \\ 4x_1 & + & 5x_2 & + & 6x_3 & = & 11 \\ 7x_1 & + & 8x_2 & + & 9x_3 & = & 12 \end{array}$$

Here is the augmented matrix for this system

$$[A|\mathbf{b}] = \left(\begin{array}{ccc:c} 1 & 2 & 3 & 10 \\ 4 & 5 & 6 & 11 \\ 7 & 8 & 9 & 12 \end{array}\right).$$

The first row consists of all the constants from the first equation with the coefficient of the x_1 in the first column, the coefficient of the x_2 in the second column, the coefficient of the x_3 in the third column and the constant in the final column. The second row is the constants from the second equation with the same placement and likewise for the third row. The dashed line represents where the equal sign was in the original system of equations and is not always included.
Using MATLAB commands we can define augmented matrix as follows:

```
>> A = [1 2 3; 4 5 6; 7 8 9]; b = [10; 11; 12]; Aug = [A b]
```

Homogeneous Linear System

If all of the constant terms $b_1, b_2, \ldots, b_n$ on the right-hand sides of the equal signs of the linear system (3.6) are zero, then the system (3.6) is called a *homogeneous system*, and it can be written in more compact form as

$$A\mathbf{x} = \mathbf{0}. \tag{3.8}$$

It can be seen by inspection of the homogeneous system (3.8) that one of its solution is $\mathbf{x} = \mathbf{0}$, such solution, in which all of the unknowns are zero, is called the *trivial solution* or *zero solution*. For the general nonhomogeneous linear system there are three possibilities: no solution, one solution, or an infinitely many solutions. For the general homogeneous system, there are only two possibilities: either the zero solution is the only solution or there are an infinitely many solutions (called non-trivial solutions). Of course, it is usually non-trivial solutions that are of interest in physical problems. A non-trivial solution to the homogeneous system can occurs with certain conditions on the coefficient matrix A.

3.2 Properties of Matrices and Determinant

To discuss the solution of the linear systems, it will be necessary to introduce the basic algebraic properties of matrices which make it possible to describe the linear systems in a concise way that makes solving a system of n linear equations as easy as possible.

3.2.1 Introduction of Matrices

A matrix can be described as a rectangular array of elements that can be represented as follows:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}. \tag{3.9}$$

The number $a_{11}, a_{12}, \ldots, a_{mn}$ that make up the array are called the elements of the matrix. The first subscript for the element denotes the row and the second denotes the column in which the element appear. The elements of a matrix may take many forms. They could be all numbers (real or complex), or variables, or functions, or integrals, or derivatives, or even matrices themselves.
The *order* or *size* of a matrix is specified by the number of rows (m) and column (n); thus the matrix A in (3.9) is of order m by n, usually written as $m \times n$.
A vector can be considered as a special case of a matrix having only one row or one column. A row vector containing n elements is a $1 \times n$ matrix, called a (*row matrix*), and a column vector of n elements is an $n \times 1$ matrix, called a (*column matrix*). A matrix of order 1×1 is called a *scalar*.

Definition 3.3 (Matrix Equality)

Two matrices $A = (a_{ij})$ and $B = (b_{ij})$ are equal if they are the same size and the corresponding elements in A and B are equal, that is

$$A = B \quad \textit{if and only if} \quad a_{ij} = b_{ij},$$

for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. For example, the following matrices

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 1 & 3 & 2 \\ 2 & 4 & 3 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} 1 & -1 & z \\ 1 & 3 & 2 \\ x & y & w \end{pmatrix},$$

are equal if and only if $x = 2, y = 4, z = 2$ and $w = 3$. •

Definition 3.4 (Addition of Matrices)

Let $A = (a_{ij})$ and $B = (b_{ij})$ are both $m \times n$ matrices, then the sum $A + B$ of two matrices of same size is a new matrix $C = (c_{ij})$ each of whose elements is the sum of the two corresponding elements in the original matrices, that is

$$c_{ij} = a_{ij} + b_{ij}, \quad \textit{for } i = 1, 2, \ldots, m \quad \textit{and} \quad j = 1, 2, \ldots, n.$$

For example, let

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix},$$

then

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 3 \\ 8 & 6 \end{pmatrix} = C.$$

•

Using MATLAB commands adding two matrices A and B of same size, results in the answer C another matrix of same size, are:

```
>> A = [1 2;3 4]; B = [4 1;5 2]; C = A + B
```

Definition 3.5 (Difference of Matrices)

Let A and B are $m \times n$ matrices, we write $A + (-1)B$ as $A - B$ and call this the difference of two matrices of same size is a new matrix C each of whose elements is the difference of the two corresponding elements in the original matrices. For example, let

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix}.$$

Then

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} - \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix} = \begin{pmatrix} -3 & 1 \\ -2 & 2 \end{pmatrix} = C.$$

Note that $(-1)B = -B$ is obtained by multiplying each entries of matrix B by (-1), called the scalar multiple of matrix B by -1. The matrix $-B$ is called negative of the matrix B. •

Definition 3.6 (Multiplication of Matrices)

The multiplication of two matrices is defined only when the number of columns in the first matrix is equal to the number of rows in the second. If an $m \times n$ matrix A is multiplied by an $n \times p$ matrix B, then the product matrix C is an $m \times p$ matrix where each term is defined by

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj},$$

for each $i = 1, 2, \ldots, m$, and $j = 1, 2, \ldots, p$. For example, let

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix}.$$

Then

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 4 & 1 \\ 5 & 2 \end{pmatrix} = \begin{pmatrix} 4+10 & 1+4 \\ 12+20 & 3+8 \end{pmatrix} = \begin{pmatrix} 14 & 5 \\ 32 & 11 \end{pmatrix} = C.$$

Note that even AB is defined, the product BA may not be defined. Moreover, a simple multiplication of two square matrices of same size will show that even BA is defined, it need not equal to AB, that is, they do not commute. For example, if

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix},$$

then

$$AB = \begin{pmatrix} 2 & 3 \\ -2 & 2 \end{pmatrix} \quad \textit{while} \quad BA = \begin{pmatrix} 1 & 7 \\ -1 & 3 \end{pmatrix}.$$

Thus $AB \neq BA$. •

Using MATLAB commands matrix multiplication has the standard meaning as well. Multiplying two matrices A and B of size $m \times p$ and $p \times n$ respectively, results in the answer C another matrix of size $m \times n$, are:

$$>> A = [1\ 2; -1\ 3];\ B = [2\ 1; 0\ 1];\ C = A * B$$

3.2.2 Some Special Matrix Forms

There are many special types of matrices that are encountered frequently in engineering analysis. We discuss some of the them in the following.

Definition 3.7 (Square Matrix)

A matrix A which has the same number of rows m and columns n ,that is, $m = n$, defined as

$$A = (a_{ij}), \qquad for \quad i = 1, 2, \ldots, n \quad and \quad j = 1, 2, \ldots, n,$$

is called a square matrix. For example, the following matrices

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix} \quad and \quad B = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 5 \end{pmatrix},$$

are square matrices because both have the same numbers of rows and columns. •

Definition 3.8 (Null Matrix)

It is a matrix in which all elements are zero, that is

$$A = (a_{ij}) = \mathbf{0}, \qquad for \quad i = 1, 2, \ldots, n \quad and \quad j = 1, 2, \ldots, n.$$

It is also called zero matrix. It may be either rectangular or square. For example, the following matrices

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad and \quad B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

are the zero matrices. •

Definition 3.9 (Identity Matrix)

It is a square matrix in which the main diagonal elements are equal to 1, and is defined as follows

$$\mathbf{I} = (a_{ij}) = \begin{cases} a_{ij} = 0, & if \quad i \neq j, \\ a_{ij} = 1, & if \quad i = j. \end{cases}$$

An example of 4×4 *identity matrix may be written as*

$$\mathbf{I}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The Identity matrix (also called unit matrix) serves somewhat the same purpose in matrix algebra as does the number one (unity) in scalar algebra. It is called the identity matrix because multiplication of a matrix by it will result in a same matrix. For a square matrix A *of order* n*, it can be seen that*

$$\mathbf{I}_n A = A\mathbf{I}_n = A.$$

Similarly, for rectangular matrix B *of order* $m \times n$*, we have*

$$\mathbf{I}_m B = B\mathbf{I}_n = B.$$

The multiplication of an identity matrix by itself results in a same identity matrix. •

In MATLAB identity matrices are created with **eye** function, which can take either one or two input arguments.

$>> I = eye(n);\ I = eye(m, n)$

Definition 3.10 (Transpose Matrix)

The transpose of a matrix A*, which is a new matrix formed by interchanging the rows and columns of the original matrix. If the original matrix* A *is of order* $m \times n$*, then the transpose matrix, as* A^T*, will be of order* $n \times m$*, that is*

$$If \quad A = (a_{ij}), \quad for \quad i = 1, 2, \ldots, m \quad and \quad j = 1, 2, \ldots, n,$$

then

$$A^T = (a_{ji}), \quad for \quad i = 1, 2, \ldots, n \quad and \quad j = 1, 2, \ldots, m.$$

•

The transpose of a matrix A can be found by using MATLAB command as follows:

$>> A = [1\ \ 2\ \ 3;\ \ 4\ 5\ \ 6;\ \ 7\ \ 8\ \ 9]; B = A'$

It is to be noted that

1. $(A^T)^T = A$
2. $(A_1 + A_2)^T = A_1^T + A_2^T$
3. $(A_1 A_2)^T = A_2^T A_1^T$
4. $(\alpha A)^T = \alpha A^T, \quad \alpha$ is a scalar.

Definition 3.11 (Inverse Matrix)

An $n \times n$ matrix A has an inverse or is invertible if there exists an $n \times n$ matrix B such that

$$AB = BA = I_n.$$

Then the matrix B is called the inverse of A and is denoted by A^{-1}. For example, let

$$A = \begin{pmatrix} 2 & 3 \\ 2 & 2 \end{pmatrix} \quad \textit{and} \quad B = A^{-1} = \begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix}.$$

Then we have

$$AB = BA = \mathbf{I}_2,$$

which means that B is an inverse of A. The invertible matrix is also called, nonsingular matrix. •

To find the inverse of the square matrix A using MATLAB commands we do as follows:

$>> A = [2\ 3; 2\ 2];\ Ainv = INVMAT(A)$ *or* $Ainv = inv(A)$

Program 3.1
MATLAB m-file for finding inverse of a matrix
function [Ainv]=INVMAT(A)
[n,n]=size(A); I=zeros(n,n); for i=1:n; I(i,i)=1; end; m(1:n,1:n)=A; $m(1:n, n+1:2*n) = I$;
for i=1:n; $m(i, 1:2*n) = m(i, 1:2*n)/m(i,i)$; for k=1:n; if i˜ =k
$m(k, 1:2{*}n) = m(k, 1:2{*}n) - m(k,i) {*} m(i, 1:2{*}n)$; end;end;end; $invrs = m(1:n, n{+}1:2{*}n)$;

MATLAB built-in function **inv(A)** can be also use to calculate the inverse of a square matrix A if A is invertible. If the matrix A is not invertible, then the matrix A is called *singular*. There are some well-known properties of the invertible matrix which are define as follows.

Theorem 3.2 *If the matrix A is invertible, then*

1. *It has exactly one inverse. If B and C are the inverses of A, then $B = C$.*
2. *Its inverse matrix A^{-1} is also invertible and $(A^{-1})^{-1} = A$.*
3. *Its product with another invertible matrix is invertible, and the inverse of the product is the product of the inverses in the reverse order. If A and B are invertible matrices of the same size, then AB is invertible and $(AB)^{-1} = B^{-1}A^{-1}$.*
4. *Its transpose matrix A^T is invertible and $(A^T)^{-1} = (A^{-1})^T$.*
5. *The kA for any non-zero k is invertible, that is, $(kA)^{-1} = \frac{1}{k}A^{-1}$.*
6. *The A^k for any k is also invertible, that is, $(A^k)^{-1} = (A^{-1})^k$.*
7. *Its size 1×1 is invertible when it is nonzero. If $A = (a)$, then $A^{-1} = (\frac{1}{a})$.*

8. The formula for A^{-1} when $n = 2$ is

$$A^{-1} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix},$$

provided that $a_{11}a_{22} - a_{12}a_{21} \neq 0$. •

Definition 3.12 (Diagonal Matrix)

It is a square matrix having all elements equal to zero except those on main diagonal, that is

$$A = (a_{ij}) = \begin{cases} a_{ij} = 0, & \text{if } i \neq j, \\ a_{ij} \neq 0, & \text{if } i = j. \end{cases}$$

Note that all diagonal matrices are invertible if all diagonal entries are nonzero. •

The MATLAB **diag** function is used to either create a diagonal matrix from a vector or it extract the diagonal entries of a matrix. If the input argument of the **diag** function is a vector, MATLAB uses the vector to create a diagonal matrix:

```
>> x = [2, 2, 2]; A = diag(x)
>> B = [2 -4 1; 6 10 -3; 0 5 8]; M = diag(B)
```

The matrix A is called the *scalar* matrix because it has all the elements on the main diagonal equal to the same scalars 2. Multiplication of a square matrix and a scalar matrix is commutative, and the product is also a diagonal matrix.

Definition 3.13 (Upper-Triangular Matrix)

It is a square matrix which has zero elements below and to the left of the main diagonal. The diagonal as well as the above diagonal elements can take on any value, that is

$$U = (u_{ij}), \quad \text{where} \quad u_{ij} = 0, \quad \text{if} \quad i > j.$$

An example of such a matrix is

$$U = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{pmatrix}.$$

The upper-triangular matrix is called upper-unit-triangular matrix if the diagonal elements are equal to one. This type of matrix is used in solving linear algebraic equations by LU decomposition with Crout's method. Also, if the main diagonal elements of the upper-triangular matrix are zero, then

$$A = \begin{pmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{pmatrix},$$

is called the strictly upper-triangular matrix. This type of matrix will be used in solving linear systems by iterative methods. •

Using MATLAB command **triu**(A) we can create an upper triangular matrix from a matrix A as

```
>> A = [1 2 3; 4 5 6; 7 8 9]; U = triu(A)
```

Also we can create strictly upper-triangular matrix, that is, an upper-triangular matrix with zero diagonal, from a given matrix A by using MATLAB built-in function **triu**(A,I) as follows:

```
>> A = [1 2 3; 4 5 6; 7 8 9]; U = triu(A, I)
```

Definition 3.14 (Lower-Triangular Matrix)

It is a square matrix which has zero elements above and to the right of the main diagonal and the rest of the elements can take on any value, that is

$$L = (l_{ij}), \quad where \quad l_{ij} = 0, \quad if \quad i < j.$$

An example of such a matrix is

$$L = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 1 & 0 \\ 4 & 5 & 3 \end{pmatrix}.$$

The lower-triangular matrix is called lower-unit-triangular matrix if the diagonal elements are equal to one. This type of matrix is used in solving linear algebraic equations by LU Decomposition with Doolittle's method. Also, if the main diagonal elements of the lower-triangular matrix are zero, then the matrix

$$A = \begin{pmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{pmatrix},$$

is called the strictly lower-triangular matrix. We will use this type of matrix in solving the linear systems by using iterative methods. •

In similar way like upper-triangular matrices we can create lower-triangular matrix and strictly lower-triangular matrix from a given matrix A by using MATLAB built-in functions **tril**(A) and **tril**(A,I) respectively.

Note that all the triangular matrices (upper or lower) with nonzero diagonal entries are invertible.

Definition 3.15 (Symmetric Matrix)

A symmetric matrix is one in which the elements a_{ij} of a matrix A, in the ith row and jth column equal to the element a_{ji} in the jth row and ith column which means that

$$A^T = A, \quad that\ is \quad a_{ij} = a_{ji}, \quad for \quad i \neq j.$$

Note that any diagonal matrix, including the identity, is symmetric. A lower- or upper-triangular matrix is symmetric if and only if it is, in fact , a diagonal matrix.
One way to generate a symmetric matrix is to multiply a matrix by its transpose, since $A^T A$ is

symmetric for any A. To generate a symmetric matrix using MATLAB commands we do as

```
>> A = [1:4; 5:8; 9:12]; B = A' * A; C = A * A
```

Example 3.1 *Find all the values of a, b and c for which the following matrix is symmetric:*

$$A = \begin{pmatrix} 4 & a+b+c & 0 \\ -1 & 3 & b-c \\ -a+2b-2c & 1 & b-2c \end{pmatrix}.$$

Solution. *If the given matrix is symmetric, then $A = A^T$, that is*

$$A = \begin{pmatrix} 4 & a+b+c & 0 \\ -1 & 3 & b-c \\ -a+2b-2c & 1 & b-2c \end{pmatrix} = \begin{pmatrix} 4 & -1 & -a+2b-2c \\ a+b+c & 3 & 1 \\ 0 & b-c & b-2c \end{pmatrix} = A^T,$$

which implies that

$$0 = -a+2b-2c, \qquad -1 = a+b+c, \qquad 1 = b-c.$$

Solving above system, we get, $a = 2, \quad b = -1, \quad c = -2$ and using these values, we have the given matrix of the form

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

•

Theorem 3.3 *If A and B are symmetric matrices with same size, and if k is any scalar, then*

1. *A^T is also symmetric.*
2. *$A + B$ and $A - B$ are symmetric.*
3. *kA is also symmetric.*

Note that product of symmetric matrices is not symmetric in general but the product is symmetric if and only if the matrices commute. Also, note that if A is a square matrix, then the matrices A, AA^T and A^TA are either all nonsingular or all singular. •

If for a matrix A, the $a_{ij} = -a_{ji}$ for $i \neq j$ and the main diagonal elements are not all zero, then the matrix A is called skew matrix. If all the elements on the main diagonal of a skew matrix are zero, then the matrix is called skew symmetric, that is

$$A = -A^T, \quad \text{with} \quad a_{ij} = -a_{ji}, \quad \text{for} \quad i \neq j \quad \text{and} \quad a_{ii} = 0.$$

Any square matrix may be split into the sum of a symmetric and a skew symmetric matrix. Thus

$$A = \frac{1}{2}(A + A^T) + \frac{1}{2}(A - A^T),$$

where $\frac{1}{2}(A + A^T)$ *is symmetric matrix and* $\frac{1}{2}(A - A^T)$ *is skew symmetric matrix. The following matrices*

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{pmatrix}, \quad \begin{pmatrix} 1 & 2 & 3 \\ -2 & 4 & -5 \\ -3 & 5 & 6 \end{pmatrix}, \quad \begin{pmatrix} 0 & 2 & 3 \\ -2 & 0 & 5 \\ -3 & 5 & 0 \end{pmatrix},$$

are the examples of symmetric, skew and skew symmetric matrices respectively. •

Definition 3.16 (Band Matrix)

A $n \times n$ *square matrix* A *is called a band matrix if there exists positive integers* p *and* q*, with* $1 < p$ *and* $q < n$ *such that*

$$a_{ij} = 0 \quad for \quad p \leq j - i \quad or \quad q \leq i - j.$$

The number p *describes the number of diagonals above, and including, the main diagonal on which nonzero entries may lie. The number* q *describes the number of diagonals below, and including, the main diagonal on which nonzero entries may lie. The number* $p + q - 1$ *is called the bandwidth of the matrix* A*, which tells us how many of the diagonals can contain nonzero entries. For example, the following matrix*

$$A = \begin{pmatrix} 1 & 2 & 3 & \\ 2 & 3 & 4 & 5 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 7 & 8 \end{pmatrix},$$

is banded with $p = 3$ *and* $q = 2$*, and so the bandwidth is equal to* 4*. An important property of the band matrix is called the tridiagonal matrix, in this case* $p = q = 2$*, that is, all nonzero elements lie either on or directly above or below the main diagonal. For such type of matrix, the Gaussian elimination is particular simpler. In general, the nonzero elements of a tridiagonal matrix lie in three bands: the superdiagonal, diagonal and subdiagonal. For example, the following matrix*

$$A = \begin{pmatrix} 1 & 2 & & & & \\ 2 & 3 & 1 & & & \\ & 3 & 2 & 1 & & \\ & 2 & 4 & 3 & & \\ & & 1 & 2 & 3 & \\ & & & 1 & 6 & 4 \\ & & & & 3 & 4 \end{pmatrix},$$

is a tridiagonal matrix. A matrix which is predominantly zero is called a sparse matrix. A band matrix or a tridiagonal matrix is a sparse matrix but the nonzero elements of a sparse matrix are not necessarily near the diagonal. •

3.2.3 The Determinant of Matrix

The determinant is a certain kind of a function that associates a real number with a square matrix. We will denote the determinant of a square matrix A by $\det(A)$ or $|A|$.

Definition 3.17 (Determinant of Matrix)

Let $A = (a_{ij})$ be an $n \times n$ square matrix then a determinant of A is given by:

1. $\det(A) = a_{11}$, *if* $n = 1$.
2. $\det(A) = a_{11}a_{22} - a_{12}a_{21}$, *if* $n = 2$. •

For example, if

$$A = \begin{pmatrix} 4 & 2 \\ -3 & 7 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 6 & 3 \\ 2 & 5 \end{pmatrix},$$

then

$$\det(A) = (4)(7) - (-3)(2) = 34 \quad \text{and} \quad \det(B) = (6)(5) - (3)(2) = 24.$$

Notice that the determinant of a 2×2 matrix is given by the difference of the products of the two diagonals of a matrix. The determinant of a 3×3 matrix is defined in terms of determinants of 2×2 matrices and the determinant of a 4×4 matrix is defined in terms of determinants of 3×3 matrices and so on.
MATLAB function **det**(A) calculated the determinant of the square matrix A as:

```
>> A = [2 2;6 7]; B = det(A)
```

Other way to find the determinants of only 2×2 and 3×3 matrices can be found easily and quickly using diagonals (or direct evaluation). For 2×2 matrix, the determinant can be obtained by forming the product of the entries on the line from left to right and subtracting from this number the product of the entries on the line from right to left. For a matrix of size 3×3, the diagonals of an array consisting of the matrix with the two first columns added to the right are used. Then the determinant can be obtained by forming the sum of the products of the entries on the lines from left to right, and subtract from this number the products of the entries on the lines from right to left, as shown in Figure 3.2.
Thus for 2×2 matrix

$$|A| = a_{11}a_{22} - a_{12}a_{21},$$

and for 3×3 matrix

$$|A| = \underbrace{a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}}_{\text{(diagonal products from left to right)}} - \underbrace{a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}}_{\text{(diagonal products from right to left)}}$$

For example, the determinant of 2×2 matrix can be computed as

$$|A| = \begin{vmatrix} 12 & 5 \\ -7 & 6 \end{vmatrix} = (12)(6) - (5)(-7) = 72 + 35 = 107,$$

and the determinant of 3×3 matrix can be obtained as

$$\begin{aligned} |A| = \begin{vmatrix} 4 & 5 & 6 \\ -3 & 8 & 2 \\ 4 & 9 & 7 \end{vmatrix} &= [(4)(8)(7) + (5)(2)(4) + (6)(-3)(9)] \\ &- [(6)(8)(4) + (4)(2)(9) + (5)(-3)(7)] = 102 - 159 = -57. \end{aligned}$$

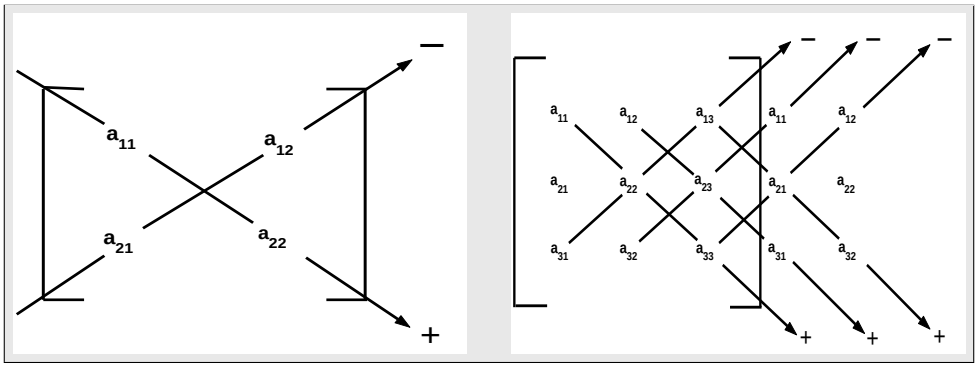


Figure 3.2: Direct evaluation of 2×2 and 3×3 matrices determinants.

For finding the determinants of the higher-order matrices, we will define the following concepts of *minor* and *cofactor* of the matrices.

Definition 3.18 *(***Minors of a Matrix)**

The minor M_{ij} of all elements a_{ij} of a matrix A of order $n \times n$ as the determinant of the sub-matrix of order $(n-1) \times (n-1)$ obtained from A by deleting the ith row and jth column (also called ijth minor of A). For example, let

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 5 & 3 & 2 \\ 4 & -2 & 4 \end{pmatrix},$$

then, the minor M_{11} will be obtained by deleting the first row and the first column of the given matrix A, that is,

$$M_{11} = \begin{vmatrix} 3 & 2 \\ -2 & 4 \end{vmatrix} = (3)(4) - (-2)(2) = 12 + 4 = 16.$$

Similarly, we can find the other possible minors of the given matrix as follows:

$$M_{12} = 12, \quad M_{13} = -22, \quad M_{21} = 10, \quad M_{22} = 12,$$

and

$$M_{23} = -16, \quad M_{31} = 9, \quad M_{32} = 9, \quad M_{33} = -9,$$

which are the required minors of the given matrix. •

Definition 3.19 *(***Cofactor of a Matrix)**

The cofactor A_{ij} of all elements a_{ij} of a matrix A of order $n \times n$ is given by

$$A_{ij} = (-1)^{i+j} M_{ij},$$

where M_{ij} is the minor of all elements a_{ij} of a matrix A. For example, the cofactor A_{11} of the elements a_{11} of the matrix

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 5 & 3 & 2 \\ 4 & -2 & 4 \end{pmatrix},$$

is computed as follows

$$A_{11} = (-1)^{1+1} M_{11} = M_{11} = 16.$$

Similarly, for other elements, we have

$$A_{12} = -12,\ A_{13} = -22,\ A_{21} = -10,\ A_{22} = 12,$$

$$A_{23} = 16,\ A_{31} = 9,\ A_{32} = -9,\ A_{33} = -9.$$

Program 3.2
MATLAB m-file for finding minors and cofactors of a matrix
function CofA = cofactor(A,i,j)
[m,n] = size(A); if m ~ = n error('Matrix must be square') end
A1 = A([1:i-1,i+1:n],[1:j-1,j+1:n]); Minor = det(A1); CofA = (-1)^ (i+j)*det(Minor);

Definition 3.20 *(***Cofactor Expansion of Determinant of a Matrix)**

Let A be a square matrix, then we define determinant of A is the sum of the products of the elements of the first row and their cofactors. If A is 3×3 matrix, then its determinant is define as

$$\det(A) = |A| = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13}.$$

Similarly, more general for $n \times n$ matrix, we define as

$$\det(A) = |A| = \sum_{1}^{n} a_{ij}A_{ij}, \quad n > 2, \tag{3.10}$$

where summation is on i for any fixed value of jth column $(1 \leq j \leq n)$, or on j for any fixed value of ith row $(1 \leq i \leq n)$ and A_{ij} is the cofactor of element a_{ij}. •

Example 3.2 *Find the minors and cofactors of the matrix A and use it to evaluate the determinant of the matrix*

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

Solution. *The minors of A are calculated as follows*

$$M_{11} = 16, \qquad M_{12} = 10, \qquad M_{13} = 3,$$

and from these values of the minors, we can calculate the cofactors of the elements of the given matrix as follows

$$A_{11} = 16, \qquad A_{12} = -10, \qquad A_{13} = 3.$$

Now by using the cofactor expansion along the first row, we can find the determinant of the matrix as follows

$$\det(A) = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13} = (3)(16) + (1)(-10) + (-4)(3) = 26.$$ •

Note that in the above Example 3.2, we computed the determinant of the matrix by using the cofactor expansion along the first row but it can also be found along the first column of the matrix. To get the results of the Example 3.2, we use the MATLAB command window as follows:

```
>> A = [3 1 -4; 2 5 6; 1 4 8]; DetA = CofFexp(A);
```

Program 3.3
MATLAB m-file for determinant of a matrix by cofactor expansion

```
function DetA = CofFexp(A)
[m,n] = size(A); if m ~= n error('Matrix must be square') end; a = A(1,:);c = [ ];
for i=1:n; c1i = cofactor(A,1,i); c = [c;c1i]; end; DetA = a*c;
```

Theorem 3.4 (The Laplace Expansion Theorem)

The determinant of an $n \times n$ matrix $A = \{a_{ij}\}$, when $n \geq 2$, can be computed as

$$\det(A) = a_{i1}A_{i1} + a_{i2}A_{i2} + \cdots + a_{in}A_{in} = \sum_{j=1}^{n} a_{ij}A_{ij},$$

which is called the cofactor expansion along the ith row and also as

$$\det(A) = a_{1j}A_{1j} + a_{2j}A_{2j} + \cdots + a_{nj}A_{nj} = \sum_{i=1}^{n} a_{ij}A_{ij},$$

is called cofactor expansion along jth column. It is called Laplace Expansion Theorem. •

Note that the cofactor and minor of an element a_{ij} differs only in sign, that is, $A_{ij} = \pm M_{ij}$. A quick way for determining whether to use the $+$ or $-$ is to use the fact that the sign relating A_{ij} and M_{ij} is in the *ith* row and *jth* column of the *checkerboard* array

$$\begin{pmatrix} + & - & + & - & + & \cdots \\ - & + & - & + & - & \cdots \\ + & - & + & - & + & \cdots \\ - & + & - & + & - & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

For example, $A_{11} = M_{11}, A_{21} = -M_{21}, A_{12} = -M_{12}, A_{22} = M_{22}$ and so on.

Definition 3.21 (Cofactor Matrix)

If A is any $n \times n$ matrix and A_{ij} is the cofactor of a_{ij}, then the matrix

$$Cof(A) = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{pmatrix},$$

is called the matrix of cofactor from A. For example, the cofactor of the matrix

$$A = \begin{pmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{pmatrix},$$

can be calculated as follows:

$$A_{11} = 12, \quad A_{12} = 6, \quad A_{13} = -16, \quad A_{21} = 4, \quad A_{22} = 2,$$

$$A_{23} = 16, \quad A_{31} = 12, \quad A_{32} = -10, \quad A_{33} = 16.$$

So that the matrix

$$Cof(A) = \begin{pmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{pmatrix}.$$

is the required of the cofactor matrix. •

Definition 3.22 (Adjoint of a Matrix)
If A is any $n \times n$ matrix and A_{ij} is the cofactor of a_{ij} of A, then the transpose of this matrix is called the adjoint of A and is denoted by Adj(A). For example, the cofactor matrix of the following matrix

$$A = \begin{pmatrix} 3 & 2 & -1 \\ 1 & 6 & 3 \\ 2 & -4 & 0 \end{pmatrix},$$

is calculated as

$$Cof(A) = \begin{pmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{pmatrix}.$$

So by taking its transpose, we get the matrix

$$\begin{pmatrix} 12 & 6 & -16 \\ 4 & 2 & 16 \\ 12 & -10 & 16 \end{pmatrix}^T = \begin{pmatrix} 12 & 4 & 12 \\ 6 & 2 & -10 \\ -16 & 16 & 16 \end{pmatrix} = Adj(A),$$

which is called the adjoint of the given matrix A. •

Example 3.3 *Find the determinant of the following matrix using cofactor expansion and show that* $\det(A) = 0$ *when* $x = 4$

$$A = \begin{pmatrix} x+2 & x & 2 \\ 1 & x-1 & 3 \\ 4 & x+1 & x \end{pmatrix}.$$

Solution. *Using the cofactor expansion along the first row, we compute the determinant of the given matrix as follows:*

$$|A| = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13},$$

where

$$A_{11} = M_{11} = x^2 - 4x - 3,\ A_{12} = -M_{12} = -x + 12,\ A_{13} = -3x + 5.$$

Thus

$$|A| = (x+2)[x^2 - 4x - 3] + x[-x + 12] + 2[-3x + 5] = x^3 - 3x^2 - 5x + 4.$$

Now taking $x = 4$*, we get*

$$|A| = (4)^3 - 3(4)^2 - 5(4) + 4 = 64 - 48 - 20 + 4 = 0,$$

which is the required determinant of the matrix at $x = 4$. •

The following are special properties which will be helpful in reducing the amount of work involved in evaluating determinants.

Theorem 3.5 *(***Properties of the Determinant***)*

Let A *be an* $n \times n$ *matrix:*

1. *The determinant of a matrix* A *is zero if any row or column is zero or equal to a linear combination of other rows and columns.*
   For example, if
   $$A = \begin{pmatrix} 3 & 1 & 0 \\ 2 & 1 & 0 \\ 4 & 3 & 0 \end{pmatrix},$$
   then $\det(A) = 0$.

2. *A determinant of a matrix* A *is changed in sign if the two rows or two columns are interchange.*
   For example, if
   $$A = \begin{pmatrix} 3 & 2 \\ 4 & 5 \end{pmatrix},$$
   then $\det(A) = 7$*, but for the matrix*
   $$B = \begin{pmatrix} 4 & 5 \\ 3 & 2 \end{pmatrix},$$
   obtained from the matrix A *by interchanging its rows, we have* $\det(B) = -7$.

3. *The determinant of a matrix* A *is equal to the determinant of its transposed. For example, if*
   $$A = \begin{pmatrix} 5 & 3 \\ 4 & 4 \end{pmatrix},$$
   then $\det(A) = 8$*, and for the matrix*
   $$B = \begin{pmatrix} 5 & 4 \\ 3 & 4 \end{pmatrix},$$
   obtained from the matrix A *by taking its transpose, we have*
   $$\det(B) = 8 = \det(A).$$

5. *If the matrix B is obtained from the matrix A by multiplying every element in one row or in one column by k, then determinant of the matrix B is equal to k times the determinant of A. For example, if*

$$A = \begin{pmatrix} 6 & 5 \\ 3 & 4 \end{pmatrix},$$

then $\det(A) = 9$, *but for the matrix*

$$B = \begin{pmatrix} 12 & 10 \\ 3 & 4 \end{pmatrix},$$

obtained from the matrix A by multiplying its first row by 2, we have

$$\det(B) = 18 = 2(9) = 2\det(A).$$

6. *If the matrix B is obtained from the matrix A by adding to a row (or a column) of a multiple of another row (or another column) of A, then determinant of the matrix B is equal to the determinant of A. For example, if*

$$A = \begin{pmatrix} 4 & 3 \\ 5 & 4 \end{pmatrix}$$

then $\det(A) = 1$, *and for the matrix*

$$B = \begin{pmatrix} 4 & 3 \\ 13 & 10 \end{pmatrix},$$

obtained from the matrix A by adding to its second row 2 times the first row, we have

$$\det(B) = 1 = \det(A).$$

7. *If two rows or two columns of a matrix A are identical, then the determinant is zero. For example, if*

$$A = \begin{pmatrix} 2 & 3 \\ 2 & 3 \end{pmatrix}, \quad \textit{then} \quad \det(A) = 0.$$

8. *The determinant of a product of matrices is the product of the determinants of all matrices. For example, if*

$$A = \begin{pmatrix} 3 & 4 & 5 \\ 3 & 2 & 1 \\ 2 & 1 & 6 \end{pmatrix} \quad \textit{and} \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 2 & 3 \\ 1 & 3 & 5 \end{pmatrix},$$

then $\det(A) = -36$ *and* $\det(A) = -3$. *Also,*

$$AB = \begin{pmatrix} 24 & 29 & 46 \\ 12 & 13 & 20 \\ 12 & 24 & 39 \end{pmatrix},$$

then $\det(AB) = 108$. *Thus*

$$\det(A)\det(B) = (-36)(-3) = 108 = \det(AB).$$

9. *The determinant of a triangular matrix (upper-triangular or lower-triangular matrix) is equal to the product of all their main diagonal elements. For example, if*

$$A = \begin{pmatrix} 3 & 4 & 5 \\ 0 & 4 & 7 \\ 0 & 0 & 5 \end{pmatrix}, \quad \textit{then} \quad \det(A) = (3)(4)(5) = 60.$$

10. *The determinant of an $n \times n$ matrix A times scalar multiple k equal to k^n times the determinant of the matrix A, that is* $\det(kA) = k^n \det(A)$. *For example, if*

$$A = \begin{pmatrix} 3 & 4 & 5 \\ 2 & 3 & 6 \\ 1 & 0 & 5 \end{pmatrix},$$

then $\det(A) = 14$, *and for the matrix*

$$B = 2A = \begin{pmatrix} 6 & 8 & 10 \\ 4 & 6 & 12 \\ 2 & 0 & 10 \end{pmatrix},$$

obtained from the matrix A by multiply by 2, we have

$$\det(B) = 112 = 8(14) = 2^3 \det(A).$$

11. *The determinant of the kth power of a matrix A equal to the kth power of the determinant of the matrix A, that is* $\det(A^k) = (\det(A))^k$. *For example, if*

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

then $\det(A) = 12$, *and for the matrix*

$$B = A^3 = \begin{pmatrix} -18 & -30 & 12 \\ 24 & -3 & -9 \\ 3 & -12 & 3 \end{pmatrix},$$

obtained by taking cubic power of the matrix A, we have

$$\det(B) = 1728 = (12)^3 = (\det(A))^3.$$

12. *The determinant of a scalar matrix* (1×1) *is equal to the element itself. For example, if* $A = (8)$, *then* $\det(A) = 8$.

Example 3.4 *Find all the values of α for which* $\det(A) = 0$, *where*

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

Solution. *We find the determinant of the given matrix by using the cofactor expansion along the first row, so we compute*

$$\begin{aligned} |A| &= a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13} \\ &= (\alpha-3)\begin{vmatrix} \alpha-1 & 1 \\ 2 & \alpha \end{vmatrix} - 1\begin{vmatrix} 0 & 1 \\ 0 & \alpha \end{vmatrix} + 0\begin{vmatrix} 0 & \alpha-1 \\ 0 & 2 \end{vmatrix} \\ &= (\alpha-3)(\alpha+1)(\alpha-2). \end{aligned}$$

Given $\det(A) = 0$*, implies that,* $\alpha = -1,\ 2,\ 3$*, the required values of* α*.* •

Theorem 3.6 *If* A *is an invertible matrix, then*

1. $\det(A) \neq 0$ *2.* $\det(A^{-1}) = \dfrac{1}{\det(A)}$ *3.* $A^{-1} = \dfrac{Adj(A)}{\det(A)}$.

4. $(adj(A))^{-1} = \dfrac{A}{\det(A)} = adj(A^{-1})$ *5.* $\det(adj(A)) = \det(A)^{n-1}$.

By using the Theorem 3.6 we can find the inverse of a matrix by showing that determinant of a matrix not equal to zero and by using adjoint and determinant of the given matrix A*.* •

Example 3.5 *For what values of* α *the following matrix has an inverse:*

$$A = \begin{pmatrix} 1 & 0 & \alpha \\ 2 & 2 & 1 \\ 0 & 2\alpha & 1 \end{pmatrix}.$$

Solution. *We find the determinant of the given matrix by using the cofactor expansion along the first row as follows:*

$$|A| = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13},$$

which is equal to

$$|A| = (1)A_{11} + (0)A_{12} + (\alpha)A_{13} = A_{11} + \alpha A_{13} = 2 - 2\alpha + 4\alpha^2.$$

From the Theorem 3.6 we know that the matrix has an inverse if $\det(A) \neq 0$*, so*

$$|A| = 2 - 2\alpha + 4\alpha^2 = 2(2\alpha+1)(\alpha-1) \neq 0.$$

Hence the given matrix has an inverse if $\alpha \neq -1/2$ *and* $\alpha \neq 1$*.* •

Example 3.6 *Use the adjoint method to compute the inverse of the following matrix*

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

Also, find the inverse and determinant of the adjoint matrix.

Solution. *First we compute the determinant of the given matrix as follows:*

$$\det(A) = |A| = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13} = (1)(-4) - (2)(3) + (-1)(5) = -15,$$

and the compute the nine cofactors as follows:

$$A_{11} = -4,\ A_{12} = -3,\ A_{13} = 5,\ A_{21} = -6,\ A_{22} = 3,\ A_{23} = 0,\ A_{31} = 1,\ A_{32} = -3,\ A_{33} = -5.$$

Thus we have the cofactor matrix and the adjoint matrix as follows

$$Cof(A) = \begin{pmatrix} -4 & -3 & 5 \\ -6 & 3 & 0 \\ 1 & -3 & -5 \end{pmatrix},\ adj(A) = \begin{pmatrix} -4 & -3 & 5 \\ -6 & 3 & 0 \\ 1 & -3 & -5 \end{pmatrix}^T = \begin{pmatrix} -4 & -6 & 1 \\ -3 & 3 & -3 \\ 5 & 0 & -5 \end{pmatrix}.$$

To get adjoint of the matrix of the Example 3.6, we use MATLAB command window as:

```
>> A = [1 2 -1;2 -1 1;1 2 2]; AdjA = Adjoint(A);
```

Program 3.4
MATLAB m-file for adjoint of a matrix

```
function AdjA = Adjoint(A)
[m,n] = size(A); if m ~= n error('Matrix must be square') end; A1 = [ ];
for i = 1:n; for j=1:n; A1 = [A1;cofactor(A,i,j)];end;end; AdjA = reshape(A1,n,n);
```

Then by using the Theorem 3.6 we can have the inverse of the matrix as follows:

$$A^{-1} = \frac{Adj(A)}{\det(A)} = -\frac{1}{15}\begin{pmatrix} -4 & -6 & 1 \\ -3 & 3 & -3 \\ 5 & 0 & -5 \end{pmatrix} = \begin{pmatrix} 4/15 & 2/5 & -1/15 \\ 1/5 & -1/5 & 1/5 \\ -1/3 & 0 & 1/3 \end{pmatrix}.$$

Using the Theorem 3.6 we can compute the inverse of the adjoint matrix as follows:

$$(adj(A))^{-1} = \frac{A}{\det(A)} = \begin{pmatrix} -1/15 & -2/15 & 1/15 \\ -2/15 & 1/15 & -1/15 \\ -1/15 & -2/15 & -2/15 \end{pmatrix},$$

and $\det(adj(A)) = (\det(A))^{3-1} = (-15)^2 = 225.$ •

Example 3.7 *If* $\det(A) = 3$ *and* $\det(B) = 4$*, then show that*

$$det(A^2B^{-1}A^TB^3) = 432.$$

Solution. *By using the properties of the determinant of the matrix, we have*

$$det(A^2B^{-1}A^TB^3) = \det(A^2)\det(B^{-1})\det(A^T)\det(B^3),$$

which can be also written as

$$det(A^2B^{-1}A^TB^3) = (\det(A))^2\frac{1}{\det(B)}(\det(A))(\det(B))^3.$$

Now using the given information, we get

$$det(A^2B^{-1}A^TB^3) = (3)^2\frac{1}{4}(3)(4)^3 = 3^34^2 = 432,$$

the required solution. •

3.2.4 Matrix Inversion Method

If matrix A is nonsingular, then the linear system (3.6) always has a unique solution for each $\mathbf{b}$, since the inverse matrix A^{-1} exists, so the solution of the system (3.6) can formally expressed as

$$A^{-1}A\mathbf{x} = A^{-1}\mathbf{b}, \quad \text{or} \quad \mathbf{I}\mathbf{x} = A^{-1}\mathbf{b},$$

$$\mathbf{x} = A^{-1}\mathbf{b}. \tag{3.11}$$

If A is a square invertible matrix, there exists a sequence of elementary row operations that carry A to the identity matrix $\mathbf{I}$ of the same size, that is, $A \longrightarrow \mathbf{I}$. This same sequence of row operations carries $\mathbf{I}$ to A^{-1}, that is, $\mathbf{I} \longrightarrow A^{-1}$. This can be also written as

$$[A|\mathbf{I}] \longrightarrow [\mathbf{I}|A^{-1}].$$

Example 3.8 *Use matrix inversion method to find unique solution the linear system* $A\mathbf{x} = \mathbf{b}$*, where*

$$A = \begin{pmatrix} 1 & 2 & 0 \\ -2 & 1 & 2 \\ -1 & 1 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Solution. *First we compute the inverse of the given matrix which has the form*

$$A^{-1} = \begin{pmatrix} 1 & 2 & -4 \\ 0 & -1 & 2 \\ 1 & 3 & -5 \end{pmatrix},$$

and then we can find unique solution of the given system as

$$\mathbf{x} = A^{-1}\mathbf{b} = \begin{pmatrix} 1 & 2 & -4 \\ 0 & -1 & 2 \\ 1 & 3 & -5 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix},$$

which is the solution of the given system by the matrix inversion method. •

Thus, when the matrix inverse A^{-1} of the coefficient matrix A is computed, the solution vector $\mathbf{x}$ of linear system is simply the product of inverse matrix A^{-1} and the right-hand side vector $\mathbf{b}$.

MATLAB commands to solve the linear system by matrix inverse method is defined as:

```
>> A = [1 2 0; -2 1 2; -1 1 1]; b = [1; 1; 1]; x = inv(A) * b
```

Theorem 3.7 *For an* $n \times n$ *matrix* A*, the following properties are equivalent:*

1. *The inverse of matrix* A *exists, that is,* A *is nonsingular.*
2. *The determinant of matrix* A *is nonzero.*
3. *The homogeneous system* $A\mathbf{x} = \mathbf{0}$ *has a trivial solution* $\mathbf{x} = \mathbf{0}$*.*

4. The nonhomogeneous system $A\mathbf{x} = \mathbf{b}$ *has a unique solution.* •

Not all matrices have inverses. Singular matrices don't have inverse and thus the corresponding system of equations does not have a unique solution. The inverse of a matrix can also be computed by using the following numerical methods for linear systems, called, Gauss-elimination method, Gauss-Jordan method and LU-decomposition method but the best and simplest method for finding the inverse of a matrix is to perform the Gauss-Jordan method on the augmented matrix with identity matrix of same size.

3.3 Solutions of Linear Systems of Equations

Now we shall discuss numerical methods for solving system of linear equations. We shall discuss both direct and indirect (iterative) methods for the solution of given linear systems. In direct method we shall discuss the familiar technique called the *method of elimination* to find the solution of linear systems. This method starts with the augmented matrix of the given linear system and obtain a matrix of a certain form. This new matrix represents a linear system that has exactly the same solutions as the given origin system. In indirect methods we shall discuss Jacobi and Gauss-Seidel methods.
The following basic theorems on the solvability of linear systems are proved in linear algebra.

Theorem 3.8 *A homogeneous system of n equations in n unknowns has a solution other than the trivial solution if and only if the determinant of the coefficients matrix A vanishes, that is matrix A is singular.* •

Theorem 3.9 (Necessary and Sufficient Condition for a unique solution)

A nonhomogeneous system of n equations in n unknowns has a unique solution if and only if the determinant of a coefficients matrix A is not vanishes, that is, A is nonsingular. •

Before, we discuss numerical methods for solving linear system, we introduce the most important numerical quantity associated with a matrix.

Definition 3.23 (Rank of a Matrix)

The rank of a matrix A is the number of pivots. An $m \times n$ *matrix will, in general, have a rank* r*, where* r *is an integer and* $r \leq min\{m, n\}$*. If* $r = min\{m, n\}$*, then the matrix is said to be full rank. If* $r < min\{m, n\}$*, then the matrix is said to be rank deficient.* •

In principle, the rank of a matrix can be determined by using the Gaussian elimination process in which the coefficient matrix A is reduced to upper-triangular form U. After reducing the matrix to triangular form, we find that the rank is the number of columns with nonzero values on the diagonal of U. In practice, especially for large matrices, round-off errors during the row operation may cause a loss of accuracy in this method of rank computation.

Theorem 3.10 *For a system of n equations in n unknowns written in the form* $A\mathbf{x} = \mathbf{b}$*, then solution* $\mathbf{x}$ *of a system exists and is unique for any* $\mathbf{b}$ *if and only if rank*$(A) = n$. •

Conversely, if rank$(A) < n$ for an $n \times n$ matrix A, then the system of equations $A\mathbf{x} = \mathbf{b}$ may or may not be consistent. Such a system may not have solution, or the solution, if it exists, will not be unique. For example, the rank of the following matrix is 3.

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 1 & 5 \\ 1 & 1 & 6 \end{pmatrix}.$$

In MATLAB command, the built-in **rank** function can be use to estimate the rank of a matrix:

```
>> A = [1 2 4;1 1 5;1 1 6]; rank(A)
```

Note that:

$$\begin{array}{lcl} \text{rank}(AB) & \leq & min(\text{rank}(A), \text{rank}(B)) \\ \text{rank}(A+B) & \leq & \text{rank}(A) + \text{rank}(B) \\ \text{rank}(AA^T) & = & \text{rank}(A) = \text{rank}(A^TA) \end{array}$$

Although the rank of a matrix is very useful to categorize the behaviour of matrices and systems of equations, the rank of a matrix is usually not computed. •

3.4 Direct Numerical Methods for Solving Linear Systems

To solve the systems of linear equations using the numerical methods, there are two types of methods available. Methods of first type are called *direct methods* or *elimination methods*. The other type of the numerical methods are called *iterative methods*. In this chapter we will discuss both type of the numerical methods. The first type of methods find the solution in a finite number of steps. These methods are guaranteed to succeed and are recommended for general purpose. Here, we will consider *Gaussian elimination method* and its variants and *LU decomposition*, by Doolittle's and Crout's methods.
The direct method refers to a procedure for computing a solution from a form that is mathematically exact. We shall begin with simple method, called Gaussian elimination method and its variants and then continue with the methods involving triangular matrices, symmetric and tridiagonal matrices.

3.4.1 Gaussian Elimination Method

It is one of the most popular and widely used direct method for solving linear systems of algebraic equations. No method of solving linear systems requires fewer operations than the Gaussian procedure. The goal of the Gaussian elimination method for solving linear systems is to convert the original system into the equivalent upper-triangular system and from which each unknown is determined by backward substitution.
The Gaussian elimination procedure start with *forward elimination*, in which the first equation in the linear system is used to eliminate the first variable from the rest of $(n-1)$ equations. Then the new second equation is used to elimination second variable from the rest of $(n-2)$ equations, and so on. If $(n-1)$ such elimination is performed then the resulting system will be the triangular form. Once this forward elimination is completed, we can determine whether the system is overdetermined or underdetermined or has a unique solution. If it has a unique solution,

then the *backward substitution* is used to solve the triangular system easily and one can find the unknown variables involve in the system.
Now we shall describe the method in detail for a system of n linear equations. Consider the following a system of n linear equations:

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & + & \cdots & + & a_{3n}x_n & = & b_3 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & & a_{n3}x_3 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array} \tag{3.12}$$

Forward Elimination

Consider first equation of the given system (3.12)

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1, \tag{3.13}$$

as first pivotal equation with first pivot element a_{11}. Then the first equation times multiples $m_{i1} = (a_{i1}/a_{11}),\ i = 2, 3, \ldots, n$, is subtracted from the *ith* equation to eliminate first variable x_1, producing an equivalent system

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ & & a_{22}^{(1)}x_2 & + & a_{23}^{(1)}x_3 & + & \cdots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ & & a_{32}^{(1)}x_2 & + & a_{33}^{(1)}x_3 & + & \cdots & + & a_{3n}^{(1)}x_n & = & b_3^{(1)} \\ & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ & & a_{n2}^{(1)}x_2 & + & a_{n3}^{(1)}x_3 & + & \cdots & + & a_{nn}^{(1)}x_n & = & b_n^{(1)} \end{array} \tag{3.14}$$

Now consider a second equation of the system (3.14), which is

$$a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \cdots + a_{2n}^{(1)}x_n = b_2^{(1)}, \tag{3.15}$$

as second pivotal equation with second pivot element $a_{22}^{(1)}$. Then the second equation times multiples $m_{i2} = (a_{i2}^{(1)}/a_{22}^{(1)}),\ i = 3, \ldots, n$, is subtracted from the *ith* equation to eliminate second variable x_2, producing an equivalent system

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ & & a_{22}^{(1)}x_2 & + & a_{23}^{(1)}x_3 & + & \cdots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ & & & & a_{33}^{(2)}x_3 & + & \cdots & + & a_{3n}^{(2)}x_n & = & b_3^{(2)} \\ & & & & \vdots & & \vdots & & \vdots & & \vdots \\ & & & & a_{n3}^{(2)}x_3 & + & \cdots & + & a_{nn}^{(2)}x_n & = & b_n^{(2)} \end{array} \tag{3.16}$$

Now consider a third equation of the system (3.16), which is

$$a_{33}^{(2)}x_3 + \cdots + a_{3n}^{(2)}x_n = b_3^{(2)}, \tag{3.17}$$

as the third pivotal equation with third pivot element $a_{33}^{(2)}$. Then the third equation times multiples $m_{i3} = (a_{i3}^{(2)}/a_{33}^{(2)}),\ i = 4, \ldots, n$, is subtracted from the *ith* equation to eliminate third variable x_3. Similarly, after *(n-1)th* steps, we have the *nth* pivotal equation which have only one unknown variable x_n, that is

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ & + & a_{22}^{(1)}x_2 & + & a_{23}^{(1)}x_3 & + & \cdots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ & & & + & a_{33}^{(2)}x_3 & + & \cdots & + & a_{3n}^{(2)}x_n & = & b_3^{(2)} \\ & & & & & & & & \vdots & & \vdots \\ & & & & & & & & a_{nn}^{(n-1)}x_n & = & b_n^{(n-1)} \end{array} \tag{3.18}$$

with *nth* pivotal element $a_{nn}^{(n-1)}$. After getting the upper-triangular system which is equivalent to the original system, the forward elimination is completed.

Backward Substitution

After the triangular set of equations has been obtained, the last equation of the system (3.18) yields the value of x_n directly. The value is then substituted into the equation next to the last one of the system (3.18) to obtain a value of x_{n-1}, which is, in turn, used along with the value of x_n in the second to the last equation to obtain a value of x_{n-2}, and so on. Mathematical formula can be obtain for the backward substitution

$$\left.\begin{array}{rcl} x_n & = & \dfrac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \\ x_{n-1} & = & \dfrac{1}{a_{n-1n-1}^{(n-2)}}\left(b_{n-1}^{(n-2)} - a_{n-1n}^{(n-2)}x_n\right) \\ & \vdots & \\ x_1 & = & \dfrac{1}{a_{11}}\left(b_1 - \displaystyle\sum_{j=2}^{n} a_{1j}x_j\right) \end{array}\right\} \tag{3.19}$$

The Gaussian elimination can be carried out by writing only the coefficients and the right-hand side terms in a matrix form, which means the augmented matrix form. Indeed, this is exactly what a computer program for the Gaussian elimination does. Even for hand calculation, the augmented matrix form is more convenient than writing all set of equations. The augmented matrix is formed as follows

$$\left(\begin{array}{ccccccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \vdots & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & \vdots & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & \vdots & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & \vdots & b_n \end{array}\right). \tag{3.20}$$

The operations used in the Gaussian elimination method can now be applied to the augmented matrix. Consequently system (3.18) is now written directly as follows:

$$\left(\begin{array}{ccccccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & \vdots & b_1 \\ & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} & \vdots & b_2^{(1)} \\ & & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} & \vdots & b_3^{(2)} \\ & & & \vdots & \vdots & \vdots & \\ & & & & a_{nn}^{(n-1)} & \vdots & b_n^{(n-1)} \end{array}\right), \tag{3.21}$$

from which the unknowns are determined as before by using backward substitution. The number of multiplications and divisions for the Gaussian elimination method for one **b** vector is approximately

$$N = \left(\frac{n^3}{3}\right) + n^2 - \left(\frac{n}{3}\right). \tag{3.22}$$

3.4.1.1 Simple Gaussian Elimination Method (or Without Pivoting)

Firstly, we will solve the linear system using the simplest variation of the Gaussian elimination method, called the *simple* Gaussian elimination or the Gaussian elimination *without pivoting*. The basic of this variation is that all the possible diagonal elements (called *pivot elements*) should be nonzero. If at any stage it becomes zero, then interchange that row with any below row with nonzero element at that position. After getting upper-triangular matrix, we use backward substitution to get the solution of the given linear system. Note that $\det(A) = (-1)^k \det(U)$, where k is number of times rows are interchanged.

Example 3.9 *Solve the following linear system using the simple Gaussian elimination method*

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & + & x_3 & = & 2 \\ 2x_1 & + & 5x_2 & + & 3x_3 & = & 1 \\ x_1 & + & 3x_2 & + & 4x_3 & = & 5 \end{array}$$

solution. *The process begins with the augmented matrix form*

$$\left(\begin{array}{cccccc} 1 & 2 & 1 & \vdots & 2 \\ 2 & 5 & 3 & \vdots & 1 \\ 1 & 3 & 4 & \vdots & 5 \end{array}\right).$$

Since $a_{11} = 1 \neq 0$, so we wish to eliminate the elements a_{21} and a_{31} by subtracting from the second and third rows the appropriate multiples of the first row. In this case the multiples are $m_{21} = \dfrac{2}{1} = 2$ and $m_{31} = \dfrac{1}{1} = 1$. Hence

$$\left(\begin{array}{cccccc} 1 & 2 & 1 & \vdots & 2 \\ 0 & 1 & 1 & \vdots & -3 \\ 0 & 1 & 3 & \vdots & 3 \end{array}\right).$$

As $a_{22}^{(1)} = 1 \neq 0$, *therefore, we wish to eliminate entry in* $a_{32}^{(1)}$ *position by subtracting the multiple* $m_{32} = \dfrac{1}{1} = 1$ *of the second row from the third row, to get*

$$\left(\begin{array}{cccc|c} 1 & 2 & 1 & \vdots & 2 \\ 0 & 1 & 1 & \vdots & -3 \\ 0 & 0 & 2 & \vdots & 6 \end{array}\right).$$

Obviously, the original set of equations has been transformed to an upper-triangular form. Since all the diagonal elements of the obtaining upper-triangular matrix are nonzero, which means that the coefficient matrix of the given system is nonsingular ($\det(A) = (-1)^0 \det(U) = 2$*) and therefore, the given system has a unique solution. Now expressing the set in algebraic form yields*

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & x_3 & = & 2 \\ & & x_2 & + & x_3 & = & -3 \\ & & & & 2x_3 & = & 6 \end{array}$$

Now using backward substitution, we get

$$\begin{array}{lcll} 2x_3 = 6, & & \textit{gives} & x_3 = 3, \\ x_2 = -x_3 - 3 = -(3) - 3 = -6, & & \textit{gives} & x_2 = -6, \\ x_1 = 2 - 2x_2 - x_3 = 2 - 2(-6) - 3, & & \textit{gives} & x_1 = 11, \end{array}$$

which is the required solution of the given system. •

The above results can be obtained using MATLAB commands as follows:

```
>> B = [1 2 1 2;2 5 3 1;1 3 4 5]; x = WP(B); disp(x)
```

Program 3.5

MATLAB m-file for the Simple Gaussian Elimination Method

```
function x=WP(B)
[n,t]=size(B); U=B; for k=1:n-1; for i=k:n-1; m=U(i+1,k)/U(k,k);for j=1:t;
U(i+1,j)=U(i+1,j)-m*U(k,j);end;end end;  i=n; x(i,1)=U(i,t)/U(i,i);  for i=n-1:-1:1; s=0;
for k=n:-1:i+1; s = s + U(i,k) * x(k,1); end; x(i,1)=(U(i,t)-s)/U(i,i); end; B; U; x; end
```

In the simple description of Gaussian elimination without pivoting just given, we used the *kth* equation to eliminate variable x_k from equations $k+1, \ldots, n$ during the *kth* step of the procedure. This is possible only if at the beginning of the *kth* step, the coefficient $a_{kk}^{(k-1)}$ of x_k in equation k is not zero. Since these coefficients are used as denominators both in the multipliers m_{ij} and in the backward substitution equations. But this does not necessarily mean that the linear system is not solvable, but that the procedure of solution must be altered. Using Gaussian elimination method $\det(A) = (-1)^k \det(U)$, where k is number of times we interchange the rows.

Example 3.10 *Solve the following linear system using the simple Gaussian elimination method.*

$$\begin{array}{rcrcrcl} & & x_2 & + & x_3 & = & 1 \\ x_1 & + & 2x_2 & + & 2x_3 & = & 1 \\ 2x_1 & + & x_2 & + & 2x_3 & = & 3 \end{array}$$

Solution. *Writing the given system in the augmented matrix form*

$$\left(\begin{array}{cccc} 0 & 1 & 1 & \vdots \ 1 \\ 1 & 2 & 2 & \vdots \ 1 \\ 2 & 1 & 2 & \vdots \ 3 \end{array}\right).$$

To solve this system, the simple Gaussian elimination method will fail immediately because the element in the first row on the leading diagonal, the pivot, is zero. Thus it is impossible to divide that row by the pivot value. Clearly, this difficulty can be overcome by rearranging the order of the rows; for example by making the first row the second, gives

$$\left(\begin{array}{cccc} 1 & 2 & 2 & \vdots \ 1 \\ 0 & 1 & 1 & \vdots \ 1 \\ 2 & 1 & 2 & \vdots \ 3 \end{array}\right).$$

Now we use the usual elimination process. The first elimination step is to eliminate the element $a_{31} = 2$ *from the third row by subtracting a multiple* $m_{31} = \dfrac{2}{1} = 2$ *of row 1 from row 3, gives*

$$\left(\begin{array}{cccc} 1 & 2 & 2 & \vdots \ 1 \\ 0 & 1 & 1 & \vdots \ 1 \\ 0 & -3 & -2 & \vdots \ 1 \end{array}\right).$$

We finished with the first elimination step since the element a_{21} *is already eliminated from second row. The second elimination step is to eliminate the element* $a_{32}^{(1)} = -3$ *from the third row by subtracting a multiple* $m_{32} = \dfrac{-3}{1}$ *of row 2 from row 3, gives*

$$\left(\begin{array}{cccc} 1 & 2 & 2 & \vdots \ 1 \\ 0 & 1 & 1 & \vdots \ 1 \\ 0 & 0 & 1 & \vdots \ 4 \end{array}\right).$$

Note that the $\det(A) = (-1)^1 \det(U) = -1$. *Obviously, the original set of equations has been transformed to an upper-triangular form. Now expressing the set in algebraic form yields*

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & + & 2x_3 & = & 1 \\ & & x_2 & + & x_3 & = & 1 \\ & & & & x_3 & = & 4 \end{array}$$

Using backward substitution, we get, $x_1 = -1,\ x_2 = -3,\quad x_3 = 4$, *the solution of the system.* •

Example 3.11 *Solve the linear system using the simple Gaussian elimination method*

$$\begin{array}{rcrcrcl} x_1 & + & x_2 & + & x_3 & = & 3 \\ 2x_1 & + & 2x_2 & + & 3x_3 & = & 7 \\ x_1 & + & 2x_2 & + & 3x_3 & = & 6 \end{array}$$

Solution. *Writing the given system in the augmented matrix form*

$$\left(\begin{array}{cccc} 1 & 1 & 1 & 3 \\ 2 & 2 & 3 & 7 \\ 1 & 2 & 3 & 6 \end{array}\right).$$

First elimination step is to eliminate the elements $a_{21} = 2$ *and* $a_{31} = 1$ *from second and third rows by subtracting multiples* $m_{21} = \dfrac{2}{1} = 2$ *and* $m_{31} = \dfrac{1}{1} = 1$ *of row 1 from row 2 and row 3 respectively, gives*

$$\left(\begin{array}{cccc} 1 & 1 & 1 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{array}\right).$$

We finished with the first elimination step. To start the second elimination step, since we note that the element $a_{22}^{(1)} = 0$*, called the second pivot element, so the simple Gaussian elimination cannot continue in its present form. Therefore, we interchange the rows 2 and 3, to get*

$$\left(\begin{array}{cccc} 1 & 1 & 1 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 1 \end{array}\right).$$

We finished with the second elimination step since the element $a_{32}^{(1)}$ *is already eliminated from third row. Obviously, the original set of equations has been transformed to an upper-triangular form. Now expressing the set in algebraic form yields*

$$\begin{array}{rcrcrcl} x_1 & + & x_2 & + & x_3 & = & 3 \\ & & x_2 & + & 2x_3 & = & 3 \\ & & & & x_3 & = & 1 \end{array}$$

Using backward substitution, we get, $x_1 = 1,\ x_2 = 1, \quad x_3 = 1$*, the solution of the system.* •

Example 3.12 *Use the simple Gaussian elimination method, find all values of* a *and* b *for which the following linear system is consistent or inconsistent.*

$$\begin{array}{rcrcrcl} 2x_1 & - & x_2 & + & 3x_3 & = & 1 \\ 4x_1 & + & 2x_2 & + & 2x_3 & = & 2a \\ 2x_1 & + & x_2 & + & x_3 & = & b \end{array}$$

Solution. *Using* $m_{21} = 2$, $m_{31} = 1$, *and* $m_{32} = \dfrac{1}{2}$, *we get*

$$[A|b] = \begin{pmatrix} 2 & -1 & 3 & 1 \\ 4 & 2 & 2 & 2a \\ 2 & 1 & 1 & b \end{pmatrix} \equiv \begin{pmatrix} 2 & -1 & 3 & 1 \\ 0 & 4 & -4 & 2a-2 \\ 0 & 2 & -2 & b-1 \end{pmatrix} \equiv \begin{pmatrix} 2 & -1 & 3 & 1 \\ 0 & 4 & -4 & 2a-2 \\ 0 & 0 & 0 & b-a \end{pmatrix}.$$

We finished with the second column. So third row of the equivalent upper-triangular system is

$$0x_1 + 0x_2 + 0x_3 = b - a. \tag{3.23}$$

Firstly, if (3.23) has no constraint on unknowns x_1, x_2, *and* x_3, *then the upper-triangular system represents only two non-trivial equations, namely*

$$\begin{array}{rcrcrcr} 2x_1 & - & x_2 & + & 3x_3 & = & 1 \\ & & 4x_2 & - & 4x_3 & = & 2a-2 \end{array}$$

in three unknowns. As a result, one of the unknowns can be chosen arbitrarily, say $x_3 = x_3^*$, *then* x_2^* *and* x_1^* *can be obtained by using backward substitution*

$$x_2^* = a/2 - 1/2 + x_3^*; \quad x_1^* = \frac{1}{2}(1 + a/2 - 1/2 - 2x_3^*).$$

Hence

$$\mathbf{x}^* = [\frac{1}{2}(1/2 + a/2 - 2x_3^*), 1/2a - 1/2 + x_3^*, x_3^*]^T,$$

is an approximation solution of given system for any value of x_3^* *for any real value of* a. *Hence the given linear system is consistent (infinitely many solutions).*
Secondly, when $b - a \neq 0$, *in this case (3.23) puts a restriction on unknowns* x_1, x_2 *and* x_3 *that is impossible to satisfy. So the system cannot have any solutions and therefore, it is inconsistent.* •

Example 3.13 *Use the simple Gaussian elimination method, find all values of* a *and* b *for which the following linear system is consistent or inconsistent. Find the solutions when the system is consistent.*

$$\begin{array}{rcrcrcr} x_1 & - & 2x_2 & + & 3x_3 & = & 4 \\ 2x_1 & - & 3x_2 & + & ax_3 & = & 5 \\ 3x_1 & - & 4x_2 & + & 5x_3 & = & b \end{array}$$

Solution. *Writing the given system in the augmented matrix form*

$$[A|b] = \begin{pmatrix} 1 & -2 & 3 & 4 \\ 2 & -3 & a & 5 \\ 3 & -4 & 5 & b \end{pmatrix} \equiv \begin{pmatrix} 1 & -2 & 3 & 4 \\ 0 & 1 & a-6 & -3 \\ 0 & 2 & -4 & b-12 \end{pmatrix} \equiv \begin{pmatrix} 1 & -2 & 3 & 4 \\ 0 & 1 & a-6 & -3 \\ 0 & 0 & -2a+8 & b-6 \end{pmatrix}.$$

CASE I. *Inconsistent system (no solution), if we take* $a = 4$ *and* $b \neq 6$, *gives*

$$\begin{array}{rcrcrcr} x_1 & - & 2x_2 & + & 3x_3 & = & 4 \\ & & x_2 & + & (a-6)x_3 & = & -3 \\ & & & & (-2a+8)x_3 & = & (b-6) \end{array}$$

CASE II. *Consistent system (infinitely many solutions), if we take $a = 4$ and $b = 6$, gives*

$$\begin{array}{rcrcrcr} x_1 & - & 2x_2 & + & 3x_3 & = & 4 \\ & & x_2 & + & (a-6)x_3 & = & -3 \\ & & & & (-2a+8)x_3 & = & (b-6) \end{array}$$

gives

$$\begin{array}{rcrcrcr} x_1 & - & 2x_2 & + & 3x_3 & = & 4 \\ & & x_2 & + & (a-6)x_3 & = & -3 \\ & & & & 0x_3 & = & 0 \end{array}$$

Thus the infinitely many solutions

$$x_1 = -2 + t, \quad x_2 = -3 + 2t, \quad x_3 = t, \quad t \in R.$$

CASE III. *Consistent system (exactly one solution), if we take $a \neq 4$ and $b \in R$, gives*

$$\begin{array}{rcrcrcr} x_1 & - & 2x_2 & + & 3x_3 & = & 4 \\ & & x_2 & + & (a-6)x_3 & = & -3 \\ & & & & (-2a+8)x_3 & = & (b-6) \end{array}$$

$x_1 = \dfrac{16a + 9b - 2ab - 70}{-2a+8}$, $x_2 = \dfrac{12a + 6b - ab - 60}{-2a+8}$, $x_3 = \dfrac{b-6}{-2a+8}$, *the unique solution.* •

Example 3.14 *Show that for α_1 and α_2 the following linear system has unique solution. Find α_1 and α_2 and compute the solution in each case.*

$$\begin{array}{rcrcrcr} x_1 & + & x_2 & + & x_3 & = & \alpha^2 \\ 4x_1 & - & x_2 & + & x_3 & = & \alpha \\ x_1 & + & x_2 & + & 2x_3 & = & 1 \\ x_1 & + & 6x_2 & + & 5x_3 & = & 1 \end{array}$$

Solution. *Using the multiples $m_{21} = 4$, $m_{31} = 1$, and $m_{41} = 1$, gives matrix form*

$$[A|b] = \begin{pmatrix} 1 & 1 & 1 & \alpha^2 \\ 4 & -1 & 1 & \alpha \\ 1 & 1 & 2 & 1 \\ 1 & 6 & 5 & 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & 1 & 1 & \alpha^2 \\ 0 & -5 & -3 & \alpha - 4\alpha^2 \\ 0 & 0 & 1 & 1 - \alpha^2 \\ 0 & 5 & 4 & 1 - \alpha^2 \end{pmatrix}.$$

Using the multiples $m_{42} = -1$, gives

$$[A|\mathbf{b}] \equiv \begin{pmatrix} 1 & 1 & 1 & \alpha^2 \\ 0 & -5 & -3 & \alpha - 4\alpha^2 \\ 0 & 0 & 1 & 1 - \alpha^2 \\ 0 & 0 & 1 & 1 - 5\alpha^2 + \alpha \end{pmatrix}.$$

Last two equations gives, $1 - \alpha^2 = 1 - 5\alpha^2 + \alpha$ and we have $4\alpha^2 - \alpha = \alpha(4\alpha - 1) = 0$. Thus $\alpha = 0$ or $\alpha = \frac{1}{4}$.
Now let $\alpha = 0$, we have

$$[A|\mathbf{b}] \equiv \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & -5 & -3 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = [U|c].$$

Using backward substitution we get the solution $[x_1, x_2, x_3]^T = [-2/5, -3/5, 1]$. *When* $\alpha = \frac{1}{4}$, *then we have*

$$[A|\mathbf{b}] \equiv \begin{pmatrix} 1 & 1 & 1 & \frac{1}{16} \\ 0 & -5 & -3 & 0 \\ 0 & 0 & 1 & \frac{15}{16} \end{pmatrix} = [U|c].$$

Again using backward substitution we obtain the solution $[x_1, x_2, x_3]^T = [-5/16, -9/16, 15/16]$. •

Example 3.15 *For what values of* α *the following linear system has (i) Unique solution, (ii) No solution, (iii) Infinitely many solutions, by using the simple Gaussian elimination method. Use smallest positive integer value of* α *to get the unique solution of the system.*

$$\begin{array}{rcrcrcl} x_1 & + & 3x_2 & + & \alpha x_3 & = & 4 \\ 2x_1 & - & x_2 & + & 2\alpha x_3 & = & 1 \\ \alpha x_1 & + & 5x_2 & + & x_3 & = & 6 \end{array}$$

Solution. *Using the multiples* $m_{21} = 2$, $m_{31} = \alpha$, *and* $m_{32} = \dfrac{5 - 3\alpha}{-7}$, *gives matrix form*

$$[A|\mathbf{b}] = \begin{pmatrix} 1 & 3 & \alpha & 4 \\ 2 & -1 & 2\alpha & 1 \\ \alpha & 5 & 1 & 6 \end{pmatrix} \equiv \begin{pmatrix} 1 & 3 & \alpha & 4 \\ 0 & -7 & 0 & -7 \\ 0 & 5-3\alpha & 1-\alpha^2 & 6-4\alpha \end{pmatrix} \equiv \begin{pmatrix} 1 & 3 & \alpha & 4 \\ 0 & -7 & 0 & -7 \\ 0 & 0 & 1-\alpha^2 & 1-\alpha \end{pmatrix} = [U|c].$$

So if $1 - \alpha^2 \neq 0$, *then we have the unique solution of the given system while for* $\alpha = \pm 1$, *we have no unique solution. If* $\alpha = 1$, *then we have infinitely many solution because third row of above matrix gives*

$$0x_1 + 0x_2 + 0x_3 = 0,$$

and when $\alpha = -1$, *we have*

$$0x_1 + 0x_2 + 0x_3 = 2,$$

which is not possible, so no solution.
Since we can not take $\alpha = 1$ *for the unique solution, so can take next positive integer* $\alpha = 2$, *which gives us upper-triangular system of the form*

$$\begin{array}{rcrcrcl} x_1 & + & 3x_2 & + & 2x_3 & = & 4 \\ & - & 7x_2 & & & = & -7 \\ & & & - & 3x_3 & = & -1 \end{array}$$

Solving this system using backward substitution, we get, $x_1 = 1/3$, $x_2 = 1$, $x_3 = 1/3$, *the required unique solution of the given system using smallest positive integer value of* α. •

Theorem 3.11 *An upper-triangular matrix A is nonsingular if and only if all its diagonal elements are different from zero.* •

Example 3.16 *Use the simple Gaussian elimination method to find all the values of* α *which make the following matrix singular.*

$$A = \begin{pmatrix} 1 & -1 & \alpha \\ 2 & 2 & 1 \\ 0 & \alpha & -1.5 \end{pmatrix}.$$

Then use the smallest positive integer value of α to find the unique solution of the linear system $A\mathbf{x} = [1, 6, -4]^T$ by simple Gaussian elimination method.

Solution. *Using $m_{21} = 2$ and $m_{32} = \dfrac{\alpha}{4}$, we get*

$$A \equiv \begin{pmatrix} 1 & -1 & \alpha \\ 0 & 4 & 1-2\alpha \\ 0 & \alpha & -1.5 \end{pmatrix} \equiv \begin{pmatrix} 1 & -1 & \alpha \\ 0 & 4 & 1-2\alpha \\ 0 & 0 & -1.5 - \dfrac{\alpha(1-2\alpha)}{4} \end{pmatrix} = U.$$

To show that the given matrix is singular, we have to set the third diagonal element equal to zero (by Theorem 3.11), that is

$$-1.5 - \frac{\alpha(1-2\alpha)}{4} = 0, \quad or \quad 2\alpha^2 - \alpha - 6 = 0.$$

Solving the above quadratic equation, we get, $\alpha = -\dfrac{3}{2}$ and $\alpha = 2$, the possible values of α which make the given matrix singular.
To find the unique solution we take the smallest positive integer value $\alpha = 1$ and using $m_{21} = 2$ and $m_{32} = \dfrac{1}{4}$, gives:

$$[A|\mathbf{b}] \left(\begin{array}{ccc:c} 1 & -1 & 1 & 1 \\ 2 & 2 & 1 & 6 \\ 0 & 1 & -1.5 & -4 \end{array}\right) \equiv \left(\begin{array}{ccc:c} 1 & -1 & 1 & 1 \\ 0 & 4 & -1 & 4 \\ 0 & 1 & -1.5 & -4 \end{array}\right) \equiv \left(\begin{array}{ccc:c} 1 & -1 & 1 & 1 \\ 0 & 4 & -1 & 4 \\ 0 & 0 & -5/4 & -5 \end{array}\right) [U|c].$$

Now expressing the set in algebraic form yields

$$\begin{array}{rcrcrcl} x_1 & - & x_2 & + & x_3 & = & 1 \\ & & 4x_2 & - & x_3 & = & 4 \\ & & & & -5/4x_3 & = & -5 \end{array}$$

Using backward substitution, we get, $x_1 = -1,\ x_2 = 2,\ x_3 = 4$, the unique solution. •

3.4.1.2 Inverse of a Matrix by Simple Gaussian Elimination Method

The inverse of the nonsingular matrix A can be easily determined by using the simple Gaussian elimination method. Here, we have to consider the augmented matrix as a combination of the given matrix A and the identity matrix $\mathbf{I}$ (same size as of A). To find the inverse matrix $B = A^{-1}$ we must solve the linear system in which the *jth* column of the matrix B is the solution of the linear system with right-hand side the *jth* column of the matrix $\mathbf{I}$.

Example 3.17 *Use the simple Gaussian elimination method to find the inverse of the following matrix*

$$A = \begin{pmatrix} 2 & -1 & 3 \\ 4 & -1 & 6 \\ 2 & -3 & 4 \end{pmatrix}.$$

Then use A^{-1} to find the unique solution of the system $A\mathbf{x} = [1, 2, 2]^T$.

Solution. *Suppose that the inverse $A^{-1} = B$ of the given matrix exists and let*

$$AB = \begin{pmatrix} 2 & -1 & 3 \\ 4 & -1 & 6 \\ 2 & -3 & 4 \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I}.$$

Now to find the elements of the matrix B, we apply the simple Gaussian elimination on the augmented matrix

$$[A|\mathbf{I}] = \left(\begin{array}{ccc:ccc} 2 & -1 & 3 & 1 & 0 & 0 \\ 4 & -1 & 6 & 0 & 1 & 0 \\ 2 & -3 & 4 & 0 & 0 & 1 \end{array}\right).$$

Using $m_{21} = \dfrac{4}{2} = 2$, $m_{31} = \dfrac{2}{2} = 1$, and $m_{32} = \dfrac{-2}{1} = -2$, gives

$$\left(\begin{array}{ccc:ccc} 2 & -1 & 3 & 1 & 0 & 0 \\ 0 & 1 & 0 & -2 & 1 & 0 \\ 0 & -2 & 1 & -1 & 0 & 1 \end{array}\right) \equiv \left(\begin{array}{ccc:ccc} 2 & -1 & 3 & 1 & 0 & 0 \\ 0 & 1 & 0 & -2 & 1 & 0 \\ 0 & 0 & 1 & -5 & 2 & 1 \end{array}\right).$$

We solve the first system

$$\begin{pmatrix} 2 & -1 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ -5 \end{pmatrix},$$

by using backward substitution, we get

$$\begin{array}{rcrcrcr} 2b_{11} & - & b_{21} & + & 3b_{31} & = & 1 \\ & & b_{21} & & & = & -2 \\ & & & & b_{31} & = & -5 \end{array}$$

which gives $b_{11} = 7,\ b_{21} = -2,\ b_{31} = -5$. Similarly, the solution of the second linear system

$$\begin{pmatrix} 2 & -1 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_{12} \\ b_{22} \\ b_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix},$$

can be obtained as follows:

$$\begin{array}{rcrcrcr} 2b_{12} & - & b_{22} & + & 3b_{32} & = & 0 \\ & & b_{22} & & & = & 1 \\ & & & & b_{32} & = & 2 \end{array}$$

which gives $b_{12} = -5/2,\ b_{22} = 1,\ b_{32} = 2$. Finally, the solution of the third linear system

$$\begin{pmatrix} 2 & -1 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_{13} \\ b_{23} \\ b_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

can be obtained as follows:

$$\begin{array}{rcrcrcl} 2b_{13} & - & b_{23} & + & 3b_{33} & = & 0 \\ & & b_{23} & & & = & 0 \\ & & & & b_{33} & = & 1 \end{array}$$

and it gives $b_{13} = -3/2,\ b_{23} = 0,\ b_{33} = 1$. *Hence the elements of the inverse matrix* B *are*

$$B = A^{-1} = \begin{pmatrix} 7 & -5/2 & -3/2 \\ -2 & 1 & 0 \\ -5 & 2 & 1 \end{pmatrix},$$

which is the required inverse of the given matrix A. *Now to find the solution of the system, we do as*

$$\mathbf{x} = A^{-1}\mathbf{b} = \begin{pmatrix} 7 & -5/2 & -3/2 \\ -2 & 1 & 0 \\ -5 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix},$$

the required solution. •

Procedure 3.1 [Gaussian Elimination Method]

1. *Form the augmented matrix,* $B = [A|\mathbf{b}]$.
2. *Check first pivot element* $a_{11} \neq 0$, *then move to the next step; otherwise, interchange rows so that* $a_{11} \neq 0$.
3. *Multiply row one by multiplier* $m_{i1} = \dfrac{a_{i1}}{a_{11}}$ *and subtract to the ith row for* $i = 2, 3, \ldots, n$.
4. *Repeat the steps 2 and 3 for the remaining pivots elements unless coefficient matrix* A *becomes upper-triangular matrix* U.
5. *Use backward substitution to solve* x_n *from the nth equation* $x_n = \dfrac{b_n^{n-1}}{a_{nn}}$ *and solve the other (n-1) unknowns variables by using (3.19).*

The use of non-zero pivots is sufficient for the theoretical correctness of the simple Gaussian elimination, but more care must be taken if one is to obtain reliable results.

Example 3.18 *Consider a linear system*

$$\begin{array}{rcrcl} 0.0002x_1 & + & 1.471x_2 & = & 1.473 \\ 0.2346x_1 & - & 1.317x_2 & = & 1.029 \end{array}$$

which has exact solution $\mathbf{x} = [10.0, 1.0]^T$. *Now we solve this system by the simple Gaussian elimination. The first elimination step is to eliminate first variable* x_1 *from second equation by subtracting multiple* $m_{21} = \dfrac{0.2346}{0.0002} = 1173$ *of first equation from second equation, gives*

$$\begin{array}{rcrcl} 0.0002x_1 & + & 1.471x_2 & = & 1.473 \\ & - & 1726x_2 & = & -1727 \end{array}$$

Using backward substitution to get the solution $\mathbf{x}^* = [5.0, 1.001]^T$. *Thus a computational disaster has occurred. But if we interchange the equations, we obtain*

$$\begin{aligned} 0.2346x_1 &- 1.317x_2 &= 1.029 \\ 0.0002x_1 &+ 1.471x_2 &= 1.473 \end{aligned}$$

Apply the Gaussian elimination again, and we got the solution $\mathbf{x}^* = [10.0, 1.0]^T$. *This solution is as good as one would hope. So, we conclude from this example that it is not enough just to avoid zero pivot, one must also avoid relatively small one.* •

Here we need some pivoting strategies which help us to over come these difficulties facing during the process of simple Gaussian elimination.

3.4.1.3 Pivoting Strategies Using Gaussian Elimination Method

Since we know that simple Gaussian elimination is applied to a problem with no pivotal elements zero. However, the method does not work if the first coefficient of the first equation or if a diagonal coefficients becomes zero in the process of solution because they are used as denominators in a forward elimination.
Pivoting is used to change sequential order of the equations for two purposes, first to prevent diagonal coefficients from becoming zero, and second, to make each diagonal coefficient larger in magnitude than any other coefficient below it, that is, to decrease the round-off errors. The equations are not mathematical affected by changes of the sequential order, but changing the order makes coefficient become non-zero. Even when all diagonal coefficients are non-zero, the changes of order increases accuracy of the computations. The standard pivoting strategy which handled these difficulties easily are explained below.

3.4.1.4 Partial Pivoting

Here we develop an implementation of the Gaussian elimination which utilizes the pivoting strategy discussed above. In using the Gaussian elimination by partial pivoting(or row pivoting), the basic approach is to use the largest (in absolute value) element on or below the diagonal in the column of current interest as the pivotal element for elimination in the rest of that column.
One immediate effect of this will be to force all the multiples used to be not greater than 1 in absolute value. This will inhibit the growth of error in the rest of elimination phase and in subsequent backward substitution.
At stage k of forward elimination, it is necessary, therefore, to be able to identify the largest element from $|a_{kk}|, |a_{k+1,k}|, \ldots, |a_{nk}|$, where these a_{ik}'s are the elements in the current partially triangularized coefficient matrix. If this maximum occurs in row p, then *pth* and *kth* rows of the augmented matrix are interchange and the elimination proceed as usual. In solving n linear equations, a total of $N = \dfrac{n(n+1)}{2}$ coefficients must be examined.

Example 3.19 *Solve the following system using the Gaussian elimination with partial pivoting*

$$\begin{aligned} x_1 &+ x_2 &+ x_3 &= 1 \\ 2x_1 &+ 3x_2 &+ 4x_3 &= 3 \\ 4x_1 &+ 9x_2 &+ 16x_3 &= 11 \end{aligned}$$

Solution. *For the first elimination step, since 4 is the largest absolute coefficient of first variable x_1, therefore, the first row and the third row are interchange, giving us*

$$\begin{array}{rcrcrcr} 4x_1 & + & 9x_2 & + & 16x_3 & = & 11 \\ 2x_1 & + & 3x_2 & + & 4x_3 & = & 3 \\ x_1 & + & x_2 & + & x_3 & = & 1 \end{array}$$

Eliminate first variable x_1 from the second and third rows by subtracting the multiples $m_{21} = \frac{2}{4}$ and $m_{31} = \frac{1}{4}$ of row 1 from row 2 and row 3 respectively, gives

$$\begin{array}{rcrcrcr} 4x_1 & + & 9x_2 & + & 16x_3 & = & 11 \\ & - & 3/2x_2 & - & 4x_3 & = & -5/2 \\ & - & 5/4x_2 & - & 3x_3 & = & -7/4 \end{array}$$

For the second elimination step, $-3/2$ is the largest absolute coefficient of second variable x_2, so eliminate second variable x_2 from the third row by subtracting the multiple $m_{32} = \frac{5}{6}$ of row 2 from row 3, gives

$$\begin{array}{rcrcrcr} 4x_1 & + & 9x_2 & + & 16x_3 & = & 11 \\ & - & 3/2x_2 & - & 4x_3 & = & -5/2 \\ & & & & 1/3x_3 & = & 1/3 \end{array}$$

Obviously, the original set of equations has been transformed to an equivalent upper-triangular form. Now using backward substitution, gives, $x_1 = 1,\ x_2 = -1,\ x_3 = 1$, the required solution. •

The following MATLAB commands will gives the same results as we obtained in the preceding Example 3.19 of the Gaussian elimination method with partial pivoting:

```
>> B = [1 1 1 1;2 3 4 3;4 9 16 11]; x = PP(B); disp(x)
```

Program 3.6

MATLAB m-file for Gaussian Elimination by Partial Pivoting

```
function x=PP(B)
% B = input('input matrix in form[A/b]');
[n,t] = size(B); U = B; for M = 1:n-1; mx(M) = abs(U(M,M)); r = M;
for i = M+1:n; if mx(M) < abs(U(i,M)); mx(M)=abs(U(i,M)); r = i; end; end
rw1(1,1:t)=U(r,1:t); rw2(1,1:t)=U(M,1:t); U(M,1:t)=rw1 ; U(r,1:t)=rw2; for k=M+1:n
m=U(k,M)/U(M,M); for j=M:t; U(k,j) = U(k,j) - m * U(M,j); end;end;
i=n; x(i)=U(i,t)/U(i,i); for i=n-1:-1:1; s=0; for k=n:-1:i+1;
s = s + U(i,k) * x(k); end; x(i)=(U(i,t)-s)/U(i,i); end; B; U; x; end
```

Procedure 3.2 [Partial Pivoting]

1. *Suppose we are about to work on the ith column of the matrix. Then we search that portion of the ith column below and including the diagonal, and find the element that has the largest absolute value. Let p denote the index of the row that contains this element.*

2. Interchange row i and p.

3. Proceed with the elimination Procedure 3.1.

3.4.1.5 Inverse of a Matrix by Gauss Elimination with Partial Pivoting

Example 3.20 *Compute the inverse A^{-1}, where*

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 4 & 1 & 2 \end{pmatrix},$$

by solving the system $AB = \mathbf{I}$, using Gauss elimination by partial pivoting where $B = A^{-1}$. Then use it to solve the system $A\mathbf{x} = [1, 1, 2]^T$.

Solution. *Suppose that the inverse $A^{-1} = B$ of the given matrix exists and let*

$$AB = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 2 & 3 \\ 4 & 1 & 2 \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I}.$$

Now to find the elements of the matrix B, we apply the Gaussian elimination using partial pivoting on the augmented matrix

$$[A|\mathbf{I}] = \left(\begin{array}{ccc:ccc} 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 2 & 3 & 0 & 1 & 0 \\ 4 & 1 & 2 & 0 & 0 & 1 \end{array}\right).$$

For the first elimination step, since 4 is the largest absolute coefficient of first variable x_1, therefore, the first row and the third row are interchange, giving us

$$\equiv \left(\begin{array}{ccc:ccc} 4 & 1 & 2 & 0 & 0 & 1 \\ 1 & 2 & 3 & 0 & 1 & 0 \\ 2 & 1 & 2 & 1 & 0 & 0 \end{array}\right).$$

Using all three possible multiples $m_{21} = \dfrac{1}{4}$, $m_{31} = \dfrac{1}{2}$ and $m_{32} = \dfrac{4}{14} = \dfrac{2}{7}$, gives

$$\equiv \left(\begin{array}{ccc:ccc} 4 & 1 & 2 & 0 & 0 & 1 \\ 0 & 7/4 & 5/2 & 0 & 1 & -1/4 \\ 0 & 1/2 & 1 & 1 & 0 & -1/2 \end{array}\right) \equiv \left(\begin{array}{ccc:ccc} 4 & 1 & 2 & 0 & 0 & 1 \\ 0 & 7/4 & 5/2 & 0 & 1 & -1/4 \\ 0 & 0 & 2/7 & 1 & -2/7 & -3/7 \end{array}\right) = [U|C].$$

Obviously, the original set of equations has been transformed to an equivalent upper-triangular form. We solve the first upper triangular linear system

$$[U|\mathbf{b}_1] = \begin{pmatrix} 4 & 1 & 2 \\ 0 & 7/4 & 5/2 \\ 0 & 0 & 2/7 \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{21} \\ b_{31} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = c_1,$$

by using backward substitution, we get

$$\begin{array}{rcrcrcl} 4b_{11} & + & b_{21} & + & 2b_{31} & = & 0 \\ & & 7/4b_{21} & + & 5/2b_{31} & = & 0 \\ & & & & 2/7b_{31} & = & 1 \end{array}$$

which gives, $\mathbf{b_1} = [b_{11}, b_{21}, b_{31}]^T = [-0.5, -5.0, 3.5]^T$. *Similarly, the solution of the second upper triangular linear system*

$$[U|\mathbf{b}_2] = \begin{pmatrix} 4 & 1 & 2 \\ 0 & 7/4 & 5/2 \\ 0 & 0 & 2/7 \end{pmatrix} \begin{pmatrix} b_{12} \\ b_{22} \\ b_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -2/7 \end{pmatrix} = c_2,$$

can be obtained by using backward substitution, as

$$\begin{array}{rcrcrcr} 4b_{12} & + & b_{22} & + & 2b_{32} & = & 0 \\ & & 7/4b_{22} & + & 5/2b_{32} & = & 1 \\ & & & & 2/7b_{32} & = & -2/7 \end{array}$$

which gives, $\mathbf{b_2} = [b_{12}, b_{22}, b_{32}]^T = [0.0, 2.0, -1.0]^T$. *Finally, the solution of the third upper triangular linear system*

$$[U|\mathbf{b}_3] = \begin{pmatrix} 4 & 1 & 2 \\ 0 & 7/4 & 5/2 \\ 0 & 0 & 2/7 \end{pmatrix} \begin{pmatrix} b_{13} \\ b_{23} \\ b_{33} \end{pmatrix} = \begin{pmatrix} 1 \\ -1/4 \\ -3/7 \end{pmatrix} = c_3,$$

can be obtained by using backward substitution, as

$$\begin{array}{rcrcrcr} 4b_{13} & + & b_{23} & + & 2b_{33} & = & 1 \\ & & 7/4b_{23} & + & 5/2b_{33} & = & -1/4 \\ & & & & 2/7b_{33} & = & -3/7 \end{array}$$

which gives, $\mathbf{b_3} = [b_{13}, b_{23}, b_{33}]^T = [0.5, 2.0, -1.5]^T$. *Hence the elements of the inverse matrix* B *are*

$$B = A^{-1} = \begin{pmatrix} -0.5 & 0 & 0.5 \\ -5 & 2 & 2 \\ 3.5 & -1 & -1.5 \end{pmatrix},$$

which is the required inverse of the given matrix A..
Thus using A^{-1}, *we can solve the linear system as*

$$\mathbf{x} = A^{-1}\mathbf{b} = \begin{pmatrix} -0.5 & 0 & 0.5 \\ -5 & 2 & 2 \\ 3.5 & -1 & -1.5 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \\ -0.5 \end{pmatrix},$$

gives the required solution of the given system. •

3.4.2 Gauss-Jordan Method

This method is a modification of the Gaussian elimination method. The Gauss-Jordan method is although inefficient for practical calculation but is often useful for theoretical purposes. The basic of this method is to convert the given matrix into a diagonal form. The forward elimination of the Gauss-Jordan method is identical to that of Gaussian elimination method. However, Gauss-Jordan elimination uses backward elimination rather than backward substitution. In the Gauss-Jordan method the forward elimination and backward elimination need not be separated. This is possible because a pivot element can be used to eliminate the coefficients not only below but also above at the same time. If this approach is taken, the form of the coefficients matrix become diagonal when elimination by the last pivot are completed. The Gauss-Jordan method simply yields a transformation of the augmented matrix of the form

$$[A|\mathbf{b}] \to [\mathbf{I}|\mathbf{c}],$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{c}$ is the column matrix, which represents the possible solution of the given linear system.
The Gauss-Jordan method particularly well suited to compute the inverse of a matrix through the transformation

$$[A|\mathbf{I}] \to [\mathbf{I}|A^{-1}].$$

Note if the inverse of the matrix can be found, then the solution of the linear system can be computed easily from the product of matrix A^{-1} and column matrix $\mathbf{b}$, that is

$$\mathbf{x} = A^{-1}\mathbf{b}. \tag{3.24}$$

Note that one can get easily the solution of linear system $A\mathbf{x} = \mathbf{b}$ and inverse of the coefficient matrix A together by the Gauss-Jordan method using the augmented matrix of the form

$$[A|\mathbf{b}|\mathbf{I}] \to [\mathbf{I}|\mathbf{x}|A^{-1}].$$

Example 3.21 *Apply the Gauss-Jordan method to find the inverse of the coefficient matrix and also the solution of the linear system $A\mathbf{x} = \mathbf{b}$, where*

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 3 \end{pmatrix} \quad \textit{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Solution. *Consider the following augmented matrix*

$$[A|\mathbf{b}|\mathbf{I}] = \left(\begin{array}{cc:c:cc:cc} 1 & 2 & 1 & 1 & 0 \\ 1 & 3 & 2 & 0 & 1 \end{array}\right).$$

Then we have

$$\equiv \left(\begin{array}{cc:c:cc:cc} 1 & 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & -1 & 1 \end{array}\right) \equiv \left(\begin{array}{cc:c:cc:cc} 1 & 0 & -1 & 3 & -2 \\ 0 & 1 & 1 & -1 & 1 \end{array}\right).$$

Thus we obtain the inverse of the matrix A

$$A^{-1} = \begin{pmatrix} 3 & -2 \\ -1 & 1 \end{pmatrix},$$

and $x_1 = -1,\ x_2 = 1$, *the solution of the given system.* •

The above results can be obtained using MATLAB commands, we do the following:

```
>> Ab = [A|b|I] = [1 2 1 1 0;1 3 2 0 1]; [I|inv(A)] = GaussJ(Ab);
```

3.4.3 LU Decomposition Method

This is another direct method to find the solution of the system of linear equations. The LU decomposition (or factorization method) is a modification of the elimination method. Here we decompose or factorize the coefficient matrix A into the product of two triangular matrices in the form

$$A = LU, \tag{3.25}$$

where L is a lower-triangular matrix and U is the upper-triangular matrix. Both are of same size as the coefficients matrix A. To solve a number of linear equations sets in which the coefficients matrices are all identical but the right-hand side are different, then the LU decomposition is more efficient than elimination method. Specifying the diagonal elements of either L and U makes the factoring unique. The procedure based on unity elements on the diagonal of matrix L is called *Doolittle's method*(or Gauss factorization), while the procedure based on unity elements on the diagonal of matrix U is called *Crout's method*.
The general forms of L and U are written as

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}, \tag{3.26}$$

such that $l_{ij} = 0$ for $i < j$ and $u_{ij} = 0$ for $i > j$.

Consider a linear system

$$A\mathbf{x} = \mathbf{b}, \tag{3.27}$$

and let A be factored into the product of L and U, as shown by (3.26). Then the linear system (3.27) becomes

$$LU\mathbf{x} = \mathbf{b},$$

which can be written as

$$L\mathbf{y} = \mathbf{b}, \quad \text{where} \quad \mathbf{y} = U\mathbf{x}.$$

The unknown elements of matrix L and matrix U are computed by equating corresponding elements in matrices A and LU in a systematic way. Once the matrices L and U have been constructed, the solution of system (3.27) can be computed in the following two steps:

1. Solve the lower-triangular system $L\mathbf{y} = \mathbf{b}$.

   By using the *forward elimination*, we will find the components of the unknown vector $\mathbf{y}$, by using the following steps:

$$\left.\begin{aligned} y_1 &= b_1, \\ y_i &= b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \qquad i = 2, 3, \ldots, n \end{aligned}\right\}. \tag{3.28}$$

2. Solve the upper-triangular system $U\mathbf{x} = \mathbf{y}$.

   By using the *backward substitution*, we will find the components of the unknown vector $\mathbf{x}$, by using the following steps:

$$\left.\begin{aligned} x_n &= \frac{y_n}{u_{nn}}, \\ x_i &= \frac{1}{u_{ii}}\left[y_i - \sum_{j=i+1}^{n} u_{ij} x_j\right], \quad i = n-1, n-2, \ldots, 1 \end{aligned}\right\}. \tag{3.29}$$

Thus the relationship of the matrices L and U to the original matrix A is given by the following theorem.

Theorem 3.12 *If the Gaussian elimination can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ without row interchanges, then the matrix A can be factored into the product of a lower-triangular matrix L and an upper-triangular matrix U, that is*

$$A = LU,$$

where the matrices L and U are of the same size as A. •

Theorem 3.13 *Let A be an $n \times n$ matrix that has an LU factorization, that is*

$$A = LU.$$

If A has rank n (that is, all pivots are non-zeros), then L and U are uniquely determined by A. •

Now we discuss the two possible variations of the LU decomposition to find the solution of the nonsingular linear system in the following.

3.4.3.1 Doolittle's Method

In Doolittle's method(which is also called the Gauss factorization), the upper-triangular matrix U is obtained by forward elimination of the Gaussian elimination method and the lower-triangular matrix L containing the multiples used in the Gaussian elimination process as the elements below the diagonal with unity elements on the main diagonal, that is,

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

Example 3.22 *Construct the LU decomposition of the following matrix A by using the Gauss factorization (that is, the LU decomposition by Doolittle's method):*

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 3 \\ 1 & 3 & 4 \end{pmatrix}.$$

Solution. *Applying the forward elimination step of Simple Gauss-elimination to the given matrix*

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 3 \\ 1 & 3 & 4 \end{pmatrix},$$

using the multiples $m_{21} = 2 = l_{21}$, $m_{31} = 1 = l_{31}$, and $m_{32} = 1 = l_{32}$, we obtain

$$\equiv \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 3 \end{pmatrix} \equiv \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} = U.$$

Hence we obtained the LU-decomposition of the given matrix as follows

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 5 & 3 \\ 1 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix},$$

where the unknown elements of matrix L are the used multiples and the matrix U is same as we obtained in forward elimination process. •

Example 3.23 *Construct the LU decomposition of the following matrix A by using the Gauss factorization (that is, the LU decomposition by Doolittle's method). Find the value(s) of α for which the following matrix*

$$A = \begin{pmatrix} 1 & -1 & \alpha \\ -1 & 2 & -\alpha \\ \alpha & 1 & 1 \end{pmatrix},$$

is singular. Also, find the unique solution of the linear system $A\mathbf{x} = [1, 1, 2]^T$ by using the smallest positive integer value of α.

Solution. *Since we know that*

$$A = \begin{pmatrix} 1 & -1 & \alpha \\ -1 & 2 & -\alpha \\ \alpha & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = LU.$$

Using $m_{21} = -1 = l_{21}$, $m_{31} = \alpha = l_{31}$, and $m_{32} = \dfrac{1+\alpha}{1} = l_{32}$, gives

$$A \equiv \begin{pmatrix} 1 & -1 & \alpha \\ 0 & 1 & 0 \\ 0 & 1+\alpha & 1-\alpha^2 \end{pmatrix} \equiv \begin{pmatrix} 1 & -1 & \alpha \\ 0 & 1 & 0 \\ 0 & 0 & 1-\alpha^2 \end{pmatrix} = U.$$

Obviously, the original set of equations has been transformed to an upper-triangular form. Thus

$$A = \begin{pmatrix} 1 & -1 & \alpha \\ -1 & 2 & -\alpha \\ \alpha & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ \alpha & 1+\alpha & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & \alpha \\ 0 & 1 & 0 \\ 0 & 0 & 1-\alpha^2 \end{pmatrix} = LU,$$

which is the required decomposition of A. The matrix will be singular if the third diagonal element $1-\alpha^2$ *of the upper-triangular U is equal to zero (Theorem 3.11), gives,* $\alpha = \pm 1$.
To find the unique solution of the given system we take $\alpha = 2$ *and it gives*

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

Write MATLAB m-file **Decompd**.*m to factored a nonsingular matrix A into a unit lower triangular matrix L and an upper triangular matrix U and using the following MATLAB commands as:*

```
>> A = [1 1 -2; -1 2 -2; 2 1 1]; Sol = Decomp(A);
```

Program 3.7
MATLAB m-file for Decomposition of Matrix

```
function Sol = Decomp(A)
[n,n] = size(A); U=A; L=eye(n); for i=1:n; for k = i=1:n; L(k,i)=U(k,i)/U(i,i);
U(k,i:n)=U(k,i:n)-U(i,i:n)*L(k,i);end;end; Sol=[L,U]; detA = prod(diag(U))
```

Now solving the lower-triangular system $L\mathbf{y} = \mathbf{b}$ *for unknown vector* $\mathbf{y}$, *that is*

$$L\mathbf{y} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = \mathbf{b}.$$

Performing forward substitution yields

$$\begin{array}{rcrcrcll} y_1 & & & & & = & 1, & \textit{gives} \quad y_1 = 1, \\ -y_1 & + & y_2 & & & = & 1, & \textit{gives} \quad y_2 = 2, \\ 2y_1 & + & 3y_2 & + & y_3 & = & 2, & \textit{gives} \quad y_3 = -6. \end{array}$$

Using the m-file ForwardSubs.m

```
function y = ForwardSubs(L, b)
[n, n] = size(L); y = zeros(n, 1);
for k = 1 : n; y(k)=(b(k)-L(k,1:k-1)*y(1:k-1))/L(k, k) end
```

and the following MATLAB commands to generate the solution of lower-triangular system as:

```
>> L = [1 0 0; -1 1 0; 2 3 1]; b = [1; 1; 2]; sol = ForwardSubs(L, b);
```

Then solving the upper-triangular system $U\mathbf{x} = \mathbf{y}$ *for unknown vector* $\mathbf{x}$*, that is*

$$U\mathbf{x} = \begin{pmatrix} 1 & -1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ -6 \end{pmatrix} = \mathbf{y}.$$

Performing backward substitution yields

$$\begin{array}{rcrcrcrcl} x_1 & - & x_2 & + & 2x_3 & = & 1, & gives & x_1 = -1, \\ & & x_2 & & & = & 2, & gives & x_2 = 2, \\ & & & - & 3x_3 & = & -6, & gives & x_3 = 2, \end{array}$$

the approximate solution of the given system. •

Using the m-file BackwardSubs.m

```
functionx = BackwardSubs(U, y)
[n, n] = size(U); x = zeros(n, 1); x(n) = y(n)/U(n, n);
fork = n − 1 : −1 : 1; x(k) = (y(k) − U(k, k + 1 : n) ∗ x(k + 1 : n))/U(k, k); end
```

and the following MATLAB commands to generate the solution of upper-triangular system as:

```
>> U = [1 −1 2; 0 1 0; 0 0 −3]; y = [1; 1; 2]; sol = BackwardSubs(U, y);
```

There is an other way to find the values of the unknown elements of the matrices L and U, which we describe in the following example.

Example 3.24 *Construct the LU decomposition of the following matrix using Doolittle's method*

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix}.$$

Solution. *Since*

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Performing the multiplication on the right-hand side, gives

$$\begin{pmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{pmatrix}.$$

Then equating elements of first column to obtain

$$\begin{array}{rcrcl} 1 & = & u_{11}, & u_{11} & = 1, \\ 1 & = & l_{21}u_{11}, & l_{21} & = 1, \\ 2 & = & l_{31}u_{11}, & l_{31} & = 2. \end{array}$$

Now equating elements of second column to obtain

$$\begin{array}{rclrcl} 2 &=& u_{12}, & u_{12} &=& 2, \\ 3 &=& l_{21}u_{12} + u_{22}, & u_{22} &=& 3 - 2 = 1, \\ 2 &=& l_{31}u_{12} + l_{32}u_{22}, & l_{32} &=& 2 - 4 = -2. \end{array}$$

Finally, equating elements of third column to obtain

$$\begin{array}{rclrcl} 4 &=& u_{13}, & u_{13} &=& 4, \\ 3 &=& l_{21}u_{13} + u_{23}, & u_{23} &=& 3 - 4 = -1, \\ 2 &=& l_{31}u_{13} + l_{32}u_{23} + u_{33}, & u_{33} &=& 2 - 10 = -8. \end{array}$$

Thus we obtain

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & -1 \\ 0 & 0 & -8 \end{pmatrix} = LU,$$

the factorization of the given matrix. •

The general formula for getting elements of L and U corresponding to the coefficient matrix A for a set of n linear equations can be written as

$$\left.\begin{array}{rclll} u_{ij} &=& a_{ij} - \displaystyle\sum_{k=1}^{i-1} l_{ik}u_{kj}, & & 2 \le i \le j \\ l_{ij} &=& \dfrac{1}{u_{ii}}\left[a_{ij} - \displaystyle\sum_{k=1}^{j-1} l_{ik}u_{kj}\right], & & i > j \ge 2 \\ u_{ij} &=& a_{1j}, & & i = 1 \\ l_{ij} &=& \dfrac{a_{i1}}{u_{11}} = \dfrac{a_{i1}}{a_{11}}, & & j = 1 \end{array}\right\}. \tag{3.30}$$

Example 3.25 *Solve the following linear system by LU decomposition using Doolittle's method*

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} \quad \textit{and} \quad \mathbf{b} = \begin{pmatrix} -2 \\ 3 \\ -6 \end{pmatrix}.$$

Solution. *Since the factorization of the coefficient matrix A has been already constructed in the Example 3.24 as*

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & -1 \\ 0 & 0 & -8 \end{pmatrix} = LU.$$

Then solving the first system $L\mathbf{y} = \mathbf{b}$ *for unknown vector* $\mathbf{y}$*, that is*

$$L\mathbf{y} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} -2 \\ 3 \\ -6 \end{pmatrix} = \mathbf{b}.$$

Performing forward substitution yields

$$\begin{array}{rcrcrcrcr} y_1 & & & & & = & -2, & gives & y_1 = -2, \\ y_1 & + & y_2 & & & = & 3, & gives & y_2 = 5, \\ 2y_1 & - & 2y_2 & + & y_3 & = & -6, & gives & y_3 = 8. \end{array}$$

Then solving the second system $U\mathbf{x} = \mathbf{y}$ *for unknown vector* $\mathbf{x}$*, that is*

$$U\mathbf{x} = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & -1 \\ 0 & 0 & -8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -2 \\ 5 \\ 8 \end{pmatrix} = \mathbf{y}.$$

Performing backward substitution yields

$$\begin{array}{rcrcrcrcr} x_1 & + & 2x_2 & + & 4x_3 & = & -2, & gives & x_1 = -6, \\ & & x_2 & - & x_3 & = & 5, & gives & x_2 = 4, \\ & & & - & 8x_3 & = & 8, & gives & x_3 = -1, \end{array}$$

the approximate solution of the given system. •

We can also write MATLAB m-file called, Doolittle.m to get the solution of the linear system by LU decomposition by using Doolittle's method. In order to reproduce above results using MATLAB commands, we do the following:

```
>> A = [1 2 4; 1 3 3; 2 2 2]; b = [-2 3 -6]; sol = Doolittle(A,b);
```

Program 3.8

MATLAB m-file for using the Doolittle's Method

```
function sol = Doolittle(A,b)
[n,n]=size(A); u=A;l=zeros(n,n);
for i=1:n-1; if abs(u(i,i))> 0;  for i1=i+1:n; m(i1,i)=u(i1,i)/u(i,i);
for j=1:n;  u(i1,j) = u(i1,j) - m(i1,i) * u(i,j);end;end;end;end
for i=1:n; l(i,1)=A(i,1)/u(1,1); end;  for j=2:n; for i=2:n; s=0;
for k=1:j-1; s = s + l(i,k) * u(k,j); end
l(i,j)=(A(i,j)-s)/u(j,j); end; end y(1)=b(1)/l(1,1);
for k=2:n; sum=b(k);  for i=1:k-1; sum = sum - l(k,i) * y(i); end
y(k)=sum/l(k,k); end;  x(n)=y(n)/u(n,n);  for k=n-1:-1:1; sum=y(k);
for i=k+1:n; sum = sum - u(k,i) * x(i); end;  x(k)=sum/u(k,k);end; l; u; y; x
```

Procedure 3.3 [LU Decomposition by Doolittle's Method]

1. *Take the nonsingular matrix, A.*
2. *If possible, decompose the matrix $A = LU$ using (3.30).*
3. *Solve linear system $L\mathbf{y} = \mathbf{b}$ using (3.28).*
4. *Solve linear system $U\mathbf{x} = \mathbf{y}$ using (3.29).*

Example 3.26 *Which of the following matrices cannot be factored as $A = LU$ using Doolittle's method*

$$A = \begin{pmatrix} 1 & 2 & 6 \\ 4 & 8 & -1 \\ -2 & 3 & 5 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 6 \\ -2 & 3 & 5 \\ 4 & 8 & -1 \end{pmatrix}.$$

Solution. *Since*

$$A = LU = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Performing the multiplication on the right-hand side, gives

$$\begin{pmatrix} 1 & 2 & 6 \\ 4 & 8 & -1 \\ -2 & 3 & 5 \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{pmatrix}.$$

Then equating elements of first column to obtain

$$\begin{array}{rclrcl} 1 &=& u_{11}, & u_{11} &=& 1, \\ 4 &=& l_{21}u_{11}, & l_{21} &=& 4, \\ -2 &=& l_{31}u_{11}, & l_{31} &=& -2. \end{array}$$

Now equating elements of second column to obtain

$$\begin{array}{rclrcl} 2 &=& u_{12}, & u_{12} &=& 2, \\ 8 &=& l_{21}u_{12} + u_{22}, & u_{22} &=& 8 - 8 = 0, \\ 3 &=& l_{31}u_{12} + l_{32}u_{22}, & 3 &=& (-2)(2) + l_{32}(0) = -4, \end{array}$$

which is a contradiction. Therefore, A does not have a LU factorization.
Now consider

$$B = LU = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}.$$

Performing the multiplication on the right-hand side, gives

$$\begin{pmatrix} 1 & 2 & 6 \\ -2 & 3 & 5 \\ 4 & 8 & -1 \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} \end{pmatrix}.$$

Then equating elements of first column to obtain

$$\begin{array}{rclrcl} 1 &=& u_{11}, & u_{11} &=& 1, \\ -2 &=& l_{21}u_{11}, & l_{21} &=& -2, \\ 4 &=& l_{31}u_{11}, & l_{31} &=& 4. \end{array}$$

Now equating elements of second column to obtain

$$\begin{array}{rclrcl} 2 &=& u_{12}, & u_{12} &=& 2, \\ 3 &=& l_{21}u_{12}+u_{22}, & u_{22} &=& 3+4=7, \\ 8 &=& l_{31}u_{12}+l_{32}u_{22}, & l_{32} &=& 0, \end{array}$$

Finally, equating elements of third column to obtain

$$\begin{array}{rclrcl} 6 &=& u_{13}, & u_{13} &=& 6, \\ 5 &=& l_{21}u_{13}+u_{23}, & u_{23} &=& 5+12=17, \\ -1 &=& l_{31}u_{13}+l_{32}u_{23}+u_{33}, & u_{33} &=& -1-24=-25. \end{array}$$

Thus the matrix B

$$B=\begin{pmatrix} 1 & 2 & 6 \\ -2 & 3 & 5 \\ 4 & 8 & -1 \end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 6 \\ 0 & 7 & 17 \\ 0 & 0 & -25 \end{pmatrix}=LU,$$

has LU factorization. •

Example 3.27 *Use LU-factorization method with Doolittle's method ($l_{ii}=1$) to find values of α for which the following linear system has unique solution and infinitely many solutions. Write down the solution for both cases.*

$$\begin{array}{rcrcrcl} x_1 &+& 0.5x_2 &+& \alpha x_3 &=& 0.5 \\ 2x_1 &-& 3x_2 &+& x_3 &=& -1 \\ -x_1 &-& 1.5x_2 &+& 2.5x_3 &=& -1 \end{array}$$

Solution. *We use Simple Gauss-elimination method to convert the following matrix of the given system by using the multiples $m_{21}=2, m_{31}=-1$ and $m_{32}=1/4$,*

$$A=\begin{pmatrix} 1 & 0.5 & \alpha \\ 2 & -3 & 1 \\ -1 & -1.5 & 2.5 \end{pmatrix},$$

into equivalent an upper-triangular matrix form

$$A\equiv\begin{pmatrix} 1 & 0.5 & \alpha \\ 0 & -4 & 1-2\alpha \\ 0 & 0 & 1.5\alpha+2.25 \end{pmatrix}=U,$$

to get LU-factorization of A in the following form

$$A=\begin{pmatrix} 1 & 0.5 & \alpha \\ 2 & -3 & 1 \\ -1 & -1.5 & 2.5 \end{pmatrix}=\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0.25 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0.5 & \alpha \\ 0 & -4 & 1-2\alpha \\ 0 & 0 & 1.5\alpha+2.25 \end{pmatrix}=LU.$$

Then by solving the lower-triangular system of the form $L\mathbf{y}=[0.5,-1,-1]^T$ and obtained the solution $\mathbf{y}=[0.5,-2,0]^T$. Now solving the upper-triangular system $U\mathbf{x}=\mathbf{y}$ of the form

$$U\mathbf{x}=\begin{pmatrix} 1 & 0.5 & \alpha \\ 0 & -4 & 1-2\alpha \\ 0 & 0 & 1.5\alpha+2.25 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}=\begin{pmatrix} 0.5 \\ -2 \\ 0 \end{pmatrix}=\mathbf{y}.$$

From last equation we have

$$(1.5\alpha + 2.25)x_3 = 0,$$

so for unique solution of the given system $(1.5\alpha + 2.25) \neq 0$ *(nonsingular), which implies that* $x_3 = 0$. *Using backward substitution, we have* $x_2 = 0.5$ *and* $x_1 = 0.25$. *Thus,* $[0.25, 0.5, 0]^T$ *is the unique solution of the given system.*
If $(1.5\alpha + 2.25) = 0$ *(singular), that is,* $\alpha = -1.5$, *then for this we must have infinitely many solutions. So to get the infinitely many solutions, we have to solve the following resulting system*

$$\begin{array}{rcrcrcr} x_1 & + & 0.5x_2 & + & \alpha x_3 & = & 0.5 \\ - & & 4x_2 & + & (1-2\alpha)x_3 & = & -2 \\ & & & & 0x_3 & = & 0 \end{array}$$

By taking $\alpha = -1.5$ *and if we choose* $x_3 = t \in R,\ t \neq 0$, *then we have* $x_2 = 0.5 + t$ *and* $x_1 = 0.25 + t$, *so* $\mathbf{x}^* = [0.25 + t, 0.5 + t, t]^T$ *is the infinitely many solutions of the given system.* •

Example 3.28 *Use LU-factorization method with Doolittle's method* ($l_{ii} = 1$) *to find the constant* α *such that the following nonhomogeneous linear system has non-trivial solutions (infinite many solutions). Find these solutions.*

$$\begin{array}{rcrcrcr} x_1 & + & x_2 & & & = & 1 \\ 3x_1 & + & \alpha x_2 & + & 5x_3 & = & 8 \\ & & 7x_2 & + & 3x_3 & = & 3 \end{array}$$

Solution. *Using Simple Gauss-elimination method, we can easily find factorization of A as*

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 3 & \alpha & 5 \\ 0 & 7 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 7/(\alpha-3) & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 0 & (\alpha-3) & 5 \\ 0 & 0 & 3 - 35/(\alpha-3) \end{pmatrix} = LU.$$

Since by one of the property of the determinant

$$\det(A) = \det(LU) = \det(L)\det(U).$$

So when using LU decomposition by Doolittle's method, then

$$\det(A) = \det(U) = \prod_{i=1}^{n} u_{ii} = (u_{11}u_{22}\cdots u_{nn}),$$

where $\det(L) = 1$ *because L is lower-triangular matrix and all its diagonal elements are unity. Thus the determinant of the given matrix A is*

$$|A| = |U| = (\alpha - 3)(3 - 35/(\alpha - 3)) = 3\alpha - 44, \quad \alpha \neq 3.$$

So $|A| = 0$, *gives,* $\alpha = 44/3$ *and for this value of* α *we have non-trivial solutions. By solving the lower-triangular system of the form* $L\mathbf{y} = [1, 8, 3]^T$ *of the form*

$$L\mathbf{y} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 3/5 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 8 \\ 3 \end{pmatrix} = \mathbf{b},$$

we obtained the solution $\mathbf{y} = [1, 5, 0]^T$. *Now solving the upper-triangular system* $U\mathbf{x} = \mathbf{y}$ *of the form*

$$U\mathbf{x} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 35/3 & 5 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix} = \mathbf{y}.$$

If we choose $x_3 = t \in R,\ t \neq 0$, *then,* $x_2 = 3(1-t)/7$ *and* $x_1 = (4+3t)/7$, *then the non-trivial solutions of the given system is* $\mathbf{x}^* = [(4+3t)/7, 3(1-t)/7, t]^T$. •

3.4.3.2 Inverse of a Matrix by using Doolittle's Method

LU factorization by using Doolittle's method is also used to invert matrices. Their usefulness for this purpose is based on the fact that triangular matrices are easily inverted. Once the factorization has been effected, the inverse of a matrix A is found from the formula

$$A^{-1} = (LU)^{-1} = U^{-1}L^{-1}. \tag{3.31}$$

Then

$$UA^{-1} = L^{-1}, \qquad \text{where} \qquad LL^{-1} = I.$$

A practical way of calculating the determinant is to use the forward elimination process of the Gaussian elimination or, alternatively, the LU decomposition.
If no pivoting is used, calculation of the determinant using the LU decomposition is very easy.

Example 3.29 *Find determinant and inverse of the following matrix using LU decomposition by Doolittle's method.*

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

Solution. *Since we know that*

$$A = \begin{pmatrix} 1 & -2 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = LU.$$

Using $m_{21} = 1$, $m_{31} = 1$, *and* $m_{32} = 3$, *we get*

$$\begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Obviously, the original set of equations has been transformed to an upper-triangular form. Thus

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

which is the required decomposition of A.
Now we find the determinant of the matrix A as follows:

$$\det(A) = \det(U) = u_{11}u_{22}u_{33} = (1)(1)(1) = 1.$$

To find the inverse of the matrix A, first we will compute the inverse of the lower-triangular matrix L^{-1} *from*

$$LL^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} l'_{11} & 0 & 0 \\ l'_{21} & l'_{22} & 0 \\ l'_{31} & l'_{32} & l'_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I,$$

by using the forward substitution.
To solve the first system

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} l'_{11} \\ l'_{21} \\ l'_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

by using forward substitution, we get, $l'_{11} = 1,\ l'_{21} = -1,\ l'_{31} = 2$. *Similarly, the solution of the second linear system*

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ l'_{22} \\ l'_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

can be obtained as, $l'_{22} = 1,\ l'_{32} = -3$. *Finally, the solution of the third linear system*

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ l'_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

gives $l'_{33} = 1$. *Hence the elements of the matrix* L^{-1} *are*

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix},$$

which is the required inverse of the lower-triangular matrix L.
To find the inverse of the given matrix A, we will solve the system

$$UA^{-1} = \begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{21} & a'_{22} & a'_{23} \\ a'_{31} & a'_{32} & a'_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix} = L^{-1},$$

by using backward substitution.
We solve the first system

$$\begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a'_{11} \\ a'_{21} \\ a'_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix},$$

by using backward substitution, we get, $a'_{11} = -3,\ a'_{21} = -1,\ a'_{31} = 2$. *Similarly, the solution of the second linear system*

$$\begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a'_{12} \\ a'_{22} \\ a'_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -3 \end{pmatrix},$$

can be obtained as, $a'_{12} = 5,\ a'_{22} = 1,\ a'_{32} = -3$. *Finally, the solution of the third linear system*

$$\begin{pmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a'_{13} \\ a'_{23} \\ a'_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

can be obtained as, $a'_{13} = -1,\ a'_{23} = 0,\ a'_{33} = 1$. *Hence the elements of the inverse matrix* A^{-1} *are*

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

3.4.3.3 Crout's Method

The Crout's method, in which matrix U has unity on the main diagonal, is similar to Doolittle's method in all other aspects. The L and U matrices are obtained by expanding the matrix equation $A = LU$ term by term to determine the elements of the L and U matrices.

Example 3.30 *Construct the* LU *decomposition of the following matrix using Crout's method*

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 2 & 5 & 6 \end{pmatrix}.$$

Solution. *Since*

$$A = LU = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{pmatrix}.$$

Performing the multiplication on the right-hand side, gives

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 2 & 5 & 6 \end{pmatrix} = \begin{pmatrix} l_{11} & l_{11}u_{12} & l_{11}u_{13} \\ l_{21} & l_{21}u_{12} + l_{22} & l_{21}u_{13} + l_{22}u_{23} \\ l_{31} & l_{31}u_{12} + l_{32} & l_{31}u_{13} + l_{32}u_{23} + l_{33} \end{pmatrix}.$$

Then equating elements of first column to obtain, $1 = l_{11}, 6 = l_{21}, 2 = l_{31}$. *Then equating elements of second column to obtain*

$$\begin{array}{rclrcl} 2 &=& l_{11}u_{12}, & u_{12} &=& 2, \\ 5 &=& l_{21}u_{12} + l_{22}, & l_{22} &=& 5 - 12 = -7, \\ 5 &=& l_{31}u_{12} + l_{32}, & l_{32} &=& 5 - 4 = 1. \end{array}$$

Finally, then equating elements of third column to obtain

$$\begin{array}{rclcrcl} 3 &=& l_{11}u_{13}, & & u_{13} &=& 3, \\ 4 &=& l_{21}u_{13}+l_{22}u_{23}, & & u_{23} &=& (4-18)/-7 = \ 2, \\ 6 &=& l_{31}u_{13}+l_{32}u_{23}+l_{33}, & & l_{33} &=& (6-6-2) \ = -2. \end{array}$$

Thus we get

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 2 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 6 & -7 & 0 \\ 2 & 1 & -2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} = LU,$$

the factorization of the given matrix. •

The general formula for getting elements of L and U corresponding to the coefficient matrix A for a set of n linear equations can be written as

$$\left.\begin{array}{rcll} l_{ij} &=& a_{ij} - \displaystyle\sum_{k=1}^{j-1} l_{ik}u_{kj}, & i \geq j, \quad i = 1, 2, \ldots, n \\ u_{ij} &=& \dfrac{1}{l_{ii}}[a_{ij} - \displaystyle\sum_{k=1}^{i-1} l_{ik}u_{kj}], & i < j, \quad j = 2, 3, \ldots, n \\ l_{ij} &=& a_{i1}, & j = 1 \\ u_{ij} &=& \dfrac{a_{ij}}{a_{11}}, & i = 1 \end{array}\right\}. \tag{3.32}$$

Example 3.31 *Solve the following linear system by LU decomposition using Crout's method*

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 2 & 5 & 6 \end{pmatrix} \qquad \textit{and} \qquad \mathbf{b} = \begin{pmatrix} 1 \\ -1 \\ 5 \end{pmatrix}.$$

Solution. *Since the factorization of the coefficient matrix A has been already constructed in the Example (3.30) as*

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

Then solving the first system $L\mathbf{y} = \mathbf{b}$ *for unknown vector* $\mathbf{y}$*, that is*

$$L\mathbf{y} \begin{pmatrix} 1 & 0 & 0 \\ 6 & -7 & 0 \\ 2 & 1 & -2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 5 \end{pmatrix} = \mathbf{b}.$$

Performing forward substitution yields

$$\begin{array}{rcrcrcl} y_1 & & & & = & 1, \ \textit{gives} & y_1 = 1, \\ 6y_1 & - & 7y_2 & & = & -1, \ \textit{gives} & y_2 = 1, \\ 2y_1 & + & y_2 & - \ 2y_3 = & 5, \ \textit{gives} & y_3 = -1. \end{array}$$

Then solving the second system $U\mathbf{x} = \mathbf{y}$ *for unknown vector* $\mathbf{x}$*, that is*

$$U\mathbf{x} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = \mathbf{y}.$$

Performing backward substitution yields

$$\begin{array}{rcrcrcrcrcl} x_1 & + & 2x_2 & + & 3x_3 & = & 1, & \textit{gives} & x_1 & = & -2, \\ & & x_2 & + & 2x_3 & = & 1, & \textit{gives} & x_2 & = & 3, \\ & & & & x_3 & = & -1, & \textit{gives} & x_3 & = & -1, \end{array}$$

and we obtained the solution $\mathbf{x} = [-2, 3, -1]^T$ *of the given system.* •

The above results can be reproduced by using MATLAB command as follows:

```
>> A = [1 2 3;6 5 4;2 5 6]; b = [1 -1 5]; sol = Crout(A,b);
```

Program 3.9

MATLAB m-file for the Crout's Method

```
function sol = Crout(A, b)
[n,n]=size(A); u=zeros(n,n); l=u; for i=1:n; u(i,i)=1; end; l(1,1)=A(1,1);
for i=2:n; u(1,i)=A(1,i)/l(1,1); l(i,1)=A(i,1); end; for i=2:n; for j=2:n; s=0;
if i <= j; K=i-1; else; K=j-1; end; for k=1:K; s = s + l(i,k) * u(k,j); end
if j > i; u(i,j)=(A(i,j)-s)/l(i,i); else l(i,j)=A(i,j)-s; end;end;end
y(1)=b(1)/l(1,1); for k=2:n; sum=b(k); for i=1:k-1; sum = sum - l(k,i) * y(i); end
y(k)=sum/l(k,k);end x(n)=y(n)/u(n,n); for k=n-1:-1:1; sum=y(k);
for i=k+1:n; sum = sum - u(k,i) * x(i); end; x(k)=sum/u(k,k); end; l; u; y; x;
```

Note that we can also find the LU-decomposition of a matrix A by using simple Gauss-elimination method. We start with the product matrices of the form $\mathbf{I}A$ and convert it to the equivalent form LU, that is, we have to convert right matrix A to unit upper-triangular matrix U. We describe the procedure in the following example.

Example 3.32 *Solve the following system using LU-decomposition by Crout's method*

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

Solution. *The Crout's method makes LU factorization a byproduct of Gaussian elimination. To illustrate, let the given matrix of the system is*

$$A = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 0 & -2 \\ -3 & -5 & 1 \end{pmatrix}.$$

The process begins with the product matrices form

$$\mathbf{I}A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ -1 & 0 & -2 \\ -3 & -5 & 1 \end{pmatrix}.$$

In each of the steps below, we arrange so that the product of the two matrices is always equal to the original matrix A. Now the first step of Gaussian elimination on the right factor is to divide the first row by the pivot element. Then the Crout's rule copies the pivot element to the matching element of the left factor at the same time we divide. The next step in Gaussian elimination is to eliminate all the elements below the pivot element. This is done by multiplying the first row by below $(n-1)$ eliminating elements, subtracting the product from the $(n-1)$ rows, and putting the result in the $(n-1)$ rows. The Crout's rule copies all those eliminating elements into the matching elements of the left factor. We repeat the same procedure for the remaining pivot elements. Thus we obtain

$$\mathbf{I}A \equiv \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 2 & -2 \\ 0 & 1 & 1 \end{pmatrix}.$$

The product of matrices is still equal to A.

$$\mathbf{I}A \equiv \begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ -3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{pmatrix}.$$

The product of matrices is still equal to A.

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

The product is still A and we have achieved the desired factorization.
Now solving the first system $L\mathbf{y} = \mathbf{b}$ for unknown vector $\mathbf{y}$, that is

$$L\mathbf{y} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ -3 & 1 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} = \mathbf{b}.$$

Performing forward substitution yields

$$\begin{array}{rcrcrcl} y_1 & & & & = & 3, & \textit{gives } y_1 = 3, \\ -y_1 & + & 2y_2 & & = & 1, & \textit{gives } y_2 = 2, \\ -3y_1 & + & y_2 & + \; 2y_3 & = & 1, & \textit{gives } y_3 = 4. \end{array}$$

Then solving the second system $U\mathbf{x} = \mathbf{y}$ for unknown vector $\mathbf{x}$, that is

$$U\mathbf{x} = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix} = \mathbf{y}.$$

Performing backward substitution yields

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & & = & 3, & \textit{gives} \quad x_1 = -9, \\ & & x_2 & - \; x_3 & = & 2, & \textit{gives} \quad x_2 = 6, \\ & & & x_3 & = & 4, & \textit{gives} \quad x_3 = 4, \end{array}$$

and we obtained $\mathbf{x} = [-9, 6, 4]^T$*, the solution of the given system.* •

Example 3.33 *Use LU decomposition by Crout's method to find the value(s) of* α *for which the following matrix*

$$A = \begin{pmatrix} 1 & 1 & \alpha \\ 1 & \alpha & 1 \\ \alpha & 1 & 1 \end{pmatrix},$$

is singular. Compute the unique solution of the linear system $A\mathbf{x} = [1, 1, -2]^T$ *by using the smallest positive integer value of* α*.*

solution. *Using the procedure done in Example (3.32), we obtained the factored of the matrix*

$$A = \mathbf{I}A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & \alpha \\ 1 & \alpha & 1 \\ \alpha & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & \alpha - 1 & 0 \\ \alpha & 1 - \alpha & (2 - \alpha - \alpha^2) \end{pmatrix} \begin{pmatrix} 1 & 1 & \alpha \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} = LU.$$

Since $|A| = |L| = (\alpha - 1)(2 - \alpha - \alpha^2) = 0$*, gives,* $\alpha = -2$*,* $\alpha = 1$*, and* $\alpha = 1$*, which make* A *singular. To find unique solution we have to choose* $\alpha = 2$*, and then solve the lower-triangular system*

$$L\mathbf{y} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -1 & -4 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix} = \mathbf{b},$$

and it gives, $y_1 = 1, \quad y_2 = 0, \quad y_3 = 1$*. Now solve the upper-triangular system*

$$U\mathbf{x} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \mathbf{y},$$

and we obtained $\mathbf{x} = [-2, 1, 1]^T$*, the solution of the given system. One can check that for* $\alpha = 1$ *we have no solution and for* $\alpha = -2$ *we have infinitely many solutions.* •

Procedure 3.4 [LU Decomposition by the Crout's Method]

1. *Take the nonsingular matrix,* A*.*
2. *If possible, decompose the matrix* $A = LU$ *using (3.32).*
3. *Solve linear system* $L\mathbf{y} = \mathbf{b}$ *using (3.28).*
4. *Solve linear system* $U\mathbf{x} = \mathbf{y}$ *using (3.29).*

3.4.3.4 Inverse of a Matrix by using Crout's Method

The Crout's method can also be use to find the inverse of a matrix A from the formula

$$A^{-1} = (LU)^{-1} = U^{-1}L^{-1}, \tag{3.33}$$

where U^{-1} and L^{-1} can be obtained as

$$U^{-1}U = \mathbf{I} \qquad \text{and} \qquad L^{-1}L = \mathbf{I}.$$

Example 3.34 *Use Crout's method to find L^{-1}, U^{-1}, and the determinant of the inverse of the following matrix.*

$$A = \begin{pmatrix} 2 & -2 & 1 \\ 5 & 1 & -3 \\ 3 & 4 & 1 \end{pmatrix}.$$

Solution. *First we compute the LU decomposition of A, which is*

$$A = \begin{pmatrix} 2 & -2 & 1 \\ 5 & 1 & -3 \\ 3 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 5 & 6 & 0 \\ 3 & 7 & 71/12 \end{pmatrix} \begin{pmatrix} 1 & -1 & 1/2 \\ 0 & 1 & -11/12 \\ 0 & 0 & 1 \end{pmatrix} = LU.$$

$$\det(A) = \det(L) = l_{11}l_{22}l_{33} = (2)(6)(71/12) = 71.$$

To find the inverse of the matrix A, first we will compute the inverse of the lower-triangular matrix L^{-1} from

$$LL^{-1} = \begin{pmatrix} 2 & 0 & 0 \\ 5 & 6 & 0 \\ 3 & 7 & 71/12 \end{pmatrix} \begin{pmatrix} l'_{11} & 0 & 0 \\ l'_{21} & l'_{22} & 0 \\ l'_{31} & l'_{32} & l'_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I,$$

by using the forward substitution. To solve the first system

$$\begin{pmatrix} 2 & 0 & 0 \\ 5 & 6 & 0 \\ 3 & 7 & 71/12 \end{pmatrix} \begin{pmatrix} l'_{11} \\ l'_{21} \\ l'_{31} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

by using forward substitution, we get, $l'_{11} = 1/2,\ l'_{21} = -5/12,\ l'_{31} = 17/71$. Similarly, the solution of the second linear system

$$\begin{pmatrix} 2 & 0 & 0 \\ 5 & 6 & 0 \\ 3 & 7 & 71/12 \end{pmatrix} \begin{pmatrix} 0 \\ l'_{22} \\ l'_{32} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

can be obtained as, $l'_{22} = 1/6,\ l'_{32} = -14/71$. Finally, the solution of the third linear system

$$\begin{pmatrix} 2 & 0 & 0 \\ 5 & 6 & 0 \\ 3 & 7 & 71/12 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ l'_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

gives $l'_{33} = 12/71$. *Hence the elements of the matrix* L^{-1} *are*

$$L^{-1} = \begin{pmatrix} 1/2 & 0 & 0 \\ -5/12 & 1/6 & 0 \\ 17/71 & -14/71 & 12/71 \end{pmatrix},$$

which is the required inverse of the lower-triangular matrix L. *To find the inverse* U^{-1}, *let*

$$U^{-1} = \begin{pmatrix} 1 & a'_{12} & a'_{13} \\ 0 & 1 & a'_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad \textit{then} \quad U^{-1}U = \begin{pmatrix} 1 & -1 + a'_{12} & 1/2 - 11/12 + a'_{13} \\ 0 & 1 & -11/12 + a'_{23} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{I},$$

and by equating the corresponding entries, we obtained

$$U^{-1} = \begin{pmatrix} 1 & 1 & 5/12 \\ 0 & 1 & 11/12 \\ 0 & 0 & 1 \end{pmatrix}.$$

Hence the elements of the inverse matrix A^{-1} *are*

$$A^{-1} = U^{-1}L^{-1} = \begin{pmatrix} 1 & 1 & 5/12 \\ 0 & 1 & 11/12 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & 0 & 0 \\ -5/12 & 1/6 & 0 \\ 17/71 & -14/71 & 12/71 \end{pmatrix} = \frac{1}{71} \begin{pmatrix} 13 & 6 & 5 \\ -14 & -1 & 11 \\ 17 & -14 & 12 \end{pmatrix}.$$

Thus the determinant of the inverse matrix A^{-1} *is*

$$\det(A^{-1}) = \det(U^{-1}) \det(L^{-1}) = \det(L^{-1}) = (1/2)(1/6)(12/71) = \frac{1}{71}.$$

For the LU decomposition we have not used pivoting for the sake of simplicity. However, pivoting is important for the same reason as in the Gaussian elimination. We know that pivoting in the Gaussian elimination is equivalent to interchanging the rows of coefficients matrix together with the terms on the right-hand side. This indicates that pivoting may be applied to the LU decomposition as long as the interchanging is applied to the left and right terms in the same way. When performing pivoting in the LU decomposition, the changes in the order of the rows are recorded. The same reordering is then applied to the right-hand side terms before starting the solution in accordance with the forward elimination and backward substitution steps. •

Since we know that not every matrix has a direct LU decomposition. We define the following matrix which gives the sufficient condition for the LU decomposition of the matrix. It also, helps us for the convergence of the iterative methods for solving linear systems.

Definition 3.24 (Strictly Row Diagonally Dominant Matrix)

A square matrix is said to be strictly row diagonally dominant (SRDD) if in every row of the matrix the absolute value of each element on the main diagonal is greater than the sum of the absolute values of all the other elements in that row. Thus, strictly row diagonally dominant matrix is defined as

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad \textit{for} \quad i = 1, 2, \ldots, n. \tag{3.34}$$

Example 3.35 *The matrix*

$$A = \begin{pmatrix} 7 & 3 & 1 \\ 1 & 6 & 3 \\ -2 & 4 & 8 \end{pmatrix},$$

is strictly row diagonally dominant since

$$\begin{array}{lllllll} |7| & > & |3| + |1|, & \textit{that is}, & 7 & > & 4, \\ |6| & > & |1| + |3|, & \textit{that is}, & 6 & > & 4, \\ |8| & > & |-2| + |4|, & \textit{that is}, & 8 & > & 6, \end{array}$$

but the following matrix

$$B = \begin{pmatrix} 6 & -3 & 4 \\ 3 & 7 & 3 \\ 5 & -4 & 10 \end{pmatrix},$$

is not strictly row diagonally dominant since

$$|6| \not> |-3| + |4|.$$

Strictly row diagonally dominant matrix occurs naturally in a wide variety of practical applications, and when solving a strictly row diagonally dominant system by Gauss-elimination method, partial pivoting is never required. •

Theorem 3.14 *If a matrix A is strictly row diagonally dominant, then:*

1. *Matrix A is nonsingular.*
2. *Gaussian elimination without row interchange can be performed on the linear system* $A\mathbf{x} = \mathbf{b}$.
3. *Matrix A has LU factorization.* •

Definition 3.25 (Strictly Column Diagonally Dominant Matrix)

A square matrix is said to be strictly column diagonally dominant (SCDD) if in every column of the matrix the absolute value of each element on the main diagonal is greater than the sum of the absolute values of all the other elements in that column. Thus, strictly column diagonally dominant matrix is defined as

$$|a_{ii}| > \sum_{\substack{i=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad \textit{for} \quad j = 1, 2, \ldots, n. \tag{3.35}$$

A matrix is strictly column diagonally dominant if A^T *is strictly row diagonally dominant.*

Example 3.36 *The matrix*

$$A = \begin{pmatrix} 7 & 3 & 1 \\ 1 & 6 & 3 \\ -2 & 2 & 8 \end{pmatrix},$$

is strictly column diagonally dominant since

$$\begin{array}{llll} |7| > |1| + |-2|, & \textit{that is}, & 7 > 3, \\ |6| > |3| + |2|, & \textit{that is}, & 6 > 5, \\ |8| > |3| + |1|, & \textit{that is}, & 8 > 4, \end{array}$$

but the following matrix

$$B = \begin{pmatrix} 6 & -3 & 4 \\ 3 & 7 & 3 \\ 5 & -4 & 10 \end{pmatrix},$$

is not strictly column diagonally dominant since

$$|6| \not> |3| + |5|.$$

Obviously, strict row or column diagonal dominance may be achieved (or disrupted) by re-ordering the system.
Strictly diagonally dominant matrix occurs naturally in a wide variety of practical applications, and when solving a strictly diagonally dominant system by Gauss-elimination method, partial pivoting is never required. •

Theorem 3.15 *If a matrix A is strictly column diagonally dominant, then:*

1. *Matrix A is nonsingular.*
2. *Gaussian elimination without row interchange can be performed on the linear system $A\mathbf{x} = \mathbf{b}$.*
3. *Matrix A has LU factorization.* •

Example 3.37 *Solve the following linear system using the simple Gaussian elimination method and also, find the LU decomposition of the matrix using Doolittle's method and Crout's method*

$$\begin{array}{rcrcrcr} 5x_1 & + & x_2 & + & x_3 & = & 7 \\ 2x_1 & + & 6x_2 & + & x_3 & = & 9 \\ x_1 & + & 2x_2 & + & 9x_3 & = & 12 \end{array}$$

solution. *Start with the augmented matrix form*

$$\left(\begin{array}{ccc:c} 5 & 1 & 1 & 7 \\ 2 & 6 & 1 & 9 \\ 1 & 2 & 9 & 12 \end{array}\right),$$

and since $a_{11} = 5 \neq 0$, so we can eliminate the elements a_{21} and a_{31} by subtracting from the second and third rows the appropriate multiples of the first row. In this case the multiples are given

$$m_{21} = \frac{2}{5} \quad \textit{and} \quad m_{31} = \frac{1}{5}.$$

Hence

$$\left(\begin{array}{ccccc} 5 & 1 & 1 & \vdots & 7 \\ 0 & 28/5 & 3/5 & \vdots & 31/5 \\ 0 & 9/5 & 44/5 & \vdots & 53/5 \end{array}\right).$$

As $a_{22}^{(1)} = 28/5 \neq 0$, *therefore, we eliminate entry in* $a_{32}^{(1)}$ *position by subtracting the multiple* $m_{32} = \dfrac{1.8}{5.6} = 9/28$ *of the second row from the third row, to get*

$$\left(\begin{array}{ccccc} 5 & 1 & 1 & \vdots & 7 \\ 0 & 28/5 & 3/5 & \vdots & 31/5 \\ 0 & 0 & 43/5 & \vdots & 43/5 \end{array}\right).$$

Obviously, the original set of equations has been transformed to an upper-triangular form. Since all the diagonal elements of the obtaining upper-triangular matrix are nonzero, which means that the coefficient matrix of the given system is nonsingular and therefore, the given system has a unique solution. Now expressing the set in algebraic form yields

$$\begin{array}{rcrcrcl} 5x_1 & + & x_2 & + & x_3 & = & 7 \\ & & (28/5)x_2 & + & (3/5)x_3 & = & 31/5 \\ & & & & (43/5)x_3 & = & 43/5 \end{array}$$

Now using backward substitution to get the solution of the system as

$$\begin{array}{lcll} (43/5)x_3 & = & 43/5, & \textit{gives} \quad x_3 = 1, \\ (28/5)x_2 & = & -(3/5)x_3 + 31/5, & \textit{gives} \quad x_2 = 1, \\ 5x_1 & = & 7 - x_2 - x_3, & \textit{gives} \quad x_1 = 1. \end{array}$$

Since we know that in using LU decomposition by Doolittle's method the unknown elements of matrix L are the used multiples and the matrix U is same as we obtained in forward elimination process of the simple Gauss elimination. Thus the LU decomposition of the matrix A can be obtained by using Doolittle's method as follows:

$$A = \left(\begin{array}{ccc} 5 & 1 & 1 \\ 2 & 6 & 1 \\ 1 & 2 & 9 \end{array}\right) = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 2/5 & 1 & 0 \\ 1/5 & 9/28 & 1 \end{array}\right)\left(\begin{array}{ccc} 5 & 1 & 1 \\ 0 & 28/5 & 3/5 \\ 0 & 0 & 43/5 \end{array}\right) = LU.$$

Similarly, the LU decomposition of the matrix A by using Crout's method can be obtained as

$$A = \left(\begin{array}{ccc} 5 & 1 & 1 \\ 2 & 6 & 1 \\ 1 & 2 & 9 \end{array}\right) = \left(\begin{array}{ccc} 5 & 0 & 0 \\ 2 & 28/5 & 0 \\ 1 & 9/5 & 43/5 \end{array}\right)\left(\begin{array}{ccc} 1 & 1/5 & 1/5 \\ 0 & 1 & 1/10 \\ 0 & 0 & 1 \end{array}\right) = LU.$$

Thus the conditions of the Theorem 3.15 are satisfied. •

Definition 3.26 (Regular Matrix)

A square matrix is said to be regular matrix if it is both the strictly row diagonally dominant matrix and strictly column diagonally dominant matrix. For example, the matrix

$$A = \begin{pmatrix} 4 & 2 & 1 \\ 1 & 6 & 2 \\ 1 & 2 & 5 \end{pmatrix},$$

is a regular matrix but the following matrix (only strictly row diagonally dominant matrix)

$$B = \begin{pmatrix} 3 & 2 & 0 \\ 2 & 4 & 1 \\ 2 & 3 & 6 \end{pmatrix},$$

is not regular matrix and the following matrix (only strictly column diagonally dominant matrix)

$$C = \begin{pmatrix} 5 & 2 & 3 \\ 1 & 6 & 5 \\ 2 & 3 & 9 \end{pmatrix},$$

is also not regular matrix. •

Note that:

- A regular matrix is a square matrix that has an inverse.
- A regular matrix A, is described as a square matrix that for all positive integer n, is such that A^n has positive entries.
- If A has an entry that is 0 or negative, then A is not regular.

3.5 Norms of Vectors and Matrices

For solving linear systems, we discuss a method for quantitatively measuring the distance between vectors in $\mathbf{R}^n$, the set of all column vectors with real components, to determine whether the sequence of vectors that results from using an direct method converges to a solution of the system. To define a distance in $\mathbf{R}^n$, we use the notation of the *norm* of a vector.

3.5.0.1 Vector Norms

It is sometimes useful to have a scalar measure of the magnitude of a vector. Such a measure is called a *vector norm* and for a vector $\mathbf{x}$ is written as $\|\mathbf{x}\|$.
A vector norm on $\mathbf{R}^n$ is a function, from $\mathbf{R}^n$ to $\mathbf{R}$ satisfying:

1. $\|\mathbf{x}\| > 0$ for all $\mathbf{x} \in \mathbf{R}^n$.
2. $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$.

3. $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|, \quad \text{for all} \quad \alpha \in \mathbf{R}, \quad \mathbf{x} \in \mathbf{R}^n.$

4. $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|, \quad \text{for all} \quad \mathbf{x}, \mathbf{y} \in \mathbf{R}^n.$

There are three norms in $\mathbf{R}^n$ that are most commonly used in applications, called l_1-norm, l_2-norm, and l_∞-norm, and are defined for the given vectors $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ as

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}, \qquad \|\mathbf{x}\|_\infty = \max_{1\leq i\leq n} |x_i|.$$

The l_1-norm is called the *absolute norm*, the l_2-norm is frequently called the *Euclidean norm* as it is just the formula for distance in ordinary three-dimensional Euclidean space extended to dimension n. Finally, the l_∞-norm is called the *maximum norm* or occasionally the *uniform norm*. All these three norms are also called the *natural norms*.

Example 3.38 *Compute l_p-norms ($p = 1, 2, \infty$) of the vector $\mathbf{x} = [-5, 3, -2]^T$ in $\mathbf{R}^3$.*

Solution. *These l_p-norms ($p = 1, 2, \infty$) of the given vector are:*

$$\begin{aligned}
\|\mathbf{x}\|_1 &= |x_1| + |x_2| + |x_3| = |-5| + |3| + |-2| = 10.\\
\|\mathbf{x}\|_2 &= (x_1^2 + x_2^2 + x_3^2)^{1/2} = \left[(-5)^2 + (3)^2 + (-2)^2\right]^{1/2} \approx 6.1644.\\
\|\mathbf{x}\|_\infty &= \max\{|x_1|, |x_2|, |x_3|\} = \max\{|-5|, |3|, |-2|\} = 5.
\end{aligned}$$

Note that:

$$\begin{aligned}
\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{n}\|\mathbf{x}\|_\infty\\
\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2\\
\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_1 \leq n\|\mathbf{x}\|_\infty
\end{aligned}$$

In MATLAB command the built-in **norm** function computes l_p-norms of vectors. If only one argument is passed to norm, the l_2-norm is returned and for two arguments, the second one is used to specify the value of p. For example,

```
>> x = [-5 3 -2]; v = norm(x); v = norm(x,2); v = norm(x,1), v = norm(x,inf)
```

The internal MATLAB constant *inf* is used to select the l_∞-norm.

3.5.0.2 Matrix Norms

A matrix norm is a measure of how well one matrix approximates another, or, more accurately, of how well their difference approximates the zero matrix. An iterative procedure for inverting a matrix produces a sequence of approximate inverses. Since in practices such a process must be terminated, it is desirable to have some measure of the error of approximate inverse.
So a matrix norm on the set of all $n \times n$ matrices is a real-valued function, $\|.\|$, defined on this set, satisfying for all $n \times n$ matrices A and B and all real number α as follows:

1. $\|A\| > 0, \qquad A \neq \mathbf{0}.$
2. $\|A\| = 0, \qquad A = \mathbf{0}.$
3. $\|I\| = 1, \qquad I \text{ is the identity matrix.}$
4. $\|\alpha A\| = |\alpha|\|A\|, \quad \text{for some scalar} \quad \alpha \in \mathbf{R}.$
5. $\|A + B\| \leq \|A\| + \|B\|.$
6. $\|AB\| \leq \|A\|\|B\|.$
7. $\|A - B\| \geq \Big|\|A\| - \|B\|\Big|.$

Several norms for matrices have been defined, we shall use the following three natural norms l_1, l_2, and l_∞ for a square matrix of order n:

$$\|A\|_1 = \max_j \left(\sum_{i=1}^{n} |a_{ij}|\right) = \text{maximum column-sum.}$$

$$\|A\|_2 = \max_{\|x\|_2=1} \|A\mathbf{x}\|_2 = \text{spectral norm.}$$

$$\|A\|_\infty = \max_i \left(\sum_{j=1}^{n} |a_{ij}|\right) = \text{row-sum norm.}$$

The l_1-norm and l_∞-norm are widely used because they are easy to calculate. The matrix norm $\|A\|_2$ that corresponds to the l_2-norm is related the eigenvalues of the matrix. It sometimes has special utility because no other norm is smaller than this norm. It therefore, provides the best measure of the size of a matrix, but is also the most difficult to compute. We will discuss this natural norm later in the chapter.

For $m \times n$ matrix, we can paraphrase the *Frobenius norm* (or *Euclidean norm*), which is not a natural norm and is define as

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2} = \sqrt{tr(A^T A)},$$

where $tr(A^T A)$ is the *trace* of a matrix $A^T A$, that is, the sum of the diagonal entries of $A^T A$. The Frobenius norm of a matrix is a good measure of the magnitude of a matrix. It is to be noted that $\|A\|_F \neq \|A\|_2$. For a diagonal matrix, all norms have the same values.

Example 3.39 *Compute l_p-norms ($p = 1, \infty, F$) of the following matrix*

$$A = \begin{pmatrix} 4 & 2 & -1 \\ 3 & 5 & -2 \\ 1 & -2 & 7 \end{pmatrix}.$$

Solution. *The l_1, l_∞, l_F-norms of the matrix are as follows:*

$$\|A\|_1 = max\{8, 9, 10\} = 10.$$

$$\|A\|_\infty = max\{7, 10, 10\} = 10.$$

$$\|A\|_F = (16 + 4 + 1 + 9 + 25 + 4 + 1 + 4 + 49)^{1/2} \approx 10.6301.$$

Like l_p-norms of vectors, in MATLAB command the built-in **norm** function can be used to compute l_p-norms of matrices. The l_1-norm, l_∞-norm and Frobenius norm of a matrix can be find as

$$>> A = [4\ 2\ -1; 3\ 5\ -2; 1\ -2\ -7];\ norm(A,1);\ norm(A,inf);\ norm(A,inf)$$

3.6 Iterative Methods for Solving Linear Systems

The methods discussed in the previous section for the solution of the system of linear equations have been direct, which required a finite number of arithmetic operations. The elimination methods of solving such systems usually yield sufficiently accurate solutions for approximately 20 to 25 simultaneous equations, where most of the unknowns are present in all of the equations. When the coefficients matrix is sparse (has many zeros), a considerably large number of equations can be handled by the elimination methods. But these methods are generally impractical when many hundreds or thousands of equations must be solved simultaneously.
There are, however, several methods which can be used to solve large numbers of simultaneous equations. These methods are, called *iterative methods* by which an approximation to the solution of a system of linear equations may be obtained. The iterative methods are used most often for large sparse systems of linear equations and efficient in terms of computer storage and time requirement. Systems of this type arise frequently in the numerical solution of boundary value problems and partial differential equations. Unlike the direct methods, the iterative methods may not always yield a solution, even if the determinant of the coefficients matrix is not zero. Here, we consider just two of these iterative methods. These two forms the basis of a family of methods which are designed either to accelerate the convergence or to suit some particular computer architecture.

3.6.1 Jacobi Iterative Method

This is one of the easiest iterative method to find the approximate solution of the system of linear equations (3.42). To explain its procedure, consider a system of three linear equations as follows:

$$\begin{array}{rcrcrcl} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3 \end{array}$$

The solution process starts by solving for the first variable x_1 from first equation, second variable x_2 from second equation and third variable x_3 from third equation, gives

$$\begin{array}{rcrcrcl} a_{11}x_1 & = & b_1 & - & a_{12}x_2 & - & a_{13}x_3 \\ a_{22}x_2 & = & b_2 & - & a_{21}x_1 & - & a_{23}x_3 \\ a_{33}x_3 & = & b_3 & - & a_{31}x_1 & - & a_{32}x_2 \end{array}$$

Divide both sides of the above three equations by their diagonal elements, a_{11}, a_{22} and a_{33} respectively, to have

$$\begin{array}{rcrcrcl} x_1 & = & 1/a_{11}(b_1 & - & a_{12}x_2 & - & a_{13}x_3) \\ x_2 & = & 1/a_{22}(b_2 & - & a_{21}x_1 & - & a_{23}x_3) \\ x_3 & = & 1/a_{33}(b_3 & - & a_{31}x_1 & - & a_{32}x_2) \end{array}$$

Let $\mathbf{x}^{(k)} = \left[x_1^{(k)}, x_2^{(k)}, x_3^{(k)}\right]^T$ be an initial solution of the exact solution $\mathbf{x}$ of the linear system (3.42), then we define an iterative sequence

$$\begin{array}{rcl} x_1^{(k+1)} &=& 1/a_{11}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}) \\ x_2^{(k+1)} &=& 1/a_{22}(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}) \\ x_3^{(k+1)} &=& 1/a_{33}(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}) \end{array} \tag{3.36}$$

where k is the number of iterative steps. Then the form (3.36) is called the Jacobi formula for system of three equations. For a general system of n linear equations, the Jacobi method is defined by

$$\begin{array}{rcl} x_i^{(k+1)} &=& \dfrac{1}{a_{ii}}\left[b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right] \\ i &=& 1, 2, \ldots, n, \quad k = 0, 1, 2, \ldots, \end{array} \tag{3.37}$$

provided that the diagonal elements $a_{ii} \neq 0$ for each $i = 1, 2, \ldots, n$. If the diagonal elements equal to zero, then reordering of the equations can be performed so that no element in the diagonal position equal to zero. As usual with iterative methods, an initial approximation $x_i^{(0)}$ must be supplied. If we don't have knowledge about the exact solution, it is conventional to start with $x_i^{(0)} = \mathbf{0}$ for all i. The iterations defined by (3.37) are stopped when

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \epsilon, \tag{3.38}$$

or by using other possible stopping criteria

$$\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k+1)}\|} < \epsilon, \tag{3.39}$$

where ϵ is a preassigned small positive number. For this purpose, any convenient norm can be used, the most usual being the l_∞-norm.

Example 3.40 *Solve the following system of equations using the Jacobi iterative method, using $\epsilon = 10^{-6}$ in the l_∞-norm.*

$$\begin{array}{rcrcrcl} 5x_1 &-& x_2 &+& x_3 &=& 10 \\ 2x_1 &+& 8x_2 &-& x_3 &=& 11 \\ -x_1 &+& x_2 &+& 4x_3 &=& 3 \end{array}$$

Start with the initial solution $\mathbf{x}^{(0)} = [0, 0, 0]^T$.

Solution. *The Jacobi iterative method for the given system has the form*

$$\begin{array}{rcl} x_1^{(k+1)} &=& 1/5(10 + x_2^{(k)} - x_3^{(k)}) \\ x_2^{(k+1)} &=& 1/8(11 - 2x_1^{(k)} + x_3^{(k)}) \\ x_3^{(k+1)} &=& 1/4(3 + x_1^{(k)} - x_2^{(k)}) \end{array}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000000	0.000000	0.000000
1	2.000000	1.375000	0.750000
2	2.125000	0.968750	0.906250
⋮	⋮	⋮	⋮
15	2.000000	0.999999	1.000000
16	2.000000	1.000000	1.000000

Table 3.1: Solution of the Example 3.40

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000000	0.000000	0.000000
1	5.500000	-10.0000	0.750000
2	45.87500	18.25000	4.625000
3	-65.1875	224.0000	7.656250

Table 3.2: Solution of the Example 3.41

and starting with initial approximation $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$*, then for* $k = 0$*, we obtain*

$$\begin{aligned} x_1^{(1)} &= 1/5(10 + x_2^{(0)} - x_3^{(0)}) = 1/5(10 + 0 - 0) = 2, \\ x_2^{(1)} &= 1/8(11 - 2x_1^{(0)} + x_3^{(0)}) = 1/8(11 - 0 + 0) = 1.375, \\ x_3^{(1)} &= 1/4(3 + x_1^{(0)} - x_2^{(0)}) = 1/4(3 + 0 - 0) = 0.75. \end{aligned}$$

The first and subsequent iterations are listed in Table 3.1.

Note that the Jacobi method converges and after 16 iterations we obtained what is obviously the exact solution. Ideally the iteration should stop automatically when we obtained the required accuracy using one of the stopping criteria mentioned by (3.38) or (3.39).
To get the above results using MATLAB command, we do the following:

```
>> Ab = [A|b] = [5 -1 1 10; 2 8 -1 11; -1 1 4 3];
>> x = [0 0 0]; acc = 0.5e-6; JacobiM(Ab, x, acc);
```

Example 3.41 *Solve the following system of equations using the Jacobi iterative method.*

$$\begin{aligned} 2x_1 + 8x_2 - x_3 &= 11 \\ 5x_1 - x_2 + x_3 &= 10 \\ -x_1 + x_2 + 4x_3 &= 3 \end{aligned}$$

Start with the initial solution $\mathbf{x}^{(0)} = [0, 0, 0]^T$.

Solution. *Results for this linear system are listed in Table 3.2.* •

Notice that Jacobi method diverges rapidly. Although the given linear system is same as the linear system of the previous Example 3.40 except the first and second equations are interchanged. From this example we concluded that Jacobi iterative method is not always convergent.

Program 3.10
MATLAB m-file for the Jacobi Iterative Method for Linear System
function x=JacobiM(Ab,x,acc)
[n,t]=size(Ab); b=Ab(1:n,t); R=1; k=1; d(1,1:n+1)=[0 x]; while $R > acc$
for i=1:n; sum=0; for j=1:n; if j ~ =i
$sum = sum + Ab(i,j) * d(k,j+1)$; end; $x(1,i) = (1/Ab(i,i)) * (b(i,1) - sum)$;end;end
k=k+1; d(k,1:n+1)=[k-1 x]; R=max(abs((d(k,2:n+1)-d(k-1,2:n+1))));
if $k > 10$ & $R > 100$ ('Jacobi Method is diverges') break; end; end; x=d;

Procedure 3.5 [Jacobi Method]

1. *Check the coefficient matrix A is strictly diagonally dominant (for guaranteed convergence).*
2. *Initialize the first approximation* $\mathbf{x}^{(0)}$ *and pre-assigned accuracy* ϵ.
3. *Compute the constant* $\mathbf{c} = D^{-1}\mathbf{b} = \dfrac{b_i}{a_{ii}}, \quad for \quad i = 1, 2, \ldots, n.$
4. *Compute the Jacobi iteration matrix* $T_J = -D^{-1}(L+U)$.
5. *Solve for the approximate solutions* $\mathbf{x}_i^{(k+1)} = T_J\mathbf{x}_i^{(k)} + \mathbf{c}, \quad i = 1, 2, \ldots, n$ *and* $k = 0, 1, \ldots$
6. *Repeat step 5 until* $\|\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}\| < \epsilon$.

3.6.2 Gauss-Seidel Iterative Method

This is one of the most popular and widely used iterative method to find the approximate solution of the system of linear equations. This iterative method is a modification of the Jacobi iterative method and give us good accuracy by using the most recently calculated values.
From the Jacobi iterative formula (3.37), it is seen that the new estimates for solution $\mathbf{x}$ are computed from the old estimates and only when all the new estimates have been determined are then used in the right-hand side of the equation to perform the next iteration. But the Gauss-Seidel method is to make use of the new estimates in the right-hand side of the equation as soon as they become available. For example, the Gauss-Seidel formula for the system of three equations can be define an iterative sequence

$$\begin{array}{rcl} x_1^{(k+1)} &=& 1/a_{11}(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}) \\ x_2^{(k+1)} &=& 1/a_{22}(b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)}) \\ x_3^{(k+1)} &=& 1/a_{33}(b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)}) \end{array} \tag{3.40}$$

For a general system of n linear equations, the Gauss-Seidel iterative method defined as

$$\begin{array}{rcl} x_i^{(k+1)} &=& \dfrac{1}{a_{ii}}\left[b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right] \\ i &=& 1, 2, \ldots, n, \quad k = 0, 1, 2, \ldots \end{array} \tag{3.41}$$

The Gauss-Seidel iterative method is sometimes called the method of *successive iteration*, because the most recent values of all $\mathbf{x}_i$ are used in the calculation.

Example 3.42 *Solve the following system of equations using the Gauss-Seidel iterative method, with* $\epsilon = 10^{-6}$ *in* l_∞*-norm.*

$$\begin{array}{rcrcrcr} 5x_1 & - & x_2 & + & x_3 & = & 10 \\ 2x_1 & + & 8x_2 & - & x_3 & = & 11 \\ -x_1 & + & x_2 & + & 4x_3 & = & 3 \end{array}$$

Start with the initial solution $\mathbf{x}^{(0)} = [0, 0, 0]^T$.

Solution. *The Gauss-Seidel iteration for the given system is*

$$\begin{array}{rcrcrcr} x_1^{(k+1)} & = & 1/5(10 & + & x_2^{(k)} & - & x_3^{(k)}) \\ x_2^{(k+1)} & = & 1/8(11 & - & 2x_1^{(k+1)} & + & x_3^{(k)}) \\ x_3^{(k+1)} & = & 1/4(3 & + & x_1^{(k+1)} & - & x_2^{(k+1)}) \end{array}$$

and starting with initial approximation $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$*, then for* $k = 0$*, we obtain*

$$\begin{array}{rcrcrcl} x_1^{(1)} & = & 1/5(10 & + & x_2^{(0)} & - & x_3^{(0)}) = 1/5(10 + 0 - 0) = 2, \\ x_2^{(1)} & = & 1/8(11 & - & 2x_1^{(1)} & + & x_3^{(0)}) = 1/8(11 - 4 + 0) = 0.875, \\ x_3^{(1)} & = & 1/4(3 & + & x_1^{(1)} & - & x_2^{(1)}) = 1/4(3 + 2 - 0.875) = 1.03125. \end{array}$$

The first and subsequent iterations are listed in Table 3.3.

The above results can be obtained using MATLAB command as follows:

```
>> Ab = [A|b] = [5 -1 1 10; 2 8 -1 11; -1 1 4 3];
>> x = [0 0 0]; acc = 0.5e-6; GaussSM(Ab, x, acc);
```

Note that the Gauss-Seidel method converged and required 10 iterations to obtain the correct solution for the given system, which is 6 iterations less than required by the Jacobi method for the same Example 3.40.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000000	0.000000	0.000000
1	2.000000	0.875000	1.031250
2	1.968750	1.011719	0.989258
3	2.004492	0.997534	1.001740
⋮	⋮	⋮	⋮
9	2.000000	0.999999	1.000000
10	2.000000	1.000000	1.000000

Table 3.3: Solution of the Example 3.42

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000000	0.000000	0.000000
1	5.500000	17.5000	-2.25000
2	-65.6250	-340.375	69.43750
3	1401.719	7068.031	-1415.83

Table 3.4: Solution of the Example 3.43

Example 3.43 *Solve the following system of equations using the Gauss-Seidel iterative method.*

$$\begin{array}{rcrcrcr} 2x_1 & + & 8x_2 & - & x_3 & = & 11 \\ 5x_1 & - & x_2 & + & x_3 & = & 10 \\ -x_1 & + & x_2 & + & 4x_3 & = & 3 \end{array}$$

Start with the initial solution $\mathbf{x}^{(0)} = [0, 0, 0]^T$.

Solution. *Results for this linear system are listed in Table 3.4. Note that in this case the Gauss-Seidel method diverges rapidly. Although the given linear system is same as the linear system of the previous Example 3.42 except the first and second equations are interchanged. From this example we concluded that the Gauss-Seidel iterative method is not always convergent.* •

Program 3.11

MATLAB m-file for the Gauss-Seidel Iterative Method for Linear System
function x=GaussSM(Ab,x,acc)
[n,t]=size(Ab); b=Ab(1:n,t);R=1; k=1; d(1,1:n+1)=[0 x]; k=k+1; while $R > acc$
for i=1:n; sum=0; for j=1:n
if $j <= i-1; sum = sum + Ab(i,j) * d(k,j+1);$ elseif $j >= i+1$
$sum = sum + Ab(i,j) * d(k-1,j+1);$ end; end; $x(1,i) = (1/Ab(i,i)) * (b(i,1) - sum);$
d(k,1)=k-1; d(k,i+1)=x(1,i); end; R=max(abs((d(k,2:n+1)-d(k-1,2:n+1))));
k=k+1; if $R > 100$ & $k > 10$; ('Gauss-Seidel method is Diverges') break ;end;end;x=d;

Procedure 3.6 [Gauss-Seidel Method]

1. *Check the coefficient matrix A is strictly diagonally dominant (for guaranteed convergence).*
2. *Initialize the first approximation* $\mathbf{x}^{(0)} \in \mathbf{R}$ *and pre-assigned accuracy* ϵ.
3. *Compute the constant* $\mathbf{c}_G = (D + L)^{-1}\mathbf{b}$.
4. *Compute the Gauss-Seidel iteration matrix* $T_G = -(D + L)^{-1}U$.
5. *Solve for the approximate solutions* $x_i^{(k+1)} = T_G x_i^{(k)} + \mathbf{c}, \quad i = 1, 2, \ldots, n$ *and* $k = 0, 1, \ldots$
6. *Repeat step 5 until* $\|\mathbf{x}_i^{(k+1)} - \mathbf{x}_i^{(k)}\| < \epsilon$.

From the Examples 3.40 and (3.42), we noted that the solution by the Gauss-Seidel method converges more quickly than the Jacobi method. In general, we may state that **if both the Jacobi method and the Gauss-Seidel method are converge, then the Gauss-Seidel method converges more quickly**. This is generally the case but not always true. In fact, there are some linear systems for which the Jacobi method converges but the Gauss-Seidel method does not, and others for which the Gauss-Seidel method converges but the Jacobi method does not.

Example 3.44 *Use the Jacobi and Gauss-Seidel method to solve the simultaneous linear equations obtaining x_1, x_2 and x_3 correct to the nearest whole number using initial values $x_1^{(0)} = 1, x_2^{(0)} = 1, x_3^{(0)} = 1$.*

$$\begin{array}{rcrcrcl} 5x_1 & + & x_2 & - & x_3 & = & 4 \\ x_1 & + & 4x_2 & + & 2x_3 & = & 15 \\ x_1 & - & 2x_2 & + & 5x_3 & = & 12 \end{array}$$

Solution. *Using the Jacobi iterative method and we obtained the approximation as follows:*

$$\begin{array}{rcrcrcl} x_1^{(k+1)} & = & 0.8 & - & 0.2x_2^{(k)} & + & 0.2x_3^{(k)} \\ x_2^{(k+1)} & = & 3.75 & - & 0.25x_1^{(k)} & - & 0.5x_3^{(k)} \\ x_3^{(k+1)} & = & 2.4 & - & 0.2x_1^{(k)} & + & 0.4x_2^{(k)} \end{array}$$

$$\begin{array}{lll} x_1^{(1)} = 0.8, & x_2^{(1)} = 3.0, & x_3^{(1)} = 2.6 \\ x_1^{(2)} = 0.72, & x_2^{(2)} = 2.25, & x_3^{(2)} = 3.44 \\ x_1^{(3)} = 1.038, & x_2^{(3)} = 1.85, & x_3^{(3)} = 3.156 \end{array}$$

The results of the last two iterations both give $x_1 = 1, x_2 = 2, x_3 = 3$, when rounded to the nearest whole number. In fact, these whole numbers are clearly seen to be the exact solution.

Using the Gauss-Seidel iterative method and we obtained the approximation as follows:

$$\begin{array}{rcrcrcl} x_1^{(k+1)} & = & 0.8 & - & 0.2x_2^{(k)} & + & 0.2x_3^{(k)} \\ x_2^{(k+1)} & = & 3.75 & - & 0.25x_1^{(k+1)} & - & 0.5x_3^{(k)} \\ x_3^{(k+1)} & = & 2.4 & - & 0.2x_1^{(k+1)} & + & 0.4x_2^{(k+1)} \end{array}$$

$$\begin{array}{lll} x_1^{(1)} = 0.8, & x_2^{(1)} = 3.05, & x_3^{(1)} = 3.46 \\ x_1^{(2)} = 0.88, & x_2^{(2)} = 1.80, & x_3^{(2)} = 2.94 \\ x_1^{(3)} = 1.03, & x_2^{(3)} = 2.02, & x_3^{(3)} = 3.00 \end{array}$$

In this particular example, the rate of convergence of both methods is about the same, giving $x_1 = 1, x_2 = 2, x_3 = 3$ to the nearest whole number; but we would normally expect the Gauss-Seidel method to converge at a faster rate. •

Example 3.45 *Use the Jacobi and Gauss-Seidel method to solve the simultaneous linear equations obtaining approximation of $[x_1, x_2, x_3]^T = [2.7373, 0.9873, -1.6525]^T$ correct to the 10^{-1} accuracy using starting values $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$.*

$$\begin{array}{rcrcrcl} 9x_1 & + & 2x_2 & + & 4x_3 & = & 20 \\ x_1 & + & 10x_2 & + & 4x_3 & = & 6 \\ 2x_1 & - & 4x_2 & + & 10x_3 & = & -15 \end{array}$$

Solution. *Using the Jacobi iterative method and we obtained the approximation as follows:*

$$\begin{array}{rcrcrcl} x_1^{(k+1)} & = & 2.2222 & - & 0.2222x_2^{(k)} & + & 0.4444x_3^{(k)} \\ x_2^{(k+1)} & = & 0.6 & - & 0.1x_1^{(k)} & - & 0.4x_3^{(k)} \\ x_3^{(k+1)} & = & -1.5 & - & 0.2x_1^{(k)} & + & 0.4x_2^{(k)} \end{array}$$

$$\begin{aligned} x_1^{(1)} &= 2.2222, \quad x_2^{(1)} = 0.6000, \quad x_3^{(1)} = -1.5000 \\ x_1^{(2)} &= 2.7556, \quad x_2^{(2)} = 0.9778, \quad x_3^{(2)} = -1.7044 \\ x_1^{(3)} &= 2.7625, \quad x_2^{(3)} = 1.0062, \quad x_3^{(3)} = -1.6600 \end{aligned}$$

Using the Gauss-Seidel iterative method and we obtained the approximation as follows:

$$\begin{aligned} x_1^{(k+1)} &= 2.2222 - 0.2222x_2^{(k)} + 0.4444x_3^{(k)} \\ x_2^{(k+1)} &= 0.6 - 0.1x_1^{(k+1)} - 0.4x_3^{(k)} \\ x_3^{(k+1)} &= -1.5 - 0.2x_1^{(k+1)} + 0.4x_2^{(k+1)} \end{aligned}$$

$$\begin{aligned} x_1^{(1)} &= 2.2222, \quad x_2^{(1)} = 0.3778, \quad x_3^{(1)} = -1.7933 \\ x_1^{(2)} &= 2.9353, \quad x_2^{(2)} = 0.9970, \quad x_3^{(2)} = -1.7044 \\ x_1^{(3)} &= 2.7403, \quad x_2^{(3)} = 2.02, \quad x_3^{(3)} = -1.6493 \\ x_1^{(4)} &= 2.7337, \quad x_2^{(4)} = 0.9863, \quad x_3^{(4)} = -1.6522 \end{aligned}$$

The rate of convergence of the Jacobi method is faster than the Gauss-Seidel method ($\|T_J\| = \|T_G\| = 0.6667$) for the given accuracy. •

3.6.2.1 Main Difference Between the Jacobi and the Gauss-Seidel Method

Both the Jacobi and Gauss-Seidel methods are iterative methods for solving the linear system $A\mathbf{x} = \mathbf{b}$. In the Jacobi method the updated vector x is used for the computations only after all the variables (i.e. all components of the vector $\mathbf{x}$) have been updated. On the other hand in the Gauss-Seidel method, the updated variables are used in the computations as soon as they are updated. Thus in the Jacobi method, during the computations for a particular iteration, the "known" values are all from the previous iteration. However in the Gauss-Seidel method, the "known" values are a mix of variable values from the previous iteration (whose values have not yet been evaluated in the current iteration), as well as variable values that have already been updated in the current iteration. Even though the Gauss-Seidel's method uses the improved values as soon as they are computed, this does not ensure that the Gauss-Seidel's method would converge faster than Jacobi iterations.

3.6.3 Matrix Forms of Iterative Methods for a Linear System

The iterative methods to solve the system of linear equations

$$A\mathbf{x} = \mathbf{b}, \tag{3.42}$$

start with an initial approximation $\mathbf{x}^{(0)} \in \mathbf{R}$ to the solution $\mathbf{x}$ of the linear system (3.42), and generates a sequence of vectors $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converges to $\mathbf{x}$. Most of these iterative methods involve a process that converts the system (3.42) into an equivalent system of the form

$$\mathbf{x} = T\mathbf{x} + \mathbf{c}, \tag{3.43}$$

for some square matrix T and vector $\mathbf{c}$. After the initial vector $\mathbf{x}^{(0)}$ is selected, the sequence of approximate solutions vector is generated by computing

$$\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}, \qquad \text{for} \quad k = 0, 1, 2, \ldots \tag{3.44}$$

The sequence is terminated when the error is sufficiently small, that is

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \epsilon, \quad \text{for small positive} \quad \epsilon. \tag{3.45}$$

Note that a matrix T is called iteration matrix and a vector $\mathbf{c}$ is a column matrix. We can find the forms of these matrices easily for both iterative methods as follows. Let a matrix A can be written as

$$A = L + D + U, \tag{3.46}$$

where L is strictly lower-triangular, U is strictly upper-triangular, and D is the diagonal parts of the coefficients matrix A, that is

$$L = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{pmatrix}, \qquad U = \begin{pmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{23} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

and

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{pmatrix}.$$

Then the linear system (3.42) can be written as

$$(L + D + U)\mathbf{x} = \mathbf{b}. \tag{3.47}$$

Now we find forms of both matrices T and $\mathbf{c}$ which help us to solve the linear system.

3.6.3.1 Jacobi Iterative Method

The equation (3.47) can be written as

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

Since matrix D is nonsingular, so we can write above equation as

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

which can be put in the form

$$\mathbf{x}^{(k+1)} = T_J\mathbf{x}^{(k)} + \mathbf{c}_j, \qquad \text{for} \quad k = 0, 1, 2, \ldots, \tag{3.48}$$

which is the matrix form of Jacobi iterative method and where

$$T_J = -D^{-1}(L + U) \quad \text{and} \quad \mathbf{c}_j = D^{-1}\mathbf{b}, \tag{3.49}$$

are called Jacobi iteration matrix and Jacobi constant column matrix, respectively and their elements are defined by

$$\begin{aligned} t_{ij} &= \frac{a_{ij}}{a_{ii}}, \qquad i,j = 1,2,\ldots,n, \quad i \neq j, \\ t_{ij} &= 0, \qquad i = j, \\ c_i &= \frac{b_i}{a_{ii}}, \qquad i = 1,2,\ldots,n. \end{aligned}$$

Note that the diagonal elements of Jacobi iteration matrix T_J are always zero.

3.6.3.2 Gauss-Seidel Iterative Method

The equation (3.47) can also be written as

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

Since lower-triangular matrix $(L+D)$ is nonsingular, so we can write above equation as

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

which can be put in the form

$$\mathbf{x}^{(k+1)} = T_G\mathbf{x}^{(k)} + \mathbf{c}_G, \qquad \text{for} \quad k = 0,1,2,\ldots, \tag{3.50}$$

which is the matrix form of Gauss-Seidel iterative method and where

$$T_G = -(L+D)^{-1}U \quad \text{and} \quad \mathbf{c}_G = (L+D)^{-1}\mathbf{b}, \tag{3.51}$$

are called Gauss-Seidel iteration matrix and Gauss-Seidel constant column matrix, respectively.

Example 3.46 *Consider the following system*

$$\begin{array}{rcrcrcr} 6x_1 & + & 2x_2 & & & = & 1 \\ x_1 & + & 7x_2 & - & 2x_3 & = & 2 \\ 3x_1 & - & 2x_2 & + & 9x_3 & = & -1 \end{array}$$

(a) Find the matrix form of iterative (Jacobi and Gauss-Seidel) methods.
(b) If $\mathbf{x}^{(k)} = [x_1^{(k)}, x_2^{(k)}, x_3^{(k)}]^T$*, then writing the iterative forms of part(a) in the component forms and find the exact solution of the given system.*
(c) Find formulas for the error $\mathbf{e}^{(k+1)}$ *in the* $(n+1)th$ *step .*
(d) Find the second approximation of the error $e^{(2)}$ *using part (c) if* $\mathbf{x}^{(0)} = [0,0,0]^T$.

Solution. *Since the given matrix A is*

$$A = \begin{pmatrix} 6 & 2 & 0 \\ 1 & 7 & -2 \\ 3 & -2 & 9 \end{pmatrix},$$

and so

$$A = L + U + D = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 3 & -2 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 6 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 9 \end{pmatrix}.$$

Jacobi Iterative Method

(a) Since the matrix form of the Jacobi iterative method can be written as

$$\mathbf{x}^{(k+1)} = T_J \mathbf{x}^{(k)} + \mathbf{c}_J, \qquad k = 0, 1, 2, \ldots,$$

where

$$T_J = -D^{-1}(L + U) \qquad \textit{and} \qquad \mathbf{c}_J = D^{-1}\mathbf{b}.$$

One can easily compute the Jacobi iteration matrix T_J and the vector $\mathbf{c}_J$ as follows:

$$T_J = -\begin{pmatrix} 1/6 & 0 & 0 \\ 0 & 1/7 & 0 \\ 0 & 0 & 1/9 \end{pmatrix}\begin{pmatrix} 0 & 2 & 0 \\ 1 & 0 & -2 \\ 3 & -2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -2/6 & 0 \\ -1/7 & 0 & 2/7 \\ -3/9 & 2/9 & 0 \end{pmatrix} \quad \textit{and} \quad \mathbf{c} = \begin{pmatrix} 1/6 \\ 2/7 \\ -1/9 \end{pmatrix}.$$

Thus the matrix form of Jacobi iterative method is

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 0 & -2/6 & 0 \\ -1/7 & 0 & 2/7 \\ -3/9 & 2/9 & 0 \end{pmatrix}\mathbf{x}^{(k)} + \begin{pmatrix} 1/6 \\ 2/7 \\ -1/9 \end{pmatrix}, \quad k = 0, 1, 2.$$

(b) Now by writing the above iterative matrix form of in the component form, we have

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & -1/3 & 0 \\ -1/7 & 0 & 2/7 \\ -1/3 & 2/9 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 1/6 \\ 2/7 \\ -1/9 \end{pmatrix},$$

and it is equivalent to

$$\begin{array}{rcrcrcr} x_1 & = & & - & 1/3x_2 & + & 1/6 \\ x_2 & = & -1/7x_1 & + & 2/7x_3 & + & 2/7 \\ x_3 & = & -1/3x_1 & + & 2/9x_2 & - & 1/9 \end{array}$$

Solving for x_1, x_2 and x_3, we get, $x_1 = 1/12,\ x_2 = 1/4,\ x_3 = -1/12$, the exact solution.
(c) Since the error in the $(n+1)$th step is defined as, $\mathbf{e}^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k+1)}$,

$$\mathbf{e}^{(k+1)} = \begin{pmatrix} 1/12 \\ 1/4 \\ -1/12 \end{pmatrix} - \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix}\mathbf{x}^{(k)} - \begin{pmatrix} 16 \\ 27 \\ -19 \end{pmatrix}.$$

This can be also written as

$$\mathbf{e}^{(k+1)} = \begin{pmatrix} 1/12 \\ 1/4 \\ -1/12 \end{pmatrix} - \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix}\begin{pmatrix} 1/12 \\ 1/4 \\ -1/12 \end{pmatrix} + \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix}\mathbf{e}^{(k)} - \begin{pmatrix} 16 \\ 27 \\ -19 \end{pmatrix}.$$

$$\mathbf{e}^{(k+1)} = \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix} \mathbf{e}^{(k)},$$

(because $\mathbf{x}^{(k)} = \mathbf{x} - \mathbf{e}^{(k)}$*) which is the required error in the* $(n+1)th$ *step.*
(d) Now finding the first approximation of the error, we have to compute the following

$$\mathbf{e}^{(1)} = \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix} \mathbf{e}^{(0)}, \quad \textit{where} \quad \mathbf{e}^{(0)} = \mathbf{x} - \mathbf{x}^{(0)}.$$

Using $\mathbf{x}^{(0)} = [0, 0, 0]^T$*, we have*

$$\mathbf{e}^{(0)} = \begin{pmatrix} 1/12 \\ 1/4 \\ -1/12 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/12 \\ 1/4 \\ -1/12 \end{pmatrix}.$$

Thus

$$\mathbf{e}^{(1)} = \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix} \begin{pmatrix} 1/12 \\ 1/4 \\ -1/12 \end{pmatrix} = \begin{pmatrix} -1/12 \\ -1/28 \\ 1/36 \end{pmatrix}.$$

Similarly, for the second approximation of the error, we have to compute the following

$$\mathbf{e}^{(2)} = \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix} \mathbf{e}^{(1)}, \quad \textit{or} \quad \mathbf{e}^{(2)} = \begin{pmatrix} 0 & -26 & 0 \\ -17 & 0 & 27 \\ -39 & 29 & 0 \end{pmatrix} \begin{pmatrix} -1/12 \\ -1/28 \\ 1/36 \end{pmatrix} = \begin{pmatrix} 1/84 \\ 5/252 \\ 5/252 \end{pmatrix},$$

which is the required second approximation of the error.

Gauss-Seidel Iterative Method

(a) Now by using Gauss-Seidel method, first we compute the Gauss-Seidel iteration matrix T_G *and the vector* $\mathbf{c}_G$ *as follows:*

$$T_G = \begin{pmatrix} 0 & -1/3 & 0 \\ 0 & 1/21 & 2/7 \\ 0 & 23/189 & 4/63 \end{pmatrix} \quad \textit{and} \quad \mathbf{c}_G = \begin{pmatrix} 1/6 \\ 11/42 \\ -41/378 \end{pmatrix}.$$

Thus the matrix form of Gauss-Seidel iterative method is

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 0 & -1/3 & 0 \\ 0 & 1/21 & 2/7 \\ 0 & 23/189 & 4/63 \end{pmatrix} \mathbf{x}^{(k)} + \begin{pmatrix} 1/6 \\ 1142 \\ -41378 \end{pmatrix}, \quad k = 0, 1, 2.$$

(b) Writing the above iterative form in component form, we get

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & -1/3 & 0 \\ 0 & 1/21 & 2/7 \\ 0 & 23/189 & 4/63 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 1/6 \\ 11/42 \\ -41/378 \end{pmatrix},$$

and it is equivalent to

$$\begin{array}{rcrcrcr} x_1 & = & & & -1/3x_2 & + & 1/6 \\ x_2 & = & 1/21x_2 & + & 2/7x_3 & + & 11/42 \\ x_3 & = & 23/189x_2 & + & 4/63x_3 & - & 41/378 \end{array}$$

Now solving for x_1, x_2 and x_3, we get, $[x_1, x_2, x_3]^T = [1/12, 1/4, -1/12]^T$, the exact solution.
(c) The error in the $(n+1)th$ step can be easily computed as

$$\mathbf{e}^{(k+1)} = \begin{pmatrix} 0 & -1/3 & 0 \\ 0 & 1/21 & 2/7 \\ 0 & 23/189 & 4/63 \end{pmatrix} \mathbf{e}^{(k)}.$$

(d) The first and second approximations of the error can be calculated as follows:

$$\mathbf{e}^{(1)} = \begin{pmatrix} 0 & -1/3 & 0 \\ 0 & 1/21 & 2/7 \\ 0 & 23/189 & 4/63 \end{pmatrix} \mathbf{e}^{(0)} = [-1/12, -1/84, 19/756]^T,$$

and

$$\mathbf{e}^{(2)} = \begin{pmatrix} 0 & -1/3 & 0 \\ 0 & 1/21 & 2/7 \\ 0 & 23/189 & 4/63 \end{pmatrix} \mathbf{e}^{(1)} = [1/252, 5/756, 1/6804]^T.$$

3.6.4 Convergence Criteria of Iterative Methods

The rate of convergence of an iterative method determines on how fast the error $\|x - x^{(k)}\}$ goes to zero as k, the number of iterations, increases. Since we noted that the Jacobi method and the Gauss-Seidel method do not always converge to the solution of the given system of linear equations. Here we need some conditions which make the both methods converge. The sufficient conditions for the convergence of both iterative methods are discussed in the following theorems.

Theorem 3.16 (First Sufficient Condition for Convergence)

If the matrix A is strictly row diagonally dominant (SRDD), then for any choice of initial approximation $\mathbf{x}^{(0)} \in \mathbf{R}$ both the Jacobi method and the Gauss-Seidel method give sequence $\{x^{(k)}\}_{k=0}^{\infty}$ of approximations that converge to the solution of the linear system. •

Example 3.47 *Rearrange the following linear system of equations*

$$\begin{array}{rcrcrcr} x_1 & + & 6x_2 & - & 3x_3 & = & 4 \\ 2x_1 & + & 2x_2 & + & 6x_3 & = & 7 \\ 5x_1 & + & 2x_2 & - & x_3 & = & 6 \end{array}$$

such that the convergence of both iterative methods (Jacobi and Gauss-Seidel) is guaranteed. Use initial solution $\mathbf{x}^{(0)} = [0, 0, 0]^T$, compute approximation solution within accuracy 10^{-2}.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000	0.000	0.000
1	1.200	0.667	1.167
2	1.167	1.050	0.544
⋮	⋮	⋮	⋮
7	0.994	0.777	0.572
8	1.004	0.787	0.576

Table 3.5: Solution by J. Method

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.000	0.000	0.000
1	1.200	0.667	1.167
2	1.167	1.050	0.544
⋮	⋮	⋮	⋮
7	0.994	0.777	0.572
8	1.004	0.787	0.576

Table 3.6: Solution by Gauss-Seidel Method

Solution. *For the guarantee convergence of iterative methods, the system must be SRDD form, so rearrange the given system in the following form*

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

Jacobi Iterative Method:

$$\begin{aligned} x_1^{(k+1)} &= 0.2(6 - 2x_2^{(k)} + x_3^{(k)}) \\ x_2^{(k+1)} &= 0.1667(4 - x_1^{(k)} + 3x_3^{(k)}) \\ x_3^{(k+1)} &= 0.1667(7 - 2x_1^{(k)} - 2x_2^{(k)}) \end{aligned}$$

Starting with $\mathbf{x}^{(0)} = [0, 0, 0]^T$*, the first and subsequent iterations are listed in Table 3.5.*

Gauss-Seidel Iterative Method:

$$\begin{aligned} x_1^{(k+1)} &= 0.2(6 - 2x_2^{(k)} + x_3^{(k)}) \\ x_2^{(k+1)} &= 0.1667(4 - x_1^{(k+1)} + 3x_3^{(k)}) \\ x_3^{(k+1)} &= 0.1667(7 - 2x_1^{(k+1)} - 2x_2^{(k+1)}) \end{aligned}$$

Starting with initial approximation $\mathbf{x}^{(0)} = [0, 0, 0]^T$*, the first and subsequent iterations are listed in Table 3.6. Note that Gauss-Seidel iterative method converges faster than Jacobi iterative method.*

Note that the following system

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

is not a strictly row diagonally dominant matrix, and yet both methods converge to the solution $x_1 = 1$ and $x_2 = 1$ when we use an initial approximation $[0, 0]^T$.
There is another sufficient condition for the convergence of both iterative methods which is defined in the following theorem.

Theorem 3.17 (Second Sufficient Condition for Convergence)

For any initial approximation $\mathbf{x}^{(0)} \in \mathbf{R}$*, the sequence* $\{x^{(k)}\}_{k=0}^{\infty}$ *of approximations defined by*

$$\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}, \quad \textit{for each} \quad k \geq 0, \quad \textit{and} \quad \mathbf{c} \neq 0, \tag{3.52}$$

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ *if* $\|T\| < 1$ *for any natural matrix norm, and the following* **error bounds** *hold:*

$$\begin{aligned} \|\mathbf{x} - \mathbf{x}^{(k)}\| &\leq \|T\|^k \|\mathbf{x}^{(0)} - \mathbf{x}\|, \\ \|\mathbf{x} - \mathbf{x}^{(k)}\| &\leq \frac{\|T\|^k}{1 - \|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|. \end{aligned} \tag{3.53}$$

If $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \epsilon$*, then we need the number of iterations*

$$k \geq \frac{\ln(\epsilon/M)}{\ln(\|T\|)}, \qquad M = \frac{\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|}{1 - \|T\|}.$$

Note that smaller the value of the $\|T\|$, faster the convergence of the iterative methods. But the convergence is **linear**, which means, $\lim_{k\to\infty} \dfrac{\|e^{(k+1)}\|}{\|e^{(k)}\|} = M \neq 0.$ •

Example 3.48 *Consider the following nonhomogeneous linear system* $A\mathbf{x} = \mathbf{b}$*, where*

$$A = \begin{pmatrix} 5 & 0 & -1 \\ -1 & 3 & 0 \\ 0 & -1 & 4 \end{pmatrix} \quad \textit{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}.$$

Find the matrix form of iterative (Jacobi and Gauss-Seidel) methods and show that Gauss-Seidel iterative method converges faster than Jacobi iterative method for the given system.

Solution. *Here we will show that the* l_∞*-norm of the Gauss-Seidel iteration matrix* T_G *is less than the* l_∞*-norm of the Jacobi iteration matrix* T_J*, that is*

$$\|T_G\|_\infty < \|T_J\|_\infty.$$

The Jacobi iteration matrix T_J *can be obtained from the given matrix* A *as follows*

$$T_J = -D^{-1}(L+U) = \begin{pmatrix} 0 & 0 & 1/5 \\ 1/3 & 0 & 0 \\ 0 & 1/4 & 0 \end{pmatrix}, \quad c_J = D^{-1}\mathbf{b} = \begin{pmatrix} 1/5 \\ 2/3 \\ 1 \end{pmatrix}.$$

Thus the matrix form of Jacobi iterative method is

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 0 & 0 & 1/5 \\ 1/3 & 0 & 0 \\ 0 & 1/4 & 0 \end{pmatrix} \mathbf{x}^{(k)} + \begin{pmatrix} 1/5 \\ 2/3 \\ 1 \end{pmatrix}, \quad k \geq 0.$$

Similarly, Gauss-Seidel iteration matrix T_G *is defined as*

$$T_G = -(D+L)^{-1}U == \begin{pmatrix} 0 & 0 & 1/5 \\ 0 & 0 & 1/15 \\ 0 & 0 & 1/60 \end{pmatrix}, \quad C_G = \begin{pmatrix} 1/5 \\ 11/15 \\ 71/60 \end{pmatrix}.$$

So the matrix form of Gauss-Seidel iterative method is

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 0 & 0 & 1/5 \\ 0 & 0 & 1/15 \\ 0 & 0 & 1/60 \end{pmatrix} \mathbf{x}^{(k)} + \begin{pmatrix} 1/5 \\ 11/15 \\ 71/60 \end{pmatrix}, \quad k \geq 0.$$

Since the l_∞-norm of the matrix T_J is

$$\|T_J\|_\infty = \max\left\{\frac{1}{5}, \frac{1}{3}, \frac{1}{4}\right\} = \frac{1}{3} = 0.3333 < 1,$$

and the l_∞-norm of the matrix T_G is

$$\|T_G\|_\infty = \max\left\{\frac{1}{5}, \frac{1}{15}, \frac{1}{60}\right\} = \frac{1}{5} = 0.2000 < 1.$$

Since $\|T_G\|_\infty < \|T_J\|_\infty$, which shows that Gauss-Seidel method will converge faster than Jacobi method for the given linear system. •

Note that the condition $\|T\| < 1$ is equivalent to the condition that a matrix A is to be strictly row diagonally dominant.
For Jacobi method for a general matrix A, the norm of Jacobi iteration matrix is defined as

$$\|T_J\| = \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left|\frac{a_{ij}}{a_{ii}}\right|.$$

Thus for $\|T_J\| < 1$ is equivalent to requiring

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| < |a_{ii}|,$$

that is, a matrix A is strictly row diagonally dominant.

Example 3.49 *Show that the matrix form of Gauss-Seidel iterative method for the following system*

$$\begin{array}{rcrcl} 2x_1 & + & x_2 & = & 4 \\ x_1 & + & 2x_2 & = & 5 \end{array}$$

is

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 0 & -0.5 \\ 0 & 0.25 \end{pmatrix} \mathbf{x}^{(k)} + \begin{pmatrix} 2 \\ 1.5 \end{pmatrix}, \quad k \geq 0.$$

Use this form to compute the second approximation $\mathbf{x}^{(2)}$ using $\mathbf{x}^{(0)} = [0.5, 0.5]^T$. If $\mathbf{x} = [1, 2]^T$ be the exact solution of the given system, then find the l_∞ norm of the exact error. Determine the number of iterations needed to get an accuracy within 10^{-2}.

Solution. *The Gauss-Seidel iteration matrix T_G is defined as*

$$T_G = -(D + L)^{-1}U = -\begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

and it gives

$$T_G = -\begin{pmatrix} 1/2 & 0 \\ -1/4 & 2/4 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -1/2 \\ 0 & 1/4 \end{pmatrix}.$$

Also,

$$c_G = (D+L)^{-1}\mathbf{b} = \begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}^{-1}\begin{pmatrix} 4 \\ 5 \end{pmatrix},$$

and it gives

$$c_G = \begin{pmatrix} 1/2 & 0 \\ -1/4 & 2/4 \end{pmatrix}\begin{pmatrix} 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3/2 \end{pmatrix}.$$

So the matrix form of Gauss-Seidel iterative method is

$$\mathbf{x}^{(k+1)} = \begin{pmatrix} 0 & -0.5 \\ 0 & 0.25 \end{pmatrix}\mathbf{x}^{(k)} + \begin{pmatrix} 2 \\ 1.5 \end{pmatrix}, \quad k \geq 0.$$

Now we use this matrix form to find the second approximation using given initial approximation, we have

$$\mathbf{x}^{(1)} = \begin{pmatrix} 0 & -0.5 \\ 0 & 0.25 \end{pmatrix}\mathbf{x}^{(0)} + \begin{pmatrix} 2 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 0 & -0.5 \\ 0 & 0.25 \end{pmatrix}\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} + \begin{pmatrix} 2 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 1.25 \\ 1.875 \end{pmatrix}.$$

$$\mathbf{x}^{(2)} = \begin{pmatrix} 0 & -0.5 \\ 0 & 0.25 \end{pmatrix}\mathbf{x}^{(1)} + \begin{pmatrix} 2 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 0 & -0.5 \\ 0 & 0.25 \end{pmatrix}\begin{pmatrix} 1.25 \\ 1.875 \end{pmatrix} + \begin{pmatrix} 2 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 1.0625 \\ 1.9688 \end{pmatrix}.$$

The l_∞ norm of exact error is

$$\|e\| = \|\mathbf{x} - \mathbf{x}^{(2)}\| = \|[1,2]^T - [1.0625, 1.9688]^T\| = \|[-0.0625, 0.0312]^T\| = 0.0625.$$

To find the number of iterations, we use the formula (3.53) as

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T_G\|^k}{1 - \|T_G\|}\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\| \leq 10^{-2}.$$

Since the l_∞-norm of the matrix T_G is

$$\|T_G\|_\infty = \max\{|-0.5|, |0.25|\} = 0.5000 < 1.$$

and it gives

$$\frac{(0.5)^k}{0.5}(1.375) \leq 10^{-4}, \quad \textit{or} \quad (0.5)^{(k-1)} \leq \frac{10^{-2}}{1.375}.$$

Taking ln *on both sides, we obtain*

$$(k-1)\ln(0.5) \leq \ln\left(\frac{10^{-2}}{1.375}\right), \quad \textit{gives} \quad k \geq 8.1027, \quad \textit{or} \quad k = 9.$$

•

Example 3.50 *Consider the following linear system of equations*

$$\begin{array}{rcrcrcr} 4x_1 & - & x_2 & + & x_3 & = & 12 \\ -x_1 & + & 3x_2 & + & x_3 & = & 1 \\ x_1 & + & x_2 & + & 5x_3 & = & -14 \end{array}$$

(a) Show that both iterative methods (Jacobi and Gauss-Seidel) will converge by using $\|T\|_\infty < 1$.
(b) Find second approximation $\mathbf{x}^{(2)}$ *when the initial solution is* $\mathbf{x}^{(0)} = [4, 3, -3]^T$.
(c) Compute the error bounds for your approximations.
(d) How many iterations needed to get an accuracy within 10^{-4}.

Solution. *From (3.46), we have*

$$A = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 3 & 1 \\ 1 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix} = L + U + D.$$

Jacobi Method:

(a) Since the Jacobi iteration matrix is defined as

$$T_J = -D^{-1}(L + U),$$

and by using the given information, we have

$$T_J = -\begin{pmatrix} 1/4 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/5 \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1/4 & -1/4 \\ 1/3 & 0 & -1/3 \\ -1/5 & -1/5 & 0 \end{pmatrix}.$$

Then the l_∞ *norm of the matrix* T_J *is*

$$\|T_J\|_\infty = \max\left\{\frac{2}{4}, \frac{2}{3}, \frac{2}{5}\right\} = \frac{2}{3} = 0.6667 < 1.$$

Thus the Jacobi method will converge for the given linear system.

(b) The Jacobi method for the given system is

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}\left[12 + x_2^{(k)} - x_3^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{3}\left[1 + x_1^{(k)} - x_3^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{5}\left[-14 - x_1^{(k)} - x_2^{(k)}\right] \end{aligned}$$

Starting with initial approximation $x_1^{(0)} = 4, x_2^{(0)} = 3, x_3^{(0)} = -3$, *and for* $k = 0, 1$, *we obtain the first and the second approximations as*

$$\mathbf{x}^{(1)} = [4.5, 2.6667, -4.2]^T \quad \textit{and} \quad \mathbf{x}^{(2)} = [4.7167, 3.2333, -4.2333]^T.$$

(c) Using the error bound formula (3.53), we obtain

$$\|\mathbf{x}-\mathbf{x}^{(2)}\| \le \frac{(2/3)^2}{1-2/3}\left\|\begin{pmatrix} 4.5 \\ 2.6667 \\ -4.2 \end{pmatrix} - \begin{pmatrix} 4 \\ 3 \\ -3 \end{pmatrix}\right\| \le \frac{4}{3}(1.2) = 1.6.$$

(d) To find the number of iterations, we use the formula (3.53) as

$$\|\mathbf{x}-\mathbf{x}^{(k)}\| \le \frac{\|T_J\|^k}{1-\|T_J\|}\|\mathbf{x}^{(1)}-\mathbf{x}^{(0)}\| \le 10^{-4}.$$

It gives

$$\frac{(2/3)^k}{1/3}(1.2) \le 10^{-4}, \quad \textit{or} \quad (2/3)^k \le \frac{10^{-4}}{3.6}.$$

Taking ln *on both sides, we obtain*

$$k\ln(2/3) \le \ln\left(\frac{10^{-4}}{3.6}\right), \quad \textit{gives} \quad k \ge 25.8789, \quad \textit{or} \quad k = 26.$$

Gauss-Seidel Method:

(a) Since the Gauss-Seidel iteration matrix is defined as

$$T_G = -(D+L)^{-1}U,$$

and by using the given information, we have

$$T_G = -\begin{pmatrix} 1/4 & 0 & 0 \\ 1/12 & 1/3 & 0 \\ -4/60 & -1/15 & 1/5 \end{pmatrix}\begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1/4 & -1/4 \\ 0 & 1/12 & -5/12 \\ 0 & -4/60 & 8/60 \end{pmatrix}.$$

Then the l_∞ norm of the matrix T_G is

$$\|T_G\|_\infty = \max\left\{\frac{2}{4}, \frac{6}{12}, \frac{12}{60}\right\} = \frac{1}{2} < 1.$$

Thus the Gauss-Seidel method will converge for the given linear system.

(b) The Gauss-Seidel method for the given system is

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}\left[12 + x_2^{(k)} - x_3^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{3}\left[1 + x_1^{(k+1)} - x_3^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{5}\left[-12 - x_1^{(k+1)} - x_2^{(k+1)}\right] \end{aligned}$$

Starting with initial approximation $x_1^{(0)} = 4, x_2^{(0)} = 3, x_3^{(0)} = -3$, *and for* $k = 0, 1$, *we obtain the first and the second approximations as*

$$\mathbf{x}^{(1)} = [4.5, 2.8333, -4.2667]^T \quad \textit{and} \quad \mathbf{x}^{(2)} = [4.775, 3.3472, -4.4244]^T.$$

(c) Using the error bound formula (3.53), we obtain

$$\|\mathbf{x} - \mathbf{x}^{(2)}\| \leq \frac{(1/2)^2}{1 - 1/2} \left\| \begin{pmatrix} 4.5 \\ 2.8333 \\ -4.2667 \end{pmatrix} - \begin{pmatrix} 4 \\ 3 \\ -3 \end{pmatrix} \right\| \leq \frac{1}{2}(1.2667) = 0.6334.$$

(d) To find the number of iterations, we use the formula (3.53) as

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T_G\|^k}{1 - \|T_G\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\| \leq 10^{-4}.$$

It gives

$$\frac{(1/2)^k}{1/2}(1.2667) \leq 10^{-4}, \quad \textit{or} \quad (1/2)^k \leq \frac{10^{-4}}{2.5334}.$$

Taking ln *on both sides, we obtain*

$$k \ln(1/2) \leq \ln\left(\frac{10^{-4}}{2.5334}\right), \quad \textit{gives} \quad k \geq 14.6084 \quad \textit{or} \quad k = 15,$$

which is the required number of iterations. •

Example 3.51 *Consider the following linear system of equations*

$$\begin{array}{rcrcrcr} 2x_1 & + & \alpha x_2 & & & = & 3 \\ \alpha x_1 & + & 8x_2 & + & \alpha x_3 & = & 10 \\ & & \alpha x_2 & + & 2x_3 & = & 3 \end{array}$$

(a) Find the bound for α *such that the Jacobi iterative method will converge using* $\|T\|_\infty < 1$.
(b) Find second approximation $\mathbf{x}^{(2)}$ *when the initial solution is* $\mathbf{x}^{(0)} = [0.5, 0.5, 0.5]^T$.
(c) Compute the error bounds for your approximations.
(d) How many iterations needed to get an accuracy within 10^{-4}.
(e) If $\mathbf{x} = [1, 1, 1]^T$ *be the exact solution of the given system, then find the* l_∞ *norm of the exact error.*

Solution. *From (3.46), we have*

$$A = \begin{pmatrix} 2 & \alpha & 0 \\ \alpha & 8 & \alpha \\ 0 & \alpha & 2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ \alpha & 0 & 0 \\ 0 & \alpha & 0 \end{pmatrix} + \begin{pmatrix} 0 & \alpha & 0 \\ 0 & 0 & \alpha \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 2 \end{pmatrix} = L + U + D.$$

(a) Since the Jacobi iteration matrix is defined as

$$T_J = -D^{-1}(L + U),$$

and by using the given information, we have

$$T_J = \begin{pmatrix} 0 & -\alpha/2 & 0 \\ -\alpha/8 & 0 & -\alpha/8 \\ 0 & -\alpha/2 & 0 \end{pmatrix}.$$

Then the l_∞ norm of the matrix T_J is

$$\|T_J\|_\infty = \max\left\{|\frac{\alpha}{2}|, |\frac{\alpha}{4}|, |\frac{\alpha}{2}|\right\} = |\frac{\alpha}{2}| < 1.$$

Thus the Jacobi method will converge for the given linear system for $-2 < \alpha < 2$.

(b) The Jacobi method for the given system using $\alpha = 1$ is

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{2}\left[3 \quad - \quad x_2^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{8}\left[10 \quad - \quad x_1^{(k)} \quad - \quad x_3^{(k)}\right] \\ x_3^{(k+1)} &= \frac{1}{2}\left[3 \quad - \quad x_2^{(k)}\right] \end{aligned}$$

Starting with initial approximation $x_1^{(0)} = 0.5, x_2^{(0)} = 0.5, x_3^{(0)} = 0.5$, and for $k = 0, 1$, we obtain the first and the second approximations as

$$\mathbf{x}^{(1)} = [1.25, 1.25, 1.25]^T \quad \textit{and} \quad \mathbf{x}^{(2)} = [0.9375, 0.9375, 0.9375]^T.$$

(c) Using the error bound formula (3.53), we obtain

$$\|\mathbf{x} - \mathbf{x}^{(2)}\| \leq \frac{(1/2)^2}{1 - 1/2} \left\| \begin{pmatrix} 1.25 \\ 1.25 \\ 1.25 \end{pmatrix} - \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \right\| \leq \frac{1}{2}(0.75) = 0.375.$$

(d) To find the number of iterations, we use the formula (3.53) as

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T_J\|^k}{1 - \|T_J\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\| \leq 10^{-4}.$$

It gives

$$\frac{(1/2)^k}{1/2}(0.75) \leq 10^{-4}, \quad \textit{or} \quad (1/2)^k \leq \frac{10^{-4}}{1.5}.$$

Taking ln *on both sides, we obtain, $k \geq 13.8727$, or $k = 14$.*
(e) Since

$$\|\mathbf{x} - \mathbf{x}^{(2)}\| = \|[0.0625, 0.0625, 0.0625]^T\| = 0.0625. \qquad \bullet$$

Example 3.52 *Consider the following system*

$$\begin{array}{rcrcrcl} 4x_1 & + & x_2 & & & = & 7 \\ x_1 & + & 2x_2 & & & = & 0 \\ & & 2x_2 & + & 3x_3 & = & 1 \end{array}$$

If $x^{(0)} = [0, 0, 0]^T$*, then compute an error bound* $\|x - x^{(10)}\|$ *for the approximation using Gauss-Seidel method.*

Solution. *Since we know that error bound formula for the gauss-Seidel method is*

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T_G\|^k}{1 - \|T_G\|}\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|,$$

and given $k = 10$*, we have*

$$\|\mathbf{x} - \mathbf{x}^{(10)}\| \leq \frac{\|T_G\|^{10}}{1 - \|T_G\|}\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|.$$

So we have to find $\|T_G\|$ *and the first approximation* $\mathbf{x}^{(1)}$*.*
Since the Gauss-Seidel iteration matrix is defined as

$$T_G = -(D + L)^{-1}U,$$

and by using the given information, we have

$$T_G = -\begin{pmatrix} 4 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 2 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

To find the inverse of the matrix $(D + L)$*, we will use Gauss-Jordan method as follows:*

$$[(D + L)|I] = \left(\begin{array}{ccc:ccc} 4 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 \end{array}\right) \equiv \left(\begin{array}{ccc:ccc} 1 & 0 & 0 & 1/4 & 0 & 0 \\ 0 & 2 & 0 & -1/4 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 \end{array}\right)$$

$$\equiv \left(\begin{array}{ccc:ccc} 1 & 0 & 0 & 1/4 & 0 & 0 \\ 0 & 1 & 0 & -1/8 & 1/2 & 0 \\ 0 & 0 & 3 & 1/4 & -1 & 1 \end{array}\right) \equiv \left(\begin{array}{ccc:ccc} 1 & 0 & 0 & 1/4 & 0 & 0 \\ 0 & 1 & 0 & -1/8 & 1/2 & 0 \\ 0 & 0 & 1 & 1/12 & -1/3 & 1/3 \end{array}\right).$$

Thus

$$T_G = -\begin{pmatrix} 1/4 & 0 & 0 \\ -18 & 1/2 & 0 \\ 1/12 & -1/3 & 1/3 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -1/4 & 0 \\ 0 & 1/8 & 0 \\ 0 & -1/12 & 0 \end{pmatrix}.$$

Then the l_∞ *norm of the matrix* T_G *is*

$$\|T_G\|_\infty = \max\left\{\frac{1}{4}, \frac{1}{8}, \frac{1}{12}\right\} = \frac{1}{4} = 0.25 < 1.$$

Now to find the first approximation using Gauss-Seidel method, we will the following formula

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{4}\Big[7 - x_2^{(k)} \Big] \\ x_2^{(k+1)} &= \frac{1}{2}\Big[0 - x_1^{(k+1)} \Big] \\ x_3^{(k+1)} &= \frac{1}{3}\Big[1 - 2x_2^{(k+1)}\Big] \end{aligned}$$

Starting with initial approximation $x_1^{(0)} = 0, x_2^{(0)} = 0, x_3^{(0)} = 0$ *and for* $k = 0$*, we obtain the first approximation as*

$$\mathbf{x}^{(1)} = [1.7500, -0.8750, 0.9167]^T.$$

Thus

$$\|\mathbf{x} - \mathbf{x}^{(10)}\| \leq \frac{(0.25)^{10}}{0.75}(1.75) = 2.2252 \times 10^{-6},$$

the required an error bound. •

Theorem 3.18 *If* A *is a symmetric positive definite matrix with positive diagonal entries, then the Gauss-Seidel method converges to unique solution of the linear system* $A\mathbf{x} = \mathbf{b}$*.* •

Theorem 3.19 *Let* A *be the coefficient matrix of the system* $A\mathbf{x} = \mathbf{b}$*. The Gauss-Seidel method guaranteed converges to unique solution of the linear system if* $A^TA\mathbf{x} = A^T\mathbf{b}$*.* •

Example 3.53 *Solve the following system of equations using the Gauss-Seidel iterative method.*

$$\begin{array}{rcrcrcr} 2x_1 & + & 8x_2 & - & x_3 & = & 11 \\ 5x_1 & - & x_2 & + & x_3 & = & 10 \\ -x_1 & + & x_2 & + & 4x_3 & = & 3 \end{array}$$

Start with the initial solution $\mathbf{x}^{(0)} = [0, 0, 0]^T$*.*

Solution. *We noted that this example (Example 3.42), the Gauss-Seidel method diverges rapidly. Using above theorem condition (*$A^TA\mathbf{x} = A^T\mathbf{b}$*), we got the equivalent system as*

$$\begin{array}{rcrcrcr} 30x_1 & + & 10x_2 & - & x_3 & = & 69 \\ 10x_1 & + & 66x_2 & - & 5x_3 & = & 81 \\ -x_1 & - & 5x_2 & + & 18x_3 & = & 11 \end{array}$$

for which the method converges very fast to exact solution $\mathbf{x} = [2, 1, 1]^T$ *just after 5 iterations.* •

3.7 Errors in Solving Linear Systems

Any computed solution of a linear system must, because of round-off and other errors, be considered an approximate solution. Here we shall consider the most natural method for determining the accuracy of a solution of the linear system. One obvious way of estimating the accuracy of the computed solution $\mathbf{x}^*$ is to compute $A\mathbf{x}^*$ and to see how close $A\mathbf{x}^*$ comes to $\mathbf{b}$. Thus if $\mathbf{x}^*$ is an approximate solution of the given system $A\mathbf{x} = \mathbf{b}$, we compute a vector

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}^*, \tag{3.54}$$

which is called the *residual vector* and can be easily calculated. The quantity

$$\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} = \frac{\|\mathbf{b} - A\mathbf{x}^*\|}{\|\mathbf{b}\|},$$

is called the *relative residual*.

Program 3.12
MATLAB m-file for finding Residual Vector

```
function r=RES(A,b,x0)
[n,n]=size(A);
for i=1:n; R(i) = b(i); for j=1:n
R(i)=R(i)-A(i,j)*x0(j);end; RES(i)=R(i); end; r=RES'
```

The smallness of the residual then provides a measure of the goodness of the approximate solution $\mathbf{x}^*$. If every component of vector $\mathbf{r}$ vanishes, then $\mathbf{x}^*$ is the exact solution. If $\mathbf{x}^*$ is a good approximation then we would expect each component of $\mathbf{r}$ to be small, at least in a relative sense. For example, the following linear system

$$\begin{aligned} x_1 &+ 2x_2 &= 3 \\ 1.0001x_1 &+ 2x_2 &= 3.0001 \end{aligned}$$

has the exact solution $\mathbf{x} = [1, 1]^T$ but has a poor approximate solution $\mathbf{x}^* = [3, 0]^T$. To see how good this solution is, we compute the residual, $\mathbf{r} = [0, -0.0002]^T$, and so $\|\mathbf{r}\|_\infty = 0.0002$. Although the norm of the residual vector is small, the approximate solution $\mathbf{x}^* = [3, 0]^T$ is obviously quite poor; in fact $\|\mathbf{x} - \mathbf{x}^*\|_\infty = 2$.
To get above results using MATLAB command, we do the following:

```
>> A = [1 2; 1.0001 2]; b = [3 3.0001]; x0 = [3 0];
>> RESID(A, b, x0); x = [1 1]; Error = norm((x - x0), inf);
```

We can conclude from the residual that the approximate solution is correct to at most three decimal places.
Intuitively it would seem reasonable to assume that when $\|\mathbf{r}\|$ is small for a given vector norm, then the error $\|\mathbf{x} - \mathbf{x}^*\|$ would be small as well. In fact this is true for some systems. However, there are systems of equations which do not satisfy this property. Such systems are said to be *ill-conditioned*.

3.7.1 Ill-Conditioned Linear Systems

In solving the linear system numerically we have to see the problem conditioning, algorithm stability, and cost. Above we discussed efficient elimination schemes to solve a linear system and these schemes are stable when pivoting is employed. But there are some ill-conditioned systems which are tough to solve by any method.
A system of equations is considered to be well-conditioned if a small change in the coefficient matrix or a small change in the right hand side results in a small change in the solution vector.
A system of equations is considered to be ill-conditioned if a small change in the coefficient matrix or a small change in the right hand side results in a large change in the solution vector.
Before introduction of ill-conditioned system, let us consider the following system of equation

$$\begin{aligned} x_1 &+ 3x_2 = 4 \\ \tfrac{1}{3}x_1 &+ x_2 = 1.33 \end{aligned}$$

Note that this system of equations has no solution. But, if we take the approximate value of $\frac{1}{3} = 0.3$, then above system becomes

$$\begin{aligned} x_1 &+ 3x_2 = 4 \\ 0.3x_1 &+ x_2 = 1.33 \end{aligned}$$

The solution of this system is $x_1 = 0.1,\ x_2 = 1.3$. If we take the approximate value of $\frac{1}{3} = 0.33$, then the system becomes

$$\begin{aligned} x_1 &+ 3x_2 = 4 \\ 0.33x_1 &+ x_2 = 1.33 \end{aligned}$$

whose solution is $x_1 = 10,\ x_2 = -2$. The system

$$\begin{aligned} x_1 &+ 3x_2 = 4 \\ 0.3333x_1 &+ x_2 = 1.33 \end{aligned}$$

gives the solution $x_1 = 100,\ x_2 = -32$.
So all these systems of equation "looks" ill-conditioned (**unstable or ill-posed system**) because a small change in the coefficient matrix resulted in a large change in the solutions vector.
Consider another linear system

$$\begin{aligned} x_1 &+ 2x_2 = 4 \\ 2x_1 &+ 3x_2 = 7 \end{aligned}$$

The exact solution is easily verified to be $x_1 = 2,\ x_2 = 1$. Now make a small change in the right hand side vector of the equations

$$\begin{aligned} x_1 &+ 2x_2 = 4.001 \\ 2x_1 &+ 3x_2 = 7.001 \end{aligned}$$

gives the solution $x_1 = 1.999, x_2 = 1.001$. Make a small change in the coefficient matrix of the equations

$$\begin{aligned} 1.001x_1 &+ 2.001x_2 = 4 \\ 2.001x_1 &+ 3.001x_2 = 7.999 \end{aligned}$$

gives the solution $x_1 = 2.003, x_2 = 0.997$.
So this system of equation "looks" well conditioned (**stable or well-posed system**) because small

changes in the coefficient matrix or the right hand side resulted in small changes in the solution vector.
In practical problems we can expect the coefficients in the system to be subject to small errors, either because of round-off or because of physical measurement. If the system is ill-conditioned, the resulting solution may be grossly in error. Errors of this type, unlike those caused by round-off error accumulation, can not be avoided by careful programming.
Note that for ill-conditioned systems the residual is not necessarily a good measure of the accuracy of a solution. How then can we tell when a system is ill-conditioned ? In the following we discuss the some possible indicators of ill-conditioned system.

Definition 3.27 (Condition Number of a Matrix)

The number $\|A\|\|A^{-1}\|$ is called the condition number of a nonsingular matrix A and is denoted by $K(A)$, that is

$$cond(A) = K(A) = \|A\|\|A^{-1}\|. \tag{3.55}$$

Note that the condition number $K(A)$ for A depends on the matrix norm used and can, for some matrices, vary considerably as the matrix norm is changed. Since

$$1 = \|I\| = \|AA^{-1}\| \leq \|A\|\|A^{-1}\| = K(A),$$

therefore, the condition number is always in the range $1 \leq K(A) \leq \infty$ regardless of any natural norm. The lower limit is attained for identity matrices and $K(A) = \infty$ if A is singular. So the matrix A is *well-behaved* (well-conditioned) if $K(A)$ is close to 1 and is increasingly *ill-conditioned* when $K(A)$ is significantly greater than 1, that is, $K(A) \to \infty$. But, how large does $K(A)$ have to be before a system is regarded as ill-conditioned? There is no clear threshold. However, to assess the effects of ill-conditioning, a rule of thumb can be used. For a system with condition number $K(A)$, expect a loss of roughly $log_{10}K(A)$ decimal places in the accuracy of the solution. For the above first system the matrix and its inverse

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3.999 \end{bmatrix},\ A^{-1} = \begin{bmatrix} -3999 & 2000 \\ 2000 & -1000 \end{bmatrix},\ K(A) = \|A\|_\infty\|A^{-1}\|_\infty = 5.999 \times 5999 = 35990,$$

expect to lose 4 decimal places ($log_{10}(35990) = 4.55618$) in accuracy. On the other hand the above second system the matrix and its inverse

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} -3 & 2 \\ 2 & -1 \end{bmatrix}, \quad K(A) = \|A\|_\infty\|A^{-1}\|_\infty = 5 \times 5 = 25,$$

expect to lose 1 decimal place ($log_{10}(25) = 1.39794$) in accuracy.

The condition numbers provide bounds for the sensitivity of the solution of a set of equations to changes in the coefficient matrix. Unfortunately, the evaluation of any of the condition numbers of a matrix A is not a trivial task since it is necessary first to obtain its inverse.

Example 3.54 *Compute the condition number of the following matrix using the l_∞-norm*

$$A = \begin{pmatrix} 2 & -1 & 0 \\ 2 & -4 & -1 \\ -1 & 0 & 2 \end{pmatrix}.$$

Solution. *Since the condition number of a matrix is defined as*

$$K(A) = \|A\|_\infty \|A^{-1}\|_\infty.$$

First we calculate the inverse of the given matrix which is

$$A^{-1} = \begin{pmatrix} \frac{8}{13} & -\frac{2}{13} & -\frac{1}{13} \\ \frac{3}{13} & -\frac{4}{13} & -\frac{2}{13} \\ \frac{4}{13} & -\frac{1}{13} & \frac{6}{13} \end{pmatrix}.$$

Then

$$\|A\|_\infty = 7, \quad \textit{and} \quad \|A^{-1}\|_\infty = \frac{11}{13}.$$

Therefore,

$$K(A) = \|A\|_\infty \|A^{-1}\|_\infty = (7)\left(\frac{11}{13}\right) \approx 5.9231.$$

Depending on the application, we might consider this number to be reasonably small and conclude that the given matrix A is reasonably well-conditioned. •

To get above results using MATLAB commands, we do the following:

```
>> A = [2 -1 0; 2 -4 -1; -1 0 2]; Ainv = inv(A)
>> K(A) = norm(A, inf) * norm(Ainv, inf)
```

Example 3.55 *If the condition number of following matrix A is* 8.8671*, then find the* l_∞*-norm of its inverse matrix, that is,* $\|A^{-1}\|_\infty$

$$A = \begin{pmatrix} 10.2 & 2.4 & 4.5 \\ -2.3 & 7.7 & 11.1 \\ -5.5 & -3.2 & 0.9 \end{pmatrix}.$$

Solution. *Since the condition number of a matrix is defined as*

$$K(A) = \|A\|_\infty \|A^{-1}\|_\infty.$$

First we calculate the l_∞*-norm of the given matrix A which is the maximum of the absolute row sums, we have*

$$\|A\|_\infty = max\{17.1000, 21.1000, 9.6\} = 21.1000,$$

and as it is given $K(A) = 8.8671$*, so we have*

$$8.8671 = (21.1000)\|A^{-1}\|_\infty.$$

Simplifying this, we get $\|A^{-1}\|_\infty = 0.4202$. •

We might think that if the determinant of a matrix is close to zero, then the matrix is ill-conditioned. However, this is false. Consider the following matrix

$$A = \begin{pmatrix} 10^{-7} & 0 \\ 0 & 10^{-7} \end{pmatrix},$$

for which $\det A = 10^{-14} \approx 0$. One can easily find the condition number of the given matrix as

$$K(A) = \|A\|_\infty \|A^{-1}\|_\infty = (10^{-7})(10^{7}) = 1.$$

The matrix A is therefore perfectly conditioned. Thus a small determinant is necessary but not sufficient for a matrix to be ill-conditioned.
The condition number of a matrix $K(A)$ using l_2-norm can be computed by the built-in function **cond** command in the MATLAB as follows:

```
>> A = [1  -1 2; 3 1  -1; 2 0 1];  K(A) = cond(A)
```

Theorem 3.20 (Error in Linear Systems)

Suppose that $\mathbf{x}^*$ *is an approximation to the solution* $\mathbf{x}$ *of the linear system* $A\mathbf{x} = \mathbf{b}$ *and* A *is a nonsingular matrix and* $\mathbf{r}$ *is the residual vector for* $\mathbf{x}^*$*. Then for any natural norm, the error is*

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \|\mathbf{r}\| \|A^{-1}\|, \tag{3.56}$$

and the relative error is

$$\frac{\|\mathbf{x} - \mathbf{x}^*\|}{\|\mathbf{x}\|} \leq K(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}, \quad \textit{provided that} \quad \mathbf{x} \neq 0,\ \mathbf{b} \neq 0. \tag{3.57}$$

Proof. Since $\mathbf{r} = \mathbf{b} - A\mathbf{x}^*$ and A is nonsingular, then

$$A\mathbf{x} - A\mathbf{x}^* = \mathbf{b} - (\mathbf{b} - \mathbf{r}) = \mathbf{r},$$

which implies that

$$A(\mathbf{x} - \mathbf{x}^*) = \mathbf{r}, \quad \text{or} \quad \mathbf{x} - \mathbf{x}^* = A^{-1}\mathbf{r}. \tag{3.58}$$

Taking norm on both side, gives

$$\|\mathbf{x} - \mathbf{x}^*\| = \|A^{-1}\mathbf{r}\| \leq \|A^{-1}\| \|\mathbf{r}\|.$$

Moreover, since $\mathbf{b} = A\mathbf{x}$, then

$$\|\mathbf{b}\| \leq \|A\| \|\mathbf{x}\|, \qquad \text{or}, \quad \|\mathbf{x}\| \geq \frac{\|\mathbf{b}\|}{\|A\|}.$$

Hence

$$\frac{\|\mathbf{x} - \mathbf{x}^*\|}{\|\mathbf{x}\|} \leq \frac{\|A^{-1}\| \|\mathbf{r}\|}{\|\mathbf{b}\| / \|A\|} \leq K(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

The inequalities (3.56) and (3.57) imply that the quantities $\|A^{-1}\|$ and $K(A)$ can be used to give an indication of the connection between the residual vector and the accuracy of the approximation. If the quantity $K(A) \approx 1$, the relative error will be fairly close to the relative residual. But if $K(A) >> 1$, then the relative error could be many times larger than the relative residual. •

Example 3.56 *Find the condition number of the following matrix (for $n = 2, 3, \ldots$)*

$$A_n = \begin{bmatrix} 1 & 1 \\ 1 & 1 - 1/n \end{bmatrix}.$$

If $n = 2$ and $x^* = [-1.99, 2.99]^T$ be the approximate solution of the linear system $A\mathbf{x} = [1, -0.5]^T$, then find the relative error.

Solution. *We can easily find the inverse of the given matrix as*

$$A_n^{-1} = \frac{1}{(1 - 1/n) - 1} \begin{bmatrix} 1 - 1/n & -1 \\ -1 & 1 \end{bmatrix} = -n \begin{bmatrix} 1 - 1/n & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 - n & n \\ n & -n \end{bmatrix}.$$

Then the l_∞-norm of both matrices A_n and A_n^{-1} are

$$\|A_n\|_\infty = 2 \quad \textit{and} \quad \|A_n^{-1}\|_\infty = 2n,$$

and so the condition number of the matrix can be computed as follows:

$$K(A) = \|A_n\|_\infty \|A_n^{-1}\|_\infty = (2)(2n) = 4n \qquad \textit{and} \qquad \lim_{n\to\infty} K(A) = \infty,$$

which shows that the matrix A_n is obviously ill-conditioned. Here we expect that the relative error in the calculated solution to a linear system of the form $A_n\mathbf{x} = \mathbf{b}$ could be as much as $4n$ times the relative residual.
The residual vector (by taking $n = 2$) can be calculated as

$$\mathbf{r} = \mathbf{b} - A_2\mathbf{x}^* = \begin{pmatrix} 1 \\ -0.5 \end{pmatrix} - \begin{pmatrix} 1 & 1 \\ 1 & 0.5 \end{pmatrix} \begin{pmatrix} -1.99 \\ 2.99 \end{pmatrix} = \begin{pmatrix} 0.000 \\ -0.005 \end{pmatrix},$$

and it gives $\|\mathbf{r}\|_\infty = 0.005$. Now using (3.57), we obtain

$$\frac{\|\mathbf{x} - \mathbf{x}^*\|}{\|\mathbf{x}\|} \le K(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} = (8) \frac{0.005}{1} = 0.0400,$$

which is the required relative error. •

Example 3.57 *Consider a following linear system*

$$\begin{array}{rcrcrcr} x_1 & + & x_2 & - & x_3 & = & 1 \\ x_1 & + & 2x_2 & - & 2x_3 & = & 0 \\ -2x_1 & + & x_2 & + & x_3 & = & -1 \end{array}$$

(a) Discuss the ill-conditioning of the given linear system.
(b) If $\mathbf{x}^* = [2.01, 1.01, 1.98]^T$ be an approximate solution of the given system, then find the residual vector $\mathbf{r}$ and its norm $\|\mathbf{r}\|_\infty$.
(c) Estimate the relative error using (3.57).

Solution. *(a) The matrix A and its inverse is*

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 2 & -1 & 0 \\ 1.5 & -0.5 & 0.5 \\ 2.5 & -1.5 & 0.5 \end{pmatrix}.$$

Then

$$\|A\|_\infty = 5, \ \|A^{-1}\|_\infty = 4.5, \qquad K(A) = \|A\|_\infty \|A^{-1}\|_\infty = (5)(4.5) = 22.5 >> 1,$$

which shows that the matrix is ill-conditioned. Thus the given system is ill-conditioned.

```
>> A = [1 1 -1; 1 2 -2; -2 1 1]; K(A) = norm(A, inf) * norm(inv(A), inf)
```

(b) The residual vector can be calculated as

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}^* = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} - \begin{pmatrix} 1 & 1 & -1 \\ 1 & 2 & -2 \\ -2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2.01 \\ 1.01 \\ 1.98 \end{pmatrix} = \begin{pmatrix} -0.04 \\ -0.07 \\ 0.03 \end{pmatrix},$$

and it gives

$$\|\mathbf{r}\|_\infty = 0.07.$$

```
>> A = [1 1 -1; 1 2 -2; -2 1 1]; b = [1 0 -1]';
>> x0 = [2.01 1.01 1.98]'; r = RES(A, b, x0); rnorm = norm(r, inf);
```

(c) From (3.57), we have

$$\frac{\|\mathbf{x} - \mathbf{x}^*\|}{\|\mathbf{x}\|} \leq K(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

By using above parts (a) and (b) and the value $\|\mathbf{b}\|_\infty = 1$, we obtain

$$\frac{\|\mathbf{x} - \mathbf{x}^*\|}{\|\mathbf{x}\|} \leq (22.5) \frac{(0.07)}{1} = 1.575.$$

```
>> RelErr = (K(A) * rnorm)/norm(b, inf);
```

which is the required relative error. •

Example 3.58 *Consider a linear system $A\mathbf{x} = \mathbf{b}$, where*

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 4 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad \textit{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}.$$

(a) Discuss the conditioning of the given linear system.
(b) Suppose that $\mathbf{b}$ is changed to $\mathbf{b}^* = [1, 1, 1.99]^T$. How large a relative change can this change produce in the solution to $A\mathbf{x} = \mathbf{b}$?

Solution. *(a) Since the matrix A and its inverse is*

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 4 & 0 \\ 1 & 2 & 1 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 4/3 & 1 & -8/3 \\ -1/3 & 0 & 2/3 \\ -2/3 & -1 & 7/3 \end{pmatrix}.$$

Then

$$\|A\|_\infty = 5, \|A^{-1}\|_\infty = 5, \qquad K(A) = \|A\|_\infty \|A^{-1}\|_\infty = (5)(5) = 25.$$

(b) Since the change from $\mathbf{b}$ *to* $\mathbf{b}^*$ *is an error* $\delta\mathbf{b}$*, that is,* $\mathbf{b}^* = \mathbf{b} + \delta\mathbf{b}$*, so*

$$\delta\mathbf{b} = \begin{pmatrix} -0.01 \\ 0 \\ 0 \end{pmatrix} = -\mathbf{r},$$

and the l_∞*-norm of this column matrix is,* $\|\delta\mathbf{b}\|_\infty = 0.01$*. From the equation (3.57), we get*

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{25(0.01)}{2} = 0.1250,$$

the possible relative change in the solution to the given linear system. •

3.7.2 Method to Solve Ill-Conditioned System

If the system is ill-conditioned, an approximate solution can be improved by an iterative technique, called the method of **residual correction** (or **iterative refinement method**). The procedure of the method is defined below.
Let $\mathbf{x}^{(1)}$ be an approximate solution to the system

$$A\mathbf{x} = \mathbf{b}, \tag{3.59}$$

and let $\mathbf{y}$ be a correction to $\mathbf{x}^{(1)}$ so that the exact solution $\mathbf{x}$ satisfies

$$\mathbf{x} = \mathbf{x}^{(1)} + \mathbf{y}.$$

Then by substituting into (3.59), we find that $\mathbf{y}$ must satisfies

$$A\mathbf{y} = \mathbf{r} = \mathbf{b} - A\mathbf{x}^{(1)}, \tag{3.60}$$

where $\mathbf{r}$ is the residual. The system (3.60) can now be solved to give correction $\mathbf{y}$ to the approximation $\mathbf{x}^{(1)}$. Thus the new approximation

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \mathbf{y},$$

will be closer to the solution as compared to $\mathbf{x}^{(1)}$. If necessary, we compute new residual

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}^{(2)},$$

and solve again system (3.60) to get new corrections. Normally two or three iterations are enough for getting exact solution. This iterative method can be used to obtain an improved solution whenever an approximate solution has been obtained by any means.

Example 3.59 *The following linear system*

$$\begin{array}{rcrcl} 1.01x_1 & + & 0.99x_2 & = & 2 \\ 0.99x_1 & + & 1.01x_2 & = & 2 \end{array}$$

has the exact solution $\mathbf{x} = [1, 1]^T$. *The approximate solution due to the Gaussian elimination method is* $\mathbf{x}^{(1)} = [1.01, 1.01]^T$, *and residual* $\mathbf{r}^{(1)} = [-0.02, -0.02]^T$. *Then the solution to the system*

$$A\mathbf{y} = \mathbf{r}^{(1)},$$

using the simple Gaussian elimination method is $\mathbf{y}^{(1)} = [-0.01, -0.01]^T$. *So the new approximation is*

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \mathbf{y}^{(1)} = [1, 1]^T,$$

which is equal to the exact solution just after one iteration. •

3.8 Exercises

1. Solve the following system using the matrix inversion method.

$$\begin{array}{rcrcrcr} x_1 & + & 3x_2 & - & x_3 & = & 4 \\ 5x_1 & - & 2x_2 & - & x_3 & = & -2 \\ 2x_1 & + & 2x_2 & + & x_3 & = & 9 \end{array}$$

2. Use the simple Gaussian elimination method to show that the following system does not have a solution.

$$\begin{array}{rcrcrcr} 3x_1 & + & x_2 & & & = & 1.5 \\ 2x_1 & - & x_2 & - & x_3 & = & 2 \\ 4x_1 & + & 3x_2 & + & x_3 & = & 0 \end{array}$$

3. Solve the Problem 1 using the simple Gaussian elimination method.

4. Solve the following system using the simple Gaussian elimination method.

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

5. For what values of a and b the following linear system has no solution or infinitely many solutions.

$$\begin{array}{rcrcrcr} 2x_1 & + & x_2 & + & x_3 & = & 2 \\ -2x_1 & + & x_2 & + & 3x_3 & = & a \\ 2x_1 & & & - & x_3 & = & b \end{array}$$

6. Find the inverse of each of the following matrix by using simple Gauss elimination method:

$$A = \begin{pmatrix} 3 & 3 & 3 \\ 0 & 2 & 2 \\ 2 & 4 & 5 \end{pmatrix}, \quad B = \begin{pmatrix} 5 & 3 & 2 \\ 3 & 2 & 2 \\ 2 & 6 & 5 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 3 & 4 & 3 \end{pmatrix}.$$

7. Solve Problem 4 using the Gaussian elimination with partial pivoting.

8. For what value(s) of α each of the following matrix A is singular using Doolittle's method.

$$(\mathbf{a})\ A=\begin{pmatrix} 1 & -1 & 2 \\ -1 & 3 & -1 \\ \alpha & -2 & 3 \end{pmatrix},\ (\mathbf{b})\ A=\begin{pmatrix} 1 & 5 & 7 \\ 4 & 4 & \alpha \\ -2 & \alpha & 9 \end{pmatrix},\quad (\mathbf{c})\ A=\begin{pmatrix} 2 & -4 & \alpha \\ 2 & 4 & 3 \\ 4 & -2 & 5 \end{pmatrix}.$$

9. Find the determinant of each of the following matrix using the LU decomposition by Doolittle's method:

$$(\mathbf{a})\quad A=\begin{pmatrix} 2 & 3 & -1 \\ 1 & 2 & 1 \\ 2 & 1 & -6 \end{pmatrix},\quad (\mathbf{b})\quad A=\begin{pmatrix} 1 & -2 & 2 \\ 2 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

10. Find the determinant of each of the following matrix using the LU decomposition by Crout's method:

$$(\mathbf{a})\quad A=\begin{pmatrix} 2 & 2 & -1 \\ 1 & 2 & 1 \\ 2 & 1 & -4 \end{pmatrix},\quad (\mathbf{b})\quad A=\begin{pmatrix} 2 & -1 & 1 \\ 1 & 2 & 2 \\ 2 & 0 & 2 \end{pmatrix}.$$

11. Find $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$ and $\|\mathbf{x}\|_\infty$ for the following vectors.
(a) $[2,-1,-6,3]^T$ **(b)** $[\sin k, \cos k, 3^k]^T$, for a fixed integer k.

12. Find the Jacobi iteration matrix and its l_∞-norm for each of the following matrix.

$$(\mathbf{a})\quad \begin{pmatrix} 11 & -3 & 2 \\ 4 & 10 & 3 \\ -2 & 5 & 9 \end{pmatrix},\qquad (\mathbf{b})\quad \begin{pmatrix} 7 & 1 & 1 \\ 3 & 13 & 2 \\ -4 & 3 & 14 \end{pmatrix},$$

13. Find the Gauss-Seidel iteration matrix and its l_∞-norm for each of the following matrix.

$$(\mathbf{a})\quad \begin{pmatrix} 3 & 0 & 1 \\ 1 & 4 & 0 \\ 0 & 2 & 5 \end{pmatrix},\qquad (\mathbf{b})\quad \begin{pmatrix} 5 & 2 & 1 \\ 4 & 9 & 2 \\ 3 & 1 & 6 \end{pmatrix}.$$

14. Solve the following linear system using the Jacobi method, start with initial approximation $\mathbf{x}^{(0)}=0$ and iterate until $\|\mathbf{x}^{(k+1)}-x^{(k)}\|_\infty \le 10^{-5}$ for each system.

$$\begin{aligned} 4x_1 &- x_2 + x_3 &= 7 \\ 4x_1 &- 8x_2 + x_3 &= -21 \\ -2x_1 &+ x_2 + 5x_3 &= 15 \end{aligned}$$

Find the Jacobi iteration matrix T_J and show that $\|T_J\| < 1$. Use Jacobi method to find first approximate solution $\mathbf{x}^{(1)}$ of the linear system by using $\mathbf{x}^{(0)}=[0,0,0]^T$. Compute error bound $\|\mathbf{x}-\mathbf{x}^{(10)}\|$. Compute the number of steps needed to get the accuracy within 10^{-5}.

15. The following linear systems have $\mathbf{x}$ as the exact solution and $\mathbf{x}^*$ is an approximate solution. Compute $\|\mathbf{x}-\mathbf{x}^*\|_\infty$ and $K(A)\dfrac{\|\mathbf{r}\|_\infty}{\|\mathbf{b}\|_\infty}$, where $\mathbf{r}=\mathbf{b}-A\mathbf{x}^*$ is the residual vector.

$$\begin{array}{rcrcll} 0.89x_1 & + & 0.53x_2 & = & 0.36 & \qquad \mathbf{x} = [1,-1]^T \\ 0.47x_1 & + & 0.28x_2 & = & 0.19 & \qquad \mathbf{x}^* = [0.702,-0.500]^T \end{array}$$

16. Discuss the ill-conditioning (stability) of the linear system

$$\begin{array}{rcrcl} 1.01x_1 & + & 0.99x_2 & = & 2 \\ 0.99x_1 & + & 1.01x_2 & = & 2 \end{array}$$

If $\mathbf{x}^* = [2,0]^t$ be an approximate solution of the system, then find the residual vector $\mathbf{r}$ and estimate the relative error.

17. The following linear systems have $\mathbf{x}$ as the exact solution and $\mathbf{x}^*$ is an approximate solution. Compute $\|\mathbf{x}-\mathbf{x}^*\|_\infty$ and $K(A)\dfrac{\|\mathbf{r}\|_\infty}{\|\mathbf{b}\|_\infty}$, where $\mathbf{r}=\mathbf{b}-A\mathbf{x}^*$ is the residual vector.

$$\begin{array}{rcrcll} 0.89x_1 & + & 0.53x_2 & = & 0.36 & \qquad \mathbf{x} = [1,-1]^T \\ 0.47x_1 & + & 0.28x_2 & = & 0.19 & \qquad \mathbf{x}^* = [0.702,-0.500]^T \end{array}$$

Chapter 4

Polynomial Interpolation and Approximation

4.1 Introduction

In this chapter we describe the numerical methods for the approximation of functions other than the elementary functions. The main purpose of these techniques is to replace a complicated function by one which is simpler and more manageable. We sometimes know the value of a function $f(x)$ at a set of points (say, $x_0 < x_1 < x_2 \cdots < x_n$) but we do not have an analytic expression for $f(x)$ that let us calculate its value at an arbitrary point. We concentrate on techniques which may be adapted if, for example, we have a table of values of function may have been obtained from some physical measurement or some experiments or long numerical calculation that can not be cast into a simple functional form. The task now is to estimate $f(x)$ for an arbitrary point x by, in some sense, drawing a smooth curve through (and perhaps beyond) the data points x_i. If the desired x is in between the largest and smallest of the data point, then the problem is called *interpolation*; and if x is outside that range, it is called *extrapolation*. In this chapter we shall restrict our attention to interpolation. It is a rational process generally used in estimating a missing functional value by taking a weighted average of known functional values at neighbouring data points.
Interpolation scheme must model the function, in between or beyond the known data point, by some plausible functional form. The form should be sufficiently general so as to be able to approximate large classes of functions which might arise in practice. The functional form are polynomials, trigonometric functions, rational functions and exponential functions. However, we shall restrict our attention to *polynomials*. The polynomial functions are widely used in practice, since they are easy to determine, evaluate, differentiate, and integrable. *Polynomial interpolation* provides some mathematical tools that can be used in developing methods for approximation theory, numerical differentiation, numerical integration, and numerical solutions of ordinary differential equations and partial differential equations. A set of data points we consider here may be *equally spaced or unequally spaced* in the independent variable x. Several procedures can be used to fit approximation polynomials for both cases. For example, Lagrange interpolatory polynomial and Newton divided-difference interpolatory polynomial. These methods are quite easy to apply. The general form of a *nth*-degree polynomial is

$$f(x) = p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n, \tag{4.1}$$

where n denotes the degree of the polynomial; and $a_0, a_1, \ldots, a_n$ are constants coefficients. Since there are $(n+1)$ coefficients, so $(n+1)$ data points are required to obtain unique value for the coefficients. The important property of polynomials that makes them suitable for approximating functions is due to the following theorem called, the *Weierstrass approximation theorem.*

Theorem 4.1 (Weierstrass Approximation Theorem)

If $f(x)$ is a continuous function in the closed interval $[a,b]$ then for every $\epsilon > 0$ there exists a polynomial $p_n(x)$, where the value of n depends on the value of ϵ, such that for all x in $[a,b]$,

$$|f(x) - p_n(x)| < \epsilon. \tag{4.2}$$

Consequently, any continuous function can be approximated to any accuracy by a polynomial of high enough degree. •

4.2 Polynomial Interpolation

Suppose we have given a set of $(n+1)$ data points relating a dependent variables $f(x)$ to an independent variable x as follows

x	x_0	x_1	$\cdots$	x_n
$f(x)$	$f(x_0)$	$f(x_1)$	$\cdots$	$f(x_n)$

Generally the data points $x_0, x_1, \ldots, x_n$ are arbitrary and assume the interval between two adjacent points is not the same (unequally spaced) and assume that the data points are organized in such a way that $x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n$. But some times this is not happen.
When the data points in a given functional relationship are not equally spaced, the interpolation problem becomes more difficult to solve. The basis for this assertion lies in the fact that the interpolating polynomial coefficient will depend on the functional values as well as on the data points given in the table.
When we are given the value of the function f at the points $x_0, x_1, x_2, \cdots, x_{n-1}, x_n$ and want to find a polynomial $p_n(x)$ of degree n that agrees with f at all points; that is,

$$p_n(x_i) = f(x_i), \qquad i = 0, 1, \ldots, n. \tag{4.3}$$

We will call this type of interpolation simple interpolation.

There are many ways in which interpolating polynomials can be found. In the method of undetermined coefficients we consider the coefficients in (4.1) as unknowns that can be determined by imposing conditions on the polynomial. In simple interpolation we have $n+1$ conditions specified by (4.3) to be satisfied, and $n+1$ coefficients in (4.1)to select. When we write this down explicitly, we get the linear system

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix}. \tag{4.4}$$

For example, to approximate $f(x) = e^x$ in $[0, 1]$ by a second degree polynomial, we can use the interpolation points $0, \frac{1}{2}$, and 1. The system to be solved then is

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{4} \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \sqrt{e} \\ e \end{pmatrix}.$$

Solving this linear system we obtain the approximation

$$p_2(x) = 1 + 0.8766x + 0.8417x^2.$$

The method of undetermined coefficients is simple and intuitive, and gives results with minimum effort. But there are reasons why it is not always suitable. One of the reasons is that the system (4.4) tends to become ill-conditioned quickly as n increases and is not suitable for large n. Another reason is that it does not give a very explicit form of the polynomial, and so is hard to use for analysis.

A second way of getting the interpolating polynomial, one that overcomes some of these limitations of the undetermined coefficients method, is the Lagrange form. Now we discuss this form in some more details.

4.2.1 Lagrange Interpolating Polynomials

It is one of the popular and well known interpolation method to approximate the functions at an arbitrary point x. The Lagrange interpolation method provides a direct approach for determining interpolated values regardless of the data points spacing, that is, it can be fitted to unequally spaced or equally spaced data. To discuss about the Lagrange interpolation method, we start with a simplest form of interpolation, that is, *linear interpolation.* The interpolated value is obtained from the equation of straight line that passes through two tabulated values, one each side of required value. This straight line is a first-degree polynomial. The problem of determining a polynomial of degree one that passes through the distinct points (x_0, y_0) and (x_1, y_1) is the same as approximating the function $f(x)$ for which $f(x_0) = y_0$ and $f(x_1) = y_1$ by means of first degree polynomial interpolation.

4.2.1.1 Linear Lagrange Interpolating Polynomial

Let us consider the construction of a linear polynomial $p_1(x)$ passing through two data points $(x_0, f(x_0))$ and $(x_1, f(x_1))$, see Figure 4.1. Consider a linear polynomial of the form

$$f(x) = p_1(x) = a_0 + a_1 x. \tag{4.5}$$

Since a polynomial of degree one has two coefficients, so one might expect to be able to choose two conditions, which satisfy

$$p_1(x_k) = f(x_k); \qquad k = 0, 1.$$

When $p_1(x)$ passes through point $(x_0, f(x_0))$, we have

$$p_1(x_0) = a_0 + a_1 x_0 = y_0 = f(x_0),$$

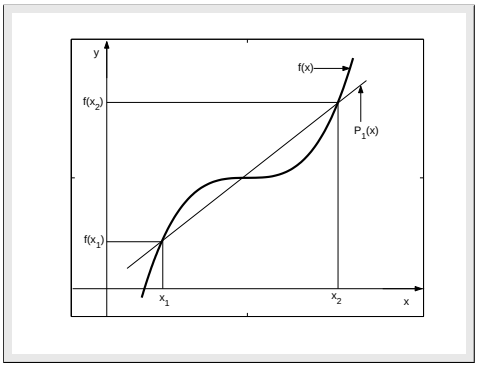


Figure 4.1: Linear Lagrange interpolation.

and if it passes through point $(x_1, f(x_1))$, we have

$$p_1(x_1) = a_0 + a_1 x_1 = y_1 = f(x_1).$$

Solving last two equations, gives a unique solution

$$a_0 = \frac{x_0 y_1 - x_1 y_0}{x_0 - x_1} \quad \text{and} \quad a_1 = \frac{y_1 - y_0}{x_1 - x_0}. \tag{4.6}$$

Putting these values in (4.5), we have

$$f(x) = p_1(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) y_0 + \left(\frac{x - x_0}{x_1 - x_0}\right) y_1 = L_0(x) f(x_0) + L_1(x) f(x_1), \tag{4.7}$$

where

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{and} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}. \tag{4.8}$$

Note that when $x = x_0$, then $L_0(x_0) = 1$ and $L_1(x_0) = 0$. Similarly, when $x = x_1$, then $L_0(x_1) = 0$ and $L_1(x_1) = 1$. The polynomial (4.7) is known as *linear Lagrange interpolating polynomial* and (4.8) is called the *Lagrange coefficient polynomials.*

4.2.1.2 Quadratic Lagrange Interpolating Polynomial

When $p_2(x)$ passes through three points $(x_0, f(x_0)), (x_1, f(x_1))$ and $(x_2, f(x_2))$, we have quadratic Lagrange polynomial as follows

$$f(x) = p_2(x) = L_0(x) f(x_0) + L_1(x) f(x_1) + L_2(x) f(x_2), \tag{4.9}$$

where the Lagrange coefficients are define as follows:

$$\begin{aligned} L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \\ L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \\ L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}. \end{aligned} \tag{4.10}$$

4.2.1.3 Cubic Lagrange Interpolating Polynomial

Similarly, when $p_3(x)$ passes through 4 points $(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2))$ and $(x_3, f(x_3))$, we have the following cubic Lagrange polynomial as follows

$$f(x) = p_3(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + L_3(x)f(x_3), \tag{4.11}$$

where the Lagrange coefficients are define as follows:

$$\begin{aligned} L_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}, \\ L_1(x) &= \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}, \\ L_2(x) &= \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}, \\ L_3(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}. \end{aligned} \tag{4.12}$$

4.2.1.4 Nth Degree Lagrange Interpolating Polynomial

To generalize the concept of the Lagrange interpolation, consider the construction of a polynomial $p_n(x)$ of degree at most n that passes through $(n+1)$ distinct points $(x_0, f(x_0)), \ldots, (x_n, f(x_n))$ (see Figure 4.2) and satisfy the interpolation conditions

$$p_n(x_k) = f(x_k); \qquad k = 0, 1, 2, \ldots, n. \tag{4.13}$$

Assume that there exists polynomial $L_k(x)$ $(k = 0, 1, 2, \ldots, n)$ of degree n having the property

$$L_k(x_j) = \begin{cases} 0 & \text{for} \quad k \neq j, \\ 1 & \text{for} \quad k = j, \end{cases} \tag{4.14}$$

and

$$\sum_{k=0}^{n} L_k(x) = 1. \tag{4.15}$$

The polynomial $p_n(x)$ is given by

$$\begin{aligned} f(x) = p_n(x) &= L_0(x)f(x_0) + L_1(x)f(x_1) + \cdots + L_{i-1}(x)f(x_{i-1}) \\ &+ L_i(x)f(x_i) + \cdots + L_n(x)f(x_n) = \sum_{k=0}^{n} L_k(x)f(x_k). \end{aligned} \tag{4.16}$$

It is clearly a polynomial of degree at most n and satisfy the conditions (4.13) since

$$\begin{aligned} p_n(x_i) = L_0(x_i)f(x_0) &+ L_1(x_i)f(x_1) + \cdots + L_{i-1}(x_i)f(x_{i-1}) \\ &+ L_i(x_i)f(x_i) + \cdots + L_n(x_i)f(x_n), \end{aligned}$$

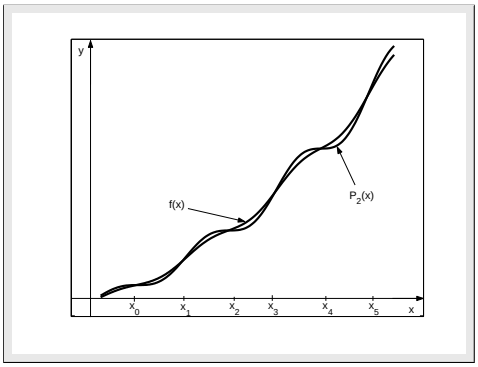


Figure 4.2: General Lagrange interpolation.

which implies that

$$p_n(x_i) = f(x_i).$$

It remains to be shown how the polynomial $L_i(x)$ can be constructed so that they satisfy (4.14). If $L_i(x)$ is to satisfies (4.14), then it must contain a factor

$$(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n). \tag{4.17}$$

Since this expression has exactly n terms and $L_i(x)$ is a polynomial of degree n, we can deduce that

$$L_i(x) = A_i(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n), \tag{4.18}$$

for some multiplicative constant A_i. Let $x = x_i$, then the value of A_i is chosen so that

$$A_i = \frac{1}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}, \tag{4.19}$$

where none of the terms in denominator can be zero from the assumption of distinct points. Hence

$$L_i(x) = \prod_{k=0}^{n} \left(\frac{x - x_k}{x_i - x_k} \right), \quad i \neq k. \tag{4.20}$$

The interpolating polynomial can now be readily evaluated by substituting (4.20) into (4.16) to give

$$f(x) = p_n(x) = \sum_{i=0}^{n} \prod_{k=0}^{n} \left(\frac{x - x_k}{x_i - x_k} \right) f(x_i), \qquad i \neq k. \tag{4.21}$$

This formula is called the Lagrange interpolation formula of degree n and the terms in (4.20) are called the Lagrange coefficient polynomials.

4.2.1.5 Uniqueness of Lagrange Interpolating Polynomial

To show the *uniqueness* of the interpolating polynomial $p_n(x)$, we suppose that in addition to the polynomial $p_n(x)$ the interpolation problem has another solution $q_n(x)$ of degree $\leq n$ whose graph passes through $(x_i, y_i),\ i = 0, 1, \ldots, n$. Then define

$$r_n(x) = p_n(x) - q_n(x),$$

of the degree not greater than n. Since

$$r_n(x_i) = p_n(x_i) - q_n(x_i) = f(x_i) - f(x_i) = 0,$$

the polynomial $r_n(x)$ vanishes at $n+1$ point. But by using the following well known result from the theory of equations: *"If a polynomial of degree n vanishes at $n+1$ distinct points, then the polynomial is identically zero"*. Hence $r_n(x)$ vanishes identically, or equivalently, $p_n(x) = q_n(x)$.

Example 4.1 *Let $f(x) = \cos(x^2) - x + 3$. Show that $f(x) = 0$ has a root in the interval $[2.5, 3.0]$. Use linear Lagrange interpolating polynomial to find its approximation.*

Solution. *Let $x = \alpha$ be the root of the equation $\cos(x^2) - x + 3 = 0$, and to show it lies in the interval $[2.5, 3.0]$, we do as follows: Since the given function $f(x) = \cos(x^2) - x + 3$ is defined in the interval $[2.5, 3.0]$ and so it is continuous on $[2.5, 3.0]$. Starting with $x_0 = 2.5$ and $x_1 = 3.0$, we compute:*

$$\begin{aligned} x_0 &= 2.5: \qquad f(2.5) = 1.499 \\ x_1 &= 3.0: \qquad f(3.0) = -0.911, \end{aligned}$$

and since $f(2.5)f(3.0) < 0$, so that a root of $f(x) = 0$ lies in the interval $[2.5, 3.0]$. Now to find its approximation using linear Lagrange interpolating polynomial, we do as:

$$p_1(x) = L_0(x)f(x_0) + L_1(x)f(x_1) = L_0(x)(1.499) + L_1(x)(-0.911),$$

where the Lagrange coefficients can be calculate as follows:

$$L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)} = \frac{(\alpha - 3)}{(2.5 - 3)}, \quad L_1(x) = \frac{(x - x_0)}{(x_1 - x_0)} = \frac{(\alpha - 2.5)}{(3 - 2.5)}.$$

Thus

$$f(x) = p_1(x) = \frac{1}{-0.5}(\alpha - 3)(1.499) + \frac{1}{0.5}(\alpha - 2.5)(-0.911) = 0.$$

Solving for α, we get $4.82\alpha = 13.549$, which gives, $\alpha \approx 2.811$, the required approximation of the root of the equation in the interval $[2.5, 3.0]$. •

Example 4.2 *Let $f(x) = 0$ be defined on the three numbers $-h, 0, h$, where $h \neq 0$. Use Lagrange interpolating polynomial to construct the polynomial $p(x)$ which interpolate $f(x)$ at the given numbers. Then show that this polynomial can be written in the following form*

$$f(x) = p(x) = \frac{1}{2h^2}[f(-h) - 2f(0) + f(h)]x^2 + \frac{1}{2h}[f(h) - f(-h)]x + f(0).$$

Solution. *Given three distinct points $x_0 = -h, x_1 = 0$ and $x_2 = h$ and using the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}f(x_2),$$

at these data points, we get

$$f(x) = p_2(x) = \frac{(x^2 - xh)}{(-h)(-2h)}f(-h) + \frac{(x^2 - h^2)}{(h)(-h)}f(0) + \frac{(x^2 + xh)}{(2h)(h)}f(h).$$

Separating the coefficients of x^2, x and constant term, we get

$$f(x) = p_2(x) = \left(\frac{f(-h)}{2h^2} + \frac{f(0)}{-h^2} + \frac{f(h)}{2h^2}\right)x^2 + \left(\frac{-hf(-h)}{2h^2} + \frac{hf(h)}{2h^2}\right)x + \left(\frac{-h^2 f(0)}{-h^2}\right),$$

after simplifying, we obtain

$$f(x) = p(x) = \frac{1}{2h^2}[f(-h) - 2f(0) + f(h)]x^2 + \frac{1}{2h}[f(h) - f(-h)]x + f(0). \quad \bullet$$

Example 4.3 *Let $p_2(x)$ be the quadratic Lagrange interpolating polynomial for the data: $(1, 2), (2, 3), (3, \alpha)$. Find α if the constant term in $p_2(x)$ is 5. Find the approximation of $f(2.5)$.*

Solution. *Consider the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = 2L_0(x) + 3L_1(x) + \alpha L_2(x),$$

where the Lagrange coefficients can be calculate as follows:

$$\begin{aligned} L_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 2)(x - 3)}{(1 - 2)(1 - 3)} = \frac{1}{2}(x^2 - 5x + 6), \\ L_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x - 1)(x - 3)}{(2 - 1)(2 - 3)} = -(x^2 - 4x + 3), \\ L_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x - 1)(x - 2)}{(3 - 1)(3 - 2)} = \frac{1}{2}(x^2 - 3x + 2). \end{aligned}$$

Thus

$$f(x) = p_2(x) = \frac{1}{2}(x^2 - 5x + 6)(2) - (x^2 - 4x + 3)(3) + \frac{1}{2}(x^2 - 3x + 2)(\alpha).$$

Separating the coefficients of x^2, x and constant term, we get

$$f(x) = p_2(x) = \left(-2 + \frac{\alpha}{2}\right)x^2 + \left(7 - \frac{3\alpha}{2}\right)x + (-3 + \alpha).$$

Since the given value of the constant term is 5, using this, we get, $(-3 + \alpha) = 5$, which gives, $\alpha = 8$. Thus by using $\alpha = 8$ and $x = 2.5$, we have

$$f(2.5) \approx p_2(2.5) = 12.50 - 12.50 + 5 = 5,$$

the required approximation of the function. $\bullet$

Example 4.4 *Let $f(x) = x + \dfrac{1}{x}$, with points $x_0 = 1, x_1 = 1.5, x_2 = 2.5$ and $x_3 = 3$. Find the quadratic Lagrange polynomial for the approximation of $f(2.7)$. Also, find the relative error.*

Solution. *Consider the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2).$$

Since the given interpolating point is $x = 2.7$, therefore, the best three points for the quadratic polynomial should be as follows:

$$x_0 = 1.5,\ f(x_0) = 2.1667, \quad x_1 = 2.5,\ f(x_1) = 2.9, \quad x_2 = 3,\ f(x_2) = 3.3333.$$

So using these values, we have

$$f(x) = p_2(x) = 2.1667L_0(x) + 2.9L_1(x) + 3.3333L_2(x),$$

where

$$\begin{aligned} L_0(x) &= \frac{(x-2.5)(x-3)}{(1.5-2.5)(1.5-3)} = \frac{1}{1.5}(x^2 - 5.5x + 7.5), \\ L_1(x) &= \frac{(x-1.5)(x-3)}{(2.5-1.5)(2.5-3)} = \frac{1}{-0.5}(x^2 - 4.5x + 4.5), \\ L_2(x) &= \frac{(x-1.5)(x-2.5)}{(3-1.5)(3-2.5)} = \frac{1}{0.75}(x^2 - 4x + 3.75). \end{aligned}$$

Using these Lagrange coefficients in the polynomial and after simplifying, gives

$$f(x) = p_2(x) = 0.0889x^2 + 0.3776x + 1.4003,$$

which is the required quadratic polynomial, and $f(2.7) \approx p_2(2.7) = 3.0679$. The relative error is

$$\frac{|f(2.7) - p_2(2.7)|}{|f(2.7)|} = \frac{|3.0704 - 3.0679|}{|3.0704|} = 0.0008. \qquad \bullet$$

Note that the sum of the Lagrange coefficients is equal to 1 as it should be

$$L_0(2.7) + L_1(2.7) + L_2(2.7) = -0.0400 + 0.7200 + 0.3200 = 1.$$

Using MATLAB command the above results can be reproduce as follows:

$>> x = [1.5\ 2.5\ 3];\ y = x + 1/x;\ x0 = 2.7;\ sol = lint(x, y, x0)$

Example 4.5 *Find the unique quadratic Lagrange polynomial $p_2(x)$ such that $f(1) = 1, f(3) = 27, f(4) = 64$. Determine the point x at which the polynomial has a minimum.*

Solution. *Consider the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2),$$

which can be written as

$$f(x) = p_2(x) = (1)L_0(x) + (27)L_1(x) + (64)L_2(x),$$

where

$$\begin{aligned} L_0(x) &= \frac{(x-3)(x-4)}{(1-3)(1-4)} = \frac{1}{6}(x^2-7x+12), \\ L_1(x) &= \frac{(x-1)(x-4)}{(3-1)(3-4)} = -\frac{27}{2}(x^2-5x+4), \\ L_2(x) &= \frac{(x-1)(x-3)}{(4-1)(4-3)} = \frac{64}{3}(x^2-4x+3). \end{aligned}$$

Using these Lagrange coefficients in the polynomial and after simplifying, gives

$$f(x) = p_2(x) = 8x^2 - 19x + 12.$$

To determine the minimum of the polynomial, we do as

$$f'(x) = 0, \qquad 16x - 19 = 0, \qquad x = 19/16 = 1.1875,$$

at which the polynomial

$$f(1.1875) = p_2(1.1875) = 8(1.1875)^2 - 19(1.1875) + 12 = 12.59375,$$

has minimum because $f''(1.1875) = 16(1.1875) = 19 > 0$. •

Example 4.6 *Using the quadratic Lagrange interpolation formula to find the numbers* A, B *and* C *such that* $p_2(1.4) = Af(0) + Bf(1) + Cf(2)$. *If* $f(0) = 1, f(1) = 2$ *and* $f(2) = 3$, *then find the approximation of* $f(1.4)$.

Solution. *Consider the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = Af(x_0) + Bf(x_1) + Cf(x_2).$$

Using the given values as $x_0 = 0, x_1 = 1, x_2 = 2$ *and the interpolating point* $x = 1.4$, *we obtain*

$$\begin{aligned} A = L_0(1.4) &= \frac{(1.4-1)(1.4-2)}{(0-1)(0-2)} = -0.12, \\ B = L_1(1.4) &= \frac{(1.4-0)(1.4-2)}{(1-0)(1-2)} = 0.84, \\ C = L_2(1.4) &= \frac{(1.4-0)(1.4-1)}{(2-0)(2-1)} = 0.28. \end{aligned}$$

Thus

$$f(1.4) \approx p_2(1.4) = (-0.12)(1) + (0.84)(2) + (0.28)(3) = 2.4,$$

the required approximation of the function at the point $x = 1.4$. •

Example 4.7 *Consider the following table:*

x	0	3	7
$f(x)$	2	4	19

(a) Construct the quadratic Lagrange polynomial $p_2(x) = ax^2 + bx + c$ to approximate $f(x)$.
(b) Use the polynomial in part (a) to interpolate the function $f(x)$ at $x = 4$.

Solution. *(a) Obviously, a quadratic polynomial can be determined so that it passes through the three points. Consider the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = 2L_0(x) + 4L_1(x) + 19L_2(x). \quad (4.22)$$

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned} L_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{1}{21}(x^2 - 10x + 21), \\ L_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = -\frac{1}{12}(x^2 - 7x), \\ L_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{1}{28}(x^2 - 3x). \end{aligned}$$

Putting these values of the Lagrange coefficients in (4.22), we have

$$f(x) = p_2(x) = \frac{1}{84}(37x^2 - 55x + 168),$$

(with $a = 37/84, b = -55/84, c = 2$) which is the required quadratic Lagrange polynomial.
(b) Now take $x = 4$ in the above polynomial, we obtain

$$f(4) \approx p_2(4) = \frac{1}{84}\Big[37(4)^2 - 55(4) + 168\Big] = 6.4286,$$

which is the required estimate value of $f(4)$. •

By using the following MATLAB commands we can easily find the value of the polynomial at the given point as

```
>> CP = [37/84  -55/84 168/84]; Sol = polyval(CP,4)
```

Note that the sum of the Lagrange coefficients is equal to 1 as it should be

$$L_0(4) + L_1(4) + L_2(4) = -\frac{1}{7} + 1 + \frac{1}{7} = 1.$$

Using MATLAB command the above results can be reproduce as follows:

```
>> x = [0 3 7]; y = [2 4 19]; x0 = 4; sol = lint(x,y,x0)
```

Program 4.1
MATLAB m-file for the Lagrange Interpolation Method
function fi=lint(x,y,x0)
dxi=x0-x; m=length(x); L=zeros(size(y));
$L(1) = prod(dxi(2:m))/prod(x(1)-x(2:m))$;
$L(m) = prod(dxi(1:m-1))/prod(x(m)-x(1:m-1))$;
for j=2:m-1
$num = prod(dxi(1:j-1)) * prod(dxi(j+1:m))$;
$dem = prod(x(j)-x(1:j-1)) * prod(x(j)-x(j+1:m))$;
L(j)=num/dem; end; $fi = sum(y.*L)$;

Example 4.8 *Construct the table for* $(\alpha, M(\alpha))$ *by evaluating the integral at* $\alpha = 1, 3, 5, 6$.

$$M(\alpha) = \int_0^1 (\alpha - e^x)\, dx.$$

Use the constructed table to find the best approximation of $M(4)$ *by using quadratic Lagrange polynomial. Compute the absolute error.*

Solution. *Since* $M(\alpha) = \int_0^1 (\alpha - e^x)\, dx = (\alpha x - e^x)\Big|_0^1 = \alpha - e + 1$. *Thus so by using the given values of* α, *we get*

α	1.00	3.00	5.00	6.00
$M(\alpha)$	−0.7183	1.2817	3.2817	4.2817

Since a quadratic polynomial can be determined so that it passes through the three points, let us consider the best form of the constructed table for the quadratic Lagrange interpolating polynomial to approximate $M(4)$ *as*

α	3.00	5.00	6.00
$M(\alpha)$	1.2817	3.2817	4.2817

So using the quadratic Lagrange interpolating polynomial

$$M(\alpha) = p_2(\alpha) = L_0(\alpha)f(\alpha_0) + L_1(\alpha)f(\alpha_1) + L_2(\alpha)f(\alpha_2), \tag{4.23}$$

$$M(4) \approx p_2(4) = 1.2817L_0(4) + 3.2817L_1(4) + 4.2817L_2(4). \tag{4.24}$$

The Lagrange coefficients can be calculate as follows:

$$L_0(4) = \frac{(4-5)(4-6)}{(3-5)(3-6)} = \frac{1}{3},\ L_1(4) = \frac{(4-3)(4-6)}{(5-3)(5-6)} = 1,\ L_2(4) = \frac{(4-3)(4-5)}{(6-3)(6-5)} = -\frac{1}{3}.$$

Putting these values of the Lagrange coefficients in (4.24), we obtain

$$M(4) \approx p_2(4) = \frac{1}{3}(1.2817) + 1(3.2817) - \frac{1}{3}(4.2817) = 2.2817,$$

which is the required approximation of $M(4)$ *by the quadratic interpolating polynomial.*
From the given integral we can obtained the exact value as follows

$$M(4) = \int_0^1 (4 - e^x)\, dx = (4x - e^x)\Big|_0^1 = 5 - e = 2.2817,$$

so $|M(4) - p_2(4)| = |2.2817 - 2.2817| = 0.0000$, *up to four decimal places.* •

Example 4.9 *Consider the following table:*

x	1	2	3
$f(x)$	2	3	5

If $f(x) = p_2(x) = a_0 + a_1x + a_2x^2$, then find the approximation of $f(1.5)$.

Solution. *Using values of $f(x)$ at $x = 1, 2$ and 3, we get*

$$a_0 + a_1 + a_2 = 2, \quad 3a_0 + 2a_1 + 4a_2 = 3, \quad a_0 + 3a_1 + 9a_2 = 4.$$

Now solving this linear system for a_0, a_1, a_2, we obtain, $a_0 = 2,\ a_1 = -0.5,\ a_2 = 0.5$. Thus

$$f(x) = p_2(x) = 2 - 0.5x + 0.5x^2, \quad f(1.5) \approx p_2(1.5) = 2 - 0.75 + 1.125 = 2.375,$$

the required approximation of $f(1.5)$. •

Example 4.10 *Find the missing term in the following table using best Lagrange polynomial:*

x	0	1	2	6
$f(x)$	1	2	?	3

Solution. *Let $x_0 = 0, x_1 = 1$ and $x_2 = 6$, then obviously, a quadratic polynomial can be determined so that it passes through these three points. Quadratic Lagrange polynomial is*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = L_0(x) + 2L_1(x) + 3L_2(x). \tag{4.25}$$

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned}
L_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-1)(x-6)}{(0-1)(0-6)} = \frac{1}{6}(x^2-7x+6),\\
L_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-0)(x-6)}{(1-0)(1-6)} = -\frac{1}{5}(x^2-6x),\\
L_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-0)(x-1)}{(6-0)(6-1)} = \frac{1}{30}(x^2-x).
\end{aligned}$$

Putting these values of the Lagrange coefficients in (4.25), we have

$$f(x) = p_2(x) = \frac{1}{6}(x^2-7x+6)(1) - \frac{1}{5}(x^2-6x)(2) + \frac{1}{30}(x^2-x)(3) = \frac{1}{15}(-2x^2+17x+15),$$

which is the required quadratic interpolating polynomial. The above polynomial at $x = 2$ is

$$f(2) \approx p_2(2) = \frac{1}{15}\Big[-2(2)^2 + 17(2) + 15\Big] = \frac{41}{15} = 2.7333,$$

which is the missing term in the table. We can also find the missing number by considering a quadratic polynomial of the form

$$f(x) = p_2(x) = a + bx + cx^2.$$

If the curve passes through the points $x = 0, 1, 6$, then we have

$$1 = a, \quad 2 = a + b + c, \quad 3 = a + 6b + 36c.$$

Solving this system for a, b and c, we obtain, $a = 1,\ b = \frac{17}{15},\ c = -\frac{2}{15}$. Thus

$$f(x) = p_2(x) = 1 + \frac{17}{15}x - \frac{2}{15}x^2, \quad f(2) \approx p_2(2) = 2.7333,$$

the required value of the missing term. •

Example 4.11 *The nonlinear equation $x - 9^{-x} = 0$ has a solution in $[0, 1]$. Compute the Lagrange polynomial on $x_0 = 0, x_1 = 0.5$ and $x_2 = 1$ and then use it to find the approximate solution to the given equation which lies in the given interval $[0, 1]$.*

Solution. *Let the function $f(x) = x - 9^{-x}$ and at the given points, we have a table of the form*

x	0	0.5	1
$f(x)$	-1	1/6	8/9

So using the quadratic Lagrange interpolating polynomial

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = -L_0(x) + \frac{1}{6}L_1(x) + \frac{8}{9}L_2(x), \quad (4.26)$$

where the values of the Lagrange coefficients can be calculate as follows:

$$\begin{aligned} L_0(x) &= \frac{(x - 0.5)(x - 1)}{(0 - 0.5)(0 - 1)} = 2x^2 - 3x + 1, \\ L_1(x) &= \frac{(x - 0)(x - 1)}{(0.5 - 0)(0.5 - 1)} = -4x^2 + 4x, \\ L_2(x) &= \frac{(x - 0)(x - 0.5)}{(1 - 0)(1 - 0.5)} = 2x^2 - x. \end{aligned}$$

Putting these values of the Lagrange coefficients in (4.26), we have

$$f(x) = p_2(x) = \frac{1}{18}(-16x^2 + 50x - 18).$$

Now setting this polynomial equal to zero (because $f(x) = 0$), we get

$$0 = p_2(x) = \frac{1}{18}(-16x^2 + 50x - 18), \qquad \textit{gives} \qquad 8x^2 - 25x + 9 = 0.$$

Now solving this quadratic equation, one can get the values, $x_1 = 2.70985$ and $x_2 = 0.41515$. Thus the approximate solution to the given equation in $[0, 1]$ is $x_2 = 0.41515$. •

Note that we can use Lagrange interpolating polynomial to express the given rational function $R(x) = \frac{f(x)}{g(x)}$ as sums of partial fractions. For this we have to construct the table for the function $f(x)$ at x values (the zeros of the denominator function $g(x)$).

Example 4.12 *Use quadratic Lagrange polynomial, express the rational function* $R(x) = \dfrac{f(x)}{g(x)} = \dfrac{2x^2 - 9x - 9}{x^3 - 9x}$ *as sums of partial fractions and determine the values of* $A, B,$ *and* C*:*

$$\frac{2x^2 - 9x - 9}{x^3 - 9x} = \frac{A}{(x+3)} + \frac{B}{(x-0)} + \frac{C}{(x-3)}.$$

Solution. *Given rational function is of the form*

$$\frac{f(x)}{g(x)} = \frac{2x^2 - 9x - 9}{x^3 - 9x} = \frac{2x^2 - 9x - 9}{(x+3)(x-0)(x-3)},$$

and the zeros of the function $g(x)$ *as* $x = -3, 0, 3$*. So the table gets the form*

x	-3	0	3
$f(x)$	36	-9	-18

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned}
L_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-0)(x-3)}{(-3-0)(-3-3)} = \frac{1}{18}(x-0)(x-3),\\
L_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x+3)(x-3)}{(0+3)(0-3)} = -\frac{1}{9}(x+3)(x-3),\\
L_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x+3)(x-0)}{(3+3)(3-0)} = \frac{1}{18}(x+3)(x-0).
\end{aligned}$$

By using the quadratic Lagrange interpolating formula

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = 36L_0(x) - 9L_1(x) - 18L_2(x_2),$$

the polynomial $f(x)$ *is given by*

$$f(x) = 2(x-0)(x-3) + (x+3)(x-3) - (x+3)(x-0).$$

Thus

$$R(x) = \frac{2(x-0)(x-3) + (x+3)(x-3) - (x+3)(x-0)}{(x+3)(x-0)(x-3)} = \frac{2}{(x+3)} + \frac{1}{(x-0)} - \frac{1}{(x-3)},$$

is the sums of partial fractions of the given rational function with $A = 2, B = 1,$ *and* $C = -1$. •

We can easily check this results by using MATLAB command as follows:

```
>> num = [0 2 -9 -9]; den = [1 0 -9 0]; [c, z] = residue(num, den)
```

4.2.1.6 Error Formula of Lagrange Polynomial

As with any numerical technique, it is important to obtain bounds for the errors involved. Now we discuss the error term when the Lagrange polynomial is used to approximate continuous function $f(x)$. It is not possible, in general, to say how accurately the interpolating polynomial p_n approximates given function f. All can be said with certainty is that $f(x)-p_n(x)=0$ at $x=x_0,x_1,\ldots,x_n$. However, it is sometimes possible to obtain a bound on the error $f(x)-p_n(x)$ at an intermediate point x using the following theorem.

Error Formulas of Linear, Quadratic and Cubic Lagrange Polynomials

If $f(x)$ has second, third and fourth derivatives on interval I and if it is approximated by the polynomials $p_1(x), p_2(x), p_3(x)$ passing respectively, through $2, 3, 4$ data points on I, then the errors E_1, E_2, E_3 are given by

$$E_1 = f(x) - p_1(x) = \frac{f''(\eta(x))}{2!}(x-x_0)(x-x_1), \quad \eta(x) \in I, \tag{4.27}$$

where $p_1(x)$ is the linear Lagrange polynomial (4.7) and a unknown point $\eta(x) \in (x_0, x_1)$.

$$E_2 = f(x) - p_2(x) = \frac{f'''(\eta(x))}{3!}(x-x_0)(x-x_1)(x-x_2), \quad \eta(x) \in I, \tag{4.28}$$

where $p_2(x)$ is the quadratic Lagrange polynomial (4.9) and a unknown point $\eta(x) \in (x_0, x_2)$.

$$E_3 = f(x) - p_3(x) = \frac{f^{(4)}(\eta(x))}{4!}(x-x_0)(x-x_1)(x-x_2)(x-x_3), \quad \eta(x) \in I, \tag{4.29}$$

where $p_3(x)$ is the cubic Lagrange polynomial (4.11) and a unknown point $\eta(x) \in (x_0, x_3)$.

Continuing in the similar manner, in the following theorem we define the error formula for the nth degree Lagrange polynomial (4.16).

Theorem 4.2 (Error Formula of *nth* Degree Lagrange Polynomial)

If $f(x)$ has $(n+1)$ derivatives on interval I and if it is approximated by a polynomial $p_n(x)$ passing through $(n+1)$ data points on I, then the error E_n is given by

$$E_n = f(x) - p_n(x) = \frac{f^{(n+1)}(\eta(x))}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n), \quad \eta(x) \in I, \tag{4.30}$$

where $p_n(x)$ is Lagrange interpolating polynomial (4.16) and a unknown point $\eta(x) \in (x_0, x_n)$. •

The error formula (4.30) is an important theoretical results because the Lagrange polynomials are used extensively for deriving numerical differentiation and integration methods. Error bounds for these techniques are obtained from Lagrange error formula.

Example 4.13 *Let $f(x) = \sqrt{x - x^2}$ and $p_2(x)$ be the quadratic Lagrange interpolating polynomial on $x_0 = 0, x_1 = \alpha$ and $x_2 = 1$. Find the largest value of α in $(0, 1)$ for which*

$$f(0.5) - p_2(0.5) = -0.25.$$

Solution. *Consider the quadratic Lagrange interpolating polynomial as follows:*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2).$$

At the given values of $x_0 = 0, x_1 = \alpha, x_2 = 1$, we have, $f(0) = 0, f(\alpha) = \sqrt{\alpha - \alpha^2}$ and $f(1) = 0$, gives

$$f(x) = p_2(x) = L_0(x)(0) + L_1(x)(\sqrt{\alpha - \alpha^2}) + L_2(x)(0),$$

where

$$L_1(x) = \frac{(x-0)(x-1)}{(\alpha-0)(\alpha-1)} = \frac{x^2 - x}{\alpha^2 - \alpha}.$$

Thus

$$f(x) = p_2(x) = \frac{x^2 - x}{\alpha^2 - \alpha}\sqrt{\alpha - \alpha^2} \quad \textit{and} \quad p_2(0.5) = \frac{-0.25}{\alpha^2 - \alpha}\sqrt{\alpha - \alpha^2}.$$

Given

$$f(0.5) - p_2(0.5) = -0.25, \quad \textit{gives} \quad p_2(0.5) = f(0.5) + 0.25 = 0.5 + 0.25 = 0.75,$$

so

$$-0.25\frac{\sqrt{\alpha - \alpha^2}}{(\alpha - \alpha^2)} = 0.75, \quad \textit{or} \quad \sqrt{\alpha - \alpha^2} = -3(\alpha - \alpha^2).$$

Thus, taking square on both sides, we get

$$\alpha - \alpha^2 = 9(\alpha - \alpha^2)^2, \quad \textit{or} \quad (\alpha - \alpha^2)[1 - 9(\alpha - \alpha^2)] = \alpha(1 - \alpha)[9\alpha^2 - 9\alpha + 1] = 0.$$

Solving this equation for α, we get, $\alpha = 0$, or $\alpha = 1$, or $\alpha = 0.127322$, or $\alpha = 0.872678$. Thus $\alpha = 0.872678$, the required largest value in the given interval $(0, 1)$. •

Example 4.14 *Let $f(x) = \dfrac{1}{x}$ be defined in the interval $[2, 4]$ and $x_0 = 2$, $x_1 = 2.5$, $x_2 = 4$. Compute the value of the unknown point η in the error formula of quadratic Lagrange interpolating polynomial for the approximation of $f(3)$ using the given points x_0, x_1, x_2.*

Solution. *Consider the quadratic Lagrange interpolating polynomial as follows:*

$$p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2),$$

At the given values of $x_0 = 2, x_1 = 2.5, x_2 = 4$, we have, $f(2) = 1/2, f(2.5) = 1/2.5$ and $f(4) = 1/4$, so using $x = 3$, we have

$$f(3) \approx p_2(3) = (1/2)L_0(3) + (1/2.5)L_1(3) + (1/5)L_2(3).$$

Then

$$L_0(3) = \frac{(3-2.5)(3-4)}{(2-2.5)(2-4)} = -\frac{1}{2}, \; L_1(3) = \frac{(3-2)(3-4)}{(2.5-2)(2.5-4)} = \frac{4}{3}, \; L_2(3) = \frac{(3-2)(3-2.5)}{(4-2)(4-2.5)} = \frac{1}{6}.$$

$$f(3) \approx p_2(3) = (1/2)(-1/2) + (1/2.5)(4/3) + (1/4)(1/6) = 0.325,$$

which is the required approximation of $f(3)$ *by the quadratic interpolating polynomial. The error is*

$$f(3) - p_2(3) = 1/3 - 0.325 = 0.0083.$$

Since the error formula of the quadratic Lagrange polynomial is

$$E = f(x) - p_2(x) = \frac{f'''(\eta)}{3!}(x - x_0)(x - x_1)(x - x_2), \quad \eta \in I,$$

and the third derivative of f *is,* $f'(\eta) = -1/\eta^2, \quad f''(\eta) = 2/\eta^3, \quad f'''(\eta) = -6/\eta^4$. *Thus*

$$0.0083 = f(3) - p_2(3) = \left(\frac{(3-2)(3-2.5)(3-4)}{6}\right)\frac{(-6)}{(\eta^4)} = \frac{0.5}{\eta^4},$$

and solving for η, *we get,* $\eta^4 = 60.2410, \quad \eta^2 = 7.7615, \quad \eta = 2.7859 \in (2, 4)$, *required value of the unknown point* η. •

Error Bound Formulas of Linear, Quadratic and Cubic Lagrange Polynomials

If $f(x)$ has second, third and fourth derivatives on interval I and if it is approximated by the polynomials $p_1(x), p_2(x), p_3(x)$ passing respectively, through $2, 3, 4$ data points on I, then the errors bound $|E_1|, |E_2|, |E_3|$ are given by

$$|E_1| = |f(x) - p_1(x)| \leq \frac{M}{2!}|(x - x_0)(x - x_1)|, \quad M = \max_{x_0 \leq x \leq x_1} |f''(x)|,$$

$$|E_2| = |f(x) - p_2(x)| \leq \frac{M}{3!}|(x - x_0)(x - x_1)(x - x_2)|, \quad M = \max_{x_0 \leq x \leq x_2} |f'''(x)|,$$

$$|E_3| = |f(x) - p_3(x)| \leq \frac{M}{3!}|(x - x_0)(x - x_1)(x - x_2)(x - x_3)|, \quad M = \max_{x_0 \leq x \leq x_3} |f^{(4)}(x)|,$$

and for degree n Lagrange polynomial

$$|E_n| = |f(x) - p_n(x)| \leq \frac{M}{(n+1)!}|(x - x_0)(x - x_1)\cdots(x - x_n)|, \quad M = \max_{x_0 \leq x \leq x_n} |f^{(n+1)}(x)|.$$

Example 4.15 *Show that a bound for the error in the linear interpolation is*

$$|f(x) - p_1(x)| \leq \frac{h^2}{8}M, \quad \textit{where} \quad M = \max_{x_0 \leq x \leq x_1} |f''(x)|, \quad \textit{and} \quad h = x_1 - x_0. \tag{4.31}$$

Solution. *Consider two points* x_0 *and* x_1, *then the linear polynomial* $p_1(x)$ *interpolating* $f(x)$ *at these points is*

$$f(x) = p_1(x) = \frac{(x - x_1)}{(x_0 - x_1)}f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)}f(x_1).$$

By using the given data point, the error formula (4.30) becomes

$$f(x) - p_1(x) = \frac{(x - x_0)(x - x_1)}{2!}f''(\eta(x)),$$

where $\eta(x)$ is a unknown point between x_0 and x_1. Hence

$$|f(x) - p_1(x)| = \left|\frac{(x-x_0)(x-x_1)}{2!}\right| |f''(\eta(x))|.$$

The value of $f''(\eta(x))$ can not be computed exactly because $\eta(x)$ is not known. But we can bound the error by computing the largest possible value for $|f''(\eta(x))|$. So bound $|f''(x)|$ on $[x_0, x_1]$ can be obtain

$$M = \max_{x_0 \le x \le x_1} |f''(x)|, \quad \textit{and} \quad |f''(\eta(x))| \le M.$$

$$|f(x) - p_1(x)| \le \frac{M}{2}|(x-x_0)(x-x_1)|.$$

Since the maximum of function $g(x) = (x-x_0)(x-x_1)$ in $[x_0, x_1]$ occurs at the critical point $x = \dfrac{(x_0+x_1)}{2}$ ($g'(x) = 0$) and so that maximum is $|(x-x_0)(x-x_1)| = \dfrac{(x_1-x_0)^2}{4}$.
This follows easily by noting that the function $(x-x_0)(x-x_1)$ is a quadratic and has two roots x_0 and x_1, hence its maximum value occurs midway between these roots. Thus, for any $x \in [x_0, x_1]$, we have

$$|f(x) - p_1(x)| \le \frac{(x_1-x_0)^2}{8}M, \quad \textit{or} \quad |f(x) - p_1(x)| \le \frac{h^2}{8}M,$$

where $h = x_1 - x_0$. •

Example 4.16 *Find the linear Lagrange polynomial that passes through the points $(0, f(0))$ and $(\pi, f(\pi))$ and then use it to approximate the function $f(x) = 2\cos x$ at $\dfrac{\pi}{2}$. Find absolute error and an bound for the error in the linear interpolation of $f(x)$.*

Solution. *Given two points $x_0 = 0$ and $x_1 = \pi$, then the linear Lagrange polynomial $p_1(x)$*

$$f(x) = p_1(x) = \frac{(x-x_1)}{(x_0-x_1)}f(x_0) + \frac{(x-x_0)}{(x_1-x_0)}f(x_1),$$

interpolating $f(x)$ at these points is

$$f(x) = p_1(x) = \frac{(x-\pi)}{(0-\pi)}f(0) + \frac{(x-0)}{(\pi-0)}f(\pi).$$

By using the function values at the given data point, we get

$$f(x) = p_1(x) = \frac{(x-\pi)}{(0-\pi)}(2) + \frac{(x-0)}{(\pi-0)}(-2) = 2 - \frac{4x}{\pi} \quad \textit{and} \quad f(\pi/2) \approx p_1(\pi/2) = 0.$$

Thus absolute error, $|2\cos(\pi/2) - p_1(\pi/2)| = 0$. Since $M = \max\limits_{0 \le x \le \pi} |f''(x)| = \max\limits_{0 \le x \le \pi} |-2\cos x| = 2$ and $h = \pi$, so by using the linear Lagrange error formula (4.31), we get

$$|f(x) - p_1(x)| \le \frac{(\pi-0)^2}{4} = \frac{\pi^2}{4},$$

which is the required bound of error in the linear interpolation of $f(x)$. •

Example 4.17 *Find the approximation of* $\sin(\pi/6)$ *using quadratic Lagrange polynomial that passes through the points* $(0, f(0)), (\pi/4, f(\pi/4))$, *and* $(\pi/2, f(\pi/2))$. *Compute an bound of error in the quadratic interpolation.*

Solution. *Using quadratic Lagrange polynomial* $p_2(x)$

$$f(x) = p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2),$$

which can be written as

$$\sin(\pi/6) \approx f(\pi/6) = p_2(\pi/6) = \frac{(\pi/6-0)(\pi/6-\pi/2)}{(\pi/4-0)(\pi/4-\pi/2)}(0.70711) + \frac{(\pi/6-0)(\pi/6-\pi/4)}{(\pi/2-0)(\pi/2-\pi/4)}(1) = 0.51743.$$

To compute the error bound, we calculate $M = \max\limits_{0\le x\le \pi/2}|f'''(x)| = \max\limits_{0\le x\le \pi/2}|-\cos x| = 1$, *so by using the quadratic Lagrange error formula, we get*

$$|f(x) - p_2(x)| \le \frac{|(\pi/6-0)(\pi/6-\pi/4)(\pi/6-\pi/2)|}{3!}(1) = 0.02392,$$

which is the required bound of error in the quadratic interpolation of $f(x)$. •

Example 4.18 *Construct the Lagrange interpolating polynomial of degree 2 for* $f(x) = x\ln x + e^{-x}$ *on the interval* $[2, 4]$ *with the points* $x_0 = 2.0,\ x_1 = 3.0,\ x_2 = 4.0$. *Find a bound for the absolute error.*

Solution. *Thus, the following table is given:*

x	2.0	3.0	4.0
$f(x)$	1.5216	3.3456	5.5635

Consider a quadratic Lagrange interpolating polynomial as

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2), \tag{4.32}$$

$$f(x) = p_2(x) = L_0(x)f(2) + L_1(x)f(3) + L_2(x)f(4).$$

$$f(x) = p_2(x) = 1.5216L_0(x) + 3.3456L_1(x) + 5.5635L_2(x). \tag{4.33}$$

We construct the basic Lagrange polynomials:

$$L_0(x) = \frac{(x-3)(x-4)}{(2-3)(2-4)} = \frac{(x^2-7x+12)}{2},$$

$$L_1(x) = \frac{(x-2)(x-4)}{(3-2)(3-4)} = \frac{(x^2-6x+8)}{-1},$$

$$L_2(x) = \frac{(x-2)(x-3)}{(4-2)(4-3)} = \frac{(x^2-5x+6)}{2}.$$

Putting these values of the Lagrange coefficients in (4.33), we have

$$f(x) = p_2(x) = 0.1970x^2 + 0.8392x - 0.9447,$$

which is the required quadratic polynomial. The error is given by

$$f(x) - p_2(x) = \frac{f^{(3)}(\eta(x))}{3!}(x - x_0)(x - x_1)(x - x_2).$$

$$f(x) = x \ln x + e^{-x} \qquad \text{and} \qquad f^{(3)}(x) = -\frac{1}{x^2} - e^{-x},$$

and the maximum of the third derivative is at 2, so

$$|f^{(3)}(\eta(x))| \leq M = \max_{2\leq x\leq 4} |f^{(3)}(x)| = \max_{2\leq x\leq 4} |-\frac{1}{x^2} - e^{-x}| = 0.3853.$$

Next we bound $|g(x)|$, where $g(x) = (x-2)(x-3)(x-4)$. To find the maximum of $|g(x)|$, we need to take the derivative:

$$g(x) = x^3 - 9x^2 + 26x - 24, \quad g'(x) = 3x^2 - 18x + 26 = 0, \quad \text{gives} \quad x_1 = 3.5774,\ x_2 = 2.4226,$$

and

$$|g(x)| = |g(3.5774)| = |-0.3849| = 0.3849 \quad \text{and} \quad |g(x)| = |g(2.4226| = |0.3849| = 0.3849.$$

Thus, obtaining the error bound:

$$|f(x) - p_2(x)| = \frac{|f^{(3)}(\eta(x))|}{3!}|(x - x_0)(x - x_1)(x - x_2)| \leq \frac{0.3853}{6}(0.3849) = 2.47 \times 10^{-3},$$

which is desired bound for the absolute error. •

Example 4.19 *Use the quadratic Lagrange interpolating polynomial by selecting the best three points from $\{-2, 0, 1, 2, 2.5\}$ on the function defined by $f(x) = (x+1)^{\frac{1}{3}}$ to estimate the cube root of $\frac{3}{2}$ (that is, $(\frac{3}{2})^{\frac{1}{3}}$) and compute an error bound and absolute error.*

Solution. *Since the given function is a cube root of $(x+1)$, so by taking $x+1 = \frac{3}{2}$, we have $x = \frac{1}{2}$, therefore, the best points for the quadratic polynomial are, $x_0 = 0, x_1 = 1$, and $x_2 = 2$. Consider the quadratic Lagrange interpolating polynomial as*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2), \tag{4.34}$$

$$f(0.5) \approx p_2(0.5) = (1)^{1/3}L_0(0.5) + (2)^{1/3}L_1(0.5) + (3)^{1/3}L_2(0.5). \tag{4.35}$$

The Lagrange coefficients can be calculate as follows:

$$L_0(0.5) = \frac{(0.5-1)(0.5-2)}{(0-1)(0-2)} = 0.375,$$

$$L_1(0.5) = \frac{(0.5-0)(0.5-2)}{(1-0)(1-2)} = 0.75,$$

$$L_2(0.5) = \frac{(0.5-0)(0.5-1)}{(2-0)(2-1)} = -0.125.$$

Putting these values of the Lagrange coefficients in (4.35), we have

$$f(0.5) \approx p_2(0.5) = (1)^{1/3}(0.375) + (2)^{1/3}(0.75) + (3)^{1/3}(-0.125) = 1.1396,$$

which is the required approximation of the given exact solution $\left(\frac{3}{2}\right)^{1/3} \approx 1.1447$.
To compute an error bound for the approximation of the given function in the interval $[0, 2]$*, we use the following quadratic error formula*

$$|f(x) - p_2(x)| = \frac{|f^{(3)}(\eta(x))|}{3!}|(x - x_0)(x - x_1)(x - x_2)|.$$

As

$$|f^{(3)}(\eta(x))| \leq M = \max_{0 \leq x \leq 2} |f^{(3)}(x)|,$$

and the first three derivatives are

$$f'(x) = \frac{1}{3}(x+1)^{-2/3}, \quad f''(x) = -\frac{2}{9}(x+1)^{-5/3}, \quad f^{(3)}(x) = \frac{10}{27}(x+1)^{-8/3},$$

$$M = \max_{0 \leq x \leq 2} \left| \frac{10}{27}(x+1)^{-8/3} \right| = \frac{10}{27}.$$

Hence

$$|f(0.5) - p_2(0.5)| \leq \frac{10/27}{6}|(0.5 - 0)(0.5 - 1)(0.5 - 2)| \leq \frac{10(0.375)}{162} = 0.0232,$$

which is desired error bound. Also, we have

$$|f(0.5) - p_2(0.5)| = |(1.5)^{1/3} - 1.1396| = |1.1447 - 1.1396| = 0.0051,$$

the desired absolute error. •

Example 4.20 *Consider* $f(x) = \sin x$ *and its values are known at five points* $\{0, 0.2, 0.4, 0.6, 0.8\}$*. If the approximation of* $\sin 0.28$ *by four degree Lagrange interpolating polynomial is* 0.2763591*, then compute the error bound and the absolute error for the approximation.*

Solution. *To compute an error bound for the approximation of the given function in the interval* $[0, 0.8]$*, we use the following error formula for Lagrange polynomial degree four*

$$|f(x) - p_4(x)| = \frac{|f^{(5)}(\eta(x))|}{5!}|(x - x_0)(x - x_1)(x - x_2)(x - x_3)(x - x_4)|,$$

or

$$|f(x) - p_4(x)| \leq \frac{M}{5!}|(x - x_0)(x - x_1)(x - x_2)(x - x_3)(x - x_4)|.$$

Since

$$|f^{(5)}(\eta(x))| \leq M = \max_{0 \leq x \leq 0.8} |f^{(5)}(x)| = \max_{0 \leq x \leq 0.8} |\cos x| = 1,$$

so

$$|f(0.28) - p_4(0.28)| \leq \frac{1}{120}|0.28(0.28-0.2)(0.28-0.4)(0.28-0.6)(0.28-0.6)(0.28-0.8)| \leq 3.7 \times 10^{-6},$$

which is desired error bound. Also, we have to compute absolute error as

$$|f(0.28) - p_4(0.28)| = |\sin 0.28 - p_4(0.28)| = |0.2763556 - 0.2763591| = 3.5 \times 10^{-6},$$

which is desired result. •

Example 4.21 *Use the best Lagrange interpolating polynomial to find the approximation of* $f(1.5)$ *if* $f(-2) = 2, f(-1) = 1.5,\ f(1) = 3.5$ *and* $f(2) = 5$. *Estimate the error bound if the maximum value of* $|f^{(4)}(x)|$ *is* 0.025 *in the interval* $[-2, 2]$.

Solution. *Since the given number of points are,* $x_0 = -2, x_1 = -1, x_2 = 1, x_3 = 2$, *therefore the best Lagrange interpolating polynomial to find the approximation of* $f(1.5)$ *will be the cubic. The cubic Lagrange interpolating polynomial for the approximation of the given function is:*

$$f(x) = p_3(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + L_3(x)f(x_3),$$

and taking $f(-2) = 2, f(-1) = 1.5, f(1) = 3.5, f(2) = 5$ *and the interpolating point* $x = 1.5$, *we have*

$$f(1.5) \approx p_3(1.5) = 2L_0(1.5) + 1.5L_1(1.5) + 3.5L_2(1.5) + 5L_3(1.5).$$

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned} L_0(1.5) &= \frac{(1.5+1)(1.5-1)(1.5-2)}{(-2+1)(-2-1)(-2-2)} = 0.0521, \\ L_1(1.5) &= \frac{(1.5+2)(1.5-1)(1.5-2)}{(-1+2)(-1-1)(-1-2)} = -0.1458, \\ L_2(1.5) &= \frac{(1.5+2)(1.5+1)(1.5-2)}{(1+2)(1+1)(1-2)} = 0.7292, \\ L_3(1.5) &= \frac{(1.5+2)(1.5+1)(1.5-1)}{(2+2)(2+1)(2-1)} = 0.3646. \end{aligned}$$

Putting these values of the Lagrange coefficients in the above equation, we get

$$f(1.5) \approx p_3(1.5) = 2(0.0521) + 1.5(-0.1458) + 3.5(0.7292) + 5(0.3646) = 4.2607,$$

which is the required cubic interpolating polynomial approximation of the function at the given point $x = 1.5$. *Note that* $L_0(1.5) + L_1(1.5) + L_2(1.5) + L_3(1.5) = 1$.
To compute an error bound for the approximation of the given function in the interval $[-2, 2]$, *we use the following cubic error formula*

$$|f(x) - p_3(x)| = \frac{|f^{(4)}(\eta(x))|}{4!}|(x - x_0)(x - x_1)(x - x_2)(x - x_3)|.$$

As

$$|f^{(4)}(\eta(x))| \leq M = \max_{-2 \leq x \leq 2} |f^{(4)}(x)| = 0.025,$$

so

$$|f(1.5) - p_3(1.5)| \leq \frac{M}{4!}|(1.5+2)(1.5+1)(1.5-1)(1.5-2)| = \frac{(0.025)(2.1875)}{24} = 0.0023,$$

which is desired error bound. •

Example 4.22 *Consider the following table having the data for* $f(x) = e^{3x} + \cos 2x$*:*

x	0.1	0.2	0.4	0.5
$f(x)$	2.3300	2.7432	4.0168	5.0220

Find the approximation of $f(0.45)$ *using the best quadratic Lagrange interpolation formula and also estimate an error bound and absolute error for the approximation.*

Solution. *Using the best possible data points* $0.2, 0.4, 0.5$*, the quadratic Lagrange formula to find the approximation of the function is*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2),$$

which implies that

$$\begin{aligned} f(x) = p_2(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) \\ &+ \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2), \end{aligned}$$

or

$$\begin{aligned} f(x) &= p_2(x) = 45.72[x^2 - 0.9x + 0.2] - 200.84[x^2 - 0.7x + 0.1] \\ &+ 167.4[x^2 - 0.6x + 0.08]. \end{aligned}$$

Thus

$$f(x) = p_2(x) = 12.28x^2 - 1.0x + 2.452 \quad \textit{and} \quad f(0.45) \approx p_2(0.45) = 4.4887. \tag{4.36}$$

Since $f(0.45) = 4.4790$ *is exact value, so the absolute error is* 0.0097*. Now to compute an error bound of the approximation, we use the following formula*

$$|f(x) - p_2(x)| = \frac{|f^{(3)}(\eta(x))|}{3!}|(x-x_0)(x-x_1)(x-x_2)|. \tag{4.37}$$

Taking the third derivative of the given function, we get

$$f'(x) = 3e^{3x} - 2\sin 2x, \quad f''(x) = 9e^{3x} - 4\cos 2x, \quad f'''(x) = 27e^{3x} + 8\sin 2x.$$

Thus

$$|f^{(3)}(\eta(x))| = |27e^{3\eta(x)} + 8\sin 2(\eta(x))|, \qquad \textit{for} \qquad \eta(x) \in (0.2, 0.5),$$

and it gives

$$|f^{(3)}(0.2)| = 52.3126 \qquad \textit{and} \qquad |f^{(3)}(0.5)| = 127.7374.$$

The value of $f^{(3)}(\eta(x))$ *can not be computed exactly because* $\eta(x)$ *is not known. But we can bound the error by computing the largest possible value for* $|f^{(3)}(\eta(x))|$*. So bound* $|f^{(3)}(x)|$ *on* $[0.2, 0.5]$ *can be obtain*

$$M = \max_{0.2 \le x \le 0.5} |f^{(3)}(x)| = 127.7374,$$

and so for $|f^{(4)}(\eta(x))| \le M$*, we have (4.37) as follows*

$$|f(x) - p_2(x)| \le (127.7374)(0.000625)/6 = 0.0133,$$

which is the required error bound for the approximation. •

Example 4.23 *Determining spacing h in a table of equally spaced value of the function $f(x) = e^x$ between smallest point $a = 1$ and largest point $b = 2$, so that interpolation with a second-degree polynomial in this table will yield a desired accuracy.*

Solution. *Suppose that the given table contains the function values $f(x_i)$, for the points $x_i = 1 + ih,\ i = 0, 1, \ldots, n$, where $n = \dfrac{(2-1)}{h}$. If $x \in [x_{i-1}, x_{i+1}]$, then we approximate the function $f(x)$ by degree 2 polynomial $p_2(x)$ which interpolates $f(x)$ at x_{i-1}, x_i, x_{i+1}. Then the error formula (4.30) for these data points becomes*

$$|f(x) - p_2(x)| = \Big|\frac{(x - x_{i-1})(x - x_i)(x - x_{i+1})}{3!}\Big|\Big|f'''(\eta(x))\Big|$$

where $\eta(x) \in (x_{i-1}, x_{i+1})$. Since the point $\eta(x)$ is unknown and so, we can not estimate $f'''(\eta(x))$, therefore, let

$$|f'''(\eta(x))| \leq M = \max_{1 \leq x \leq 2} |f'''(x)|.$$

Then

$$|f(x) - p_2(x)| \leq \frac{M}{6}|(x - x_{i-1})(x - x_i)(x - x_{i+1})|.$$

Since $f(x) = e^x$ and $f'''(x) = e^x$, therefore

$$|f'''(\eta(x))| \leq M = \max_{1 \leq x \leq 2} |e^x| = e^2 = 7.3891.$$

Now to find the maximum value of $|(x - x_{i-1})(x - x_i)(x - x_{i+1})|$, take $t = (x - x_i)$, gives

$$\max_{x \in [x_{i-1}, x_{i+1}]} |(x - x_{i-1})(x - x_i)(x - x_{i+1}| = \max_{t \in [-h, h]} |(t - h)t(t + h)| = \max_{t \in [-h, h]} |t(t^2 - h^2)|,$$

using the linear change of variables $t = x - x_i$. As we see the function $H(t) = t^3 - th^2$ vanishes at $t = -h$ and $t = h$, so the maximum value of $|H(t)|$ on $[-h, h]$ must occurs at one of the extreme of $H(t)$, which can be found by solving the equation

$$H'(t) = 3t^2 - h^2 = 0, \quad \textit{gives}, \quad t = \pm h/\sqrt{3}.$$

Hence

$$\max_{x \in [x_{i-1}, x_{i+1}]} |(x - x_{i-1})(x - x_i)(x - x_{i+1})| = \frac{2h^3}{3\sqrt{3}}.$$

Thus, for any $x \in [1, 2]$, we have

$$|f(x) - p_2(x)| \leq \frac{(2h^3/3\sqrt{3})e^2}{6} = \frac{h^3 e^2}{9\sqrt{3}},$$

if $p_2(x)$ is chosen as the polynomial of degree 2 which interpolates $f(x) = e^x$ at the three tabular points nearest x. If we wish to obtain six decimal place accuracy this way, we would have to choose h so that

$$\frac{h^3 e^2}{9\sqrt{3}} < 5 \times 10^{-7}, \quad \textit{which implies that} \quad h^3 < 10.5483 \times 10^{-7},$$

and it gives $h = 0.01$. •

If the derivatives of f can be uniformly bounded by a constant, then we can choose n appropriately large in order to force the error term to be as small as we want. So, we can find a very accurate interpolating polynomial. In particular, if we look at the case when we choose to use equally spaced points (but without pre-determining n), then bounding the derivatives allows us to choose n large enough (alternatively h small enough) to give an accurate polynomial interpolation. The form of the error bound Theorem 4.2 is not as useful as the following one for quick calculations.

Theorem 4.3 (Error Bounds for Lagrange Interpolation at Equally Spaced Points)

Assume that $f(x)$ is defined on the interval $[a, b]$, which contains equally spaced points $x_n = x_0 + hn$ or $h = (x_n - x_0)/n$. Additionally, assume that $f(x)$ and the derivatives of $f(x)$ up to the order $(n+1)$, are continuous and bounded on the special intervals $[x_0, x_1]$, $[x_0, x_2]$ and $[x_0, x_3]$, respectively; that is

$$|f^{(n+1)}(x)| \leq M \quad \textit{for} \quad x_0 \leq x \leq x_n,$$

for $n = 1, 2, 3$. Then error bounds for linear, quadratic and cubic polynomials are:

$$\begin{aligned} |E_1(x)| &\leq \frac{h^2}{8}M \qquad \textit{for} \qquad x_0 \leq x \leq x_1, \\ |E_2(x)| &\leq \frac{h^3}{9\sqrt{3}}M \qquad \textit{for} \qquad x_0 \leq x \leq x_2, \\ |E_3(x)| &\leq \frac{h^4}{24}M \qquad \textit{for} \qquad x_0 \leq x \leq x_3. \end{aligned}$$

Continue in the similar manner for the interval $[x_0, x_n]$, for $n = 1, 2, \ldots, n$, we have

$$|E_n(x)| \leq \frac{M}{4(n+1)}\left(\frac{b-a}{n}\right)^{n+1} = \frac{M}{4(n+1)}h^{n+1}, \quad \textit{for} \quad x_0 \leq x \leq x_n, \tag{4.38}$$

the general error bound formula. It has been proved that $\displaystyle\prod_{i=0}^{n}|x - x_i| \leq \frac{h^{n+1}n!}{4}$. •

Example 4.24 *Find an error bound if $f(x) = \sin x$ is approximated by an interpolation polynomial with ten (10) equally spaced data points in $[0, 1.6875]$.*

Solution. *Given $n = 9$ and $a = 0,\ b = 1.6875$, then*

$$M = \max_{0\leq x\leq 1.6875}|f^{(10)}(x)| = \max_{0\leq x\leq 1.6875}|-\sin x| \leq 1, \quad \forall\ x \in [0, 1.6875].$$

Hence, the interpolation error (using Theorem 4.38) can be bounded by

$$|E_9(x)| = |\sin x - p_9(x)| \leq \frac{1}{40}\left(\frac{1.6875}{9}\right)^{10} \approx 1.34 \times 10^{-9},$$

for all $x \in [0, 1.6875]$. •

Note that $f^{(n)}(x) = \pm \sin x$ for even n and $f^{(n)}(x) = \pm \cos x$ for odd n, so we have a uniform bound on $f^{(n)}(x)$ for all n. That is $|f^{(n)}(x)| \leq 1$ for all x and for all n. In such instance, we can force interpolation error to 0 by increasing the number of interpolation data points.
On the other hand, if the derivatives of all order for the function f are continuous but we cannot uniformly the derivatives, then increasing the interpolation data points may not result in smaller errors.

Example 4.25 *Find an error bound if* $f(x) = \dfrac{1}{x^2+1}$ *is approximated by an interpolation polynomial with ten equally spaced data points in the interval* $[-5, 5]$.

Solution. *Given* $n = 9$ *and* $a = -5, b = 5$, *then*

$$M = \max_{-5 \leq x \leq 5} |f^{(10)}(x)|,$$

One can find the maximum value of $|f^{(10)}(x)|$ *is a very large value. Note that the term*

$$\frac{1}{40}\left(\frac{10}{9}\right)^{10} \approx 0.0717,$$

so this term will not be small enough to guarantee a reasonable bound on the error. Furthermore, if we allow n *to get larger, then the magnitude of the derivatives of* f *also get very large very fast. So adding more interpolation points can increase the oscillation of the interpolating polynomial.* •

Example 4.26 *Determine an appropriate step size* h *to construct a table of* $f(x) = x + \ln(x+1)$ *on the interval* $[2, 3]$ *when the error for linear Lagrange interpolation is to be bounded by* 9×10^{-4}. *Then use the constructed table to find the approximation of* $f(2.6)$ *using linear Lagrange polynomial and compute the absolute error.*

Solution. *Since we know that error bound formula for the linear Lagrange polynomial is*

$$|E| \leq \frac{Mh^2}{8} \leq 9 \times 10^{-4}, \quad h = x_1 - x_0, \tag{4.39}$$

$$\frac{Mh^2}{8} \leq 9 \times 10^{-4}, \quad \textit{where} \quad M = \max_{2 \leq x \leq 3} |f''(x)|.$$

Since

$$f'(x) = 1 + \frac{1}{x+1} \quad \textit{and} \quad f''(x) = -\frac{1}{(x+1)^2}, \quad M = \max_{2 \leq x \leq 3} \left|\frac{-1}{(x+1)^2}\right| = \frac{1}{9}.$$

Thus

$$h^2 \leq 648 \times 10^{-4} = 0.0648, \quad \textit{gives} \quad h = 0.25.$$

Hence we obtain the required table

x	2	2.25	2.5	2.75	3
$f(x)$	3.0986	3.4287	3.7528	4.0718	4.3863

for the given function $f(x)$ on $[2,3]$ with step size $h = 0.25$.
As $x = 2.6$, so take two points $x_0 = 2.5$ and $x_1 = 2.75$, then linear Lagrange polynomial $p_1(x)$

$$f(x) = p_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1),$$

and interpolating $f(2.6)$ at these points is

$$f(2.6) \approx p_1(2.6) = \frac{(2.6 - 2.75)}{(2.5 - 2.75)}(3.7528) + \frac{(2.6 - 2.5)}{(2.75 - 2.5)}(4.0718) = 3.8804.$$

To compute absolute error in the linear interpolation of $f(x)$, we do as follows

$$|f(2.6) - p_1(2.6)| = |(2.6 + \ln(3.6)) - 3.8804| = |3.8809 - 3.8804| = 0.0005,$$

the required error. •

Example 4.27 *(a) Let $f(x) = (x+1)\ln(x+1)$ be the function defined over the interval $[1,2]$. Find the approximations of $(2.9 \ln 2.9)$ using linear, quadratic and cubic Lagrange interpolating polynomials for equally spaced data points defined over the interval $[1,2]$. Compute the error bounds for linear, quadratic and cubic Lagrange interpolating polynomials for equally spaced data points. Also, compute absolute error for each case.*
(b) Determine the step size h and the number of points to be used in the tabulation of the given function $f(x) = (x+1)\ln(x+1)$ in $[1,2]$ so that linear, quadratic and cubic interpolations will be correct to six decimal places.

Solution. *(a) For linear Lagrange polynomial, we have $h = 2 - 1 = 1$, so using $x_0 = 1, x_1 = 2$ and $x = 1.9$, in the linear Lagrange formula, we have*

$$f(1.9) \approx p_1(1.9) = L_0(1.9)f(1) + L_1(1.9)f(2) = 1.3863L_0(1.9) + 3.2958L_1(1.9). \tag{4.40}$$

The Lagrange coefficients can be calculate as follows:

$$L_0(1.9) = \frac{(1.9 - 2)}{(1 - 2)} = 0.1 \quad \textit{and} \quad L_1(1.9) = \frac{(1.9 - 1)}{(2 - 1)} = 0.9.$$

Putting these values of the Lagrange coefficients in (4.40), we have

$$f(1.9) \approx p_1(1.9) = 1.3863(0.1) + 3.2958(0.9) = 3.1049.$$

Now for quadratic Lagrange polynomial, take $h = \dfrac{2-1}{2} = 0.5$, using $x_0 = 1, x_1 = 1.5, x_2 = 2$ and $x = 1.9$, in the quadratic Lagrange formula, we have

$$\begin{aligned} f(1.9) \approx p_2(1.9) &= L_0(1.9)f(1) + L_1(1.9)f(1.5) + L_2(1.9)f(2) \\ &= 1.3863L_0(1.9) + 2.2907L_1(1.9) + 3.2958L_2(1.9). \end{aligned} \tag{4.41}$$

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned} L_0(1.9) &= \frac{(1.9-1.5)(1.9-2)}{(1-1.5)(1-2)} = -0.08, \\ L_1(1.9) &= \frac{(1.9-1)(1.9-2)}{(1.5-1)(1.5-2)} = 0.36, \\ L_2(1.9) &= \frac{(1.9-1)(1.9-1.5)}{(2-1)(2-1.5)} = 0.72. \end{aligned}$$

Putting these values of the Lagrange coefficients in (4.41), we have

$$f(1.9) \approx p_2(1.9) = 1.3863(-0.08) + 2.2907(0.36) + 3.2958(0.72) = 3.0868.$$

For cubic Lagrange polynomial, we have $h = \dfrac{2-1}{3} = 1/3$, *so using* $x_0 = 1, x_1 = 4/3, x_2 = 5/3$, $x_3 = 2$ *and* $x = 1.9$, *in the cubic Lagrange formula, we have*

$$\begin{aligned} f(1.9) \approx p_3(1.9) &= L_0(1.9)f(1) + L_1(1.9)f(4/3) + L_2(1.9)f(5/3) + L_3(1.9)f(2) \\ &= 1.3863L_0(1.9) + 1.9770L_1(1.9) + 2.6156L_2(1.9) + 3.2958L_3(1.9). \quad (4.42) \end{aligned}$$

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned} L_0(1.9) &= \frac{(1.9-4/3)(1.9-5/3)(1.9-2)}{(1-4/3)(1-5/3)(1-2)} = 0.0595, \\ L_1(1.9) &= \frac{(1.9-1)(1.9-5/3)(1.9-2)}{(4/3-1)(4/3-5/3)(4/3-2)} = -0.2835, \\ L_2(1.9) &= \frac{(1.9-1)(1.9-4/3)(1.9-2)}{(5/3-1)(5/3-4/3)(5/3-2)} = 0.6885, \\ L_3(1.9) &= \frac{(1.9-1)(1.9-4/3)(1.9-5/3)}{(2-1)(2-4/3)(2-5/3)} = 0.5355. \end{aligned}$$

Putting these values of the Lagrange coefficients in (4.42), we have

$$f(1.9) \approx p_3(1.9) = 1.386(0.0595) + 1.977(-0.2835) + 2.616(0.6885) + 3.296(0.5355) = 3.0877.$$

The derivatives of the given function $f(x) = (x+1)\ln(x+1)$ *are as follows:*

$$f'(x) = 1 + \ln(x+1), \quad f''(x) = \frac{1}{x+1}, \quad f'''(x) = -\frac{1}{(x+1)^2}, \quad f^{(4)}(x) = \frac{2}{(x+1)^3}.$$

Now for error bound of linear Lagrange polynomial, we use the formula

$$|E_1| \leq \frac{Mh^2}{8},$$

where $h = 2 - 1 = 1$ *and* $M = \max_{1\le x\le 2} |f''(x)| = \max_{1\le x\le 2} \left|\frac{1}{x+1}\right| = \frac{1}{2}$. *So*

$$|E_1| \le \frac{(1/2)(1)^2}{8} = \frac{1}{16} = 0.0625,$$

the error bound for the linear Lagrange polynomial.
Similarly, for error bound of quadratic Lagrange polynomial, we use the formula

$$|E_2| \le \frac{Mh^3}{9\sqrt{3}},$$

where $h = (2 - 1)/2 = 1/2$ *and* $M = \max_{1\le x\le 2} |f'''(x)| = \max_{1\le x\le 2} \left|-\frac{1}{(x+1)^2}\right| = \frac{1}{4}$. *So*

$$|E_2| \le \frac{(1/4)(1/2)^3}{9\sqrt{3}} = \frac{(1/32)}{9\sqrt{3}} = 0.0020,$$

the error bound for the quadratic Lagrange polynomial.
Finally, for error bound of cubic Lagrange polynomial, we use the formula

$$|E_3| \le \frac{Mh^4}{24},$$

where $h = (2 - 1)/3 = 1/3$ *and* $M = \max_{1\le x\le 2} |f^{(4)}(x)| = \max_{1\le x\le 2} \left|\frac{2}{(x+1)^3}\right| = \frac{1}{4}$. *Thus*

$$|E_3| \le \frac{(1/4)(1/3)^4}{24} = \frac{1}{7776} = 0.0001,$$

the error bound for the cubic Lagrange polynomial. Finally,

$$|f(1.9) - p_1(1.9)| = |3.0877 - 3.1049| = 0.0172,$$

$$|f(1.9) - p_2(1.9)| = |3.0877 - 3.0868| = 0.0015,$$

$$|f(1.9) - p_3(1.9)| = |3.0877 - 3.0877| = 0.0000,$$

are respectively, the absolute error for linear, quadratic and cubic polynomials.
(b) Since we know that the upper bound of error in linear polynomial is

$$|E_1| \le \frac{Mh^2}{8} \quad \textit{and} \quad M = \frac{1}{2},$$

therefore,

$$\frac{h^2}{16} \le 5 \times 10^{-6},\ h \le 0.0089, \quad n = 113.$$

As the upper bound of error in quadratic polynomial is

$$|E_2| \le \frac{Mh^3}{9\sqrt{3}} \quad \textit{and} \quad M = \frac{1}{4},$$

therefore,

$$\frac{h^3}{36\sqrt{3}} \leq 5 \times 10^{-6}, \quad or \quad h^3 \leq 311.7691 \times 10^{-6},\ h \leq 0.0678 \text{ and } n = 14.7476 \approx 15.$$

Finally, as the upper bound of error in cubic polynomial is

$$|E_3| \leq \frac{Mh^4}{24} \text{ and } M = \frac{1}{4},\ \frac{h^4}{96} \leq 5 \times 10^{-6},\ \text{or } h^4 \leq 480 \times 10^{-6}.$$

This gives, $h \leq 0.1480$ *and* $n = 6.7560 \approx 7$. *Thus we need, respectively,* 114 *points,* 16 *points and* 8 *points for the linear, quadratic and cubic interpolations.* •

Example 4.28 *Find the cubic Lagrange interpolating polynomial to find the approximation of* $f(x)$ *if* $f(1) = 0.5, f(4/3) = 1, f(5/3) = 2$ *and* $f(2) = 3$. *If* $|f^{(4)}| \leq \frac{1}{5}$ *for* $1 \leq x \leq 2$, *then show that the error estimate is given as* $|f(x) - p_3(x)| \leq \frac{1}{27}$.

Solution. *The given number of points are,* $x_0 = 1, x_1 = 4/3, x_2 = 5/3, x_3 = 2,$, *therefore the cubic Lagrange interpolating polynomial for the approximation of the given function is:*

$$f(x) = p_3(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + L_3(x)f(x_3),$$

and using the given function values, gives

$$f(x) = p_3(x) = L_0(x)(0.5) + L_1(x)(1) + L_2(x)(2) + L_3(x)(3).$$

The Lagrange coefficients can be calculate as follows:

$$\begin{aligned}
L_0(1.5) &= \frac{(x-4/3)(x-5/3)(x-2)}{(1-4/3)(1-5/3)(1-2)} = \frac{(x^3 - 5x^2 + 74/9x - 40/9)}{-2/9}, \\
L_1(1.5) &= \frac{(x-1)(x-5/3)(x-2)}{(4/3-1)(4/3-5/3)(4/3-2)} = \frac{(x^3 - 14/3x^2 + 7x - 10/3)}{2/27}, \\
L_2(1.5) &= \frac{(x-1)(x-4/3)(x-2)}{(5/3-1)(5/3-4/3)(5/3-2)} = \frac{(x^3 - 13/3x^2 + 6x - 8/3)}{-2/27}, \\
L_3(1.5) &= \frac{(x-1)(x-4/3)(x-5/3)}{(2-1)(2-4/3)(2-5/3)} = \frac{(x^3 - 4x^2 + 47/9x - 20/9)}{2/9}.
\end{aligned}$$

Putting these values of the Lagrange coefficients in the above equation, we get

$$f(x) = p_3(x) = \frac{1}{4}(-9x^3 + 45x^2 - 62x + 28),$$

which is the required cubic interpolating polynomial for the approximation of the function. Since we know the error of cubic Lagrange polynomial is

$$f(x) - p_3(x) = \frac{f^{(4)}(\eta(x))}{4!}(x - x_0)(x - x_1)(x - x_2)(x - x_3),$$

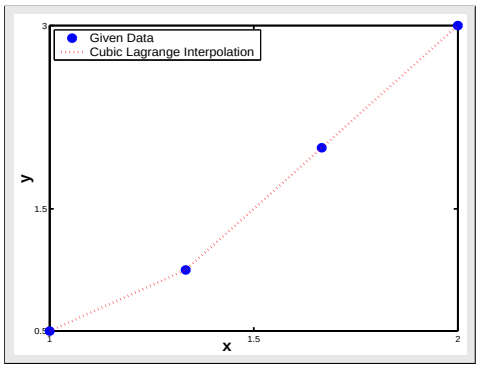


Figure 4.3: Graphical solution of the Example 4.28.

$$|f(x) - p_3(x)| = \frac{|f^{(4)}(\eta(x))|}{4!}|(x-x_0)||(x-x_1)||(x-x_2)||(x-x_3)|,$$

and $|f^{(4)}| \leq \frac{1}{5}$ *for* $1 \leq x \leq 2$*, we obtain*

$$|f(x) - p_3(x)| \leq \frac{1}{120}|(x-x_0)||(x-x_1)||(x-x_2)||(x-x_3)|.$$

Now for $1 \leq x \leq 2$*, we deduce that*

$$|x-1| \leq 1, \qquad |x-4/3| \leq 2/3, \qquad |x-5/3| \leq 2/3, \qquad |x-2| \leq 1.$$

Hence, the possible error in the cubic Lagrange polynomial is

$$|f(x) - p_3(x)| \leq \frac{1}{120}(1)(2/3)(2/3)(1) = 0.0037.$$

A third way, that is very convenient for interpolation in table, is the divided difference formula. While the Lagrange interpolation formula is at the heart of polynomial interpolation, it is not, by any stretch of the imagination, the most practical way to use it. Using this interpolation formula, there is no restriction on the spacing and order of the tabulating points $x_0, x_1, \ldots, x_n$. Also, the value of y (the dependent variable) can be calculated at any point x within minimum and maximum values of $x_0, x_1, \ldots, x_n$. But just consider for a moment that if we have to add an addition data point in the degree $p_2(x)$, in order to find cubic polynomial $p_3(x)$, we have to repeat the whole process again because we can not use the solution of the quadratic polynomial $p_2(x)$ in the construction of the cubic polynomial $p_3(x)$. Therefore one can note that the Lagrange method is not particular efficient for large values of n, the degree of the polynomial. When n is large and the data for x is ordered, some improvement in efficiency can be obtained by considering only the data pairs in the vicinity of the x values for which $f(x)$ is sought.
One will be quickly convinced that there must be better techniques available. In the following section we discuss some of the more practical approaches to polynomial interpolation. In using the following scheme the construction of the difference table plays an important role. In using the Lagrange interpolation scheme there is no need to construct difference table. •

4.2.2 Newton's General Interpolating Formula

Since we noted in the previous section that for a small number of data point one can easily use the Lagrange formula of the interpolating polynomial. However, for a large number of data points there will be many multiplication and more significantly, whenever a new data point is added to an existing set, the interpolating polynomial has to be completely recalculated. Here, we describe an efficient way of organizing the calculations so as to overcome these disadvantages.
Let us consider the *nth*-degree polynomial $p_n(x)$ that agrees with the function $f(x)$ at the distinct numbers $x_0, x_1, \ldots, x_n$. The divided differences of $f(x)$ with respect to $x_0, x_1, \ldots, x_n$ are derived to express $p_n(x)$ in the form

$$\begin{aligned} f(x) = p_n(x) = a_0 &+ a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots \\ &+ a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}), \end{aligned} \tag{4.43}$$

for appropriate constants $a_0, a_1, \ldots, a_n$.
Now to determine the constants, firstly, by evaluating $p_n(x)$ at x_0, we have

$$p_n(x_0) = a_0 = f(x_0) \tag{4.44}$$

Similarly, when $p_n(x)$ is evaluated at x_1, then

$$p_n(x_1) = a_0 + a_1(x_1 - x_0) = f(x_1), \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{4.45}$$

Divided Differences

Now we express the interpolating polynomial in terms of divided difference.

Firstly, we define the *Zeroth divided difference* at the point x_i by

$$f[x_i] = f(x_i), \tag{4.46}$$

which is simply the value of the function $f(x)$ at x_i.

The *first-order* or *first divided difference* at the points x_i and x_{i+1} can be defined by

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}. \tag{4.47}$$

In general, the *nth divided difference* $f[x_i, x_{i+1}, \ldots, x_{i+n}]$ is defined by

$$f[x_i, x_{i+1}, \ldots, x_{i+n}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+n}] - f[x_i, x_{i+1}, \ldots, x_{i+n-1}]}{x_{i+n} - x_i}. \tag{4.48}$$

By using this definition, (4.44) and (4.45) can be written as

$$a_0 = f[x_0]; \qquad a_1 = f[x_0, x_1],$$

Table 4.1: Divided difference table for a function $y = f(x)$.

k	x_k	Zero Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference
0	x_0	$f[x_0]$			
1	x_1	$f[x_1]$	$f[x_0, x_1]$		
2	x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
3	x_3	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$

respectively. Similarly, one can have the values of other constants involving in (4.43) such as

$$\begin{aligned} a_2 &= f[x_0, x_1, x_2], \\ a_3 &= f[x_0, x_1, x_2, x_3], \\ \cdots &= \cdots \\ a_n &= f[x_0, x_1, \ldots, x_n]. \end{aligned}$$

Putting the values of these constants in (4.43), we get

$$\begin{aligned} f(x) = p_n(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}), \end{aligned} \tag{4.49}$$

which can also be written as

$$f(x) = p_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}). \tag{4.50}$$

This type of polynomial is known as the *Newton's interpolatory divided difference polynomial*. Table 4.1 shows the divided difference for a function $f(x)$.

Example 4.29 *Construct the fourth divided differences table for* $f(x) = 4x^4 + 3x^3 + 2x^2 + 10$ *for the values* $x = 3, 4, 5, 6, 7, 8$.

Solution. *The result are listed in Table 4.2.*

From the results in Table 4.2, one can note that the nth divided difference for the nth polynomial equation is always constant and the (n+1)th divided difference is always zero for the nth polynomial equation. •

Using the following MATLAB command one can construct the Table 4.2 as follows:

```
>> x = [3 4 5 6 7 8]; y = 4*x.^4+3*x.^3+2*x.^2+10; D = divdiff(x,y)
```

Divided differences are now can be used to write the Newton's interpolating polynomial. Starting with the constant interpolating polynomial

$$p_0(x) = f[x_0].$$

Table 4.2: Divided differences table for $f(x) = 4x^4 + 3x^3 + 2x^2 + 10$ at given points.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference	Fourth Divided Difference	Fifth Divided difference
0	3	433					
1	4	1258	825				
2	5	2935	1677	426			
3	6	5914	2979	651	75		
4	7	10741	4827	924	91	4	
5	8	18058	7317	1245	107	4	0

4.2.2.1 Linear Newton's Interpolating Polynomial

The linear Newton's interpolating polynomial passing through two points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ can be written as

$$f(x) = p_1(x) = f[x_0] + (x - x_0)f[x_0, x_1].$$

4.2.2.2 Quadratic Newton's Interpolating Polynomial

The quadratic Newton's interpolating polynomial passing through the points $(x_0, f(x_0))$, $(x_1, f(x_1))$ and $(x_2, f(x_2))$ can be written in terms of divided differences as

$$f(x) = p_2(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2].$$

$$f(x) = p_2(x) = p_1(x) + (x - x_0)(x - x_1)f[x_0, x_1, x_2],$$

that is, the interpolating polynomial of degree 2 makes full use of the polynomial of degree 1, simply adding one extra term to $p_1(x)$. This is one of the advantages of the Newton's polynomial over Lagrange polynomial. Also, the total arithmetic operations for preparation of the table and polynomial are $3n(n + 3)/2$ which is less than the number of arithmetic operations of Lagrange polynomial $(3n^2 + 5n + 1)$.

4.2.2.3 Cubic Newton's Interpolating Polynomial

Similarly, the cubic Newton's interpolating polynomial passing through the points $(x_0, f(x_0))$, $(x_1, f(x_1)), (x_2, f(x_2))$ and $(x_3, f(x_3))$ can be written in terms of divided differences as

$$p_3(x) = f[x_0]+(x-x_0)f[x_0, x_1]+(x-x_0)(x-x_1)f[x_0, x_1, x_2]+(x-x_0)(x-x_1)(x-x_2)f[x_0, x_1, x_2, x_3].$$

This polynomial can also be written as

$$p_3(x) = p_2(x) + (x - x_0)(x - x_1)(x - x_2)f[x_0, x_1, x_2, x_3],$$

that is, the interpolating polynomial of degree 3 makes full use of the polynomial of degree 2, simply adding one extra term to $p_2(x)$. Note that using linear polynomial in quadratic polynomial, the starting point x_0 for both polynomials should be same.

4.2.2.4 Nth Degree Newton's Interpolating Polynomial

Repeating this entire process again, $p_3(x), p_4(x)$ and higher degree interpolating polynomials can be consecutively obtained in the same way. In general, the interpolating polynomial $p_n(x)$ passing through the points $(x_i, f(x_i))(i = 0, 1, \ldots, n)$, can be written in terms of divided differences as

$$\begin{aligned} f(x) = p_n(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}), \end{aligned} \tag{4.51}$$

which can also be written as

$$f(x) = p_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}), \tag{4.52}$$

or

$$f(x) = p_n(x) = f[x_0] + \sum_{k=0}^{n} f[x_0, x_1, \cdots, x_k] \prod_{i=0}^{k-1}(x - x_i). \tag{4.53}$$

This type of polynomial is known as the *Newton's interpolatory divided difference polynomial.*

Theorem 4.4 (Newton's Interpolating Polynomial)

Suppose that $x_0, x_1, \ldots, x_n$ are $(n + 1)$ distinct points in the interval $[a, b]$. There exists a unique polynomial $p_n(x)$ of degree at most n with the property that

$$f(x_i) = p_n(x_i), \quad \textit{for} \quad i = 0, 1, \ldots, n.$$

The Newton's form of this polynomial is

$$f(x) = p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}),$$

where

$$a_k = f[x_0, x_1, x_2, \cdots, x_k], \quad \textit{for} \quad k = 0, 1, 2, \ldots, n.$$

One can easily show that (4.52) is simply a rearrangement of the Lagrange form defined by (4.16). For example, the Newton divided difference interpolation polynomial of degree one is

$$f(x) = p_1(x) = f[x_0] + f[x_0, x_1](x - x_0),$$

which implies that

$$\begin{aligned} f(x) = p_1(x) &= f(x_0) + \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0}\right)(x - x_0) \\ &= \frac{(x_1 - x_0)f(x_0) + (x - x_0)f(x_1) - f(x_0)(x - x_0)}{x_1 - x_0} \\ &= \left(\frac{x - x_1}{x_0 - x_1}\right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right) f(x_1) = L_0(x)f(x_0) + L_1(x)f(x_1), \end{aligned}$$

which is the Lagrange interpolating polynomial of degree one. Similarly, one can show the equivalent for the *nth*-degree polynomial.

Example 4.30 *Find the Lagrange and the Newton forms of the interpolating polynomial for the following data*

x	0	1	3
$f(x)$	1	2	3

Write both polynomials in the form $a + bx + cx^2$ to verify that they are identical as functions.

Solution. *With $x_0 = 0, x_1 = 1$ and $x_2 = 3$, we obtain the quadratic Lagrange interpolating polynomial*

$$\begin{aligned} f(x) = p_2(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2) \\ &= \frac{(x-1)(x-3)}{(0-1)(0-3)}(1) + \frac{(x-0)(x-3)}{(1-0)(1-3)}(2) + \frac{(x-0)(x-1)}{(3-0)(3-1)}(3). \end{aligned}$$

$$f(x) = p_2(x) = 1 + \frac{7}{6}x - \frac{1}{6}x^2.$$

The result of the divided difference is listed in Table 4.3.
Since the Newton's interpolating polynomial of degree 2 is defined as

Table 4.3: Divided differences table for the Example 4.30.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference
0	0	**1**		
1	1	2	**1**	
2	3	3	$\frac{1}{2}$	$-\frac{\mathbf{1}}{\mathbf{6}}$

$$f(x) = p_2(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

By using Table 4.3, we have Newton's polynomial

$$f(x) = p_2(x) = 1 + (1)(x-0) + \left(-\frac{1}{6}\right)(x-0)(x-1) = 1 + \frac{7}{6}x - \frac{1}{6}x^2,$$

which show that both polynomials are identical as functions. •

Example 4.31 *Show that the Newton's interpolating polynomial $p_2(x)$ of degree 2 satisfies the interpolation conditions*

$$p_2(x_i) = f(x_i), \qquad i = 0, 1, 2.$$

Solution. *Since the Newton's interpolating polynomial of degree 2 is*

$$f(x) = p_2(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

First take $x = x_0$, we have

$$p_2(x_0) = f[x_0] + 0 + 0 = f(x_0).$$

Now take $x = x_1$, we have

$$p_2(x_1) = f[x_0] + f[x_0, x_1](x_1 - x_0) + 0 = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_1 - x_0) = f(x_1).$$

Finally, take $x = x_2$, we have

$$p_2(x_2) = f[x_0] + f[x_0, x_1](x_2 - x_0) + f[x_0, x_1, x_2](x_2 - x_0)(x_2 - x_1),$$

which can be written as

$$p_2(x_2) = f[x_0] + f[x_0, x_1](x_2 - x_0) + \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}(x_2 - x_0)(x_2 - x_1).$$

$$p_2(x_2) = f[x_0] + f[x_0, x_1](x_1 - x_0) + f[x_1, x_2](x_2 - x_1).$$

From (4.47), we have

$$p_2(x_2) = f[x_0] + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_1 - x_0) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x_2 - x_1).$$

$$p_2(x_2) = f(x_0) + f(x_1) - f(x_0) + f(x_2) - f(x_1) = f(x_2).$$

Table 4.4: Divided differences table for $f(x) = e^x$ at given points.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference	Fourth Divided difference
0	-1	2				
1	0	1	-1			
2	1	2	1	1		
3	2	-7	-9	-5	-2	
4	3	10	17	13	6	2

Example 4.32 *The cubic Newton's polynomial $p_3(x) = 2 - (x+1) + x(x+1) - 2x(x+1)(x-1)$ interpolates the first four points in the following table:*

x	-1	0	1	2	3
$f(x)$	2	1	2	-7	10

By adding one additional term (using point $(3, 10)$) to $p_3(x)$, find the Newton's polynomial $p_4(x)$ that interpolates the whole table and then use it to find the approximation of $f(0.5)$.

Solution. *Since the Newton's polynomial for the whole table data points is the four degree Newton's interpolating polynomial and it can be written as*

$$f(x) = p_4(x) = p_3(x) + x(x+1)(x-1)(x-2)f[x_0, x_1, x_2, x_3].$$

Now to find to find fourth divided difference $f[x_0, x_1, x_2, x_3]$, we have to construct the required divided differences table. The result are listed in Table 4.4.
Thus the Newton's interpolating polynomial passing through all the given data points is

$$f(x) = p_4(x) = 2 - (x+1) + x(x+1) - 2x(x+1)(x-1) + 2x(x+1)(x-1)(x-2).$$

Thus at $x = 0.5$, we get, $f(0.5) \approx p_4(0.5) = 3.1250$, the required approximation of the function. •

Example 4.33 *Consider the following table of date points*

x	3	1	5	6
$f(x)$	1	-3	2	4

Find the third divided difference $f[3, 1, 5, 6]$ and use it to find the Newton's form of the interpolating polynomial. Find approximation of $f(2)$.

Solution. *The third divided differences for the given data points are listed in Table 4.5. The cubic*

Table 4.5: Divided difference table for the function $y = f(x)$ at the given points.

k	x_k	Zero Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference
0	$x_0 = 3$	$f[x_0] = 1$			
1	$x_1 = 1$	$f[x_1] = -3$	$f[x_0, x_1] = 2$		
2	$x_2 = 5$	$f[x_2] = 2$	$f[x_1, x_2] = 5/4$	$f[x_0, x_1, x_2] = -3/8$	
3	$x_3 = 6$	$f[x_3] = 4$	$f[x_2, x_3] = 2$	$f[x_1, x_2, x_3] = 3/20$	$f[x_0, x_1, x_2, x_3] = 7/40$

Newton's interpolating polynomial passing through the given points can be written as

$$p_3(x) = f[x_0] + (x-x_0)f[x_0, x_1] + (x-x_0)(x-x_1)f[x_0, x_1, x_2] + (x-x_0)(x-x_1)(x-x_2)f[x_0, x_1, x_2, x_3],$$

so using Table 4.5, we have

$$f(x) = p_3(x) = 1 + 2(x - x_0) - \frac{3}{8}(x - x_0)(x - x_1) + \frac{7}{40}(x - x_0)(x - x_1)(x - x_2),$$

$$f(x) = p_3(x) = \frac{1}{40}[7x^3 - 78x^2 + 301x - 350] \quad \textit{and quad} f(2) \approx p_3(2) = -\frac{1}{10},$$

is the required approximation of the function at $x = 2$. •

Example 4.34 *Show that the cubic Newton's interpolating polynomial using the given four points $x_0 = 0$, $x_1 = 0.25$, $x_2 = 0.5$ and $x_3 = 0.75$ is $p_3(x) = \frac{1}{3}(16x^3 + 11x + 3)$. Use it to find the approximation of the function at $x = 0.65$.*

Solution. *Since*

$$f(x) = p_3(x) = \frac{1}{3}(16x^3 + 11x + 3),$$

therefore, the table of date points is

x	0	0.25	0.5	0.75
$f(x)$	1	2	3.5	6

The third divided differences for the given data points are listed in Table 4.6. The cubic Newton's interpolating polynomial passing through the given points can be written as

$$p_3(x) = f[x_0]+(x-x_0)f[x_0,x_1]+(x-x_0)(x-x_1)f[x_0,x_1,x_2]+(x-x_0)(x-x_1)(x-x_2)f[x_0,x_1,x_2,x_3],$$

so using Table 4.6, we have

$$f(x) = p_3(x) = 1 + 4(x-0) + 4(x-0)(x-0.25) + \frac{16}{3}(x-0)(x-0.25)(x-0.5),$$

$$f(x) = p_3(x) = \frac{1}{3}[16x^3 + 11x + 3].$$

$$f(0.65) \approx p_3(0.65) = \frac{1}{30}[16(0.65)^3 + 11(0.65) + 3] = 4.8480,$$

the required approximation of the function at the given point, that is, $f(0.65) = 4.8480$. •

Table 4.6: Divided difference table for $f(x) = \frac{1}{3}(16x^3 + 11x + 3)$ at the given points.

k	x_k	Zero Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference
0	$x_0 = 0$	$f[x_0] = 1$			
1	$x_1 = 0.25$	$f[x_1] = 2$	$f[x_0, x_1] = 4$		
2	$x_2 = 0.5$	$f[x_2] = 3.5$	$f[x_1, x_2] = 6$	$f[x_0, x_1, x_2] = 4$	
3	$x_3 = 0.75$	$f[x_3] = 6$	$f[x_2, x_3] = 10$	$f[x_1, x_2, x_3] = 8$	$f[x_0, x_1, x_2, x_3] = 16/3$

Example 4.35 *Consider the following table of date points*

x	-1	3	2	-2	4
$f(x)$	8	0	-1	15	3

Is this table form a polynomial ? If so, what maximum degree of the polynomial can be obtain to approximate $f(0)$.

Solution. *The result are listed in Table 4.7. Since all the second order differences are the same they equal* 1*, so that means the maximum degree of the polynomial will be quadratic. By using Table 4.7, it can be written as*

$$f(x) = p_2(x) = 8 - 2(x+1) + 1(x+1)(x-3) = x^2 - 4x + 3,$$

because of collinear data points (having a common line). Thus, $f(0) \approx p_2(0) = 3$*, the required approximation of the function at* $x = 0$. •

Table 4.7: Divided difference table for the function $y = f(x)$ at the given points.

k	x_k	Zero Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference
0	$x_0 = -1$	$f[x_0] = 8$			
1	$x_1 = 3$	$f[x_1] = 0$	$f[x_0, x_1] = -2$		
2	$x_2 = 2$	$f[x_2] = -1$	$f[x_1, x_2] = 1$	$f[x_0, x_1, x_2] = 1$	
3	$x_3 = -2$	$f[x_3] = 15$	$f[x_2, x_3] = -4$	$f[x_1, x_2, x_3] = 1$	$f[x_0, x_1, x_2, x_3] = 0$
4	$x_4 = 4$	$f[x_3] = 3$	$f[x_2, x_3] = -2$	$f[x_1, x_2, x_3] = 1$	$f[x_1, x_2, x_3, x_4] = 0$

Example 4.36 *Repeat the Example 4.35 by adding $f(0) = 5$, that is, the following table of the data points of the form*

x	0	−1	3	2	−2	4
$f(x)$	5	8	0	−1	15	3

Solution. *The result are listed in Table 4.8 shows the value of the fifth order differences, so that means the maximum degree of the polynomial will be five. By using Table 4.8, gives*

$$\begin{aligned} f(x) = p_5(x) &= 5 - 3x + 0.3333x(x+1) + 0.3333x(x+1)(x-3) + 0.1667x(x+1)(x-3)(x-2) \\ &- 0.0417x(x+1)(x-3)(x-2)(x+2), \end{aligned}$$

$$f(x) = p_5(x) = \frac{1}{10000}\left[-417x^5 + 2501x^4 - 416x^3 - 5002x^2 - 15834x - 50000\right],$$

the required fifth degree polynomial. •

Table 4.8: Divided difference table for the function $y = f(x)$.

k	x_k	ODD	1DD	2DD	3DD	4DD	5DD
0	0	5					
1	-1	8	-3				
2	3	0	-2	0.3333			
3	2	-1	1	1	0.3333		
4	-2	15	-4	1	0	0.1667	
5	4	3	-2	1	0	0	-0.0417

Example 4.37 *Construct the divided differences table for $f(x) = x^3 + 7x^2 + 1$ using the values $x = 1, 2, 3, 4, 5$. If the approximation of $f(3.5)$ by linear Newton's polynomial is 134, then use it to find the best approximation of $f(3.5)$ by using quadratic Newton's polynomial.*

Solution. *The result are listed in Table 4.9. Since the quadratic Newton's polynomial is*

$$f(x) = p_2(x) = p_1(x) + (x - x_0)(x - x_1)f[x_0, x_1, x_2].$$

After choosing the best points $x_0 = 3, x_1 = 4, x_2 = 5$, *and Table 4.12, we get*

$$f(3.5) \approx p_2(3.5) = p_1(3.5) + (3.5-3)(3.5-4)f[3,4,5] = 134 - (0.25)(19) = 129.25,$$

and the absolute error, $|f(3.5) - p_2(3.5)| = |129.625 - 129.25| = 0.375$. *From the results in Table 4.9, one can note that the nth divided difference for the nth degree polynomial equation is always constant and the (n+1)th divided difference is always zero for the nth degree polynomial equation.* •

Using the following MATLAB command one can construct the Table 4.9 as follows:

```
>> x = [1 2 3 4 5]; y = x.^ 3+7 * x.^ 2+1; D = divdiff(x, y);
```

Table 4.9: Divided differences table for given $f(x) = x^3 + 7x^2 + 1$.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference	Fourth Divided Difference
0	1	9				
1	2	37	28			
2	3	91	54	13		
3	4	177	86	16	1	
4	5	301	124	19	1	0

The main advantage of the Newton divided difference form over the Lagrange form is that new polynomial $p_n(x)$ can be calculated from the given polynomial $p_{n-1}(x)$ by adding just one extra term, since it follows from (4.52) that

$$f(x) = p_n(x) = p_{n-1}(x) + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}). \tag{4.54}$$

Example 4.38 *(a) Construct the divided difference table for the function* $f(x) = \ln(x+2)$ *in the interval* $0 \le x \le 3$ *for the stepsize* $h = 1$.

(b) Use Newton divided difference interpolation formula to construct the interpolating polynomials of degree 2 and degree 3 (using degree 2) to approximate $\ln(3.5)$.

(c) Compute error bounds for the approximations in part (b).

Solution. *(a) The results of the divided differences are listed in Table 4.10.*

(b) Firstly, we construct the second degree polynomial $p_2(x)$ *by using the quadratic Newton interpolation formula as follows*

$$f(x) = p_2(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1),$$

then with the help of the divided differences Table 4.10, we get

$$f(x) = p_2(x) = 0.6932 + 0.4055(x - 0) - 0.0589(x - 0)(x - 1),$$

Table 4.10: Divide differences table for the Example 4.38.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference
0	0	0.6932			
1	1	1.0986	0.4055		
2	2	1.3863	0.2877	- 0.0589	
3	3	1.6094	0.2232	- 0.0323	0.0089

which implies that

$$f(x) = p_2(x) = -0.0568x^2 + 0.4644x + 0.6932 \quad \textit{and} \quad p_2(1.5) = 1.2573,$$

with possible absolute error

$$|f(1.5) - p_2(1.5)| = |1.2528 - 1.2573| = 0.0045.$$

Now to construct the cubic interpolatory polynomial $p_3(x)$ that fits at all four points. We only have to add one more term to the polynomial $p_2(x)$:

$$f(x) = p_3(x) = p_2(x) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2),$$

and this gives

$$f(x) = p_3(x) = p_2(x) + 0.0089(x^3 - 3x^2 + 2x) \textit{ and } f(1.5) \approx p_3(1.5) = 1.2573 - 0.0033 = 1.2540,$$

with possible absolute error

$$|f(1.5) - p_3(1.5)| = |1.2528 - 1.2540| = 0.0012.$$

We note that the estimated value of $f(1.5)$ by cubic interpolating polynomial is more closer to the exact solution than the quadratic polynomial.

(c) Now to compute the error bounds for the approximations $p_2(x)$ and $p_3(x)$ in part (b), we use the error formula (4.30). For the polynomial $p_2(x)$, we have

$$|f(x) - p_2(x)| = \frac{|f'''(\eta(x))|}{3!}|(x - x_0)(x - x_1)(x - x_2)|.$$

The third derivative of the given function is given as

$$f'''(x) = \frac{2}{(x+2)^3} \quad \textit{and} \quad |f'''(\eta(x))| = \Big|\frac{2}{(\eta(x)+2)^3}\Big|, \quad \textit{for} \quad \eta(x) \in (0, 2).$$

Then

$$M = \max_{0 \le x \le 2} \left|\frac{2}{(x+2)^3}\right| = 0.25, \textit{ and } \quad |f(1.5) - p_2(1.5)| \le (0.375)(0.25)/6 = 0.0156.$$

Since the error bound for the cubic polynomial $p_3(x)$ is

$$|f(x) - p_3(x)| = \frac{|f^{(4)}(\eta(x))|}{4!}|(x-x_0)(x-x_1)(x-x_2)(x-x_3)|.$$

Taking the fourth derivative of the given function, we obtain

$$f^{(4)}(x) = \frac{-6}{(x+2)^4} \quad \textit{and} \quad |f^{(4)}(\eta(x))| = \Big|\frac{-6}{(\eta(x)+2)^4}\Big|, \quad \textit{for} \quad \eta(x) \in (0,3).$$

Since

$$|f^{(4)}(0)| = 0.375 \quad \textit{and} \quad |f^{(4)}(3)| = 0.0096,$$

so $|f^{(4)}(\eta(x))| \leq \max\limits_{0\leq x\leq 3}\Big|\frac{-6}{(x+2)^4}\Big| = 0.375$ *and it gives*

$$|f(1.5) - p_3(1.5)| \leq (0.5625)(0.375)/24 = 0.0088,$$

which is the required error bound for the approximation $p_3(1.5)$. •

Note that in the above Example 4.38, we used the value of the quadratic polynomial $p_2(1.5)$ in calculating the cubic polynomial $p_3(1.5)$. It was possible because the initial value for both polynomials was the same as $x_0 = 0$. But the situation will be quite different if the initial point for both polynomials will be different. For example, if we have to find the approximate value of $\ln(4.5)$, then the suitable data points for the quadratic polynomial will be $x_0 = 1, x_1 = 2, x_2 = 3$ and for the cubic polynomial will be $x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 3$. So for getting the best approximation of $\ln(4.5)$ by the cubic polynomial $p_3(2.5)$, we can not use the value of the quadratic polynomial $p_2(2.5)$ in the cubic polynomial $p_3(2.5)$. The best way is to use the following cubic polynomial form

$$\begin{aligned} p_3(2.5) &= f[0] + (2.5-0)f[0,1] + (2.5-0)(2.5-1)f[0,1,2] \\ &+ (2.5-0)(2.5-1)(2.5-2)f[0,1,2,3], \end{aligned}$$

$$f(2.5) \approx p_3(2.5) = 0.6932 + 1.0137 - 0.2208 + 0.0166 = 1.5027.$$

One can use MATLAB command reproduce the results of the Example 4.38 as follows:

$>> x = [0\ 1\ 2\ 3];\ y = log(x+2);\ x0 = 1.5;\ Y = Ndivf(x,y,x0)$

Program 4.3

MATLAB m-file for Linear Newton's Interpolation Method

function Y=Ndivf(x,y,x0)

$m = length(x); D = zeros(m,m); D(:,1) = y(:);$

for j=2:m; for i=j:m;

$D(i,j) = (D(i,j-1) - D(i-1,j-1))/(x(i) - x(i-j+1)); end; end;$

$Y = D(m,m) * ones(size(x0));$

for $i = m-1:-1:1;\ Y = D(i,i) + (x0 - x(i)) * Y;$ end

Example 4.39 *Consider the points $x_0 = 0, x_1 = 0.4, x_2 = 0.7$ and for a function $f(x)$, the divided differences are $f[x_2] = 6, f[x_1, x_2] = 10, f[x_0, x_1, x_2] = 50/7$. Use linear Newton's polynomial $p_1(x)$ to find quadratic Newton's polynomial $p_2(x)$ for the approximation of $f(0.3)$.*

Solution. *First we construct the complete divided differences table for the given data points. Since we know that the second divided difference is defined as*

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}, \qquad \frac{50}{7} = \frac{10 - f[x_0, x_1]}{0.7 - 0}.$$

Solving for $f[x_0, x_1]$, we have, $f[x_0, x_1] = 5$. We need to find the values of the zeroth order divided differences $f[x_0]$ and $f[x_1]$ which can be obtained by using the first-order divided differences $f[x_0, x_1]$ and $f[x_1, x_2]$. Firstly, we find the value of $f[x_1]$ as follows

$$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}, \qquad 10 = \frac{6 - f[x_1]}{0.7 - 0.4}, \qquad f[x_1] = 6 - 10(0.3) = 3.$$

The other zeroth divided difference $f[x_0]$ can be computed as follows

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}, \qquad 5 = \frac{3 - f[x_0]}{0.4 - 0}, \qquad f[x_0] = 3 - 5(0.4) = 1.$$

The completed divided differences table is shown by Table 4.11. We first find the linear Newton's

Table 4.11: Divided differences table for the Example 4.39.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference
0	0	1		
1	0.4	3	5	
2	0.7	6	10	$\frac{50}{7}$

polynomial to approximate $f(0.3)$ using Table 4.11 as follows:

$$f(0.3) \approx p_1(0.3) = 1 + (5)(0.3 - 0.0) = 1 + 1.5 = 2.5,$$

and then use it to find quadratic Newton's polynomial using Table 4.11, we have

$$f(0.3) \approx p_2(0.3) = p_1(0.3) + \frac{50}{7}(0.3 - 0)(0.3 - 0.4) = 2.5 - 0.2143 = 2.2857,$$

the approximation of $f(0.3)$ using quadratic Newton's polynomial. •

Example 4.40 *Let $x_0 = 0.5, x_1 = 0.7, x_2 = 0.9, x_3 = 1.1, x_4 = 1.3, x_5 = 1.5$. Construct the divided difference table for the function $f(x) = e^x$. Use Newton polynomial $p_5(x)$ of degree five to approximate the function $f(x) = e^x$ at $x = 0.6$ when $p_4(0.6) = 1.9112$. Also, compute an error bound for your approximation.*

Solution. *Using Newton polynomial $p_5(x)$ and the given data points, give*

$$f(x) = p_5(x) = p_4(x) + (x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)f[x_0,x_1,x_2,x_3,x_4,x_5],$$

$$f(0.6) \approx p_5(0.6) = p_4(0.6) + (0.1)(-0.1)(-0.3)(-0.5)(-0.7)f[0.5,0.7,0.9,1.1,1.3,1.5].$$

$$p_5(0.6) = 1.9112 + (0.0010)(0.0228) = 1.9112228.$$

Since the error bound for the fifth-degree polynomial $p_5(x)$ is

Table 4.12: Divided differences table for $y = f(x)$ at the given points.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference	Fourth Divided Difference	Fifth Divided difference
0	0.5	1.6487					
1	0.7	2.0138	1.8252				
2	0.9	2.4596	2.2293	1.0102			
3	1.1	3.0042	2.7228	1.2339	0.3728		
4	1.3	3.6693	3.3257	1.5071	0.4553	0.1032	
5	1.5	4.4817	4.0620	1.8408	0.4553	0.1260	0.0228

$$|f(x) - p_5(x)| = \frac{|f^{(6)}(\eta(x))|}{6!}|(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)|.$$

Taking the sixth derivative of the given function, we have

$$f^{(6)}(x) = e^x, \quad |f^{(6)}(0.5)| = 1.6487, |f^{(6)}(1.5)| = 4.4817, \quad |f^{(6)}(\eta(x))| \leq \max_{0.5\leq x\leq 1.5}|e^x| = 4.4817.$$

$$|f(0.6) - p_5(0.6)| \leq (0.00095)(4.4817)/720 = 0.000006,$$

which is the required error bound for the approximation $p_5(0.6)$. •

In the case of the Lagrange interpolating polynomial we derived an expression for the truncation error in the form given by (4.30), namely, that

$$R_{n+1}(x) = \frac{f^{(n+1)}(\eta(x))}{(n+1)!}L_n(x), \quad \text{where} \quad L_n(x) = (x-x_0)(x-x_1)\cdots(x-x_n).$$

For the Newton's divided difference formula, we obtain, following the same reasoning as above

$$\begin{aligned} f(x) &= f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1) + \cdots \\ &+ f[x_0,x_1,\ldots,x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1}) \\ &+ f[x_0,x_1,\ldots,x_n,x](x-x_0)(x-x_1)\cdots(x-x_{n-1})(x-x_n), \end{aligned}$$

which can also be written as

$$f(x) = p_n(x) + f[x_0,x_1,\ldots,x_n,x](x-x_0)(x-x_1)\cdots(x-x_n), \tag{4.55}$$

$$f(x) - p_n(x) = L_n(x)f[x_0,x_1,\ldots,x_n,x]. \tag{4.56}$$

Since the interpolation polynomial agreeing with $f(x)$ at $x_0, x_1, \ldots, x_n$ is unique, it follows that these two error expressions must be equal. •

Theorem 4.5 *Let $p_n(x)$ be the polynomial of degree at most n that interpolates a function $f(x)$ at a set of $n+1$ distinct points $x_0, \ldots, x_n$. If x is a point different from the points $x_0, \ldots, x_n$, then*

$$f(x) - p_n(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{j=0}^{n} (x - x_j). \tag{4.57}$$

One can easily shows that the relationship between the divided differences and the derivative. From (4.48), we have

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Now apply the Mean Value Theorem to the above equation implies that when the derivative f' exists, then we have

$$f[x_0, x_1] = f'(\eta(x)),$$

for unknown point $\eta(x)$ lies between x_0 and x_1. Following theorem generalizes this result. •

4.2.2.5 Newton's Divided Differences Interpolation at Equally Spaced Points

For the equally spaced points $x_i = x_0 + ih$, that is, $x_0, x_1, \ldots, x_k$, the divided differences of a function can be defined as:

$$\begin{aligned}
f[x_0, x_1] &= \frac{1}{1!h}[f(x_1) - f(x_0)] = \frac{1}{h}\Delta f(x_0) \\
f[x_0, x_1, x_2] &= \frac{1}{2!h^2}[f(x_2) - 2f(x_1) + f(x_0)] = \frac{1}{2h^2}\Delta^2 f(x_0) \\
f[x_0, x_1, x_2, x_3] &= \frac{1}{3!h^3}[f(x_3) - 3f(x_2) + 3f(x_1) - f(x_0)] = \frac{1}{6h^3}\Delta^3 f(x_0) \\
\cdots &= \cdots \\
f[x_0, x_1, \ldots, x_k] &= \frac{1}{k!h^k}\Delta^k f(x_0)
\end{aligned}$$

Example 4.41 *Let $f(x) = x^3$ and equally spaced points $x_0 = 1, x_1 = 2, x_2 = 3, x_3 = 4$. Use cubic Newton polynomial to approximate $(3.5)^3$ and compute the absolute error.*

Solution. *For using cubic Newton polynomial $p_3(x)$ and the given data points, with $h = 1$, give*

$$p_3(3.5) = f[1] + (3.5-1)f[1,2] + (3.5-1)(3.5-2)f[1,2,3] + (3.5-1)(3.5-2)(3.5-3)f[1,2,3,4],$$

$$\begin{aligned}
f[1,2] &= \frac{1}{1}[f(2) - f(1)] = 7 \\
f[1,2,3] &= \frac{1}{2}[f(3) - 2f(2) + f(1)] = \frac{1}{2}[27 - 2(8) + 1] = 6 \\
f[1,2,3,4] &= \frac{1}{6}[f(4) - 3f(3) + 3f(2) - f(1)] = \frac{1}{6}[64 - 81 + 24 - 1] = 1
\end{aligned}$$

Thus

$$(3.5)^3 \approx= p_3(3.5) = 1 + (3.5-1)(7) + (3.5-1)(3.5-2)(6) + (3.5-1)(3.5-2)(3.5-3)(1) = 42.875,$$

with absolute error, $|(3.5)^3 - 42.875| = |42.875 - 42.875| = 0.0000$. •

Example 4.42 *Let* $f(x) = \ln(x+2) + e^{-(x+2)}$ *defined over the interval* $[1, 2]$. *Find the approximation of* $\ln(3.8) + e^{-3.8}$ *by using the cubic Newton's polynomial if approximation by quadratic Newton's polynomial is* 1.3572. *Compute the absolute error and the number of points to get the accuracy* 10^{-6} *using the cubic Newton's polynomial.*

Solution. *Since* $h = (2-1)/3 = 1/3$; *so* $x_0 = 1, x_1 = 4/3, x_2 = 5/3, x_3 = 2$, *and* $x = 1.8$. *Using quadratic polynomial* $p_2(x)$ *to find cubic Newton's polynomial* $p_3(x)$, *we have to use the formula*

$$f(x) = p_3(x) = p_2(x) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2),$$

and we can find third order divided difference as

$$f[1, 4/3, 5/3, 2] = \frac{27}{6}[f(2) - 3f(5/3) + 3f(4/3) - f(1)] = \frac{27}{6}[1.4046 - 3(1.3248) + 3(1.2396) - 1.1484] = 0.0028.$$

This gives

$$f(1.8) = p_3(1.8) = p_2(1.8) + (0.0028)(1.8-1)(1.8-4/3)(1.8-5/3) \text{ and } f(1.8) \approx p_3(1.8) = 1.3572 + 0.00014 = 1.3573,$$

with possible absolute error

$$|f(1.8) - p_3(1.8)| = |1.3574 - 1.3573| = 0.0001.$$

Finally, as the upper bound of error in cubic polynomial is

$$|E_3| \leq \frac{Mh^4}{24},$$

where

$$M = \max_{1 \leq x \leq 2} |f^{(4)}(x)| = \max_{1 \leq x \leq 2} \left| \frac{-6}{(x+2)^4} + e^{(-x-2)} \right| = 0.0243.$$

Since

$$\frac{Mh^4}{24} \leq 10^{-6}, \quad h^4 \leq \frac{(24) \times (10^{-6})}{M},$$

gives, $h \leq 0.1773$ *and* $n = 5.6409 \approx 6$. *Thus we need,* 7 *points, for the cubic interpolations.* •

Theorem 4.6 (Divided Differences and Derivatives)

Suppose that $f \in C^n[a, b]$ *and* $x_0, x_1, \ldots, x_n$ *are distinct number in* $[a, b]$. *Then for some point* $\eta(x)$ *in the interval* (a, b) *spanned by* $x_0, x_1, \ldots, x_n$ *exists with*

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\eta(x))}{n!}. \tag{4.58}$$

Example 4.43 *Let* $f(x) = x \ln x$, *and the points* $x_0 = 1.1, x_1 = 1.2, x_2 = 1.3$. *Compute the best approximate value for unknown point* $\eta(x)$ *by using the relation (4.58).*

Solution. *Given* $f(x) = x \ln x$, *then*

$$\begin{aligned} f(1.1) &= 1.1 \ln(1.1) = 0.1048, \\ f(1.2) &= 1.2 \ln(1.2) = 0.2188, \\ f(1.3) &= 1.3 \ln(1.3) = 0.3411. \end{aligned}$$

Since the relation (4.58) for the given data points is

$$f[x_0, x_1, x_2] = \frac{f''(\eta(x))}{2!}. \tag{4.59}$$

To compute the value of the left-hand side of the relation (4.59), we have to find the values of the first-order divided differences

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{0.2188 - 0.1048}{1.2 - 1.1} = 1.1400, \\ f[x_1, x_2] &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0.3411 - 0.2188}{1.3 - 1.2} = 1.2230, \\ f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{1.2230 - 1.1400}{1.3 - 1.1} = 0.4150. \end{aligned}$$

Now we calculate the right-hand side of the relation (4.59) for the given points and which gives us

$$\frac{f''(x_0)}{2} = \frac{1}{2x_0} = 0.4546, \quad \frac{f''(x_1)}{2} = \frac{1}{2x_1} = 0.4167, \quad \frac{f''(x_2)}{2} = \frac{1}{2x_2} = 0.3846.$$

We note that the left-hand side of (4.59) is nearly equal to the right-hand side when $x_1 = 1.2$. *Hence the best approximate value of* $\eta(x)$ *is* 1.2. •

Properties of Divided Differences

Now we discuss some of the nice properties of the divided differences as follows:

1. Divided difference of a constant is zero. Let $f(x) = a$, then

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{a - a}{x_1 - x_0} = 0.$$

2. Divided difference of $h(x) = af(x)$, a is constant, is the divided difference of $f(x)$ multiplied by a. Let $h(x) = af(x)$, then

$$h[x_0, x_1] = \frac{h(x_1) - h(x_0)}{x_1 - x_0} = \frac{af(x_1) - af(x_0)}{x_1 - x_0} = a\frac{f(x_1) - f(x_0)}{x_1 - x_0} = af[x_0, x_1].$$

3. Divided difference obeys linear property. Let $F(x) = af_1(x) + bf_2(x)$, then

$$\begin{aligned} F[x_0, x_1] &= \frac{F[x_0] - F[x_1]}{x_0 - x_1} = \frac{(af_1(x_0) + bf_2(x_0)) - (af_1(x_1) + bf_2(x_1))}{x_0 - x_1} \\ &= \frac{(af_1(x_0) - af_1(x_1)) + (bf_1(x_0) - bf_2(x_1))}{x_0 - x_1} \\ &= a\left(\frac{f_1(x_0) - f_1(x_1)}{x_0 - x_1}\right) + b\left(\frac{f_2(x_0) - bf_2(x_1)}{x_0 - x_1}\right) \\ &= af_1[x_0, x_1] + bf_2[x_0, x_1]. \end{aligned}$$

4. If $p_n(x)$ is a polynomial of degree n, then the divided differences of order n is always constant and $(n+1), (n+2), \ldots$ are identically zero.

5. The divided difference is a symmetric function of its arguments, that is, $f[x_0, x_1] = f[x_1, x_0]$. Thus if $(t_0, t_1, \ldots, t_n)$ is a permutation of $(x_0, x_1, \ldots, x_n)$, then

$$f[t_0, t_1, \ldots, t_n] = f[x_0, x_1, \ldots, x_n],$$

This can be verify easily, since the divided differences on the both sides of the above equation are the coefficient of x^n in the polynomial of degree at most n that interpolates $f(x)$ at the $n+1$ distinct points $t_0, t_1, \ldots, t_n$ and $x_0, x_1, \ldots, x_n$. These two polynomials are, of course, the same.

6. The interpolating polynomial of degree n can be obtained by adding a single term to the polynomial of degree $(n-1)$ expressed in the Newton form.

$$p_n(x) = p_{n-1}(x) + f[x_0, \ldots, x_n] \prod_{j=0}^{n-1} (x - x_j).$$

7. The divided difference $f[x_0, \ldots, x_{n-1}]$ is the coefficient of x^{n-1} in the polynomial that interpolates $(x_0, f_0), (x_1, f_1), \ldots, (x_{n-1}, f_{n-1})$.

8. A sequence of divided differences may be constructed recursively from the formula

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0},$$

and the zeroth-order divided difference is defined by

$$f[x_i] = f(x_i), \quad i = 0, 1, \ldots, n.$$

9. The another useful property of the divided difference can be obtained by using the definitions of the divided differences (4.48) and (4.51) which can be extended to the case where some or all of the points x_i are coincident, provided that $f(x)$ is sufficiently differentiable. For example, define

$$f[x_0, x_0] = \lim_{\epsilon \to 0} f[x_0, x_0 + \epsilon] = \lim_{\epsilon \to 0} \frac{f(x_0 + \epsilon) - f(x_0)}{\epsilon} = f'(x_0). \tag{4.60}$$

$$\begin{aligned}
f[x_0, x_0, x_0] &= \lim_{\epsilon \to 0} f[x_0, x_0, x_0 + \epsilon] = \lim_{\epsilon \to 0} \frac{f(x_0, x_0 + \epsilon) - f[x_0, x_0]}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{\frac{f(x_0+\epsilon) - f(x_0)}{\epsilon} - f'(x_0)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{f(x_0 + \epsilon) - f(x_0) - \epsilon f'(x_0)}{\epsilon^2} \quad \left(\frac{0}{0} \text{ form}\right) \\
&= \lim_{\epsilon \to 0} \frac{f'(x_0 + \epsilon) - f'(x_0)}{2\epsilon} \quad \text{(using } L'H\hat{o}\text{pital's rule)} \\
&= \frac{1}{2} \left[\lim_{\epsilon \to 0} \frac{f'(x_0 + \epsilon) - f'(x_0)}{\epsilon}\right] = \frac{f''(x_0)}{2}.
\end{aligned}$$

For an arbitrary, $n \geq 1$, let all the points in the Theorem 4.6 approach x_0. This leads to the definition

$$f[x_0, x_0, \ldots, x_0] = \frac{f^{(n)}(x_0)}{n!},$$

where the left hand side denotes the *nth* divided difference, all of whose points are x_0.

Example 4.44 *Let $f(x) = e^{-x}$ and let $x_0 = 0, x_1 = 1$. Using (4.58) and the above divide difference property 9, calculate $f[x_0, x_1, x_0]$, $f[x_0, x_0, x_1, x_1]$ and $f[x_0, x_1, x_1, x_1]$.*

Solution. *By using (4.58), we have $f[x_0, x_0] = \frac{1}{1!} f'(x_0) = f'(x_0)$. Therefore*

$$f[x_0, x_1, x_0] = f[x_0, x_0, x_1] = \frac{f[x_0, x_1] - f[x_0, x_0]}{x_1 - x_0} = \frac{f[x_0, x_1] - f'(x_0)}{x_1 - x_0}.$$

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}, \quad \textit{gives} \quad f[0, 1] = \frac{0.368 - 1}{1 - 0} = -0.632.$$

Using $f'(x_0) = -e^{-x_0}$, $f'(0) = -1$, we obtain, $f[0, 1, 0] = f[0, 0, 1] = \frac{-0.632 + 1}{1 - 0} = 0.368$. Now to find the value of the third divided difference which is defined as

$$f[x_0, x_0, x_1, x_1] = \frac{f[x_0, x_1, x_1] - f[x_0, x_0, x_1]}{x_1 - x_0},$$

$$f[x_0, x_0, x_1, x_1] = \frac{f'(x_1) - 2f[x_0, x_1] + f'(x_0)}{(x_1 - x_0)^2}.$$

$$f[0, 0, 1, 1] = \frac{-0.368 - 2(-0.632) - 1}{(1 - 0)^2} = -0.014.$$

$$f[x_0, x_1, x_1, x_1] = \frac{f[x_1, x_1, x_1] - f[x_0, x_1, x_1]}{x_1 - x_0},$$

$$f[x_0, x_1, x_1, x_1] = \frac{f''(x_1)/2! - (f'(x_1) - f[x_0, x_1])/(x_1 - x_0)}{x_1 - x_0}.$$

$$f[x_0, x_0, x_1, x_1] = \frac{(x_1 - x_0) f''(x_1) - 2f'(x_1) + 2f[x_0, x_1]}{2(x_1 - x_0)^2}.$$

As $f''(1) = e^{-1} = 0.368$, $f[0, 1, 1, 1] = \frac{(1 - 0)(0.368) - 2(-1) + 2(-0.632)}{2(1 - 0)^2} = 1.104.$ •

Example 4.45 *Let $f(x) = \ln(x+2)$ and $x_0 = 0, x_1 = 0, x_2 = 1, x_3 = 1$, find the best approximation of $\ln(2.5)$ by using the Newton's polynomial.*

Solution. *Using $f(x) = \ln(x + 2)$ and $x_0 = 0, x_1 = 0, x_2 = 1, x_3 = 1$, the cubic Newton's interpolating polynomial has the following form*

$$p_3(x) = f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_0) f[x_0, x_1, x_2] + (x - x_0)(x - x_0)(x - x_1) f[x_0, x_1, x_2, x_3].$$

Now we find the second and third-order divided differences as follows:

$$f[0,0,1] = \frac{f[0,1]-f'(0)}{1-0} = f(1)-f(0)-f'(0) = 1.0986-0.6932-0.5 = -0.0946.$$

$$f[0,1,1] = \frac{f[1,1]-f[0,1]}{1-0} = f'(1)-f(1)+f(0) = 0.3333-1.0986+0.6932 = -0.0721,$$

$$f[0,0,1,1] = \frac{f[0,1,1]-f[0,0,1]}{1-0} = -0.0721+0.0946 = 0.0225.$$

$$p_3(0.5) = f(0)+(0.5-0)f'(0)+(0.5-0)(0.5-0)f[0,0,1]+(0.5-0)(0.5-0)(0.5-1)f[0,0,1,1],$$

$$\ln(2.5) \approx p_3(0.5) = \ln(2)+(0.5)(0.5)+(0.25)(-0.0946)+(-0.1250)(0.0225) = 0.9167,$$

the required approximation of $\ln(2.5)$ *and*

$$|f(0.5)-p_3(0.5)| = |\ln(2.5)-p_3(0.5)| = |0.9163-0.9167| = 4.0\times 10^4,$$

the possible absolute error in the approximation. •

Example 4.46 *Let* $f(x)=x^3$ *and* $\alpha \neq 1$*, then find the value of* α *such that* $f[\alpha,1,1]=1$.

Solution. *Using* $f(x)=x^3$ *and* $x_0=\alpha, x_1=1$*, we get* $f[\alpha,1,1]$ *as follows:*

$$f[x_0,x_1,x_1] = \frac{f[x_1,x_1]-f[x_0,x_1]}{x_1-x_0} = \frac{f'(x_1)-f[x_0,x_1]}{x_1-x_0},$$

$$1 = f[\alpha,1,1] = \frac{f'(1)-f[\alpha,1]}{1-\alpha}, \quad \textit{gives} \quad 1-\alpha = 3-f[\alpha,1].$$

$$2+\alpha = \frac{f[1]-f[\alpha]}{1-\alpha} = \frac{1-\alpha^3}{1-\alpha} = (\alpha^2+\alpha+1).$$

After simplifying, we get $\alpha^2=1$*, which means* $\alpha=-1$ *(because* $\alpha\neq 1$*) is the required value.* •

Example 4.47 *If* $f(x)=\dfrac{2}{x}$*, then show that the third divided difference* $f[1,1,1,2]=-1$.

Solution. *Since we know that*

$$\begin{aligned} f[1,1,1,2] &= \frac{f[1,1,2]-f[1,1,1]}{2-1} = f[1,1,2]-f[1,1,1] \\ &= \frac{f[1,2]-f[1,1]}{2-1} - \frac{f''(1)}{2!} \\ &= \frac{f(2)-f(1)}{2-1} - \frac{f'(1)}{1!} - \frac{f''(1)}{2!} \\ &= f(2)-f(1)-f'(1)-\frac{f''(1)}{2}. \end{aligned}$$

Since $f(x)=\dfrac{2}{x}$*, so we have,* $f'(x)=-\dfrac{2}{x^2}$ *and* $f''(x)=\dfrac{4}{x^3}$*. Thus*

$$f[1,1,1,2] = f(2)-f(1)-f'(1)-\frac{f''(1)}{2} = 1-2+2-2 = -1,$$

is the required value. •

A fourth way, that is also very convenient for interpolation is piecewise spline, is the linear spline interpolation which we will discuss finally.

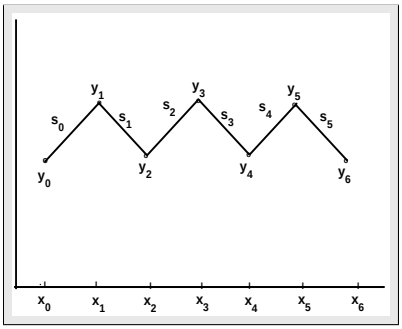


Figure 4.4: Linear spline.

4.3 Interpolation with Spline Functions

In the previous sections we studied the use of interpolation polynomials for approximating the values of the functions on closed intervals. An alternative approach is divide the interval into a collection of subintervals and construct a different approximating polynomial on each subinterval. Approximation by polynomial of this type is called *piecewise polynomial approximation.* Here, we will discuss some of the examples of a piecewise curve fitting techniques; the use of the *piecewise linear interpolation.*

Definition 4.1 (Spline Function)

Let $a = x_0 < x_1 < x_2 \cdots < x_n = b$. A function $s : [a, b] \to \mathbf{R}$ is a spline or spline function of degree m with points $x_0, x_1, \ldots, x_n$ if:
1. A function s is a piecewise polynomial such that, on each subinterval $[x_k, x_{k+1}]$, s has degree at most m.
2. A function s is $m-1$ times differentiable everywhere. •

A spline is a flexible drafting device that can be constrained to pass smoothly through a set of plotted data points. Spline functions are a mathematical tool which is an adaptation of this idea.

4.3.1 Piecewise Linear Interpolation

It is the one of the simplest piecewise polynomial interpolation for the approximation of the function, called *linear spline.* The linear spline is continuous function and the basic of it is simply connect consecutive points with straight lines. Consider the set of seven data points (x_0, y_0), (x_1, y_1), (x_2, y_2), $(x_3, y_3), (x_4, y_4), (x_5, y_5)$ and (x_6, y_6) which define six subintervals. These intervals are denoted as $[x_0, x_1], [x_1, x_2], [x_2, x_3]$, $[x_3, x_4], [x_4, x_5]$ and $[x_5, x_6]$, where $x_0, x_1, x_2, x_3, x_4, x_5$, and x_6 are distinct x-values. If we use a straight line on each subinterval (see Figure 4.4) then we can interpolate the data with a piecewise linear function, where

$$s_k(x) = p_k(x) = \frac{(x - x_{k+1})}{(x_k - x_{k+1})} y_k + \frac{(x - x_k)}{(x_{k+1} - x_k)} y_{k+1} = y_k + \frac{(y_{k+1} - y_k)}{(x_{k+1} - x_k)}(x - x_k).$$

$$s_k(x) = A_k + B_k(x - x_k), \quad \text{where} \qquad A_k = y_k \quad \text{and} \quad B_k = \frac{(y_{k+1} - y_k)}{(x_{k+1} - x_k)}. \tag{4.61}$$

Note that the linear spline must be continuous at given points $x_0, x_1, \ldots, x_n$ and

$$s(x_k) = f(x_k) = y_k, \qquad \text{for} \quad k = 0, 1, \ldots, n.$$

Example 4.48 *Find the values of unknown coefficients a and b so that the following function is a linear spline.*

$$s(x) = \begin{cases} a - x, & 0 \le x \le 1, \\ 3x - b, & 1 \le x \le 2, \\ 2x + 1, & 2 \le x \le 3. \end{cases}$$

Solution. *Since the given function is a linear spline, so s must be continuous at the internal points 1 and 2. Continuity at $x = 1$ implies that*

$$\begin{aligned} \lim_{x \to 1^-} s(x) &= \lim_{x \to 1^+} s(x), \\ \lim_{x \to 1^-} a - x &= \lim_{x \to 1^+} 3x - b, \\ a - 1 &= 3 - b, \end{aligned}$$

and it gives an equation of the from

$$a + b = 4.$$

Now continuity at $x = 2$ implies that

$$\begin{aligned} \lim_{x \to 2^-} s(x) &= \lim_{x \to 2^+} s(x), \\ \lim_{x \to 2^-} 3x - b &= \lim_{x \to 2^+} 2x + 1, \\ 6 - b &= 5, \end{aligned}$$

and it gives $b = 1$. Using this value of b, we get $a = 3$, and so

$$s(x) = \begin{cases} 3 - x, & 0 \le x \le 1, \\ 3x - 1, & 1 \le x \le 2, \\ 2x + 1, & 2 \le x \le 3, \end{cases}$$

is the linear spline function. •

Example 4.49 *Find the linear splines which interpolates the following data*

x_k	1	2	3	4
y_k	1.0	0.67	0.50	0.40

Find the approximation of the function $y(x) = \dfrac{2}{x+1}$ at $x = 2.9$. Compute absolute error.

Solution. *Given $x_0 = 1.0, x_1 = 2.0, x_2 = 3.0, x_3 = 4.0$, then using (**??**), we have*

$$A_0 = y_0 = 1.0, \ A_1 = y_1 = 0.67, \ A_2 = y_2 = 0.50, \ A_3 = y_3 = 0.4,$$

and

$$B_0 = \frac{(y_1 - y_0)}{(x_1 - x_0)} = \frac{(0.67 - 1.0)}{(2.0 - 1.0)} = -0.33,$$

$$B_1 = \frac{(y_2 - y_1)}{(x_2 - x_1)} = \frac{(0.50 - 0.67)}{(3.0 - 2.0)} = -0.17,$$

$$B_2 = \frac{(y_3 - y_2)}{(x_3 - x_2)} = \frac{(0.40 - 0.50)}{(4.0 - 3.0)} = -0.10.$$

Now using (4.61), the linear splines for three subintervals are define as

$$s(x) = \begin{cases} s_0(x) = 1.0 - 0.33(x - 1.0) = 1.33 - 0.33x, & 1 \le x \le 2, \\ s_1(x) = 0.67 - 0.17(x - 2.0) = 1.01 - 0.17x, & 2 \le x \le 3, \\ s_2(x) = 0.50 - 0.10(x - 3.0) = 0.80 - 0.10x, & 3 \le x \le 4. \end{cases}$$

The $x = 2.9$ lies in the interval $[2, 3]$, so, $f(2.9) \approx s_1(2.9) = 1.01 - 0.17(2.9) = 0.517$, and $|f(2.9) - s_1(2.9)| = |0.513 - 0.517| = 0.004$, is the required absolute error. •

Using MATLAB command window, we can reproduce above results as follows:

$>> X = [1\ 2\ 3\ 4];\ Y = [1\ 0.67\ 0.50\ 0.40];\ x = 2.9; s = LSpline(X, Y, x)$

Program 4.4
MATLAB m-file for the Linear Spline Functions
function LS=LSpline(X,Y,x)
n=length(X); for i=n-1:-1:1
$D = x - X(i)$; if $(D >= 0)$; break; end; end
$D = x - X(i)$; if $(D < 0); i = 0; D = x - X(1)$; end
$M = (Y(i + 1) - Y(i))/(X(i + 1) - X(i)); LS = Y(i) + M * D$; end

Example 4.50 *For the following table find the value of y at $x = 2.5$, using piecewise linear interpolation*

x	1	2	3	4
y	35	40	65	72

Solution. *The point $x = 2.5$ lies between 2 and 3. Then*
$y(2.5) = \frac{(25 - 3)}{(2 - 3)}40 + \frac{(2.5 - 2)}{(3 - 2)}65 = 52.5.$ •

4.4 Exercises

1. Use the Lagrange interpolation formula based on the points $x_0 = 0, x_1 = 1, x_2 = 2.5$ to find the equation of the quadratic polynomial to approximate $f(x) = \frac{2}{x + 2}$ at $x = 2.3$.

2. Let $f(x) = \cos(x\pi/4)$, where x is in radian. Use the quadratic Lagrange interpolation formula based on the points $x_0 = 0, x_1 = 1, x_2 = 2$ and $x_3 = 4$ to find the polynomial $p_2(x)$ to approximate the function $f(x)$ at $x = 0.5$ and $x = 3.5$.

3. Let $f(x) = x + 2\ln(x + 2)$. Use the quadratic Lagrange interpolation formula based on the points $x_0 = 0, x_1 = 1, x_2 = 2$ and $x_3 = 3$ to approximate $f(0.5)$ and $f(2.8)$. Also, compute the error bounds for your approximations.

4. Consider the function $f(x) = e^{x^2}$ and $x = 0, 0.25, 0.5, 1$. Then use the suitable Lagrange interpolating polynomial to approximate $f(0.75)$. Also, compute an error bound for your approximation.

5. Let $f(x) = x^4 - 2x + 1$. Use cubic Lagrange interpolation formula based on the points $x_0 = -1, x_1 = 0, x_2 = 2$ and $x_3 = 3$ to find the polynomial $p_3(x)$ to approximate the function $f(x)$ at $x = 1.1$. Also, compute an error bound for your approximation.

6. Construct the Lagrange interpolation polynomials for the following functions and compute the error bounds for the approximations:

   (a) $f(x) = x + 2^{x+1}, \quad x_0 = 0, x_1 = 1, x_2 = 2.5, x_3 = 3.$
   (b) $f(x) = 3x^3 + 2x^2 + 1, \quad x_0 = 1, x_1 = 2, x_2 = 3.$
   (c) $f(x) = \cos x - \sin x, \quad x_0 = 0, x_1 = 0.25, x_2 = 0.5, x_3 = 1.$

7. Consider the following table:

x	0	1	2	3
$f(x)$	2.0	3.72	8.39	21.06

   (a) Construct divided difference table for the tabulated function.
   (b) Compute the Newton interpolating polynomials $p_2(x)$ and $p_3(x)$ at $x = 2.2$.

8. Consider the following table:

x	1	2	3	4	5
$f(x)$	3.60	1.80	1.20	0.90	0.72

   (a) Construct divided difference table for the tabulated function.
   (b) Compute the Newton interpolating polynomials $p_3(x)$ and $p_4(x)$ at $x = 2.5, 3.5$.

9. Consider the following table of the $f(x) = \sqrt{x}$:

x	4	5	6	7	8
$f(x)$	2.0000	2.2361	2.4495	2.6458	2.8284

   (a) Construct the divided difference table for the tabulated function.
   (b) Find the Newton interpolating polynomials $p_3(x)$ and $p_4(x)$ at $x = 5.9$.
   (c) Compute error bounds for your approximations in part *(b)*.

10. Let $f(x) = e^x sinx$, with $x_0 = 0, x_1 = 2, x_2 = 2.5, x_3 = 4, x_4 = 4.5$. Then
    (a) Construct the divided-difference table for the given data points.
    (b) Find the Newton divided difference polynomials $p_2(x), p_3(x)$ and $p_4(x)$ at $x = 2.4$.
    (c) Compute error bounds for your approximations in part *(b)*.
    (d) Compute the actual error.

11. Show that if x_0, x_1 and x_2 are distinct then, $f[x_0, x_1, x_2] = f[x_1, x_2, x_0] = f[x_2, x_0, x_1]$.

12. The divided difference form of the interpolating polynomial $p_3(x)$ is

$$\begin{aligned} p_3(x) &= f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2, x_0] \\ &+ (x - x_0)(x - x_1)(x - x_2)f[x_0, x_1, x_2, x_3] \end{aligned}$$

    By expressing these divided differences in terms of the function values $f(x_i)(i = 0, 1, 2, 3)$, verify that $p_3(x)$ does pass through the points $(x_i, f(x_i))(i = 0, 1, 2, 3)$.

13. Let $f(x) = x^2 + e^x$ and $x_0 = 0, x_1 = 1$. Use the divided differences to find the value of the second divided difference $f[x_0, x_1, x_0]$.

14. Which of the following functions are linear splines ?

$$(a) \qquad s(x) = \begin{cases} x, & 0 \le x \le 1 \\ 2x - 1, & 1 \le x \le 2 \\ x + 2, & 2 \le x \le 4 \end{cases} \qquad (b) \qquad s(x) = \begin{cases} 2 - x, & 0 \le x \le 1 \\ 2x - 1, & 1 \le x \le 2 \\ x + 1, & 2 \le x \le 4 \end{cases}$$

15. Find the linear spline which interpolates the data:

$$(0, 3.5), (1, 3.9), (2, 4.7), (3, 5.8)$$

    What are its values at $x = 0.55, 1.15$ and 2.5 ?

16. Find the linear spline which interpolates the data:

$$(0, 0), (0.2, 0.18), (0.3, 0.26), (0.5, 0.41)$$

    What are its values at $x = 0.15, 0.25$, and 0.45 ?

17. Find the linear splines which interpolate the following data:

$$(0, 0), (1, 1), (16, 2), (81, 3)$$

    Compare interpolated values at $x = 0.5, 11.5$, and 30.5 to $f(x) = \sqrt[4]{x}$.

18. Find the linear splines which interpolate the following data:

$$(0, 1), (2, 0.9976), (3, 0.9945), (4, 0.9903)$$

    Compare interpolated values at $x = 1.5, 2.5$, and 3.5 to $f(x) = \cos(2x)$.

19. Find the linear splines which interpolate the following data:

$$(0, 1), (3, 2), (8, 3), (15, 4)$$

    Compare interpolated values at $x = 2.5, 5.5$, and 10.5 to $f(x) = \sqrt{x + 1}$.

Chapter 5

Numerical Differentiation and Integration

5.1 Introduction

In this chapter we deal with techniques for approximating numerically the two fundamental operations of the calculus, differentiation and integration. Both of these problems may be approached in the same way. Although both numerical differentiation and numerical integration formulas will be discussed, it should be noted that numerical differentiation is inherently much less accurate than numerical integration, and its application is generally avoided whenever possible. Nevertheless, it has been used successfully in certain applications.

Engineers are frequently confronted with the problem of differentiating functions which are defined in tabular or graphical form rather than as explicit functions. The interpretation of experimentally obtained data is a good example of this. A similar situation involves the integration of functions which have explicit forms that are difficult or impossible to integrate in terms of elementary functions. Graphical techniques, employing the construction of tangents to curves and the estimation of areas under curves, are commonly used in solving such problems, when great accuracy is not a prerequisite for the results. However, there are occasions when a higher degree of accuracy is desired, and, for these, various numerical methods are available.

5.2 Numerical Differentiation

Firstly, we discuss the numerical process for approximating the derivative of the function $f(x)$ at the given point. A function $f(x)$, known either explicitly or as a set of data points, is replaced by a simpler function. A polynomial $p(x)$ is the obvious choice of approximating function, since the operation of differentiation is then easily performed. The polynomial $p(x)$ is differentiated to obtain $p'(x)$, which is taken as an approximation to $f'(x)$ for any numerical value of x. Geometrically, this is equivalent to replacing the slope of $f(x)$, at x, by that of $p(x)$. Here, numerical differentiation are derived by differentiating interpolating polynomials.

We now turn our attention to the numerical process for approximating the derivative of a function

$f(x)$ at x, that is

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}, \quad \text{provided the limit exits.} \tag{5.1}$$

In principle, it is always possible to determine an analytic form (5.1) of a derivative for a given function. In some cases, however, the analytic form is very complicated, and a numerical approximation of the derivative may be sufficient for our purpose.
The formula (5.1) provides an obvious way to get an approximation to $f'(x)$; simply compute

$$D_h f(x) = \frac{f(x+h) - f(x)}{h}, \tag{5.2}$$

for small values of stepsize h, called numerical differentiation formula for (5.1).
Numerical differentiation is a much less satisfactory process because the seemingly obvious approximations are not always a good as they seem. Therefore, this process should, for this reason, be avoided if at all possible. We study it mainly as a means to end of solving differential equations by various numerical methods, based on the approximations we shall obtain for the derivatives of a function. We study it also, because it often happens that the thing we want to differentiate is not known function. We may, for instance, be given a table of speeds of a body observed at certain times, and wish to estimate its acceleration at these time.
Numerical differentiation is useful in estimating the derivative of a function when either function $f(x)$ is difficult to differentiate easily, or it is not known as explicit expression in x but the values of the function are described only in terms of tabulated data. Generally, it is considered that numerical differentiation is basically an *unstable process* which means that small errors made in the initial computations may cause greatly magnified errors in the final result. In fact, we may not always expect reasonable results even when the original data are known to be more accurate.
Here, we shall derive some formulas for estimating derivatives but we should avoid as far as possible, numerically calculating derivatives higher than the first, as the error in their evaluation increases with their orders. In spite of some inherent shortcomings, numerical differentiation is important to derive formulas for solving integrals and the numerical solution of both ordinary and partial differential equations.
There are three different approaches for deriving the numerical differentiation formulas. The first approach is based on the Taylor expansion of a function about a point, the second is to use difference operators, and the third approach to numerical differentiation is to fit a curve with a simple form to a function, and then to differentiate the curve-fit function. For example, the polynomial interpolation or spline methods of the Chapter 4 can be used to fit a curve to tabulated data for a function and the resulting polynomial or spline can then be differentiated. When a function is represented by a table of values, the most obvious approach is to differentiate the Lagrange interpolation formula

$$f(x) = p_n(x) + \frac{f^{(n+1)}(\eta(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i), \tag{5.3}$$

where the first term $p_n(x)$ of the right hand side is the Lagrange interpolating polynomial of degree n and the second term is its error term.
It is interesting to note that the process of numerical differentiation may be less satisfactory than interpolation the closeness of the ordinates of $f(x)$ and $p_n(x)$ on the interval of interest does not

guarantee the closeness of their respective derivatives. Note that the derivation and analysis of formulas for numerical differentiation is considerably simplifies when the data is equally spaced. It will be assumed, therefore, that the points x_i are given by $x_i = x_0 + ih, (i = 0, 1, \ldots, n)$ for some fixed tabular interval h.

5.3 Numerical Differentiation Formulas

Here, we will find the approximation of first and second derivative of a function at a given arbitrary point x. For the approximation of the first derivative of a given function we will discuss two-point formulas and three-point formulas. While for the second derivative approximation we will discuss three-point formula only.

5.3.0.1 Differentiation of the Lagrange Polynomial

Formulas for equally spaced data points that lie to right (or left) and center of given point x are called forward (or backward) and central difference formulas. These formulas can be derived by differentiation of the Lagrange interpolation polynomial. Some of the common forward-(or backward) and central-difference formulas are discuss below.

5.3.1 First Derivative Numerical Formulas

To obtain general formula for approximation of the first derivative of a function $f(x)$, we consider that $\{x_0, x_1, \ldots, x_n\}$ are $(n+1)$ distinct equally spaced points in some interval I and function $f(x)$ is continuous and its $(n+1)th$ derivatives exist in the given interval, that is, $f \in C^{n+1}(I)$. Then by differentiating (5.3) with respect to x and at $x = x_k$, we have

$$f'(x_k) = \sum_{i=0}^{n} f(x_i)L_i'(x_k) + \frac{f^{(n+1)}(\eta(x_k))}{(n+1)!} \prod_{\substack{i=0 \\ i\neq k}}^{n} (x_k - x_i). \tag{5.4}$$

The formula (5.4) is called the *(n+1)-point formula* to approximate $f'(x_k)$ with its error term. From this formula we can obtain many numerical differentiation formulas but here we shall discuss only two formulas to approximate (5.1) at given point $x = x_k$. First one is called the *two-point formula* and its error term which we can get from (5.4) by taking $n = 1$ and $k = 0$. The second numerical differentiation formula is called the *three-point formula* and its error term which can be obtained from (5.4) when $n = 2$ and $k = 0, 1, 2$.

5.3.1.1 Two-point Formula

Consider two distinct points x_0 and x_1, then, to find the approximation of (5.1), the first derivative of a function at given point, take $x_0 \in (a, b)$, where $f \in C^2[a, b]$ and that $x_1 = x_0 + h$ for some $h \neq 0$ that is sufficiently small to ensure that $x_1 \in [a, b]$. Consider the linear Lagrange interpolating polynomial $p_1(x)$ which interpolate $f(x)$ at the given points is

$$f(x) = p_1(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right) f(x_1). \tag{5.5}$$

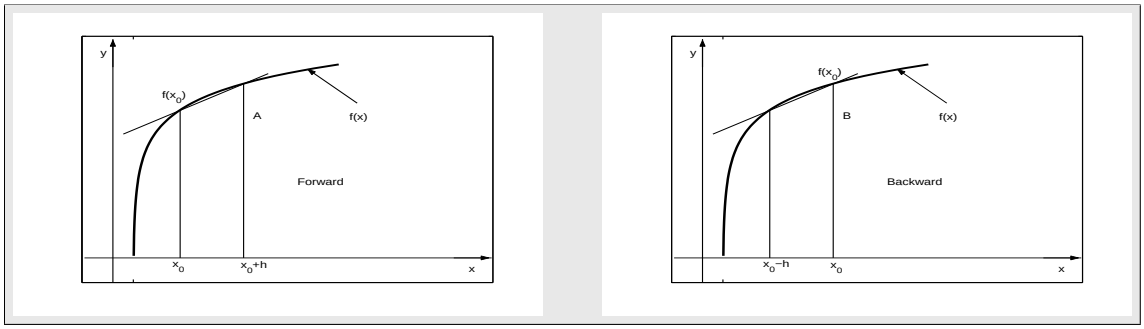
By taking derivative of (5.5) with respect to x and at $x = x_0$, we obtain

$$f'(x)|_{x=x_0} \approx p_1'(x)|_{x=x_0} = -\frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}, \quad f'(x_0) \approx -\frac{f(x_0)}{h} + \frac{f(x_0 + h)}{h},$$

which can be written as

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} = D_h f(x_0). \tag{5.6}$$

It is called the *two-point formula* for smaller values of h. For $h > 0$, sometime the formula (5.6) is also called the *two-point forward-difference formula* because it involves only differences of a function values forward from $f(x_0)$. The two-point forward-difference formula has a simple geometric interpretation as the slope of the forward secant line, as shown in Figure 5.1(a).



(a) Forward-difference approximations. (b) Backward-difference approximations.

Figure 5.1: Two-Point approximation formula.

If $h < 0$, then the formula (5.6) is also called the *two-point backward-difference formula*, which can be written as

$$f'(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h}. \tag{5.7}$$

In this case, a value of x behind the point of interest is used. The formula (5.7) is useful in cases where the independent variable represents time. If x_0 denotes the present time, the backward-difference formula uses only present and past samples, it does not rely on future data samples that may not yet be available in a real time application.
The geometric interpretation of the two-point backward-difference formula, as the slope of the backward secant line, is shown in Figure 5.1(b).

5.3.1.2 Error Term and Error Bound of Two-point Formula

The formula (5.6) is not very useful, therefore, let us attempt to find the error involves in our first numerical differentiation formula (5.6). Consider the error term for the linear Lagrange polynomial which can be written as

$$f(x) - p_1(x) = \frac{f''(\eta(x))}{2!} \prod_{i=0}^{1} (x - x_i),$$

for some unknown point $\eta(x) \in (x_0, x_1)$. By taking derivative of above equation with respect to x and at $x = x_0$, we have

$$f'(x_0) - p_1'(x_0) = \left(\frac{d}{dx} f''(\eta(x))\Big|_{x=x_0}\right) \frac{(x-x_0)(x-x_1)}{2} + \frac{f''(\eta(x_0))}{2} \left(\frac{d}{dx}(x^2 - x(x_0+h) - xx_0 + x_0(x_0+h))\Big|_{x=x_0}\right).$$

Since $\frac{d}{dx} f''(\eta(x)) = 0$ only if $x = x_0$, so error in the forward-difference formula (5.6) is

$$E_F(f,h) = f'(x_0) - D_h f(x_0) = -\frac{h}{2} f''(\eta(x)), \quad \text{where} \quad \eta(x) \in (x_0, x_0+h), \tag{5.8}$$

which is called the *error formula* of the two-point formula (5.6). Hence the formula (5.6) can be written as

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \frac{h}{2} f''(\eta(x)), \quad \text{where} \quad \eta \in (x_0, x_0+h). \tag{5.9}$$

The error bound formula of forward difference two-point formula can be written as

$$|E_F(f,h)| = |f'(x_0) - D_h f(x_0)| = \frac{h}{2} M, \tag{5.10}$$

where $|f''(\eta(x))| leM = \max_{x_0 \le x \le x_1} |f''(x)|$.
Similarly, the formula (5.7) can be written as

$$f'(x_0) = \frac{f(x_0) - f(x_0-h)}{h} + \frac{h}{2} f''(\eta(x)), \quad \text{where} \quad \eta \in (x_0-h, x_0). \tag{5.11}$$

The formula (5.9) is more useful than the formula (5.6) because now on a large class of function, an error term is available along with the basic numerical formula. Note that the error term in (5.9) has two parts; a power of h and a factor involving some higher-order derivative of $f(x)$ which gives us an indication of the class of function to which the error estimate is applicable. The h term in the error makes the entire expression converge to zero as h approaches zero. The rapidity of this convergence will depend on the power of h. These remarks apply to many error estimates in numerical analysis. There will usually be a power of h and a factor telling us to what smoothness class of the function must belong so that the estimate is valid.
Note that the formula (5.9) may also be derived from the Taylor's theorem. Expansion of function $f(x_1)$ about x_0 as far as term involving h^2 gives

$$f(x_1) = f(x_0) + hf'(x_0) + \frac{h^2}{2!} f''(\eta(x)). \tag{5.12}$$

From this the result follows by subtracting $f(x_0)$ both sides and dividing both sides by h and put $x_1 = x_0 + h$.
Note that for a linear function, $f(x) = ax + b$, the approximate formula (5.6) is exact; that is, it yields the correct value of first derivative of the function $f(x)$ for any nonzero value of h.

Example 5.1 *Let $f(x) = x^3$ be defined in the interval $[0.2, 0.3]$. Use the error formula (5.8) of two-point formula for the approximation of $f'(0.2)$ to compute a value of unknown point η.*

Solution. *Since the exact value of the first derivative of the function at $x_0 = 0.2$ is*

$$f'(x) = 3x^2 \quad \text{and} \quad f'(0.2) = 3(0.2)^2 = 0.12,$$

and the approximate value of $f'(0.2)$ using two point formula is

$$f'(0.2) \approx \frac{f(0.3) - f(0.2)}{0.1} = \frac{(0.3)^3 - (0.2)^3}{0.1} = 0.19,$$

so error E can be calculated as

$$E = 0.12 - 0.19 = -0.07.$$

Using the formula (5.6) and $f''(\eta) = 6\eta$, we have, $-0.07 = -\frac{0.1}{2}6\eta$, and solving for η, we get the value of the unknown point $\eta = 0.233$ in the interval $(0.2, 0.3)$. •

Example 5.2 *Let $f(x) = x\cos x - x^2 \sin x$ and $h = 0.1, h = 0.01$. Use two-point forward difference formula to approximate $f'(1)$. Compute bound for absolute error for each case.*

Solution. *Using the forward difference formula (5.6) with $h = 0.1, h = 0.01$ and $x_0 = 1$, we have*

$$f'(1) \approx \frac{[1.1\cos 1.1 - (1.1)^2 \sin 1.1] - [1\cos 1 - (1)^2 \sin 1]}{0.1} = \frac{-0.5794 + 0.3012}{0.1} = -2.7824.$$

$$f'(1) \approx \frac{[1.01\cos 1.01 - (1.01)^2 \sin 1.01] - [1\cos 1 - (1)^2 \sin 1]}{0.01} = \frac{-0.3267 + 0.3012}{0.01} = -2.5505.$$

Since the exact value of $f'(x) = (1-x^2)\cos x - 3x\sin x$ at $x = 1$ is, -2.5244, so the corresponding actual errors with $h = 0.1$ and $h = 0.01$ are, 0.2580 and 0.0261 respectively. This shows that the approximation obtained with $h = 0.01$ is better than the approximation with $h = 0.1$.
To find the error bound, we use the formula (5.8) as follows:

$$|E_F(f,h)| = \Big| -\frac{h}{2}\Big| |f''(\eta(x))|, \quad \text{for} \quad \eta(x) \in (x_0, x_0 + h).$$

The second derivative $f''(x)$ of the function $f(x) = x\cos x - x^2 \sin x$ can be found as

$$f'(x) = (1 - x^2)\cos x - 3x\sin x \qquad \text{and} \qquad f''(x) = -5x\cos x - (4 - x^2)\sin x.$$

So bound $|f''|$ on $[1, 1.1]$ can be obtain

$$|f''(\eta(x))| \le M = \max_{1 \le x \le 1.1} | -5x\cos x - (4 - x^2)\sin x| = 5.2259,$$

at $x = 1$, and bound $|f''|$ on $[1, 1.01]$ can be obtain

$$|f''(\eta(x))| \le M = \max_{1 \le x \le 1.01} | -5x\cos x - (4 - x^2)\sin x| = 5.2259,$$

at $x = 1$. Therefore, for $h = 0.1$ and $h = 0.01$, we have

$$|E_F(f,h)| \leq \frac{0.1}{2}M = 0.05(5.2259) = 0.2613,$$

$$|E_F(f,h)| \leq \frac{0.01}{2}M = 0.005(5.2259) = 0.02613,$$

the possible maximum errors in our approximations. •

The above results can be easily achieved with MATLAB commands as follows:

```
>> x0 = 1.0; h = 10.^ -(1 : 2);
>> df = ((x0 + h) * cos(x0 + h) - (x0 + h)^ 2  sin(x0 + h) - x0 * cos(x0) - (x0)^ 2  sin(x0))./h
```

Using MATLAB symbolic toolbox derivative of the function at $x = 1$ can be get as follows:

```
>> syms x; f = x * cos(x) - (x)^2  *sin(x); df = diff(f, 1); subs(df, 1)
```

Using the backward difference formula (5.6) with $h = 0.1, h = 0.01$ and $x_0 = 1$, we have

$$f'(1) \approx \frac{[1\cos 1 - (1)^2 \sin 1] - [0.9\cos 0.9 - (0.9)^2 \sin 0.9]}{0.1} = \frac{-0.3012 + 0.0751}{0.1} = -2.2612.$$

$$f'(1) \approx \frac{[1\cos 1 - (1)^2 \sin 1] - [0.99\cos 0.99 - (0.99)^2 \sin 0.99]}{0.01} = \frac{-0.3012 + 0.2762}{0.01} = -2.4983.$$

The corresponding actual errors with $h = 0.1$ and $h = 0.01$ are, -0.2632 and -0.0261 respectively. Note that the both errors for $h = 0.1$ and $h = 0.01$ by using the forward-difference'formula is better than the backward-difference formula for the same values of h.

Example 5.3 *Let $f(x) = x^2 \cos x$ and $h = 0.1$. Then*
(a) Compute the approximate value of $f'(1)$ using forward difference two-point formula (5.6).
(b) Compute the error bound for your approximation using the formula (5.8).
(c) Compute the absolute error.
(d) What best maximum value of stepsize h required to obtain the approximate value of $f'(1)$ correct to 10^{-2}.

Solution. *(a) Given $x_0 = 1, h = 0.1$, then by using the formula (5.6), we have*

$$f'(1) \approx \frac{f(1+0.1) - f(1)}{0.1} = \frac{f(1.1) - f(1)}{0.1} = D_h f(1).$$

Thus

$$f'(1) \approx \frac{(1.1)^2\cos(1.1) - (1)^2\cos(1)}{0.1} \approx \frac{0.5489 - 0.5403}{0.1} = 0.0860,$$

which is the required approximation of $f'(x)$ at $x = 1$.

(b) To find the error bound, we use the formula (5.8), which gives

$$E_F(f,h) = -\frac{0.1}{2} f''(\eta(x)), \quad \textit{where} \quad \eta(x) \in (1, 1.1),$$

or

$$|E_F(f,h)| = \Big| -\frac{0.1}{2}\Big| |f''(\eta(x))|, \quad \textit{for} \quad \eta \in (1, 1.1).$$

The second derivative $f''(x)$ of the function can be found as

$$f(x) = x^2 \cos x, \quad f'(x) = 2x\cos x - x^2 \sin x, \quad \textit{and} \quad f''(x) = (2 - x^2)\cos x - 4x \sin x.$$

The value of the second derivative $f''(\eta(x))$ cannot be computed exactly because $\eta(x)$ is not known. But one can bound the error by computing the largest possible value for $|f''(\eta(x))|$. So bound $|f''|$ on $[1, 1.1]$ can be obtain

$$M = \max_{1 \le x \le 1.1} |(2 - x^2)\cos x - 4x \sin x| = 3.5630,$$

at $x = 1.1$. Since $|f''(\eta(x))| \le M$, therefore, for $h = 0.1$, we have

$$|E_F(f,h)| \le \frac{0.1}{2} M = 0.05(3.5630) = 0.1782,$$

which is the possible maximum error in our approximation.
(c) Since the exact value of the derivative $f'(1)$ is $2(1)cos(1) - (1)^2 \sin(1) = 0.2391$, therefore the absolute error $|E|$ can be computed as follows:

$$|E| = |f'(1) - D_h f(1)| = |0.2391 - 0.0860| = 0.1531.$$

(d) Since the given accuracy required is 10^{-2}, so

$$|E_F(f,h)| = | -\frac{h}{2} f''(\eta(x))| \le 10^{-2},$$

for $\eta(x) \in (1, 1.1)$. This gives

$$\frac{h}{2} M \le 10^{-2}, \qquad or \qquad h \le \frac{(2 \times 10^{-2})}{3.5630} \le \frac{2}{356.3000} = 0.0056,$$

which is the best maximum value of h to get the required accuracy. •

The truncation error in the approximation of (5.9) is roughly proportional to stepsize h used in its computation. The situation is made worse by the fact that the round-off error in computing the approximate derivative (5.6) is roughly proportion to $\frac{1}{h}$. The overall error therefore is of the form

$$E = ch + \frac{\delta}{h},$$

where c and δ are constants. This places serve restriction on the accuracy that can be achieved with this formula.

Now we discuss little more about the role of the round-off error in the numerical differentiation. Consider the formula (5.6) which is

$$f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h} = D_h f(x_0).$$

If h is small, then we can reasonably assume that $f(x_0)$ and $f(x_0+h)$ have similar magnitude and, therefore, similar round-off errors. Let the actual function values used in the computation be denoted by $\tilde{f}_0$ and $\tilde{f}_1$ with

$$f(x_i) - \tilde{f}_i = \epsilon_i, \qquad \text{for} \quad i = 0, 1,$$

the errors in the function values. Thus the actual quantity calculated is

$$\tilde{D}_h f(x_0) = \frac{\tilde{f}_1 - \tilde{f}_0}{h}.$$

For the error in this quantity, replace $\tilde{f}_i$ by $f(x_i) - \epsilon_i$, for $i = 0, 1$, we obtain

$$\begin{aligned} f'(x_0) - \tilde{D}_h f(x_0) &= f'(x_0) - \frac{(f(x_1)-\epsilon_1) - (f(x_0)-\epsilon_0)}{h} \\ &= f'(x_0) - \frac{f(x_1)-f(x_0)}{h} + \frac{\epsilon_1 - \epsilon_0}{h}. \end{aligned}$$

Then the overall error is given by

$$|f'(x_0) - \tilde{D}_h f(x_0)| \le \left| f'(x_0) - \frac{f(x_1)-f(x_0)}{h} \right| + \left| \frac{\epsilon_1 - \epsilon_0}{h} \right| \le \left| \frac{h}{2} f''(\eta(x_0)) \right| + \frac{2\delta}{h},$$

the errors ϵ_0, ϵ_1 are generally random in some interval $[-\delta, \delta]$.
It is the second term on the right side of this error bound which leads to the growth of error as $h \to 0$. If

$$|f(x_i) - \tilde{f}_i| \le \frac{1}{2} \times 10^{-t} = \delta,$$

and t is the required decimal digits of accuracy, then the maximum rounding error in the two-point formula is $\dfrac{10^{-t}}{h}$. While the truncation error $\left| \dfrac{h}{2} f''(\eta(x_0)) \right|$ decreases with h, the rounding error increases. The total error, $E(h)$, therefore has a minimum with respect to h. If

$$E(h) = E_{trunc} + E_{round} = \frac{h}{2} M + \frac{10^{-t}}{h},$$

where $M = \max\limits_{x_0 \le x \le x_1} |f''(\eta(x_0))|$, then

$$\frac{dE}{dh} = \frac{M}{2} - \frac{10^{-t}}{h^2}.$$

A minimum of $E(h)$ satisfies the equation $\dfrac{dE}{dh} = 0$, that is

$$\frac{dE}{dh} = \frac{M}{2} - \frac{10^{-t}}{h^2} = 0, \quad \text{gives} \quad h = h_{opt} = \sqrt{\frac{2}{M} \times 10^{-t}},$$

which gives the optimal value for h. Thus the minimum error is

$$E(h_{opt}) = \frac{M}{2}\sqrt{\frac{2}{M}\times 10^{-t}} + \frac{10^{-t}}{\sqrt{\frac{2}{M}\times 10^{-t}}} = \sqrt{2M\times 10^{-t}}.$$

Example 5.4 *Consider $f(x) = x^2\cos x$ and $x_0 = 1$. To show the effect of rounding error, the values $\tilde{f}_i$ are obtained by rounding $f(x_i)$ to seven significant digits, compute the total error for $h = 0.1$ and also, find the optimum h.*

Solution. *Given $|\epsilon_i| \le \frac{1}{2}\times 10^{-7} = \delta$ and $h = 0.1$. Now to calculate the total error, we use*

$$E(h) = \frac{h}{2}M + \frac{10^{-t}}{h}, \quad M = \max_{1\le x\le 1.1}|(2-x^2)\cos x - 4x\sin x| = 3.5630.$$

Then

$$E(h) = \frac{0.1}{2}(3.5630) + \frac{10^{-7}}{0.1} = 0.17815 + 0.000001 = 0.178151.$$

Now to find the optimum h, we use

$$h = h_{opt} = \sqrt{\frac{2}{M}\times 10^{-t}} = \sqrt{\frac{2}{3.5630}\times 10^{-7}} = 0.00024,$$

which is the smallest value of h, below which the total error will begin to increase.

Note that for

$$\begin{aligned} h &= 0.00024, \quad & E(h) &= 0.000844,\\ h &= 0.00015, \quad & E(h) &= 0.000934,\\ h &= 0.00001, \quad & E(h) &= 0.010018. \end{aligned}$$

A similar effect is present for all numerical differentiation formulas.
The numerical differentiation formulas are often judge by the power of h in the error term. Since h is always small, so the higher power of h involve will give better approximation. In this assessment, the formula (5.9) acts poorly, as the error term involved h of power one. A superior formulas can be obtained by deriving some useful three-point formulas together with error terms which involved h^2 using the formula (5.4) by taking $n = 2$. •

5.3.1.3 Three-point Central Difference Formula

Consider the quadratic Lagrange interpolating polynomial $p_2(x)$ to the three distinct equally spaced points x_0, x_1, and x_2, with $x_1 = x_0 + h$ and $x_2 = x_0 + 2h$, for smaller value h, we have

$$f(x) = p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2).$$

Now taking the derivative of the above expression with respect to x and then take $x = x_k$, for $k = 0, 1, 2$, we have

$$f'(x_k) \approx \frac{(2x_k-x_1-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(2x_k-x_0-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(2x_k-x_0-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2). \quad (5.13)$$

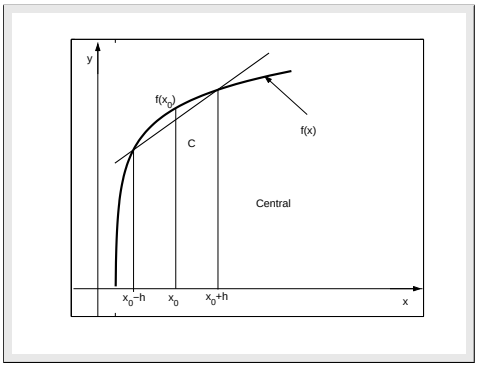


Figure 5.2: Central-difference approximations.

Three different numerical differentiation formulas can be obtained from (5.13) by putting $x_k = x_0$, or $x_k = x_1$ or $x_k = x_2$, which are use to find the approximation of the first derivative of a function defined by the formula (5.1) at the given point. Firstly, we take $x_k = x_1$, then the formula (5.13) becomes

$$f'(x_1) \approx \frac{(2x_1 - x_1 - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(2x_1 - x_0 - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(2x_1 - x_0 - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2).$$

After, simplifying, and replacing $x_0 = x_1 - h, x_2 = x_1 + h$, we obtain

$$f'(x_1) \approx \frac{f(x_1 + h) - f(x_1 - h)}{2h} = D_h f(x_1). \tag{5.14}$$

It is called the *three-point central-difference formula* (second-order) for finding the approximation of the first derivative of a function at the given point x_1.
Note that the formulation of the formula (5.14) uses data points that are centered about the point of interest x_1 even though it does not appear in the right side of (5.14).
The geometric interpretation of the central-difference formula is shown in Figure 5.2.

5.3.1.4 Error Formula and Error Bound Formula of Central Difference Formula

The formula (5.14) is not very useful, therefore, let us attempt to find the error involve in the formula (5.14) for numerical differentiation. Consider the error term for the quadratic Lagrange polynomial which can be written as

$$f(x) - p_2(x) = \frac{f'''(\eta(x))}{3!} \prod_{i=0}^{2} (x - x_i),$$

for some unknown point $\eta(x) \in (x_0, x_2)$. By taking derivative of the above equation with respect to x and then taking $x = x_1$, we have

$$f'(x_1) - p_2'(x_1) = \left(\frac{d}{dx} f'''(\eta(x)) \Big|_{x=x_1} \right) \frac{(x - x_0)(x - x_1)(x - x_2)}{6}$$

$$+ \quad \frac{f'''(\eta(x_1))}{6} \left((x - x_1)(x - x_2) + (x - x_0)(x - x_2) + (x - x_0)(x - x_1) \Big|_{x=x_1} \right).$$

Since $\frac{d}{dx}f'''(\eta(x)) = 0$ only if $x = x_1$, therefore the error formula of the central-difference formula (5.14) can be written as

$$E_C(f,h) = f'(x_1) - D_h f(x_1) = -\frac{h^2}{6}f'''(\eta(x_1)), \tag{5.15}$$

where $\eta(x_1) \in (x_1 - h, x_1 + h)$. Hence the formula (5.14) can be written as

$$f'(x_1) = \frac{f(x_1+h) - f(x_1-h)}{2h} - \frac{h^2}{6}f'''(\eta(x_1)), \tag{5.16}$$

where $\eta(x_1) \in (x_1 - h, x_1 + h)$. The formula (5.16) is more useful than the formula (5.14) because now on a large class of function, an error term is available along with the basic numerical formula. The error bound formula of central difference three-point formula can be written as

$$|E_C(f,h)| = |f'(x_1) - D_h f(x_1)| \leq \frac{h^2}{6}M, \tag{5.17}$$

where

$$|f'''(\eta(x_1))| \leq M = \max_{x_0 \leq x \leq x_2} |f'''(x)|.$$

Note that for a quadratic function, $f(x) = ax^2 + bx + c$, the approximate formula (5.14) is exact; that is, it yields the correct value of first derivative of a function $f(x)$ for any nonzero value of h.

5.3.1.5 Three-point Forward and Backward Difference Formulas with Error Formulas

Similarly, the two other three-point formulas can be obtained by taking $x_k = x_0$ and $x_k = x_2$ in the formula (5.13). Firstly, by taking $x_k = x_0$ in the formula (5.13) and then after simplifying, we have

$$f'(x_0) \approx \frac{-3f(x_0) + 4f(x_0+h) - f(x_0+2h)}{2h} = D_h f(x_0), \tag{5.18}$$

which is called the *three-point forward-difference formula* (second-order) which is use to approximate the formula (5.1) at given point $x = x_0$. The error term of this approximation formula can be obtain in the similar way as we obtained for the central-difference formula and it is

$$E_F(f,h) = \frac{h^2}{3}f'''(\eta(x_0)), \tag{5.19}$$

where $\eta(x_0) \in (x_0, x_0 + 2h)$. The error bound formula of forward difference three-point formula can be written as

$$|E_F(f,h)| = |f'(x_0) - D_h f(x_0)| \leq \frac{h^2}{3}M, \tag{5.20}$$

where $|f'''(\eta(x_0))| \leq M = \max_{x_0 \leq x \leq x_2} |f'''(x)|$.
Similarly, taking $x_k = x_2$ in the formula (5.13), and after simplifying, we obtain

$$f'(x_2) \approx \frac{f(x_2-2h) - 4f(x_2-h) + 3f(x_2)}{2h} = D_h f(x_2), \tag{5.21}$$

which is called the *three-point backward-difference formula* (second-order) which is use to approximate the formula (5.1) at given point $x = x_2$. It has the error term of the form

$$E_B(f,h) = \frac{h^2}{3} f'''(\eta(x_2)), \quad \eta(x_2) \in (x_2 - 2h, x_2). \tag{5.22}$$

The error bound formula of backward difference three-point formula can be written as

$$|E_B(f,h)| = |f'(x_2) - D_h f(x_2)| \leq \frac{h^2}{3} M, \tag{5.23}$$

where

$$|f'''(\eta(x_2))| \leq M = \max_{x_0 \leq x \leq x_2} |f'''(x)|.$$

Note that the backward-difference formula (5.21) can be obtained from the forward-difference formula by replacing h with $-h$. Also, note that the error in (5.14) is approximately half the error in (5.18) and (5.21). This is reasonable since in using the central-difference formula (5.14) data is being examined on both sides of point x_1, and for others in (5.18) and (5.21) only on one side. Note that in using the central-difference formula, a function $f(x)$ needs to be evaluated at only two points, whereas in using the other two formulas, we need the values of a function at three points. The approximations in using the formulas (5.18) and (5.21) are useful near the ends of the required interval, since the information about a function outside the interval may not be available. Thus the central-difference formula (5.14) is superior to both the forward-difference formula (5.18) and the backward-difference formula (5.21). The central-difference represents the average of the forward-difference and the backward-difference.

Example 5.5 *Let $f(x) = x^2 + \cos x$ and $h = 0.1$. Then*
(a) Compute the approximate value of $f'(1)$ by using three-point central difference formula (5.14).
(b) Compute the error bound for your approximation using (5.15).
(c) Compute the absolute error.
(d) What is the best maximum value of stepsize h required to obtain the approximate value of $f'(1)$ correct to 10^{-2}.

Solution. *(a) Given $x_1 = 1, h = 0.1$, then using the formula (5.14), we have*

$$f'(1) \approx \frac{f(1+0.1) - f(1-0.1)}{2(0.1)} = \frac{f(1.1) - f(0.9)}{0.2} = D_h f(1).$$

Then

$$f'(1) \approx \frac{[(1.1)^2 + \cos(1.1)] - [(0.9)^2 + \cos(0.9)]}{0.2} \approx \frac{1.6636 - 1.4316}{0.2} = 1.1600.$$

(b) By using the error formula (5.15), we have

$$|E_C(f,h)| = \Big| - \frac{(0.1)^2}{6} \Big| |f'''(\eta(x_1))|, \quad \textit{for} \quad \eta(x_1) \in (0.9, 1.1).$$

Since $f'''(\eta(x_1)) = \sin(\eta(x_1))$. The above formula cannot be computed exactly because $\eta(x_1)$ is not known. But one can bound the error by computing the largest possible value for $|f'''(\eta(x_1))|$. So bound $|f'''|$ on $[0.9, 1.1]$ is

$$M = \max_{0.9 \leq x \leq 1.1} |\sin x| = 0.8912,$$

at $x = 1.1$. *Thus, for* $|f'''(\eta(x_1))| \leq M$ *and* $h = 0.1$, *gives*

$$|E_C(f,h)| \leq \frac{0.01}{6}M = \frac{0.01}{6}(0.8912) = 0.0015,$$

which is the possible maximum error in our approximation.

(c) Since the exact value of the derivative $f'(1)$ *is,* $(2(1) - \sin 1) = 1.1585$, *therefore, the absolute error* $|E|$ *can be computed as follows*

$$|E| = |f'(1) - D_h f(1)| = |(2 - \sin 1) - 1.1600| = |1.1585 - 1.1600| = 0.0015.$$

(d) Since the given accuracy required is 10^{-2}, *so*

$$|E_C(f,h)| = \Big| -\frac{h^2}{6}f'''(\eta(x_1))\Big| \leq 10^{-2},$$

for $\eta(x_1) \in (0.9, 1.1)$. *Then*

$$\frac{h^2}{6}M \leq 10^{-2}, \qquad h \leq \sqrt{\frac{6\times 10^{-2}}{M}}, \qquad h \leq \sqrt{\frac{6\times 10^{-2}}{0.8912}} = 0.2312,$$

the best maximum value of h. •

To get above results using the MATLAB commands, we do the following:

```
>> x0 = 1.0; h = 0.1;
>> df = (x0 + h).^ 2 + cos(x0 + h) - (x0 - h).^ 2 + cos(x0 - h)./(2. * h);
```

Note that the formula (5.16) may also be derived from the Taylor's theorem. The second degree Taylor's expansion $f(x)$ about x_1, for $f(x_1 + h)$ and $f(x_1 - h)$, gives

$$f(x_1 + h) = f(x_1) + hf'(x_1) + \frac{h^2}{2!}f''(x_1) + \frac{h^3}{3!}f'''(\eta_1(x)),$$

$$f(x_1 - h) = f(x_1) - hf'(x_1) + \frac{h^2}{2!}f''(x_1) - \frac{h^3}{3!}f'''(\eta_2(x)).$$

Subtracting above two equations, the results is

$$f(x_1 + h) - f(x_1 - h) = 2hf'(x_1) + \frac{h^3}{3!}\left[f'''(\eta_1(x)) + f'''(\eta_2(x))\right].$$

Since $f'''(x)$ is continuous, the intermediate value theorem can be used to find a value of $\eta(x)$ so

$$\frac{f'''(\eta_1(x)) + f'''(\eta_2(x))}{2} = f'''(\eta(x)),$$

which is the required formula (5.16). •

Example 5.6 *Consider the following table for set of data points*

x	1	1.6	2	2.3	2.8	3	3.9	4	4.8	5
$f(x)$	0.00	0.47	0.69	0.83	1.03	1.10	1.36	1.39	1.57	1.61

(a) Use three-point formula for smaller value of h to find approximation of $f'(3)$.
(b) The function tabulated is $\ln x$, find error bound and absolute error for the approximation of $f'(3)$.
(c) What is the best maximum value of stepsize h required to obtain the approximate value of $f'(3)$ within the accuracy 10^{-4}.

Solution. *(a) For the given table of data points, we can use all three-points formulas as for the central difference we can take*

$$x_0 = x_1 - h = 2, \quad x_1 = 3, \quad x_2 = x_1 + h = 4, \quad \textit{gives}, \quad h = 1,$$

for the forward difference formula we can take

$$x_0 = 3, \quad x_1 = x_0 + h = 3.9, \quad x_2 = x_0 + 2h = 4.8, \quad \textit{gives}, \quad h = 0.9,$$

and for the backward difference formula we can take

$$x_0 = x_2 - 2h = 1.6, \quad x_1 = x_2 - h = 2.3, \quad x_2 = 3, \quad \textit{gives}, \quad h = 0.7.$$

Since we know that smaller the vale of h better the approximation of the derivative of the function, therefore, for the given problem, backward difference is the best formula to find approximation of $f'(3)$ as

$$f'(3) \approx \frac{f(1.6) - 4f(2.3) + 3f(3)}{2(0.7)} = \frac{(0.47 - 4(0.83) + 3(1.10))}{1.4} = 0.3214.$$

(b) Using error term of backward difference formula, we have

$$E_B(f,h) = \frac{h^2}{3} f'''(\eta), \quad \textit{or} \quad |E_B(f,h)| \leq \frac{h^2}{3} |f'''(\eta)|.$$

Taking $|f'''(\eta(x_2))| \leq M = \max\limits_{1.6 \leq x \leq 3} |f'''(x)| = \max\limits_{1.6 \leq x \leq 3} |2/x^3| = 0.4883$. Thus using $h = 0.7$, we obtain

$$|E_B(f,h)| \leq \frac{(0.7)^2}{3}(0.4883) = 0.0798,$$

the required error bounds for the approximations. To compute the absolute error we do as

$$|E| = |f'(3) - 0.3214| = |0.3333 - 0.3214| = 0.0119.$$

(c) Since the given accuracy required is 10^{-4}, so $|E_B(f,h)| = \left|\frac{h^2}{3} f'''(\eta)\right| \leq 10^{-4}$, for $\eta \in (1.6, 3)$.

Then, $\frac{h^2}{3} M \leq 10^{-4}, \qquad h^2 \leq \frac{3 \times 10^{-4}}{M}, \qquad h \leq \sqrt{\frac{3 \times 10^{-4}}{0.4883}} = 0.025.$ •

Example 5.7 *Use the best three-point formula to find approximation of $f'(1.5)$ using the following table for set of data points*

x	1	1.6	2	2.3	2.8	3	3.9	4	4.8	5
$f(x)$	0.00	0.47	0.69	0.83	1.03	1.10	1.36	1.39	1.57	1.61

Solution. *The best three-point formula for this problem is the central difference because this formula does not need the value of $f(1.5)$ which is not given in the table while the other two formulas need this value. Since $x_1 = 1.5$, so by taking $x_1 + h = 2$ and $x_1 - h = 1$, gives $h = 0.5$, we have*

$$f'(1.5) \approx \frac{f(2) - f(1)}{2(0.5)} = \frac{0.69 - 0.0}{1} = 0.69,$$

the required approximation of $f'(1.5)$. •

Example 5.8 *Let $f(x) = xe^x - (x+5)\ln(x+5)$. Use the most accurate three-point formula to determine the missing entries in the following table.*

Table 5.1: Table with the missing entries.

Formula			
x	2.0	2.1	2.2
f(x)	1.1567	3.2323	5.6416
f'(x)			
Abs. Error			
Error Bound			

Solution. *Given $f(x) = xe^x - (x+5)\ln(x+5)$ and $h = 0.1$, then:*
Forward-difference formula:

$$f'(2) \approx \frac{-3f(2) + 4f(2.1) - f(2.2)}{2h} = \frac{-3(1.1567) + 4(3.2323) - 5.6416}{0.2} = 19.0875.$$

Central-difference formula:

$$f'(2.1) \approx \frac{f(2.2) - f(2)}{2h} = \frac{5.6416 - 1.1567}{0.2} = 22.4245.$$

Backward difference formula:

$$f'(2.2) \approx \frac{f(2) - 4f(2.1) + 3f(2.2)}{2h} = \frac{1.1567 - 4(3.2323) + 3(5.6416)}{0.2} = 25.7615.$$

Since $f'(x) = (x+1) * exp(x) - \ln(x+5) - 1$, therefore:

$$Abs.Error = |E_F| = |f'(2.0) - 19.0875| = |19.2213 - 19.0875| = 0.1338.$$

$$Abs.Error = |E_C| = |f'(2.1) - 22.4245| = |22.3550 - 22.4245| = 0.0695.$$

$$Abs.Error = |E_B| = |f'(2.2) - 25.7615| = |25.9060 - 25.7615| = 0.1445.$$

Since the third derivative of the given function is $(x+3)exp(x)+(1/(x+5)^2)$*, and* $|f'''(\eta(x_0))| = M = \max_{x_0\le x\le x_2}|f'''(x)| = 46.9494$ *at* $x = 2.2$*, so the possible error bounds are:*

$$|E_F(f,h)| \le \frac{h^2}{3}M = \frac{0.01}{3}46.9494 = 0.1565.$$

$$|E_C(f,h)| \le \frac{h^2}{6}M = \frac{0.01}{6}46.9494 = 0.0782.$$

$$|E_B(f,h)| \le \frac{h^2}{3}M = \frac{0.01}{3}46.9494 = 0.1565.$$

Thus we got the the missing entries of the given table. •

Table 5.2: Solution of the Example 5.8.

Formula	Forward	Central	Backward
x	2.0	2.1	2.2
f(x)	1.1567	3.2323	5.6416
f'(x)	19.0875	22.4245	25.7615
Abs. Error	0.1338	0.0695	0.1445
Error Bound	0.1565	0.0782	0.1565

Example 5.9 *Let* $f(x) = (x\ln x + x)$ *and* $x = 0.9, 1.3, 1.6, 2.1, 2.5, 3.1$*. Then*
(a) Find the approximate value of $f'(1.9)$ *using three-point formula for smaller value of* h*.*
(b) Compute the error bound for your approximation.
(c) Compute the absolute error.
(d) What is the best maximum value of stepsize h *required to obtain the approximate value of* $f'(1.9)$ *within the accuracy* 10^{-2}*.*

Solution. *(a) For the given data points we can use all three-points difference formulas with central difference at* $x_0 = 1.3, x_1 = 1.9, x_2 = 2.5$*, forward difference at* $x_0 = 1.9, x_1 = 2.5, x_2 = 3.1$*, and backward difference at* $x_0 = 1.3, x_1 = 1.6, x_2 = 1.9$*. But the value of* $h = 0.3$ *for the backward difference formula is smaller than both the other formulas with* $h = 0.6$*. So the best three-point formula for this case is the following backward difference formula*

$$f'(x_2) \approx \frac{f(x_2-2h) - 4f(x_2-h) + 3f(x_2)}{2h} = D_h f(x_2).$$

Thus using $x_2 = 1.9, x_2 - h = 1.6$*, and* $x_2 - 2h = 1.3$*, we have*

$$f'(1.9) \approx \frac{f(1.3) - 4f(1.6) + 3f(1.9)}{2(0.3)},$$

and using $f(x) = x\ln x + x$*, we obtain the required approximation of* $f'(1.9)$ *as follows*

$$f'(1.9) \approx \frac{(1.3\ln 1.3 + 1.3) - 4(1.6\ln 1.6 + 1.6) + 3(1.9\ln 1.9 + 1.9)}{0.6} = 2.6527.$$

(b) By using the backward difference error, we get

$$|E_C(f,h)| = \Big|\frac{(0.3)^2}{3}\Big||f'''(\eta(x_1))|, \quad f'''(\eta(x_2)) = -1/(\eta(x_2))^2, \quad \text{for} \quad \eta(x_2) \in (1.3, 1.9).$$

This formula cannot be computed exactly because $\eta(x_2)$ is not known. But one can bound the error by computing the largest possible value for $|f'''(\eta(x_2))|$. So bound $|f'''|$ on $[1.3, 1.9]$ is

$$M = \max_{1.3\le x\le 1.9} |-1/x^2| = 1/(1.3)^2 = 0.5917.$$

Thus, for $|f'''(\eta(x_2))| \le M$ and $h = 0.1$, gives the possible maximum error in our approximation as

$$|E_B(f,h)| \le \frac{0.09}{3}M = \frac{0.09}{3}(0.5917) = 0.0178,$$

(c) Since the exact value of the derivative $f'(1.9)$ is, 2.6419, therefore, the absolute error $|E|$ can be computed as follows

$$|E| = |f'(1) - D_h f(1.9)| = |2.6419 - 2.6527| = 0.0108.$$

(d) Since the given accuracy required is 10^{-2}, so

$$|E_B(f,h)| = \Big|\frac{h^2}{3}f'''(\eta(x_2))\Big| \le 10^{-2}, \quad \eta(x_2) \in (1.3, 1.9).$$

Then, $\dfrac{h^2}{3}M \le 10^{-2}$, *gives* $h \le \sqrt{\dfrac{3\times 10^{-2}}{0.5917}} = 0.2252.$ •

Example 5.10 *Consider following set of data points*

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9	1.0	1.1	1.2
$f(x)$	1.00	1.10	1.18	1.26	1.32	1.38	1.43	1.47	1.52	1.54	1.55	1.56

Use the table, find the best approximation of $f'(0.75)$ and the worst approximations of $f'(0.1)$ by using three-point formulas.

Solution. *For the best approximation of $f'(0.75)$, we have to take small value of $h = 0.15$, so using the central difference three-point formula (5.14), we get*

$$f'(0.75) \approx \frac{f(0.9) - f(0.6)}{2(0.15)} \approx \frac{1.52 - 1.43}{0.3} = 0.3,$$

while the exact value of $f'(0.75)$ is 0.3184. For the worst approximation of $f'(0.1)$, we have to take big value of $h = 0.5$, so using the forward difference three-point formula (5.18), we get

$$f'(0.1) \approx \frac{-3f(0.1) + 4f(0.6) - f(1.1)}{2(0.5)} = \frac{-3(1.1) + 4(1.43) - 1.55}{1} \approx 0.8700,$$

the required worst approximation. •

Example 5.11 *Values of $f(x)$ for values of x are given as*

$$(2.0, 1.5216),\ (3.0, 3.3456),\ (4.0, 5.5635).$$

Use Lagrange interpolating polynomial of degree 2 to find $f(x)$ and then find $f'(3)$. Also, find the value of x for which $f(x)$ is maximum or minimum.

Solution. *Consider a quadratic Lagrange interpolating polynomial*

$$f(x) = p_2(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) = 1.5216L_0(x) + 3.3456L_1(x) + 5.5635L_2(x).$$

We construct the basic Lagrange polynomials:

$$\begin{aligned} L_0(x) &= \frac{(x-3)(x-4)}{(2-3)(2-4)} = \frac{(x^2-7x+12)}{2}, \\ L_1(x) &= \frac{(x-2)(x-4)}{(3-2)(3-4)} = \frac{(x^2-6x+8)}{-1}, \\ L_2(x) &= \frac{(x-2)(x-3)}{(4-2)(4-3)} = \frac{(x^2-5x+6)}{2}. \end{aligned}$$

Using these values of the Lagrange coefficients, we have

$$f(x) = p_2(x) = 0.1970x^2 + 0.8392x - 0.9447,$$

which is the required $f(x)$. Then the derivative of it at $x = 3$ is

$$f'(x) = 0.3940x + 0.8392 \quad \textit{and} \quad f'(3) = 0.3940(3) + 0.8392 = 2.0212.$$

Note that the derivative of the function $f(x) = x\ln x + e^{-x}$ is $f'(x) = \ln x + 1 - e^{-x}$ and $f'(3) = 2.0488$. Now to find value of x at which $f(x)$ has maximum or minimum, we do as

$$f'(x) = 0.3940x + 0.8392 = 0, \quad \textit{(gives)} \quad x = -2.1299,$$

and since $f''(-2.1299) = 0.3940 > 0$, so $f(x)$ has minimum at the point $x = -2.1299$. •

5.3.1.6 Differentiation of the Newton Polynomial

Now we show the relationship between the three-point formulas for approximating $f'(x)$ by considering the Newton polynomial $p_2(x)$ of degree 2 that approximates $f(x)$ using the points x_0, x_1, and x_2 is

$$p_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \tag{5.24}$$

where $a_0 = f(x_0), a_1 = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}$, and

$$a_1 = \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0}.$$

The derivative of $p_2(x)$ is

$$p_2'(x) = a_1 + a_2[(x - x_0) + (x - x_1)], \tag{5.25}$$

and when it is evaluated at $x = x_0$, the result is

$$p_2'(x_0) = a_1 + a_2(x_0 - x_1) \approx f'(x_0). \tag{5.26}$$

Observe that the points $\{x_k\}$ do not need to equally spaced for formulas (5.24) through (5.26) to hold. Choosing the x in different orders will produce different difference formulas for approximating $f'(x)$.
Firstly, if $x = x_0, x_1 = x_0 + h$, and $x_2 = x_0 + 2h$, then

$$\begin{aligned} a_1 &= \frac{f(x_0 + h) - f(x_0)}{h} \\ a_1 &= \frac{f(x_0) - 2f(x_0 + h) + f(x_0 + 2h)}{2h^2}. \end{aligned}$$

When these values are substituted into (5.26), we obtain

$$p_2'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + \frac{-f(x_0) + 2f(x_0 + h) - f(x_0 + 2h)}{2h}.$$

This is simplified to get

$$p_2'(x_0) = \frac{-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)}{2h} \approx f'(x_0), \tag{5.27}$$

which is the second-order forward-difference formula (or three-point formula) for $f'(x)$.
Secondly, if $x = x_1, x_0 = x_1 - h$, and $x_2 = x_1 + h$, then

$$\begin{aligned} a_1 &= \frac{f(x_1 + h) - f(x_1)}{h} \\ a_1 &= \frac{f(x_1) - 2f(x_1) + f(x_1 - h)}{2h^2}. \end{aligned}$$

When these values are substituted into (5.26), we obtain

$$p_2'(x_1) = \frac{f(x_1 + h) - f(x_1)}{h} + \frac{-f(x_1 + h) + 2f(x_1) - f(x_1 - h)}{2h}.$$

This is simplified to get

$$p_2'(x_0) = \frac{f(x_1 + h) - f(x_1 - h)}{2h} \approx f'(x_1), \tag{5.28}$$

which is the second-order central-difference formula (or three-point formula) for $f'(x)$.
Finally, if $x = x_2, x_1 = x_2 - h$, and $x_0 = x_2 - 2h$, then

$$\begin{aligned} a_1 &= \frac{f(x_2) - f(x_2 - h)}{h} \\ a_1 &= \frac{f(x_2) - 2f(x_2 - h) + f(x_2 - 2h)}{2h^2}. \end{aligned}$$

When these values are substituted into (5.26) and simplified to get

$$p_2'(x_2) = \frac{3f(x_2) - 4f(x_2 - h) + f(x_2 - 2h)}{2h} \approx f'(x_2), \tag{5.29}$$

which is the second-order backward-difference formula (or three-point formula) for $f'(x)$.
Similarly, we can show the relationship between the (n+1)-point formulas for approximating $f'(x)$ by considering the Newton polynomial $p_N(x)$ of degree N that approximates $f(x)$ using the points $x_0, x_1, \ldots, x_N$,

$$\begin{aligned} p_N'(x) &= f[x_0, x_1] + f[x_0, x_1, x_2][(x - x_0) + (x - x_1)] + \cdots \\ &= f[x_0, x_1, \ldots, x_N] \sum_{i=0}^{n-1} \frac{(x - x_0)(x - x_1) \cdots (x - x_{N-1})}{(x - x_i)}. \end{aligned} \tag{5.30}$$

For example, to estimate the value of the derivative of $f(x) = x^3 + 7x^2 + 1$ for $x = 1, 2, 3, 4, 5$ at the interpolating point $x = 4.5$ using the cubic Newton polynomial we do as

Table 5.3: Divided differences table for given $f(x) = x^3 + 7x^2 + 1$.

k	x_k	Zeroth Divided Difference	First Divided Difference	Second Divided Difference	Third Divided Difference	Fourth Divided Difference
0	1	9				
1	2	37	28			
2	3	91	54	13		
3	4	177	86	16	1	
4	5	301	124	19	1	0

$p_3'(x) = f[x_0, x_1] + f[x_0, x_1, x_2][(x - x_0) + (x - x_1)] + f[x_0, x_1, x_2, x_3][(x - x_1)(x - x_2) + (x - x_0)(x - x_2) + (x - x_0)(x - x_1)]$,

or

$p_3'(4.5) = f'(4.5) = 54 + 16[(4.5 - 2) + (4.5 - 3)] + 1[(4.5 - 3)(4.5 - 4) + (4.5 - 2)(4.5 - 4) + (4.5 - 2)(4.5 - 3)] = 123.7500$,

which is the exact answer. This is as we expect, because $f(x)$ is a cubic and the cubic between $x = 2$ and $x = 5$ is actually a cubic.

5.3.2 Second Derivative Numerical Formula

It is also possible to estimate second and higher order derivatives numerically. Formulas for higher derivatives can be found by differentiating the interpolating polynomial repeatedly or using the Taylor's theorem. Since the *two-point* and *three-point* formulas for the approximation of the first derivative of a function were derived by differentiating the Lagrange interpolation polynomials for $f(x)$ but the derivation of the higher-order can be tedious. Therefore, we shall use here the Taylor's theorem for finding the *three-point* central-difference formulas for finding approximation of the second derivative $f''(x)$ of a function $f(x)$ at the given point $x = x_1$. The process used to obtain numerical formulas for first and second derivatives of a function can be readily extended to third- and higher-order derivatives of a function.

5.3.2.1 Three-point Central Difference Formula

To find the three-point central-difference formula for the approximation of the second derivative of a function at given point, we use the third-order Taylor's theorem by expanding a function $f(x)$ about a point x_1 and evaluate at $x_1 + h$ and $x_1 - h$. Then

$$f(x_1 + h) = f(x_1) + hf'(x_1) + \frac{1}{2}h^2 f''(x_1) + \frac{1}{6}h^3 f'''(x_1) + \frac{1}{24}h^4 f^{(4)}(\eta_1(x)),$$

$$f(x_1 - h) = f(x_1) - hf'(x_1) + \frac{1}{2}h^2 f''(x_1) - \frac{1}{6}h^3 f'''(x_1) + \frac{1}{24}h^4 f^{(4)}(\eta_2(x)),$$

where $(x_1 - h) < \eta_2(x) < x_1 < \eta_1(x) < (x_1 + h)$.
By adding these equations and simplifies, we have

$$f(x_1 + h) + f(x_1 - h) = 2f(x_1) + h^2 f''(x_1) + \frac{(f^{(4)}(\eta_1(x)) + f^{(4)}(\eta_2(x)))}{24}h^4.$$

Solving this equation for $f''(x_1)$, we obtain

$$f''(x_1) = \frac{f(x_1 - h) - 2f(x_1) + f(x_1 + h)}{h^2} - \frac{h^4}{24}\left[f^{(4)}(\eta_1(x)) + f^{(4)}(\eta_2(x))\right].$$

If $f^{(4)}$ is continuous on $[x_1 - h, x_1 + h]$, then by using the Intermediate Value Theorem, the above equation can be written as

$$f''(x_1) = \frac{f(x_1 - h) - 2f(x_1) + f(x_1 + h)}{h^2} - \frac{h^2}{12}f^{(4)}(\eta(x_1)).$$

Then the following formula

$$f''(x_1) \approx \frac{f(x_1 - h) - 2f(x_1) + f(x_1 + h)}{h^2} = D_h^2 f(x_1), \tag{5.31}$$

is called the *three-point central-difference formula* for the approximation of the second derivative of a function $f(x)$ at the given point $x = x_1$.

5.3.2.2 Error Bound for Three Point Central Difference Formula

If $f(x) \geq 0$ on $[a, b]$ and $h = \dfrac{(b-a)}{2}$, then

$$|E_(f)| \leq \frac{h^2}{12}M, \quad M = \max_{a \leq x \leq b} |f^{(4)}(x)|.$$

Example 5.12 *Let $f(x) = x \ln x + x$ and $x = 0.9, 1.3, 2.1, 2.5, 3.2$. Then find the approximate value of $f''(x) = \dfrac{1}{x}$ at $x = 1.9$. Also, compute the absolute error.*

Solution. *Given $f(x) = x \ln x + x$, then one can easily find second derivative of the function as*

$$f'(x) = \ln x + 2 \quad \textit{and} \quad f''(x) = \frac{1}{x}.$$

To find the approximation of $f''(x) = \dfrac{1}{x}$ *at the given point* $x_1 = 1.9$*, we use the three-point formula (5.31)*

$$f''(x_1) \approx \frac{f(x_1+h) - 2f(x_1) + f(x_1-h)}{h^2} = D_h^2 f(x_1).$$

Taking the three points $1.3, 1.9$ *and* 2.5 *(equally spaced), giving* $h = 0.6$*, we have*

$$f''(1.9) \approx \frac{f(2.5) - 2f(1.9) + f(1.3)}{0.36} = \frac{4.7907 - 6.2391 + 1.6411}{0.36} = 0.5353 = D_h^2 f(1.9).$$

Since the exact value of $f''(1.9)$ *is* $\dfrac{1}{1.9} = 0.5263$*, therefore, the absolute error* $|E|$ *can be computed as follows:*

$$|E| = |f''(1.9) - D_h^2 f(1.9)| = |0.5263 - 0.5353| = 0.009.$$

Note that the *error term* of the three-point central-difference formula (5.31) for the approximation of the second derivative of a function $f(x)$ at the given point $x = x_1$ is of the form

$$E_C(f,h) = -\frac{h^2}{12} f^{(4)}(\eta(x_1)), \quad \text{for some unknown point} \quad \eta(x_1) \in (x_1 - h, x_1 + h), \tag{5.32}$$

therefore, for a cubic function, $f(x) = ax^3 + bx^2 + cx + d$, the difference formula (5.31) is exact; that is, it yields the correct value of $f''(x)$ for any nonzero value of stepsize h. The error bound formula of Three point Central difference is

$$|E_(f)| \leq \frac{h^2}{12} M, \quad M = \max_{a \leq x \leq b} |f^{(4)}(x)|.$$

Example 5.13 *Let* $f(x) = x^5 + 1$ *be defined in the interval* $[0.1, 0.2]$*. Use the error formula of three-point formula for the approximation of* $f''(0.15)$ *to find a value of the unknown point* η*.*

Solution. *Since the error formula of the three-point central-difference formula for* $f''(0.15)$ *is*

$$E = Exact - Approx. = -\frac{h^2}{12} f^{(4)}(\eta), \quad \eta \in (0.1, 0.2).$$

Since the exact value of the second derivative of the function at $x_1 = 0.15$ *is*

$$Exact = f''(0.15) = 20(0.15)^3 = 0.0675,$$

and the approximate value of $f''(0.15)$ *using three point formula is*

$$Approx. = f''(0.15) \approx \frac{f(0.2) - 2f(0.15) + f(0.1)}{(0.05)^2} = 0.0710,$$

so error E *can be calculated as*

$$E = 0.0675 - 0.0710 = -0.0038.$$

Using the error formula and $f^{(4)}(\eta) = 120\eta$*, we have,* $-0.0038 = -\dfrac{(0.05)^2}{12} 120\eta$*, and solving for* η*, we get the required value of* $\eta = 0.1520$. •

Example 5.14 *Let $f(x) = x^2 + \cos x$ (x in radian). Then*
(a) Compute the approximate value of $f''(x)$ at $x = 1$, taking $h = 0.1$ using (5.31).
(b) Compute the error bound for your approximation using (5.32).
(c) Compute the absolute error.
(d) What is the best maximum value of stepsize h required to obtain the approximate value of $f''(1)$ within the accuracy 10^{-2}.

Solution. *(a) Given $x_1 = 1, h = 0.1$, then the formula (5.31) becomes*

$$f''(1) \approx \frac{f(1+0.1) - 2f(1) + f(1-0.1)}{(0.1)^2} = D_h^2 f(1),$$

or

$$f''(1) \approx \frac{f(1.1) - 2f(1) + f(0.9)}{0.01} = \frac{1.6636 - 3.0806 + 1.4316}{0.01} \approx 1.4600 = D_h^2 f(1).$$

(b) To compute the error bound for our approximation in part (a), we use the formula (5.32) as

$$|E_C(f,h)| = \Big| -\frac{h^2}{12}\Big| |f^{(4)}(\eta(x_1))|, \quad \textit{for} \quad \eta(x_1) \in (0.9, 1.1).$$

The fourth derivative of the given function at $\eta(x_1)$ is $f^{(4)}(\eta(x_1)) = \cos\eta(x_1)$, and it cannot be computed exactly because $\eta(x_1)$ is not known. But one can bound the error by computing the largest possible value for $|f^{(4)}(\eta(x_1))|$. So bound $|f^{(4)}|$ on the interval $(0.9, 1.1)$ is

$$M = \max_{0.9 \le x \le 1.1} |\cos x)| = 0.6216,$$

at $x = 0.9$, Thus, for $|f^{(4)}(\eta(x))| \le M$, we have the possible maximum error as

$$|E_C(f,h)| \le \frac{h^2}{12} M = \frac{(0.1)^2}{12}(0.6216) = 5.1800e-004 = 0.000518.$$

By using the MATLAB symbolic commands:

$>> syms\ x;\ >> f = x.\hat{}\ 2 + cos(x);\ ddf = diff(f,2);\ dddf = diff(f,4);\ subs(ddf,1)$

(c) Since the exact value of $f''(1)$ is

$$f''(1) = 2 - \cos(1) = 1.4597,$$

therefore, the absolute error $|E|$ can be computed as follows:

$$|E| = |f''(1) - D_h^2 f(1)| = |1.4597 - 1.4600| = 0.0003.$$

(d) Since the given accuracy required is 10^{-2}, so

$$|E_C(f,h)| = \Big| -\frac{h^2}{12} f^{(4)}(\eta(x_1))\Big| \le 10^{-2},$$

for $\eta(x_1) \in (0.9, 1.1)$. Then for $|f^{(4)}(\eta(x_1))| \le M$, we have

$$\frac{h^2}{12} M \le 10^{-2}, \quad h^2 \le \frac{(12 \times 10^{-2})}{M} = \frac{(12 \times 10^{-2})}{0.6216} = 0.1931, \quad h \le 0.4394.$$

Using the following MATLAB commands, we can easily achieved the above results:

```
>> x0 = 1.0; h = 0.1;
>> ddf = ((x0 + h).^ 2 + cos(x0 + h) - 2. * (x0.^ 2 + cos(x0)) + ...
        (x0 - h).^ 2 + cos(x0 - h))./(h. ^ 2)
```

The central-difference formula is probably the most used approximation for derivatives. Many real problems are modeled by second-order differential equations, involving either ordinary or partial derivatives. These equations cannot be solved analytically. To solve the equations numerically requires the replacement of the second-order derivatives by the difference formula (5.31). •

Example 5.15 *The function $f(x)$ satisfies a given equation $f''(x) = x^2 f(x)$ and the conditions $f(0) = 1,\ f(0.2) = 3$. Use the central-difference formula (5.31) for $f''(x)$ to estimate the value of $f(0.1)$ and then find $f''(0.1)$.*

Solution. *Given $h = 0.1$, so we take $(x_1 - h) = 0, x_1 = 0.1$ and $(x_1 + h) = 0.2$, then using the central-difference formula for the second derivative of a function*

$$f''(x_1) \approx \frac{f(x_1 - h) - 2f(x_1) + f(x_1 + h)}{h^2},$$

we obtain

$$f''(0.1) = (0.1)^2 f(0.1) \approx \frac{f(0) - 2f(0.1) + f(0.2)}{0.01},$$

which is equal to

$$(0.01)f(0.1) \approx \frac{1 - 2f(0.1) + 3}{0.01}, \quad \textit{gives,} \qquad f(0.1) = \frac{4}{2 + 0.0001} = 1.9999,$$

$$f''(0.1) \approx \frac{1 - 2(1.9999) + 3}{0.01} = 0.02,$$

the required solution. •

Example 5.16 *Find the approximation of $f''(0.8)$ by using the following set of data points using three-point central difference rule:*

x	0.0	0.11	0.24	0.3	0.4	0.5	0.6	0.72	0.8	0.9	1.05	1.11	1.2
$f(x)$	1.00	1.10	1.2	1.26	1.32	1.38	1.43	1.47	1.50	1.52	1.55	1.55	1.56

The function tabulated is $f(x) = x + \cos x$, then compute an error bound for your approximation. How many subintervals approximate the given derivative to within accuracy of 10^{-6} using this differentiation rule ?

Solution. *Given $x_1 = 0.8, h = 0.4$, then the formula (5.31) becomes*

$$f''(0.8) \approx \frac{f(0.8 + 0.4) - 2f(0.8) + f(0.8 - 0.4)}{(0.4)^2} = D_h^2 f(1),$$

or

$$f''(0.8) \approx \frac{f(1.2) - 2f(0.8) + f(0.4)}{0.16} = \frac{1.56 - 2(1.50) + (1.32)}{0.16} = -0.75 = D_h^2 f(1).$$

To compute the error bound for our approximation, we use the formula as

$$|E_C(f,h)| = \Big| -\frac{h^2}{12}\Big| |f^{(4)}(\eta(x_1))|, \quad for \quad \eta(x_1) \in (0.9, 1.1).$$

The fourth derivative of the given function at $\eta(x_1)$ *is* $f^{(4)}(\eta(x_1)) = \cos\eta(x_1)$, *and it cannot be computed exactly because* $\eta(x_1)$ *is not known. But one can bound the error by computing the largest possible value for* $|f^{(4)}(\eta(x_1))|$. *So bound* $|f^{(4)}|$ *on the interval* $(0.4, 1.2)$ *is*

$$M = \max_{0.4\le x\le 1.2} |\cos x)| = 0.9211,$$

at $x = 0.4$, *Thus, for* $|f^{(4)}(\eta(x))| \le M$, *we have the possible maximum error as*

$$|E_C(f,h)| \le \frac{h^2}{12}M = \frac{(0.4)^2}{12}(0.9211) = 0.01228.$$

Since the given accuracy required is 10^{-6}, *so*

$$|E_C(f,h)| = \Big| -\frac{h^2}{12}f^{(4)}(\eta(x_1))\Big| \le 10^{-6},$$

for $\eta(x_1) \in (0.4, 1.2)$. *Then for* $|f^{(4)}(\eta(x_1))| \le M$, *we have*

$$\frac{h^2}{12}M \le 10^{-6}, \quad \frac{(1.2-0.4)^2}{n^2} \le \frac{(12\times 10^{-6})}{M},$$

solving for n, *we get*

$$n \ge \sqrt{(0.64 \times 0.9211 \times 10^6)/12} \ge 221.6424, \quad gives \quad n = 222,$$

the required subintervals for approximating the given derivative to the given accuracy . •

5.4 Numerical Integration

Numerical integration has a history extending back to the invention of calculus and before. It is used to integrate tabulated functions or to integrate functions whose integrals are either impossible or very difficult to obtain analytically. Even when analytical integration is easy, numerical integration may save time and effort if only the numerical value of the integral is desired. Consequently, numerical methods of integration represent a natural alternative whenever conventional methods fail to yield a solution.
Now for numerical integration, we wish to find an approximation to the *definite integral*

$$I(f) = \int_a^b f(x)dx, \tag{5.33}$$

assuming that $f(x)$ is integrable . If $f(x) \ge 0$ on the given interval $[a, b]$, then geometrically, the integral (5.33) is equivalent to replacing the *area* under the graph of $f(x)$, the x-axis and between the ordinates $x = a$ and $x = b$.

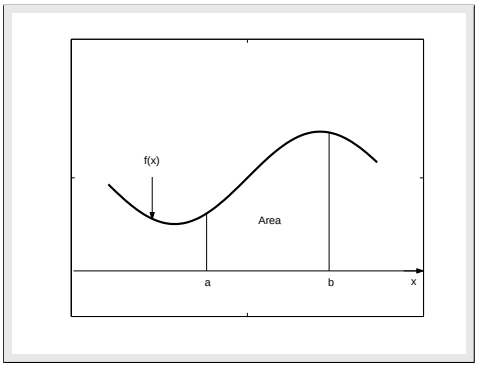


Figure 5.3: Definite integral for $f(x)$

The fundamental theorem of calculus shows that integration is the inverse process to differentiation. If we can find a function $F(x)$, called the antiderivative of $f(x)$, that is, $F'(x) = f(x)$, then we can evaluate the integral (5.33) using the relation

$$I(f) = \int_a^b f(x)dx = F(b) - F(a). \tag{5.34}$$

Sometimes considerable skill is required to obtain $F(x)$, perhaps by making a change of variable or integrating by parts. But in many cases $F(x)$ cannot be found by elementary methods. In such cases the computation of the integral (5.33) by means of formula (5.34) may be either too difficult or practically impossible. Even if $F(x)$ can be found, it may still more convenient to use a numerical method to estimate the integral (5.33) if the evaluation of $F(x)$ required a great deal of computation. Moreover, in practical applications, a function $f(x)$ often given in tabular form and then the entire concept of antiderivative is meaningless. An obvious approach is to replace a function $f(x)$ in the integral (5.33) by an approximating polynomial $p(x)$, that is

$$I(f) = \int_a^b f(x)dx \approx \int_a^b p(x)dx.$$

Numerical integration formulas are derived by integrating interpolation polynomials. Therefore, different interpolation formulas will leads to different numerical integration methods.
The definite integral (5.33) may be interpreted as the area under the curve of $y = f(x)$ from a to b as shown by Figure 5.3.
It should be noted that any areas beneath the x-axis are counted as negative. Many numerical methods for integration are based on using this interpretation of the integral to derive approximations to it by dividing the interval $[a, b]$ into a number of smaller subintervals. By making simple approximations to the curve $y = f(x)$ in the small subinterval its area may be obtained and on summing all the contributions we obtain an approximation to a integral in the interval $[a, b]$. Variations of this technique are derived by taking groups of subintervals and fitting different degree polynomials as approximations for each of these groups. The lead of accuracy obtained is dependent on the number of intervals used and the nature the approximation function.
There are several methods available in the literature for numerical integration but the most commonly methods may be classified into two groups.

(a) The Newton-Cotes formulas that employ functional values at equally spaced data points.

(b) The Gaussian quadrature formulas that employ unequally spaced data points determined by certain properties of orthogonal polynomials.

Firstly, we shall discuss the Newton-Cotes formulas which has two different types, called, the *closed Newton-Cotes* formulas and the *open Newton-Cotes* formulas. In the first type, we shall discuss in some details the two mostly usable formulas, called the *Trapezoidal rule* and the *Simpson's rule* which can be derived by integrating the Lagrange interpolating polynomials of degree 1 and 2 respectively. In the second type we shall consider some good formulas. The use of the closed Newton-Cotes and other integration formulas of order higher than the Simpson's rule is seldom necessary in most engineering applications and can be use for those cases where extremely high accuracy is required.

Example 5.17 *Find the numerical integration rule using the points $x_0 = 0, x_1 = 1$ and $x_2 = 2$ for approximating the integral $\int_0^3 f(x)dx$, for $f(x) = 1, x, x^2$. Use the resulting rule to estimate the integral $\int_0^3 (x^2+1)dx$.*

Solution. *Consider*

$$I = \int_0^3 f(x)dx \approx a_0 f(x_0) + a_1 f(x_1) + a_2 f(x_2),$$

then for $f(x) = 1,\ f(x) = x$ and $f(x) = x^2$, we get the following linear system

$$\begin{array}{lllllll} 3 & \approx & a_0 & + & a_1 & + & a_2 \\ 9/2 & \approx & 0 & + & a_1 & + & 2a_2 \\ 9 & \approx & 0 & + & a_1 & + & 4a_2 \end{array}$$

which has the solution, $a_0 = 3/4,\ a_1 = 0,\ a_2 = 9/4$. Thus the required integration rule is

$$\int_0^3 f(x)dx \approx \frac{3f(0) + 9f(2)}{4},$$

and with $f(x) = x^2 + 1$, we get

$$\int_0^3 (x^2+1)dx \approx \frac{3(0+1) + 9(4+1)}{4} = 12.$$

One can easily compute the true integral as

$$\int_0^3 (x^2+1)dx = (x^3/3 + x)\Big|_0^3 = 12.$$

Thus the above integration rule gave the exact solution for the given function. •

Using MATLAB symbolic toolbox we can easily obtained symbolic integration of function as

```
>> syms x;
>> f = x.^ 2 + 1; I = int(f);
>> a = 0; b = 3; A = int(f, a, b); subs(A);
```

Example 5.18 *Find α such that the integration formula $\int_0^1 \frac{f(x)}{\sqrt{x}} \approx Af(0) + Bf(\alpha) + Cf(1)$ may be exact for polynomials up to degree* 3.

Solution. *Consider*

$$I = \int_0^1 \frac{f(x)}{\sqrt{x}} = Af(0) + Bf(\alpha) + Cf(1),$$

then by taking $f(x) = 1, x, x^2$ and x^3, we get

$$\begin{array}{lllllll} 2 & = & A & + & B & + & C, \\ 2/3 & = & 0 & + & B\alpha & + & C, \\ 2/5 & = & 0 & + & B\alpha^2 & + & C, \\ 2/7 & = & 0 & + & B\alpha^3 & + & C. \end{array}$$

Subtracting second equation from third equation, gives, $B\alpha(\alpha - 1) = -4/15$, and then subtracting third equation from fourth equation, gives, $B\alpha^2(\alpha - 1) = -4/35$. Thus

$$-4/35 = B\alpha^2(\alpha - 1) = \alpha[B\alpha(\alpha - 1)] = \alpha(-4/15), \quad \textit{or} \quad -4/35 = \alpha(-4/15),$$

and solving for α, it gives, $\alpha = 3/7$. •

5.5 Closed Newton-Cotes Formulas

The usual strategy in developing formulas for numerical integration is similar to that for numerical differentiation. We pass a polynomial through points of a function and then integrate this polynomial approximation to a function. This allows us to integrate a function known only as a table of values. Some common formulas based on polynomial interpolation are referred to as the Newton-Cotes formulas.

An $(n+1)$-point Newton-Cotes formula for approximating the definite integral (5.33) is obtained by replacing the integrand $f(x)$ by the *nth*-degree Lagrange polynomial that interpolates the values of $f(x)$ at equally spaced data points

$$a = x_0 < x_1 < \ldots < x_n = b.$$

Note that if the end-points a and b of the given interval $[a, b]$ are in the set of interpolating points; then the Newton-Cotes formulas are called *closed*; otherwise, it is said to be *open*.

An $(n+1)$-point closed Newton-Cotes formula used points $x_i = x_0 + ih$, for, $i = 0, 1, 2, \ldots, n$, where $x_0 = a$, $x_n = b$ and $h = \dfrac{b-a}{n}$, has the form (see Figure 5.4)

$$\int_a^b f(x)dx = \int_{x_0}^{x_n} f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i), \tag{5.35}$$

where

$$a_i = \int_{x_0}^{x_n} L_i(x)dx = \int_{x_0}^{x_n} \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} dx. \tag{5.36}$$

The following theorem describes the error analysis associated with the above closed Newton-Cotes formulas.

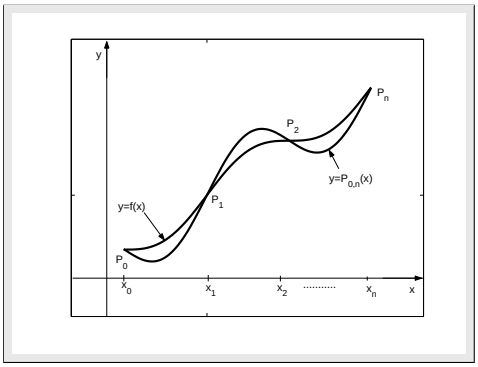


Figure 5.4: Close Newton-Cotes approximation

Theorem 5.1 (Close Newton-Cotes Formulas)

Suppose that $\sum_{i=0}^{n} a_i f(x_i)$ *denotes the* $(n+1)$*-point closed Newton-Cotes formula with* $x_0 = a, x_n = b$, *and* $h = (b-a)/n$. *There exists* $\eta(x) \in (a,b)$ *for which*

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\eta(x))}{(n+2)!} \int_0^n t^2(t-1)\cdots(t-n)dt, \tag{5.37}$$

if n *is even and* $f \in C^{n+2}[a,b]$. *For* $f \in C^{n+1}[a,b]$, *and* n *is odd, then*

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\eta(x))}{(n+1)!} \int_0^n t(t-1)\cdots(t-n)dt. \tag{5.38}$$

Different numerical integration formulas can be obtained by using the formulas (5.37) and (5.38) to approximate the definite integral (5.33). By using the formula (5.38) for $n = 1$, we have well-known numerical integration formula, called, the *Trapezoidal rule*. Similarly, by using the formula (5.37) for $n = 2$, we have one of the best integration rule called, the *Simpson's rule*. We shall discuss the formulation of both these rules and also discuss about their error terms. Later we shall also consider some more closed Newton-Cotes formulas.

5.5.1 Trapezoidal Rule

It is one of the oldest and good numerical method for approximating the definite integral (5.33). It is based on approximating a function in each subinterval by a straight line.

5.5.1.1 Simple Trapezoidal Rule

To derive the Trapezoidal rule for one-strip (one interval), let us consider the first degree Lagrange interpolating polynomial with equally spaced data points, that is, $x_0 = a, x_1 = b$ and $h = x_1 - x_0$, then

$$f(x) = p_1(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) f(x_0) + \left(\frac{x - x_0}{x_1 - x_0}\right) f(x_1). \tag{5.39}$$

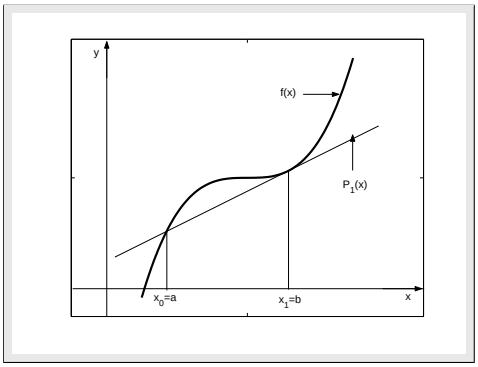


Figure 5.5: Simple Trapezoidal rule.

Taking integral on both sides of (5.39) with respect to x between the limits x_0 and x_1, we have

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{f(x_0)}{x_0 - x_1}\int_{x_0}^{x_1}(x - x_1)dx + \frac{f(x_1)}{x_1 - x_0}\int_{x_0}^{x_1}(x - x_0)dx,$$

which implies that

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{f(x_0)}{x_0 - x_1}\left[\frac{(x - x_1)^2}{2}\Big|_{x_0}^{x_1}\right] + \frac{f(x_1)}{x_1 - x_0}\left[\frac{(x - x_0)^2}{2}\Big|_{x_0}^{x_1}\right] \approx \frac{(x_1 - x_0)}{2}[f(x_0) + f(x_1)],$$

and by taking $h = x_1 - x_0$, we get

$$\int_{a=x_0}^{b=x_1} f(x)dx \approx T_1(f) = \frac{h}{2}[f(x_0) + f(x_1)]. \tag{5.40}$$

Then $T_1(f)$ is called the *simple Trapezoidal rule* or the Trapezoidal rule for one trapezoid or one strip and can be use for the approximation of the definite integral (5.33). The reason for calling this formula the Trapezoidal rule is that when $f(x)$ is a function with positive values, the integral (5.33) is approximated by the area in the trapezoid, see Figure 5.5.

Example 5.19 *Approximate the following integral* $\int_1^2 \frac{1}{x+1}dx$, *using the simple Trapezoidal rule and compute the absolute error.*

Solution. *Given* $f(x) = \frac{1}{x+1}$ *and* $h = 1$, *so using the simple Trapezoidal rule (5.40), gives*

$$T_1(f) = \frac{1}{2}\left[\frac{1}{2} + \frac{1}{3}\right] = \frac{5}{12} = 0.4167.$$

The exact solution of the given integral is, $I(f) = \ln(3/2) = 0.4055$, *so* $|0.4055 - 0.4167| = 0.0112$, *is the absolute error.* •

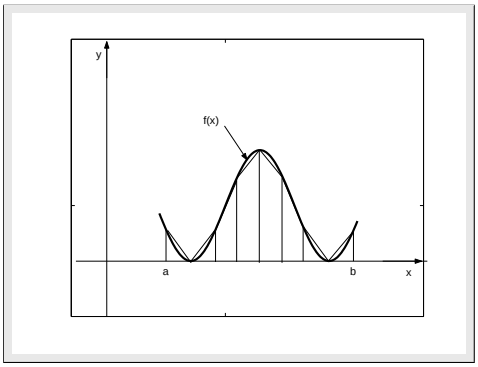


Figure 5.6: Composite Trapezoidal rule.

5.5.1.2 Composite Trapezoidal Rule

It is evident that the Newton-Cotes formulas produce accurate approximations to the definite integral (5.33) only when the limits a and b are close together, that is, the integration interval is not large. Formulas based on low-degree interpolating polynomials are clearly unsuitable since it is then necessary to use large values of h. Also, note that higher-order Newton-Cotes formulas will not necessarily produce more accurate approximations to the given integral. This difficulty can be avoided by using a piecewise approach; the integration interval is divided into subintervals and low-order formulas are applied on each of these. The corresponding integration rules are said to be in *composite form*, and the most suitable formula of this type make use of the Trapezoidal rule. The interval $[a, b]$ is partitioned into n subintervals $(x_{i-1}, x_i), \quad i = 1, 2, \ldots, n$ with $a = x_0$ and $b = x_n$ of equal width $h = (b - a)/n$ and the rule for a single interval (the simple rule (5.40)) is applied to each subinterval or a grouping of subintervals (see Figure 5.6). Since the Trapezoidal rule requires only one interval for application, there is no restriction on the integer n. We define the composite Trapezoidal rule in the form of the following theorem.

Theorem 5.2 (Composite Trapezoidal Rule)

Let $f \in C^2[a, b]$, n may be odd or even, $h = (b - a)/n$, and $x_i = a + ih$ for each $i = 0, 1, 2, \ldots, n$. Then the composite Trapezoidal rule for n subintervals can be written as

$$\int_{a=x_0}^{b=x_n} f(x)dx \approx T_n(f) = \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right]. \tag{5.41}$$

Proof. *Since for the composite form of the Trapezoidal rule, the interval is divided into n equal subintervals of width h so that $h = \dfrac{b-a}{n}$, and (n+1) distinct points $a = x_0 < x_1 < x_2 \ldots < x_n = b$, then we have*

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)dx.$$

Applying the Trapezoidal rule (5.40) for one strip to each of these integral, we have

$$\int_{a=x_0}^{b=x_n} f(x)dx \approx \frac{h}{2}[f(x_0) + f(x_1)] + \frac{h}{2}[f(x_1) + f(x_2)] + \cdots + \frac{h}{2}[f(x_{n-1}) + f(x_n)].$$

Note that each of the interior point is counted twice and therefore has a coefficient of two whereas the endpoints are counted once and therefore has a coefficient one. •

Example 5.20 *Evaluate the integral* $\int_0^1 e^{4x}dx$ *by using the Trapezoidal rule with* $n = 1, 2, 4, 8$. *Also compute the corresponding absolute errors.*

Solution. *For* $n = 1$, *we use the formula (5.40) for* $h = 1$, *as follows*

$$T_1(f) = \frac{1}{2}\Big[f(0) + f(1)\Big] = 27.7991.$$

For $n = 2$, *using the formula (5.41) and* $h = 0.5$, *we have*

$$T_2(f) = \frac{0.5}{2}\Big[f(0) + 2f(0.5) + f(1)\Big] = 17.5941.$$

For $n = 4$, *using the formula (5.41) and* $h = 0.25$, *we have*

$$T_4(f) = \frac{0.25}{2}\Big[f(0) + 2[f(0.25) + f(0.5) + f(0.75)] + f(1)\Big] = 14.4980.$$

Finally, for $n = 8$, *using (5.41) and* $h = 0.125$, *we have*

$$\begin{aligned} T_8(f) &= \frac{0.125}{2}\Big[f(0) + 2[f(0.125) + f(0.25) + f(0.375) + f(0.5) \\ &+ f(0.625) + f(0.75) + f(0.875)] + f(1)\Big] = 13.6776. \end{aligned}$$

Since the exact value of the given integral is

$$I(f) = \frac{1}{4}[e^4 - 1] = 13.4000.$$

So the corresponding absolute errors are, $14.3991, 4.1941, 1.0980$ and 0.2776, respectively, which decrease by a factor of about *four* at each stage. •

Example 5.21 *Suppose that* $f(0.25) = f(0.75) = \alpha$. *Find* α *if the composite Trapezoidal rule with* $n = 2$, *gives,* $\int_0^1 f(x)\ dx = 2$ *and with* $n = 4$, *it gives* $\int_0^1 f(x)\ dx = 1.75$.

Solution. *For* $n = 2$, *using the formula (5.41) and* $h = 0.5$, *we have*

$$\int_0^1 f(x)\ dx = 2 \approx T_2(f) = \frac{0.5}{2}\Big[f(0) + 2f(0.5) + f(1)\Big],$$

which is equivalent to

$$8 \approx f(0) + 2f(0.5) + f(1).$$

For $n = 4$, *using the formula (5.41) and* $h = 0.25$, *we have*

$$\int_0^1 f(x)\ dx = 1.75 \approx T_4(f) = \frac{0.25}{2}\Big[f(0) + 2(2\alpha) + 2f(0.5) + f(1)\Big],$$

which is equals to

$$8(1.75) \approx f(0) + 2f(0.5) + f(1) + 4\alpha, \quad \textit{or} \quad 8(1.75) \approx 8 + 4\alpha.$$

(using $8 \approx f(0) + 2f(0.5) + f(1)$*). Solving for* α, *we get* $\alpha \approx 1.5$, *the required value.* •

Note that the Trapezoidal rule of integration involves no restriction relative to number of data points involved. This is not the case with elaborate methods yet to discussed. This is one reason why it is one of the favorite numerical integration method used in mathematics and engineering.

5.5.1.3 Error Terms for Trapezoidal Rule

Now we discuss the error formulas for the Trapezoidal rule. These formulas will lead to a better understanding of the method, showing both their weakness and strengths, and will allow improvements of the method. First discuss the error for the simple Trapezoidal rule (5.40) in the from of the following theorem and then we use it to define the error for the Trapezoidal rule (5.41).

Theorem 5.3 (Error term for Simple Trapezoidal Rule)

Let $f \in C^2[a,b]$, and $h = (b-a)$. The local error that the simple Trapezoidal rule (5.40) makes in estimating the definite integral (5.33) is

$$E_{T_1}(f) = -\frac{h^3}{12} f''(\eta(x)), \quad \text{where} \qquad \eta(x) \in (a,b). \tag{5.42}$$

Proof. *Consider two points $a = x_0 < x_1 = b$ with $h = x_1 - x_0$. From the linear Lagrange interpolation formula with error term, we have*

$$f(x) = p_1(x) + \frac{f''(\eta(x)(x))}{2!} \prod_{i=0}^{1} (x - x_i). \tag{5.43}$$

By integrating (5.43) with respect to x and between x_0 and x_1, we have

$$\int_{x_0}^{x_1} f(x)dx = \int_{x_0}^{x_1} p_1(x)dx + \frac{1}{2}\int_{x_0}^{x_1} f''(\eta(x)) \prod_{i=0}^{1} (x - x_i)dx. \tag{5.44}$$

The error term for the Trapezoidal rule of one strip can be obtained as follows:

$$E_{T_1}(f) = \frac{1}{2}\int_{x_0}^{x_1} f''(\eta(x))(x - x_0)(x - x_1)dx.$$

Note that $f''(\eta(x))$ is a continuous function of x and the term $(x-x_0)(x-x_1)$ is negative on (a,b), therefore, by using the Mean Value Theorem for integrals, we have

$$E_{T_1}(f) = \frac{f''(\eta(x))}{2}\int_{x_0}^{x_1} (x - x_0)(x - x_1)dx.$$

Now to solve the integral on the right side of above equation, we use the change of variable

$$x - x_0 = uh,\ x - x_1 = (u-1)h, \quad \text{and} \quad dx = hdu$$

then we have

$$E_{T_1}(f) = \frac{f''(\eta(x))}{2}\int_0^1 huh(u-1)hdu = \frac{f''(\eta(x))h^3}{2}\int_0^1 (u^2 - u)du = -\frac{h^3 f''(\eta(x)))}{12}, \tag{5.45}$$

for some $\eta(x) \in (a,b)$. Then the formula (5.44) can be written as

$$\int_{a=x_0}^{b=x_1} f(x)dx = \frac{h}{2}[f(x_0)+f(x_1)] - \frac{h^3}{12}f''(\eta(x)), \tag{5.46}$$

for $\eta(x) \in (a,b)$, which is the simple Trapezoidal rule with its error term. •

The formula (5.45) indicates that the local error of the Trapezoidal rule is proportional to second derivative f''. So, if the Trapezoidal rule is used to integrate each of $f(x) = 1, x, x^2, x^3, \ldots$, then results have no error for $f(x) = 1$ and $f(x) = x$ but there are error for x^2 and higher power of x.

5.5.1.4 Error Bound for Simple Trapezoidal Rule

If $f(x) \geq 0$ on $[a,b]$ and $h = (b-a)$, then

$$|E_{T_1}(f)| \leq \frac{h^3}{12}M, \quad M = \max_{a\leq x\leq b}|f''(x)|.$$

Example 5.22 *Find the error bound for the Trapezoidal rule (5.40) for the integral* $\int_1^2 \frac{1}{x+1}dx$.

Solution. *Given* $f(x) = \frac{1}{x+1}$ *and* $[a,b] = [1,2]$*, then the second derivative of the function is*

$$f''(x) = \frac{2}{(x+1)^3}.$$

Since the error formula for the simple Trapezoidal rule is

$$E_{T_1}(f) = -\frac{h^3}{12}f''(\eta(x)), \quad \textit{where} \quad \eta(x) \in (1,2).$$

This formula cannot be computed exactly because $\eta(x)$ is not known. But one can bound the error by computing the largest possible value for $|f''(\eta(x))|$. Bound $|f''(\eta(x))|$ on $[1,2]$ is

$$M = \max_{1\leq x\leq 2}\Big|\frac{2}{(x+1)^3}\Big| = 0.25.$$

Then, for $|f''(\eta(x))| \leq M$, we have

$$|E_{T_1}(f)| \leq \frac{h^3}{12}M \leq \frac{(1)^3}{12}0.25 = 0.0208.$$

Comparing this with the absolute error 0.0112*, this bound is about* 2 *times the actual error.* •

Example 5.23 *Let $f(x) = x^3 + 1$ defined on the interval $[0.1, 0.2]$. Find the value of unknown point $\eta(x)$ by using the local error for the simple Trapezoidal rule (5.40).*

Solution. *Given $f(x) = x^3 + 1$, and $[a,b] = [0.1, 0.2]$, we use the formula (5.40) for $h = 0.1$, as follows*

$$ApproxValue = \frac{0.1}{2}\Big[f(0.1) + f(0.2)\Big] = \frac{0.1}{2}\Big[[(0.1)^3+1] + [(0.2)^3+1]\Big] = 0.10045.$$

We know that

$$ExactValue = \int_{0.1}^{0.2} (x^3 + 1)\ dx = (x^4/4 + x)\Big|_{0.1}^{0.2} = [(0.2)^4/4 + 0.2] - [(0.1)^4/4 + 0.1] = 0.100375,$$

so we have the error

$$E = ExactValue - ApproxValue = 0.100375 - 0.10045 = -0.000075.$$

Since the second derivative of the given function is $f''(x) = 6x$*, so by using the local error for the Trapezoidal rule (5.40), we have,* $-0.000075 = -\dfrac{(0.1)^3}{12}(6\eta(x))$*, gives the value of* $\eta(x) = 0.15$. •

5.5.1.5 Error Term for Composite Trapezoidal Rule

The *global error* of the Trapezoidal rule (5.41) equals the sum of n local errors of it, that is

$$E_{T_n}(f) = -\frac{h^3}{12}f''(\eta_1(x)) - \frac{h^3}{12}f''(\eta_2(x)) - \cdots - \frac{h^3}{12}f''(\eta_n(x)) = -\frac{h^3}{12}nf''(\eta(x)), \quad \eta_i(x) \in (x_{i-1}, x_i),$$

where $f''(\eta(x))$ is the average of the n individual values of the second derivative. Since $n = \dfrac{b-a}{h}$, thus the global error in the composite Trapezoidal rule (5.41) is

$$E_{T_n}(f) = -\frac{h^2}{12}(b-a)f''(\eta(x)), \quad \eta(x) \in (a, b). \tag{5.47}$$

Hence

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right] - \frac{h^2}{12}(b-a)f''(\eta(x)), \tag{5.48}$$

for $\eta(x) \in (a, b)$, is the composite Trapezoidal rule with its error term.
Note that whereas the simple Trapezoidal rule (5.40) has a truncation error of order h^3, the composite Trapezoidal rule (5.41) has an error of order h^2. This means that when h is halved and the number of subintervals is doubled the error decreases by a factor of approximately four (assuming that $f''(\eta(x))$ remains fairly constant throughout $[a, b]$). Of course, it is also possible to express the truncation error in terms of n rather than h. Since $h = \dfrac{b-a}{n}$, it follows that the global truncation error (5.47) is of order $O(n^2)$.

5.5.1.6 Error Bound for Composite Trapezoidal Rule

If $f(x) \geq 0$ on $[a, b]$ and $h = \dfrac{(b-a)}{n}$, then

$$|E_{T_n}(f)| \leq \frac{h^2(b-a)}{12}M, \quad M = \max_{a \leq x \leq b} |f''(x)|.$$

Example 5.24 *(a) Find approximation of* $\displaystyle\int_1^2 f(x)\ dx$*, taking* $h = 0.2$ *by using following set of data points using Trapezoidal rule:*

x	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$f(x)$	0.368	0.366	0.361	0.354	0.355	0.335	0.323	0.311	0.298	0.284	0.271

(b) The function tabulated is $f(x) = xe^{-x}$, compute an error bound and the absolute error for the approximation using Trapezoidal rule.
(c) How many subintervals approximate the given integral to an accuracy of at least 10^{-6} ?

Solution. *(a) Given $h = 0.2$, we select the following table:*

x	1.0	1.2	1.4	1.6	1.8	2.0
$f(x)$	0.368	0.361	0.355	0.323	0.298	0.271

so the composite Trapezoidal rule (5.41) for six points can be written as

$$\int_1^2 f(x)\,dx \approx T_5(f) = \frac{h}{2}\Big[f(x_0) + 2\Big(f(x_1) + f(x_2) + f(x_3) + f(x_4)\Big) + f(x_5)\Big],$$

$$\int_1^2 f(x)\,dx \approx 0.1\Big[0.368 + 2(0.361 + 0.355 + 0.323 + 0.298) + 0.271\Big] = 0.3313.$$

(b) The derivatives of the function $f(x) = xe^{-x}$ can be obtain as

$$f'(x) = (1-x)e^{-x} \quad \textit{and} \quad f''(x) = (x-2)e^{-x}.$$

Since $\eta(x)$ is unknown point in $(1, 2)$, therefore, the bound $|f''|$ on $[1, 2]$ is

$$M = \max_{1\le x\le 2}|f''(x)| = \max_{1\le x\le 2}|(x-2)e^{-x}| = 0.3679,$$

at $x = 1$. Thus the error formula (5.47) becomes

$$|E_{T_5}(f)| \le \frac{(0.2)^2(1)}{12}(0.3679) = 0.0012,$$

which is the possible maximum error in our approximation.
We can easily computed the exact value of the given integral as

$$\int_1^2 xe^{-x}\,dx = (-xe^{-x} - e^{-x})\Big|_1^2 = 0.3298.$$

Thus the absolute error $|E|$ in our approximation is given as

$$|E| = |0.3298 - T_5(f)| = |0.3298 - 0.3313| = 0.0015.$$

(c) To find the minimum subintervals for the given accuracy, we use the formula (5.47) such that

$$|E_{T_n}(f)| \le \frac{|-(b-a)^3|}{12n^2}M \le 10^{-6},$$

where $h = (b-a)/n$. Since $M = 0.3679$, then solving for n, we obtain, $n \ge 175.0952$. Hence to get the required accuracy, we need 176 subintervals or 177 points. •

Example 5.25 *Consider the integral* $I(f) = \int_1^2 \ln(x+1)dx; \qquad n = 6$ *(or 7 points).*
(a) Compute the approximation of the integral using the composite Trapezoidal rule.
(b) Compute the error bound for your approximation using the formula (5.47).
(c) Compute the absolute error.
(d) How many subintervals approximate the given integral to an accuracy of at least 10^{-4} *using the composite Trapezoidal rule ?*

Solution. *(a) Given* $f(x) = \ln(x+1), n = 6$, *and so* $h = \dfrac{2-1}{6} = \dfrac{1}{6}$, *then the composite Trapezoidal rule (5.41) for* $n = 6$, *can be written as*

$$\begin{aligned} T_6(f) &= \frac{1/6}{2}\Big[\ln(1+1) + 2\Big(\ln\Big(\frac{7}{6}+1\Big) + \ln\Big(\frac{8}{6}+1\Big) + \ln\Big(\frac{9}{6}+1\Big) \\ &+ \ln\Big(\frac{10}{6}+1\Big) + \ln\Big(\frac{11}{6}+1\Big)\Big) + \ln(2+1)\Big]. \end{aligned}$$

Hence

$$\int_1^2 \ln(x+1)dx \approx T_6(f) = \frac{1}{12}[0.6932 + 2(4.5591) + 1.0986] = 0.9092.$$

(b) The derivatives of the function $f(x) = \ln(x+1)$ *can be obtain as*

$$f'(x) = \frac{1}{(x+1)} \quad \textit{and} \quad f''(x) = \frac{-1}{(x+1)^2}.$$

Since $\eta(x)$ *is unknown point in* $(1, 2)$, *therefore, the bound* $|f''|$ *on* $[1, 2]$ *is*

$$M = \max_{1\le x\le 2} |f''(x)| = \Big|\frac{-1}{(x+1)^2}\Big| = 0.25.$$

Thus the error formula (5.47) becomes

$$|E_{T_6}(f)| \le \frac{(1/6)^2}{12}(0.25) = 0.0006,$$

which is the possible maximum error in our approximation in part (a).
(c) The absolute error $|E|$ *in our approximation is given as*

$$|E| = |(3\ln 3 - 2\ln 2 - 1) - T_6(f)| = |0.9095 - 0.9092| = 0.0003.$$

(d) To find the minimum subintervals for the given accuracy, we use the formula (5.47) such that

$$|E_{T_n}(f)| \le \frac{|-(b-a)^3|}{12n^2} M \le 10^{-4},$$

where $h = (b-a)/n$. *Since* $M = 0.25$, *then solving for* n, *we obtain,* $n \ge 14.4338$. *Hence to get the required accuracy, we need 15 subintervals.* •

To use MATLAB command for the composite Trapezoidal rule, first we define a function m-file as fn.m for the function as follows:

$function\ y = fn(x)$
$y = \log(x+1);$
$>> T6 = TrapezoidalR('fn', 1, 2, 6)$

Program 5.1
MATLAB m-file for the Composite Trapezoidal Rule
function TN=TrapezoidR(fn,a,b,n);
h=(b-a)/n; s=(feval(fn,a)+feval(fn,b))/2;
for k=1:n-1; $x = a + h * k$ s=s+feval(fn,x); end $TN = s * h;$

Example 5.26 *Find the largest value of h that can be used of the error in the estimate of the given integral* $\int_1^2 \frac{e^{-x}}{x}\,dx$ *to an accuracy no more than* 0.5×10^{-2} *using the composite Trapezoidal rule. Then find the approximate value of the integral.*

Solution. *Given* $f(x) = \frac{e^{-x}}{x}$ *and its derivatives can be obtained as*

$$f'(x) = -e^{-x}\left[\frac{1}{x} + \frac{1}{x^2}\right] \quad \textit{and} \quad f''(x) = e^{-x}\left[\frac{1}{x} + \frac{2}{x^2} + \frac{2}{x^3}\right].$$

Since f'' *is decreasing function on* $[1, 2]$*, it takes its largest value at* $x = 1$ *in the interval. Hence*

$$|f''(x)| \le e^{-1}(1 + 2 + 2) = 5e^{-1} \approx 1.84.$$

Thus the error formula (5.47) becomes

$$|E| \le \frac{h^2}{12}(1.84), \qquad \frac{h^2}{12}(1.84) \le 0.5 \times 10^{-2}, \quad \textit{or} \quad h = 0.1806 \approx 0.2, \quad \textit{gives} \quad n = 5.$$

Thus the composite Trapezoidal rule (5.41) for six points or $n = 5$ *has the form*

$$\int_1^2 f(x)\,dx \approx T_5(f) = \frac{h}{2}\Big[f(x_0) + 2\Big[f(x_1) + f(x_2) + f(x_3) + f(x_4)\Big] + f(x_5)\Big],$$

$$\int_1^2 f(x)\,dx \approx 0.1\Big[0.3679 + 2(0.2510 + 0.1761 + 0.1262 + 0.0918) + 0.0677\Big] = 0.1726,$$

is the approximation of the exact value 0.1705 *of the given integral.* •

Suppose that, due to rounding, the f_i are in error by at most $\frac{1}{2} \times 10^{-t}$. Then we see from (5.41) that the error in the composite Trapezoidal rule due to rounding is not greater than

$$\frac{1}{2}h[1 + 2 + 2 + \cdots + 2 + 1] \times 10^{-t} = nh(\frac{1}{2} \times 10^{-t}) = \frac{1}{2}(b - a) \times 10^{-t}.$$

Thus, rounding errors do not seriously affect the accuracy of this quadrature rule. This is generally true of numerical integration unlike numerical differentiation, as we saw earlier in the chapter.

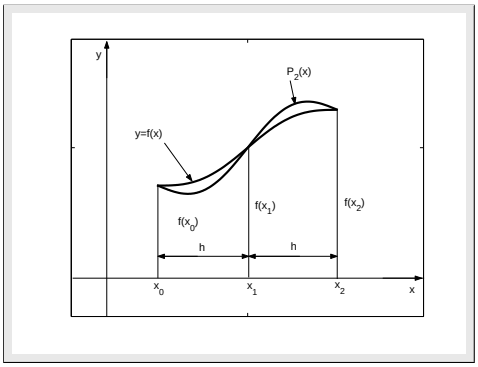


Figure 5.7: Simple Simpson's rule.

5.5.2 Simpson's Rule

The Trapezoidal rule approximates the area under a curve by the area of trapezoid formed by connecting two points on the curve by straight line. The Simpson's rule gives a more accurate approximation since it consists of connecting three points on the curve by second-degree parabola and the area under the parabola to obtain the approximate area under the curve, see Figure 5.7.

5.5.2.1 Simple Simpson's Rule

Let us consider the second-degree Lagrange interpolating polynomial, with equally spaced base points, that is, $x_0 = a, x_1 = a + h$ and $x_2 = a + 2h$, with $h = (b - a)/2$, then

$$f(x) = p_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2).$$

Taking integral on both sides of the above equation with respect to x between the limits x_0 and x_2, we have

$$\begin{aligned} \int_{x_0}^{x_2} f(x)dx &\approx \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} \int_{x_0}^{x_2} (x - x_1)(x - x_2)dx \\ &+ \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} \int_{x_0}^{x_2} (x - x_0)(x - x_2)dx \\ &+ \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \int_{x_0}^{x_2} (x - x_0)(x - x_1)dx, \end{aligned}$$

which implies that

$$\int_a^b f(x)dx \approx \frac{f(x_0)}{2h^2} I_1 + \frac{f(x_1)}{-h^2} I_2 + \frac{f(x_2)}{2h^2} I_3,$$

where

$$I_1 = \int_{x_0}^{x_2} (x - x_1)(x - x_2)dx; \quad I_2 = \int_{x_0}^{x_2} (x - x_0)(x - x_2)dx; \quad I_3 = \int_{x_0}^{x_2} (x - x_0)(x - x_1)dx.$$

Solving above three integrals by using integration by parts, we obtain the values of I_1, I_2 and I_3 as follows

$$I_1 = \frac{2h^3}{3}, \qquad I_2 = -\frac{4h^3}{3}, \qquad I_3 = \frac{2h^3}{3}.$$

By using these values, we have

$$\int_a^b f(x)dx \approx \frac{f(x_0)}{2h^2}\left(\frac{2h^3}{3}\right) + \frac{f(x_1)}{-h^2}\left(\frac{-4h^3}{3}\right) + \frac{f(x_2)}{2h^2}\left(\frac{2h^3}{3}\right).$$

Simplifying, gives

$$\int_a^b f(x)dx \approx S_2(f) = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)]. \tag{5.49}$$

which is called the *simple Simpson's rule* or Simpson's rule for two strips (or 3 points).

Example 5.27 *Approximate the following integral* $\int_1^2 \frac{1}{x+1}dx$, *using the simple Simpson's rule. Compute the actual error.*

Solution. *Since* $f(x) = \frac{1}{x+1}$ *and* $h = (2-1)/2 = 0.5$, *then by using Simpson's rule (5.49), we have*

$$\int_1^2 \frac{1}{x+1}dx \approx S_2(f) = \frac{0.5}{3}\Big[f(1) + 4f(1.5) + f(2)\Big] = (0.1667)[0.5 + 1.6 + 0.3333] = 0.4056.$$

Since the exact solution of the given integral is, $\ln 1.5 = 0.4055$, *therefore, the actual error is*

$$E_{S_2} = I(f) - S_2(f) = -0.0001.$$

To compare this error with the error got by using the simple Trapezoidal rule, the error in Simpson's rule is much smaller than for the Trapezoidal rule by a factor of about 123, *a significant increase in accuracy.* •

Example 5.28 *Let* $f(x)$ *be defined on the three numbers* $-1, 0, 1$. *Find the quadratic Lagrange polynomial and use it to shhow that*

$$\int_{-1}^{1} f(x)\,dx = \frac{1}{3}[f(-1) + 4f(0) + f(1)].$$

If $f(x) = x^2$, *then find the value of the above integral.*

Solution. *Using the quadratic Lagrange interpolation formula*

$$f(x) = p_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2),$$

at these three distinct points $x_0 = -1, x_1 = 0$, *and* $x_2 = 1$, *we get*

$$f(x) = p_2(x) = \frac{(x^2-x)}{2}f(-1) + \frac{(x^2-1)}{-1}f(0) + \frac{(x^2+x)}{2}f(1).$$

Separating the coefficients of x^2, x, *and constant term, we get*

$$f(x) = p_2(x) = \left(\frac{f(-1)}{2} + \frac{f(0)}{-1} + \frac{f(1)}{2}\right)x^2 + \left(\frac{f(1)}{2} - \frac{f(-1)}{2}\right)x + \left(\frac{-f(0)}{-1}\right),$$

after simplifying, we obtain

$$f(x) = p(x) = \frac{1}{2}[f(-1) - 2f(0) + f(1)]x^2 + \frac{1}{2}[f(1) - f(-1)]x + f(0),$$

which is the required quadratic Lagrange polynomial. Now integrating the above quadratic polynomial between -1 *and* 1, *gives*

$$\int_{-1}^{1} f(x)\,dx = \frac{1}{2}f(-1)\int_{-1}^{1}(x^2 - x)\,dx - f(0)\int_{-1}^{1}(x^2 - 1)\,dx + \frac{1}{2}f(1)\int_{-1}^{1}(x^2 + x)\,dx.$$

By evaluating the above integrals, we get

$$\int_{-1}^{1} f(x)\,dx = \frac{1}{2}f(-1)\frac{2}{3} - f(0)\frac{-4}{3} + \frac{1}{2}f(1)\frac{2}{3},$$

or

$$\int_{-1}^{1} f(x)\,dx = \frac{1}{3}[f(-1) + 4f(0) + f(1)].$$

Taking $f(x) = x^2$, *the above equation gives*

$$\int_{-1}^{1} x^2\,dx = \frac{1}{3}[(-1)^2 + 4(0)^2 + (1)^2] = \frac{2}{3},$$

the required solution. •

5.5.2.2 Error Terms for Simpson's Rule

Now we discuss the local error and the global error formulas for Simpson's rule.

Theorem 5.4 (Error Term for Simple Simpson's Rule)

Let $f \in C^4[a, b]$, *and* $h = (b - a)/2$. *The local error that the Simpson's rule makes in estimating the definite integral (5.33) is*

$$E_{S_2}(f) = -\frac{h^5}{90}f^{(4)}(\eta(x)), \quad \textit{where} \quad \eta(x) \in (a, b). \tag{5.50}$$

Proof. Consider three equally spaced points $a = x_0 < x_1 < x_2 = b$. From the quadratic Lagrange interpolation formula with error term, we have

$$f(x) = p_2(x) + \frac{f'''(\eta(x))}{3!}\prod_{i=0}^{2}(x - x_i), \tag{5.51}$$

and by integrating (5.51) with respect to x, we have

$$\int_a^b f(x)dx = \int_a^b p_2(x)dx + \frac{1}{6}\int_a^b f'''(\eta(x)(x))\prod_{i=0}^{2}(x - x_i)dx. \tag{5.52}$$

The second term on the right hand side of the (5.52)

$$E_{S_2}(f) = \frac{1}{6}\int_a^b f'''(\eta(x)(x))(x-x_0)(x-x_1)(x-x_2)dx, \tag{5.53}$$

is called the error term of the Simpson's rule for $n = 2$.
In this way it provides only an $O(h^4)$ error term, involving $f'''(\eta(x))$. By approaching the problem in another way, a higher-order term involving $f^{(4)}(\eta(x))$ can be obtained.
Consider the two intervals from x_{i-1} to x_i and x_i to x_{i+1}, with $h = (x_i - x_{i-1})$, and also, assume that $F(x)$ is the indefinite integral of a function $f(x)$ which we are trying to integrate, then the exact value of the integral from x_{i-1} to x_{i+1} is

$$I_i(f) = \int_{x_{i-1}}^{x_{i+1}} f(x)dx = F(x_{i+1}) - F(x_{i-1}). \tag{5.54}$$

The approximated value calculated by using the Simpson's rule is

$$S_{2i}(f) = \frac{h}{3}[f(x_{i-1}) + 4f(x_i) + f(x_{i+1})]. \tag{5.55}$$

Then the error define in using the Simpson's rule on this two intervals is

$$E_i(f) = I_i(f) - S_{2i}(f) \tag{5.56}$$

Now by expanding $f(x)$ about $x = x_i$ to get $f(x_{i-1})$ in terms of a function and derivatives at $x = x_i$ by using the Taylor's series, we have

$$\begin{aligned} f(x_{i-1}) &= f(x_i) + (x_{i-1}-x_i)f'(x_i) + \frac{(x_{i-1}-x_i)^2}{2!}f''(x_i) \\ &+ \frac{(x_{i-1}-x_i)^3}{3!}f'''(x_i) + \frac{(x_{i-1}-x_i)^4}{4!}f^{(4)}(x_i) + \cdots \end{aligned} \tag{5.57}$$

Since we know that $h = (x_i - x_{i-1})$, or $-h = (x_{i-1} - x_i)$, therefore

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{h^2}{2!}f''(x_i) - \frac{h^3}{3!}f'''(x_i) + \frac{h^4}{4!}f^{(4)}(x_i) - \cdots \tag{5.58}$$

Similar way we expand $f(x_{i+1})$ about x_i, we get

$$\begin{aligned} f(x_{i+1}) &= f(x_i) + (x_{i+1}-x_i)f'(x_i) + \frac{(x_{i+1}-x_i)^2}{2!}f''(x_i) \\ &+ \frac{(x_{i+1}-x_i)^3}{3!}f'''(x_i) + \frac{(x_{i+1}-x_i)^4}{4!}f^{(4)}(x_i) + \cdots, \end{aligned} \tag{5.59}$$

or

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2!}f''(x_i) + \frac{h^3}{3!}f'''(x_i) + \frac{h^4}{4!}f^{(4)}(x_i) + \cdots \tag{5.60}$$

So by using (5.58) and (5.60), we get (5.55) of the form

$$\begin{aligned} S_{2i}(f) &= \frac{h}{3}\Big[\{f(x_i) - hf'(x_i) + \frac{h^2}{2!}f''(x_i) - \frac{h^3}{3!}f'''(x_i) + \frac{h^4}{4!}f^{(4)}(x_i) - \cdots\} + \{4f(x_i)\} \\ &+ \{f(x_i) + hf'(x_i) + \frac{h^2}{2!}f''(x_i) + \frac{h^3}{3!}f'''(x_i) + \frac{h^4}{4!}f^{(4)}(x_i) + \cdots\}\Big], \end{aligned}$$

which implies that

$$\begin{aligned} S_{2i}(f) &= \frac{h}{3}\left[6f(x_i) + \frac{2h^2}{2!}f''(x_i) + \frac{2h^4}{4!}f^{(4)}(x_i) + \cdots\right] \\ &= 2hf(x_i) + 2f''(x_i)\frac{h^3}{3!} + 2f^{(4)}(x_i)\frac{h^5}{3(4!)} + \cdots \end{aligned} \tag{5.61}$$

Now we use the same procedure for $F(x)$ as we used for $f(x)$ to get

$$F(x_{i-1}) = F(x_i) - hF'(x_i) + \frac{h^2}{2!}F''(x_i) - \frac{h^3}{3!}F'''(x_i) + \frac{h^4}{4!}F^{(4)}(x_i) - \frac{h^5}{5!}F^{(5)}(x_i) + \cdots, \tag{5.62}$$

and

$$F(x_{i+1}) = F(x_i) + hF'(x_i) + \frac{h^2}{2!}F''(x_i) + \frac{h^3}{3!}F'''(x_i) + \frac{h^4}{4!}F^{(4)}(x_i) + \frac{h^5}{5!}F^{(5)}(x_i) + \cdots \tag{5.63}$$

Then by using (5.62) and (5.63), we got (5.54) of the form

$$I_i(f) = 2hF'(x_i) + 2F'''(x_i)\frac{h^3}{3!} + 2F^{(5)}(x_i)\frac{h^5}{5!} + \cdots \tag{5.64}$$

But we know that

$$F'(x) = f(x),\ F''(x) = f'(x),\ F'''(x) = f''(x),\ F^{(4)}(x) = f'''(x),\ F^{(5)}(x) = f^{(4)}(x),$$

therefore, (5.64) can be written as

$$I_i(f) = 2hf(x_i) + 2f''(x_i)\frac{h^3}{3!} + 2f^{(4)}(x_i)\frac{h^5}{5!} + \cdots \tag{5.65}$$

So by using (5.61) and (5.65), we get (5.56) of the form

$$\begin{aligned} E_i(f) &= -\left[2hf(x_i) + 2f''(x_i)\frac{h^3}{3!} + 2f^{(4)}(x_i)\frac{h^5}{3(4!)} + \cdots\right] \\ &+ \left[2hf(x_i) + 2f''(x_i)\frac{h^3}{3!} + 2f^{(4)}(x_i)\frac{h^5}{5!} + \cdots\right] \\ &= -2f^{(4)}(x_i)\frac{h^5}{3(4!)} + 2f^{(4)}(x_i)\frac{h^5}{5!} + \text{higher terms in } h^7 \ + \cdots \end{aligned} \tag{5.66}$$

Assuming that h is small, we may neglect the terms in h^7 and above, and get the approximate error as follows:

$$E_i(f) \approx f^{(4)}(x_i)h^5\left[\frac{2}{5!} - \frac{2}{3(4!)}\right] \approx -\frac{1}{90}f^{(4)}(x_i)h^5, \tag{5.67}$$

which is the desired local error for Simpson's rule. •

Since (5.67) indicates that the error of the Simpson's rule is proportional to fourth derivative $f^{(4)}$. If the Simpson's rule is used to integrate $f(x) = 1, x, x^2$ and x^3, then the results have no error.
In more general terms, the Newton-Cotes closed formula of odd order n is exact if the integrand is a polynomial of order n or less, whereas that of an even n is exact when the integrand is a polynomial of order $n + 1$ or less.

5.5.2.3 Error Bound for Simple Simpson's Rule

If $f(x) \geq 0$ on $[a, b]$ and $h = \dfrac{(b-a)}{2}$, then

$$|E_{S_2}(f)| \leq \frac{h^5}{90} M, \quad M = \max_{a \leq x \leq b} |f^{(4))}(x)|.$$

Example 5.29 *Find the error bound for the Simpson's rule using the integral* $\displaystyle\int_1^2 \frac{1}{x+1} dx$.

Solution. *Given* $f(x) = \dfrac{1}{x+1}$ *and* $[a, b] = [1, 2]$, *then the fourth derivative of the function can be obtain as*

$$f'(x) = \frac{-1}{(x+1)^2}, \quad f''(x) = \frac{2}{(x+1)^3}, \quad f'''(x) = \frac{-6}{(x+1)^4}, \quad f^{(4)}(x) = \frac{24}{(x+1)^5}.$$

Since the error formula for the Simpson's rule can be written as

$$|E_{S_2}(f)| = \Big| - \frac{h^5}{90} \Big| \Big| f^{(4)}(\eta(x)) \Big|, \qquad for \qquad \eta(x) \in (1, 2),$$

which cannot be computed exactly because $\eta(x)$ *is not known. But one can bound the error by computing the largest possible value for* $|f^{(4)}|$. *Bound* $|f^{(4)}|$ *on* $[1, 2]$ *is*

$$|f^{(4)}(\eta(x))| \leq M = \max_{1 \leq x \leq 2} = \Big| \frac{24}{(x+1)^5} \Big| = 0.75.$$

$$|E_{S_2}(f)| \leq \frac{h^5}{90} M = \frac{(0.5)^5}{90}(0.75) = 0.0003.$$

Comparing this with the absolute error 0.0001, *this bound is about* 3 *times the absolute error.* •

Example 5.30 *(a) Find approximation of* $\displaystyle\int_1^2 f(x)\, dx$, *using best integration rule:*

x	1.0	1.1	1.2	1.3	1.4	1.5	1.7	1.8	1.9	2.0
$f(x)$	0.368	0.366	0.361	0.354	0.355	0.335	0.311	0.298	0.284	0.271

(b) The function tabulated is $f(x) = xe^{-x}$, *compute an error bound for the approximation using integration rule used in part (a).*
(c) How many subintervals approximate the given integral to an accuracy of at least 10^{-6} *?*

Solution. *(a) Using equally spaced points* $x_0 = 1.0, x_1 = 1.5, x_2 = 2.0$, *gives* $h = 0.5$, *that is, we select the following table:*

x	1.0	1.5	2.0
$f(x)$	0.368	0.335	0.271

So the Simple Simpson's rule for three points can be used and it written as

$$\int_1^2 f(x)\, dx \approx S_2(f) = \frac{h}{3}\Big[f(x_0) + 4f(x_1) + f(x_2)\Big],$$

$$\int_1^2 f(x)\,dx \approx 0.1667[0.368 + 4(0.335) + 0.271] = 0.3299.$$

(b) The derivatives of the function $f(x) = xe^{-x}$ *can be obtain as*

$$f'(x) = (1-x)e^{-x}, \quad f''(x) = (x-2)e^{-x}, \quad f'''(x) = (3-x)e^{-x}, \quad f^{(4)}(x) = (x-4)e^{-x}.$$

Since $\eta(x)$ *is unknown point in* $(1, 2)$*, therefore, the bound* $|f^{(4)}|$ *on* $[1, 2]$ *is*

$$M = \max_{1\le x\le 2} |f^{(4)}(x)| = \max_{1\le x\le 2} |(x-2)e^{-x}| = 1.1036,$$

at $x = 1$*. Thus the error formula becomes*

$$|E_{S_2}(f)| \le \frac{(0.5)^5}{90}(1.1036) = 3.8319e-004,$$

which is the possible maximum error in our approximation.
(c) To find the minimum subintervals for the given accuracy, we use the error bound formula for Simple Simpson's rule such that

$$|E_{S_2}(f)| \le \frac{|-h^5|}{90} M \le 10^{-6},$$

or

$$n \ge \left(\frac{M \times 10^6}{90}\right)^{1/5},$$

where $h = (b-a)/n$*. Since* $M = 1.1036$*, then solving for* n*, we obtain,*

$$n \ge 6.5722.$$

Hence to get the required accuracy, we need 8 *subintervals or* 9 *points.* •

5.5.2.4 Composite Simpson's Rule

Just as with the simple Trapezoidal rule (5.40), the simple Simpson's rule (5.49) can be improved by dividing the integration interval $[a, b]$ into a number of subintervals of equal width h; where $h = \dfrac{b-a}{n}$. Since the simple Simpson's rule (5.49) requires a interval consisting of three points (pair of strips). In practice, we usually take more than three points and add the separate results for the different pairs of strips (see Figure 5.8). Since the simple Simpson's rule requires a pair of strips for application, so there is restriction on the integer n, which must be even. We define the *composite Simpson's rule* in the form of the following theorem.

Theorem 5.5 (Composite Simpson's Rule)

Let $f \in C^4[a, b]$*,* n *be even,* $h = (b-a)/n$*, and* $x_i = a + ih$ *for each* $i = 0, 1, 2, \ldots, n$*. Then the composite Simpson's rule for* n *subintervals can be written as*

$$\int_a^b f(x)dx \approx S_n(f) = \frac{h}{3}\left[f(a) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + f(b)\right]. \tag{5.68}$$

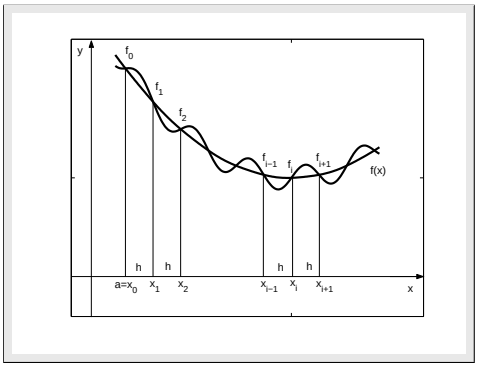


Figure 5.8: Composite Simpson's Rule.

Proof. *Since for the composite form of the Simpson's rule, the interval is divided into n equal subintervals of width h so that* $h = \dfrac{b-a}{n}$. *For this rule to work, n must be even number and the total number of (n+1) distinct points* $a = x_0 < x_1 < x_2 \ldots < x_n = b$ *should be odd. The total integral can be represented as*

$$\int_{x_0}^{x_n} f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)dx.$$

Substitute the simple Simpson's rule (5.49) for the individual integral yields

$$\begin{aligned}\int_{x_0}^{x_n} f(x)dx &\approx \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] + \frac{h}{3}[f(x_2) + 4f(x_3) + f(x_4)] \\ &+ \cdots + \frac{h}{3}[f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)].\end{aligned}$$

To avoid the repetition of terms, we summed them. Note that each of the odd interior point is counted four and therefore has a coefficient of four whereas each of the even interior point is counted two and therefore has a coefficient of two. The endpoints are counted once and therefore has a coefficient one. •

Example 5.31 *Let f be defined by*

$$f(x) = \begin{cases} x^2 - x + 1, & \text{if } 0 \le x \le 1, \\ 2x - 1, & \text{if } 1 \le x \le 2. \end{cases}$$

Approximate the integral $\int_0^2 f(x)dx$ *by using Simpson's rule with* $n = 4$.

Solution. *Since one can know that*

$$\int_0^2 f(x)dx = \int_0^1 f(x)dx + \int_1^2 f(x)dx = \int_0^1 (x^2 - x + 1)dx + \int_1^2 (2x - 1)dx = I_1(f) + I_2(f).$$

First we find the approximation of the first integral on the right hand side of above equation for $n = 4$, using the formula (5.68) and $h = 0.25$, we have

$$I_1(f) \approx \frac{0.25}{3}\Big[f(0) + 4(f(0.25) + f(0.75)) + 2f(0.5) + f(1)\Big] \approx \frac{0.25}{3}\Big[8\Big] = 0.8333.$$

Now we find the approximation of the second integral on the right hand side of above equation for $n = 4$, using the formula (5.68) and $h = 0.25$, we have

$$I_2(f) \approx \frac{0.25}{3}\Big[f(1) + 4(f(1.25) + f(1.75)) + 2f(1.5) + f(2)\Big] \approx \frac{0.25}{3}\Big[24\Big] = 2.0000.$$

Hence

$$\int_0^2 f(x)dx = I_1(f) + I_2(f) \approx 0.8333 + 2.000 = 2.83333,$$

the required approximation of the given integral. •

Example 5.32 *Suppose that* $f(1) = 0.5,\ f(1.2) = 0.9,\ [f(1.25) + f(1.75)] = \alpha,\quad f(1.5) = 1.5,$ $f(1.6) = 1.65, f(1.95) = 1.95$ *and* $f(2) = 2$. *Find the approximate value of the α if the composite Simpson's rule gives the value,* 1.35, *for the integral* $\int_1^2 f(x)\ dx$.

Solution. *Since we need the equally spaced data points, so we can take* $x_0 = 1, x_1 = 1.25, x_2 = 1.5, x_3 = 1.75$ *and* $x_4 = 2$, *gives* $n = 4$, *so* $h = \dfrac{2-1}{4} = 0.25$. *By using the composite formula (5.68) for* $n = 4$, *we have*

$$\int_1^2 f(x)\ dx \approx \frac{0.25}{3}\Big[f(1) + 4[f(1.25) + f(1.75)] + 2f(1.5) + f(2)\Big].$$

Now using the given values, we obtain

$$1.35 \approx \frac{1}{12}[0.5 + 4(\alpha) + 2(1.5) + 2], \quad or \quad 12(1.35) - 5.5 \approx 4\alpha,$$

and solving for α, gives $\alpha \approx 2.675$. •

Example 5.33 *Evaluate the integral* $\int_0^1 e^{4x}dx$ *by using the Simpson's rule with* $n = 2, 4, 8$. *Also, compute the corresponding actual errors.*

Solution. *For n = 2, using the formula (5.49) and $h = 0.5$, we have*

$$S_2(f) = \frac{0.5}{3}\Big[f(0) + 4f(0.5) + f(1)\Big] = 14.1924.$$

For n = 4, using the formula (5.68) and $h = 0.25$, we have

$$S_4(f) = \frac{0.25}{3}\Big[f(0) + 4[f(0.25) + f(0.75)] + 2f(0.5) + f(1)\Big] = 13.4659.$$

For $n = 8$, using the formula (5.68) and $h = 0.125$, we have

$$S_8(f) = \frac{0.125}{3}\Big[f(0) \quad + \quad 4[f(0.125) + f(0.375) + f(0.625) + f(0.875)] \\ + \quad 2[f(0.25) + f(0.5) + f(0.75)] + f(1)\Big] = 13.4041.$$

Note that the exact value of the given integral is 13.39995*, and so the corresponding errors are,* $0.79245, 0.06595$ *and* 0.00411 *respectively, which decrease by a factor of about 16 at each stage.* •

To use the MATLAB command for the composite Simpson's rule, first we define a function m-file as fn.m for the function as follows:

$function\ y = fn(x)$
$y = exp(4 * x);$
$>> S2 = SimpsonR('fn', 0, 1, 2)$

Program 5.2
MATLAB m-file for the Composite Simpson's Rule
function SN=SimpsonR(fn,a,b,n);
h=(b-a)/n; s=feval(fn,a)+feval(fn,b);
for k=1:2:n-1; $s = s + 4 * feval(fn, a + k * h)$; end
for k=2:2:n-2; $s = s+2*feval(fn, a+k*h)$; end; $SN = (s*h)/3$;

One can note that these results are more accurate than those obtained in the previous Example 5.20 using the Trapezoidal rule (for same number of function evaluations). The main disadvantage of the Simpson's rule is that it can only be used when the given interval $[a, b]$ is divided in to an even number of subintervals.

5.5.2.5 Error Term for Composite Simpson's Rule

Since the composite Simpson's rule (5.68) requires that the given interval $[a, b]$ is divided into even number of subintervals and each application of the simple Simpson's rule requires two subintervals, therefore, the global error of the composite Simpson's rule (5.68) is the sum of $\frac{n}{2}$ local truncation error of the simple Simpson's rule with $n = \dfrac{b-a}{h}$, that is,

$$E_{S_n}(f) = -\frac{h^5}{90}f^{(4)}(\eta_1(x)) - \frac{h^5}{90}f^{(4)}(\eta_2(x)) - \cdots - \frac{h^5}{90}f^{(4)}(\eta_{n/2}(x)) = -\frac{h^5}{90}(\frac{n}{2})\left[\frac{\sum_{i=1}^{n/2} f^{(4)}(\eta_i(x))}{n/2}\right].$$

Thus by using the Intermediate Value Theorem, we have

$$E_{S_n}(f) = -\frac{(b-a)}{180}h^4 f^{(4)}(\eta(x)), \qquad \eta(x) \in (a, b), \quad nh = b - a. \tag{5.69}$$

Then the formula (5.69) is known as the *global error* of the Simpson's rule. •

Note that the truncation error in the composite Simpson's rule is of order h^4. This means that when h is halved and the number of subintervals is doubled the error decreases by a factor of approximately 16, considerably better than the composite Trapezoidal rule.

5.5.2.6 Error Bound for Composite Simpson's Rule

If $f(x) \geq 0$ on $[a, b]$ and $h = \dfrac{(b-a)}{n}$, then

$$|E_{S_n}(f)| \leq \frac{h^4(b-a)}{180}M, \quad M = \max_{a\leq x\leq b}|f^{(4))}(x)|.$$

Example 5.34 *Approximate* $\displaystyle\int_0^1 e^{-x^2}dx$, *with an error less than* 10^{-3} *using composite Simpson's rule.*

Solution. *Since* $f(x) = e^{-x^2}$, *and the fourth derivative of* $f(x)$ *is* $(16x^4 - 48x^2 + 12)e^{-x^2}$, *we have to use the error formula (5.69) which is*

$$|E_{S_n}(f)| \leq \frac{(b-a)}{180}h^4M \leq 10^{-3}.$$

The maximum value of $|f^{(4)}(x)|$ *on the interval* $[0, 1]$ *is* 12, *and thus* $M = 12$. *Using the above error formula, we get*

$$\frac{12}{180}h^4 = \frac{12}{180n^4} \leq 10^{-3}, \quad \textit{gives} \quad n^4 \geq \frac{200}{3}, \quad n \geq \sqrt[4]{\frac{200}{3}} \approx 2.86.$$

Since even subintervals n *required is* $n = 4$, *then the error estimate is* $\dfrac{12}{180(4)^4} \leq 0.0003$.

Thus the approximation of the given integral using $h = \dfrac{1-0}{4} = \dfrac{1}{4}$ *is*

$$\int_0^1 e^{-x^2}dx \approx \frac{1/4}{3}\Big[f(0) + 4[f(1/4) + f(3/4)] + 2f(1/2) + f(1)\Big],$$

$$\int_0^1 e^{-x^2}dx \approx \frac{1}{12}\Big[1.0000 + 4(0.9394 + 0.5698) + 2(0.7788) + 0.3679\Big] = 0.746855.$$

Therefore, the true value of the given integral is lying between $0.746855 - 0.0003 = 0.746555$ *and* $0.746855 + 0.0003 = 0.7471555$, *both of which round to* 0.75. •

Example 5.35 *(a) Find the approximation of* $\displaystyle\int_0^{1.2} f(x)\,dx$ *taking* $h = 0.3$ *by using the following set of data points using Simpson's rule:*

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2
$f(x)$	1.00	1.10	1.18	1.26	1.32	1.38	1.43	1.47	1.50	1.52	1.54	1.55	1.56

(b) The function tabulated is $f(x) = x + \cos x$, compute an error bound and the absolute error for the approximation by Simpson's rule.
(c) How many subintervals approximate the given integral to within accuracy of 10^{-6} ?

Solution. *(a) Given $h = 0.3$, so to select the following set of data points for Simpson's rule as*

x	0.0	0.3	0.6	0.9	1.2
$f(x)$	1.00	1.26	1.43	1.52	1.56

so the composite Simpson's rule (5.68) for five points can be written as

$$\int_0^{1.2} f(x)\, dx \approx S_4(f) = \frac{h}{3}\Big[f(x_0) + 4(f(x_1) + f(x_3)) + 2f(x_2) + f(x_4)\Big],$$

$$\int_0^{1.2} f(x)\, dx \approx 0.1\Big[1.0 + 4(1.26 + 1.52) + 2(1.43) + 1.56\Big] = 1.6540.$$

(b) The fourth derivative of the function $f(x) = x + \cos x$ can be obtain as

$$f'(x) = 1 - \sin x, \quad f''(x) = -\cos x, \quad f'''(x) = \sin x, \quad f^{(4)}(x) = \cos x.$$

Since $\eta(x)$ is unknown point in $(0, 1.2)$, therefore, the bound $|f^{(4)}|$ on $[0, 1.2]$ is

$$M = \max_{0\le x\le 1.2} |f^{(4)}| = \max_{0\le x\le 1.2} |\cos x| = 1.0,$$

at $x = 0$. Thus the error formula (5.69) becomes

$$|E_{S_4}(f)| \le \frac{(0.3)^4(1.2)}{180}(1.0) = 0.000054.$$

We can easily computed the exact value of the given integral as

$$\int_0^{1.2} (x + \cos x)\, dx = (x^2/2 + \sin x)\Big|_0^{1.2} = 1.6520.$$

Thus the absolute error $|E|$ in our approximation is given as

$$|E| = |0.3298 - S_4(f)| = |1.6520 - 1.6540| = 0.0020.$$

(c) To find the minimum subintervals for the given accuracy, we use the formula (5.69) such that

$$|E_{S_n}(f)| \le \frac{|-(b-a)^5|}{180n^4} M \le 10^{-6},$$

where $h = (b - a)/n$. Since $M = 1$, then solving for n, we obtain, $n \ge 10.8432$. Hence to get the required accuracy, we need 12 (even) subintervals or 13 points. •

Example 5.36 *Consider the integral* $I(f) = \int_1^2 \ln(x + 1)dx; \qquad n = 6.$
(a) Find the approximation of the give integral using the composite Simpson's rule.
(b) Compute the error bound for the approximation using the formula (5.69).

(c) Compute the absolute error.

(d) How many subintervals approximate the given integral to an accuracy of at least 10^{-4} *using the composite Simpson's rule ?*

Solution. *(a) As* $f(x) = \ln(x+1), n = 6$, *and so* $h = \dfrac{1}{6}$, *the rule (5.68) becomes*

$$\begin{aligned} S_6(f) &= \frac{1/6}{3}\Big[\ln(1+1) + 4(\ln\Big(\frac{7}{6}+1\Big) + \ln\Big(\frac{9}{6}+1\Big) + \ln\Big(\frac{11}{6}+1\Big)\Big)\Big] \\ &+ \Big[2\Big(\ln\Big(\frac{8}{6}+1\Big) + \ln\Big(\frac{10}{6}+1\Big)\Big) + \ln(2+1)\Big]. \end{aligned}$$

$$\int_1^2 \ln(x+1)dx \approx S_6(f) = \frac{1}{18}\Big[0.6932 + 4(2.7309) + 2(1.8281) + 1.0986\Big] = 0.9095.$$

(b) Since the fourth derivative of the function is, $f^{(4)}(x) = \dfrac{-6}{(x+1)^4}$. *Since* $\eta(x)$ *is unknown point in* $(1,2)$, *therefore, the bound* $|f^{(4)}|$ *on* $[1,2]$ *is*

$$M = \max_{1\le x\le 2}|f^{(4)}(x)| = \Big|\frac{-6}{(x+1)^4}\Big| = 6/16 = 0.375.$$

Thus the error formula (5.69) becomes

$$|E_{T_6}(f)| \le \frac{(1/6)^4}{180}(0.375) = 0.000002.$$

(c) The absolute error $|E|$ *in our approximation is given as*

$$|E| = |(3\ln 3 - 2\ln 2 - 1) - S_6(f)| = 0.0000003.$$

(d) To find the minimum subintervals for the given accuracy, we use the error formula (5.69)

$$|E_{S_n}(f)| \le \frac{(b-a)^5}{180n^4}M \le 10^{-4}.$$

Since we know $M = 0.375$, *then we have,* $n \ge 2.136435032$. *Hence to get the required accuracy, we need* 4 *subintervals (because* n *should be even) that ensures the stipulated accuracy.* •

To find the approximate value of the given integral within the given accuracy, we use MATLAB command which gives us the approximate solution and the number of subintervals n. First we define m-file as fn.m for the function, so after finding the value of M, it simply compute n using (5.69) and then calls the previously defined SimpsonR function, we have the results:

```
function y = fn(x)
y = log(x + 1);
>> k = ErrorSR('fn', 0, 1, 0.375, 1e - 4)
```

Program 5.3
MATLAB m-file for computing Error term of the
Composite Simpson's Rule
$L = abs(b-a); n = ceil(L * sqrt(sqrt(L * M/180/eps)));$
if $mod(n,2) == 1; n = n+1;$ end; k=SimpsonR(fn,a,b,n);

Example 5.37 *Determine the number of subintervals n required to approximate the integral* $\int_0^2 \frac{1}{x+4}dx$, *with an error less than* 10^{-4} *using composite Simpson's rule. Then approximate the given integral.*

Solution. *We have to use the error formula (5.69) which is*

$$|E_{S_n}(f)| \leq \frac{(b-a)}{180}h^4 M \leq 10^{-4}.$$

Given the integrand is $f(x) = \frac{1}{x+4}$, *and we have* $f^{(4)}(x) = \frac{24}{(x+4)^5}$. *The maximum value of* $|f^{(4)}(x)|$ *on the interval* $[0,2]$ *is* $3/128$, *and thus* $M = \frac{3}{128}$. *Using the above error formula, we get*

$$\frac{3}{(90\times 128)}h^4 \leq 10^{-4}, \quad or \quad h \leq \frac{2}{5}\sqrt[4]{15} = 0.7872.$$

Since $n = \frac{2}{h} = \frac{2}{0.7872} = 2.5407$, *so the number of even subintervals n required is* $n \geq 4$. *Thus the approximation of the given integral using* $h = \frac{2-0}{4} = \frac{1}{2} = 0.5$ *is*

$$\int_0^2 \frac{1}{x+4} \approx \frac{0.5}{3}\Big[f(0) + 4[f(0.5)+f(1.5)] + 2f(1) + f(2)\Big],$$

$$\int_0^2 \frac{1}{x+4} \approx \frac{1}{6}\Big[0.25 + 4(0.2222+0.1818) + 2(0.2) + 0.1667\Big] = 0.4055,$$

which is equal to the true value of the given integral $\alpha = \ln(1.5) = 0.4055$ *up to 4 decimal places.* •

Example 5.38 *Use the best integration rule to compute the integral* $\int_{0.05}^{1} f(x)dx$, *where the table for the values of* $y = f(x)$ *is given below:*

x	0.05	0.1	0.15	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$f(x)$	0.0785	0.1564	0.2334	0.3090	0.4540	0.5878	0.7071	0.8090	0.8910	0.9511	0.9877	1.0

Solution. *Note that here the points are not given to be equidistant, so as such we can not use any of the above two formulae. However, we notice that the tabular points* $0.05, 0.10, 0.15$ *and* 0.2 *are equidistant and so are the tabular points* $0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$ *and* 1.0. *Now we can divide the interval in two subinterval:* $[0.05, 0.2]$ *and* $[0.2, 1.0]$*; thus,*

$$\int_{0.05}^{1} f(x)dx = \int_{0.05}^{0.2} f(x)dx + \int_{0.2}^{1} f(x)dx.$$

The integrals then can be evaluated in each interval. We observe that the second set has odd number of points. Thus, the first integral is evaluated by using Trapezoidal rule and the second one by Simpson's rule (of course, one could have used Trapezoidal rule in both the subintervals). For the first integral $h = 0.05$ *and for the second one* $h = 0.1$.
Thus, using Trapezoidal rule for the interval $[0.05, 0.2]$, *we get*

$$\int_{0.05}^{0.2} f(x)dx \approx \frac{0.05}{2}\left[0.0785 + 2(0.1564 + 0.2334) + 0.3090\right] = 0.0291775,$$

and using Simpson's rule for the interval $[0.2, 1]$, *we get*

$$\begin{aligned}\int_{0.2}^{1.0} f(x)dx &\approx \frac{0.1}{3}[0.3090 + 4(0.4540 + 0.7071 + 0.8910 + 0.9877) \\ &+ 2(0.5878 + 0.8090 + 0.9511) + 1.0000] = 0.6054667,\end{aligned}$$

which gives,

$$\int_{0.05}^{1} f(x)dx = \int_{0.05}^{0.2} f(x)dx + \int_{0.2}^{1} f(x)dx \approx 0.0291775 + 0.6054667 = 0.6346442.$$

In the above integral, $f(x) = \sin(\pi x/2)$ *and that the true value of the integral is* 0.6346526. •

5.6 Exercises

1. Let $f(x) = (x-1)e^x$ and take $h = 0.01$. Calculate approximation to $f'(2.3)$ using the two-point forward-difference formula and compute the actual error and an error bound for you approximation.

2. Solve the Problem 1 for the $f(x) = (x^2 + x + 1)e^{2x}$ with $h = 0.05$.

3. By using the following data: $(1.2, 11.6), (1.29, 13.8), (1.3, 14), (1.31, 14.3), (1.4, 16.8)$. Compute the best approximations of $f'(1.3)$ using the two-point forward-difference formula.

4. Let $f(x) = \sin(x+1),\ h = 0.1$. Compute the approximation of $f'(\frac{\pi}{4})$ using the two-point formula. Compute actual error and error bound.

5. Use the three-point central-difference formula to compute the approximate value for $f'(5)$ with $f(x) = (x^2+1)\ln x$ and $h = 0.05$. Compute the actual error and the error bound for you approximation.

6. Use the three-point central-difference formula to compute the approximate value for $f'(2)$ with $f(x) = e^{x/2} + 2\cos x$, and $h = 0.01$. Compute actual error and error bound.

7. Solve the Problem 3 to find the best approximation of $f'(1.3)$ using the three-point forward-difference and backward-difference formulas.

8. By using the following data: $(1.0, 2.0), (1.5, 1.94), (2.0, 2.25), (3.0, 3.11)$. Find the best approximate values for $f'(1.5), f'(1.0)$, and $f'(3.0)$ using suitable three-point formulas.

9. Use all three-point formulas to compute the approximate value for $f'(2)$ for the derivative of $f(x) = e^{x/2} + x^3$, taking $h = 0.1$. Also, compute the actual errors and error bounds for your approximation.

10. Use all three-point formulas to compute the approximate value for $f'(2.2)$ for the derivative of $f(x) = x^2e^x - x + 1$, taking $h = 0.2$. Compute actual errors and error bounds.

11. Use the most accurate three-point formula to determine approximations that will complete the following table.

x	f(x)	$f'(x)$
8.1	16.94410	
8.3	17.56492	
8.5	18.19056	
8.7	18.82091	

12. The data in the Problem 11 were taken from the function $f(x) = x\ln x$. Compute the actual errors and error bounds using the most accurate three-point formulas.

13. Let $f(x) = x + \ln(x+2)$, with $h = 0.1$. Use the three-point formula to approximate $f'(2)$. Find error bound for your approximation and compare the actual error to the bound.

14. Let $f(x) = e^{-2x}$, with points $x = 0.25, 0.5, 0.75, 1.25, 1.50$. Use the three-point central-difference formula to approximate $f'(1.0)$. Compute error bound for your approximation.

15. Approximate the integral $\int_0^2 x^2 e^{-x^2} dx$, using Trapezoidal rule with $n = 4$, and $n = 6$.

16. Approximate the integral $\int_0^2 x^2 e^{-x^2} dx$, using Simpson's rule with $n = 4$, and $n = 6$.

17. The following values of a function $f(x) = \tan x/x$ are given

x	f(x)	x	f(x)
1.00	1.5574	1.40	4.1342
1.10	1.7862	1.50	9.4009
1.20	2.1435	1.60	-21.3953
1.30	2.7709		

Find $\int_{1.0}^{1.6} f(x)dx$, using the Trapezoidal rule with $h = 0.1$.

18. Use a suitable composite integration formula to approximate the integral $\int_0^1 \frac{dx}{2e^x - 1}$, with $n = 5$.

19. Use a suitable composite integration formula for the approximation of the integral $\int_1^2 \frac{dx}{3 - x}$, with $n = 5$. Compute an upper bound for your approximation.

20. Use the composite Trapezoidal rule for the approximation of the integral $\int_1^3 \frac{dx}{7 - 2x}$ with $h = 0.5$. Also, compute an error term.

21. Find the stepsize h so that the absolute value of the error for the composite Trapezoidal rule is less than 5×10^{-4} when it is used to approximate the integral $\int_2^7 \frac{dx}{x}$.

Chapter 6

Numerical Solution of Ordinary Differential Equations

6.1 Introduction

The differential equations are of fundamental importance in engineering mathematics because many physical laws of biology, chemistry, ecology, economics, business, etc., and relations appear mathematically in the form of such equations. We know that many differential equations can be solved explicitly in terms of elementary functions of calculus. For example, the explicit solution of the differential equation

$$\frac{dy}{dx} = e^{x-y},$$

can be obtained easily as

$$y(x) = \ln(e^x + C),$$

and using initial condition $y(0) = 1$, we get

$$y(x) = \ln(e^x + e - 1).$$

We can use the MATLAB command *dsolve* (MATLAB's symbolic differential equation solver) which produces the general solution to the differential equation, or the specific solution to an associated initial-value problem.

```
>> syms x y;
>> y = dsolve('Dx = exp(x - y)','x');
>> y = dsolve('Dx = exp(x - y)','y(0) = 1','x'); pretty(y)
```

But there are many differential equations which cannot be solved explicitly in terms of the functions of calculus. For example, the solutions of the differential equation of the form

$$\frac{dy}{dx} = e^{-x^2},$$

are the integral, or antiderivatives of e^{-x^2},

$$y(x) = \int e^{-x^2}\, dx + C,$$

but it is known that these integrals cannot be expressed in terms of the functions of calculus.
We will refer to solutions that can be explicitly written in terms of elementary functions or special functions as formula or symbolic solutions-whether they were obtained via hand-calculation or via *dsolve*. When we cannot solve a differential equation in this way, or if the formula we find to complicated, we turn to numerical methods to solve as initial-value problem. This is similar to a situation in calculus: if we cannot find an antiderivative in terms of elementary functions, we turn to a numerical method such as Trapezoidal rule or Simpson's rule to evaluate a definite integral.

6.2 Ordinary Differential Equations

Here, we will discuss about the ordinary differential equations and their numerical solutions.

Definition 6.1 (Differential Equation)

An equation which involving functions and their derivatives. For example, the following equations

$$(a)\quad \frac{dy}{dx} = 3x, \qquad (b)\quad \frac{d^2y}{dx^2} + 4\frac{dy}{dx} + y = 0,$$

$$(c)\quad \frac{dy}{dx} = x^2 + y^2, \qquad (d)\quad \left(\frac{d^3y}{dx^3}\right)^2 - 5\frac{d^2y}{dx^2} + 2y = 5,$$

are differential equations. •

For the sake of completeness, we shall define some of the standard terms for differential equations.

Definition 6.2 (Dependent Variable)

It is the variable that has being differentiated. For example, in each of above differential equations (a)-(d), y is the dependent variable. •

Definition 6.3 (Independent Variable)

It is the variable with respect to which the differentiation is performed. For example, in each of above differential equations (a)-(d), x is the independent variable. •

Definition 6.4 (Order of Differential Equation)

The order of the differential equation is the order of the highest derivative involved. For example, the differential equations (a) and (c) are of first-order since the highest derivatives that appear is of first-order, whereas the differential equations (b) and (d) are respectively, the second-order and the third-order. •

Definition 6.5 (Degree of Differential Equation)

The degree of the differential equation is the power to which the highest-order derivative is raised. For example, the differential equations (a)-(c) are of degree 1 while the differential equation (d) is of degree 2. •

Definition 6.6 (Linear Differential Equation)

An differential equation is linear if

(1) The dependent variable y and all its derivatives are of the first degree, that is, the power of each term involving y or its derivatives is one.

(2) Each coefficient depends on only independent variable x or constant.

For example, the above differential equations *(a)* and *(b)* are the linear differential equations while the differential equations *(c)* and *(d)* are the nonlinear differential equations. •

Definition 6.7 (Initial Conditions)

When all of the conditions are given at starting value of independent variable x to solve a given differential equation, is called a initial condition. When the conditions are given at the endpoints of x-values, then the conditions are called the boundary conditions. •

6.2.1 Classification of Differential Equations

There are two major types of differential equations, called, *ordinary differential equations (ODE)* and *partial differential equations (PDE)*. If an equation contains only ordinary derivatives of one or more dependent variables, with respect to a single independent variable, it is then said to be an *ordinary differential equation*. For example, all the differential equations *(a)-(d)* are ordinary differential equations because there is only one independent variable, called x.
An equation involving the partial derivatives of one or more dependent variables of two or more independent variables is called it partial differential equation. For example, the following differential equation

$$\frac{\partial^2 y}{\partial x^2} = c\frac{\partial^2 y}{\partial t^2},$$

is the partial differential because it involves two independent variables, x and t. Although partial differential equations are very useful and important, their study demands a good foundation in the theory of ordinary differential equations. Consequently, in this chapter the discussion that follows we shall confine our attention to ordinary differential equations.
As a mathematical form, the ordinary differential equation is a very useful tool. It is used in modeling of a wide variety of physical phenomena, that is, chemical reactions, satellite orbits, vibrating or oscillating systems, electrical networks, and so on. In many cases, the independent variable represents time so that a differential equation describes change, with respect to time, in the system being modeled. The solution of a differential equation will be representation of the state of the system at any point in time and one can use it to study the behavior of the system. Consequently, the problem of finding the solution of a differential equation play an important role in scientific research.

The *solution* of a differential equation is the function which satisfies the differential equation. In solving a differential equation analytically, one usually compute a general solution containing arbitrary constant. The simplest form of the differential equation is

$$y' = f(x), \tag{6.1}$$

with $f(x)$ a given function. The general solution of this equation is

$$y(x) = \int f(x)dx + C, \tag{6.2}$$

where C is an arbitrary constant. For example, the differential equation of the form

$$y' = \cos x, \tag{6.3}$$

has general solution of the form

$$y(x) = \sin x + C. \tag{6.4}$$

The more general equation is

$$y' = f(x, y(x)). \tag{6.5}$$

Since the general solution of differential equation is depends on an arbitrary constant C, so this constant can be calculated by specifying the value of function $y(x)$ at a particular point x_0

$$y(x_0) = y_0.$$

The point x_0 is called initial point, and the number y_0 is called the initial value. We call the problem of solving

$$y' = \frac{dy}{dx} = f(x, y); \quad x_0 \le x \le x_n, \quad y(x_0) = y_0, \tag{6.6}$$

the initial-value problem (IVP). For example, for finding the solution of the differential equation (6.3) satisfying $y(\pi) = 1$, we have the value of the constant $C = 1$, so (6.4) becomes

$$y(x) = \sin x + 1,$$

and it is called the particular solution of the differential equation (6.3), or called the solution of the initial-value problem

$$y' = \cos x, \qquad y(\pi) = 1.$$

The main concern of this chapter is approximating the solution to the problem (6.6). The initial-value problems are problems in which the value of the dependent variable y is known at a point x_0. Such a large number of methods are available to handle problems of this type that one may have difficulty in deciding which to use. Solving initial-value problem numerically we will assume that the solution is being sought on a given finite interval $x_0 \le x \le x_n$ with $h = (b - a)/n$, where $x_0 = a, x_n = b$ and n be the number of subintervals. In this chapter the most widely used numerical methods are discussed in some details to find the solution of the initial-value problem. If the analytical process of finding a exact solution $y(x)$ is not feasible, it is still useful to know whether a solution exists and unique using numerical methods. To make precise preceding discussion, we give the following theorem which gives a sufficient condition for the existence and uniqueness of the initial-value problem (6.6).

Theorem 6.1 (Existence and Uniqueness Theorem)

Let $f(x, y)$ and $\dfrac{\partial f}{\partial y}$ be continuous functions of x and y at all points (x, y) in some neighborhood of the initial point (x_0, y_0). Then there is a unique function $y(x)$ defined on some interval $[x_0-\epsilon, x_0+\epsilon]$ and satisfying

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \quad x \in [x_0 - \epsilon, x_0 + \epsilon], \quad \epsilon > 0 \tag{6.7}$$

For example, the initial-value problem

$$y'(x) = 2xy^2, \qquad y(0) = 1,$$

has a unique solution

$$y(x) = \frac{1}{1-x^2}, \quad -1 < x < 1,$$

because the both functions

$$f(x,y) = 2xy^2, \qquad \frac{\partial f}{\partial y} = 4xy,$$

are continuous for all (x, y). Note that this example also showed that the continuity of the function $f(x, y)$ and $\dfrac{\partial f}{\partial y}$ for all (x, y) does not imply the existence of a $y(x)$ that is continuous for all x.

6.3 Numerical Methods for Solving Differential Equations

By a numerical method for solving the initial-value problem (6.6) is meant a procedure for finding approximate values $y_0, y_1, \ldots, y_n$ of the exact solution $y(x)$ at the given points $x_0 < x_1 < \ldots < x_n$. We will let y_i denote the numerical value obtained as approximation to the exact solution $y(x_i)$, with $x_i = x_0 + ih$ for $i = 0, 1, \ldots, n$, where h (constant) is the size of the interval. Numerical methods for differential equations are of great importance to the engineer and physicist because practical problems often lead to differential equations that cannot be solved by any analytical method or to equations for which the solutions in terms of formulas are so complicated that are often prefers to calculate a table of values by applying a numerical method to such an equation.
Two different types of numerical methods are available to solve initial-value problem (6.6). These are called the *single-step* and the *multi-steps* methods. The methods discussed will vary in complexity, since in general, the greater the accuracy of a method, the greater is its complexity. Here, we shall discuss only the single-step methods for solving the problem (6.6).
This type of method called self-starting, refers to estimate $y'(x)$, from a initial condition $y(x_0) = y_0$ and $y'_0 = f(x_0, y_0)$ from (6.6) and proceed step-wise. In the first-step we compute an approximate value y_1 of the solution $y(x)$ at $x = x_1 = x_0 + h$. In the second-step we compute an approximate value y_2 of that solution at $x = x_2 = x_0 + 2h$ and so on. Although these methods generally use functional evaluation information for x_i and x_{i+1}, they do not retain that information for direct use in future approximations. All the information used by these methods is consequently obtained within the interval over which the solution is being approximated. Among of them we will discuss here, the Euler's method , the Taylor's method of higher-orders, and the Runge-Kutta method of order two only.

6.3.1 Euler's Method

One of the simplest and most straight forward numerical method for solving first-order ordinary differential equation of the form (6.6) is called method of *Euler*. This method is not an efficient numerical method and so seldom used, but it is relatively easy to analysis and many of the ideas involved in the numerical solution of differential equations are introduced most simply with it.
In principle, the Euler's method uses the forward difference formula approximation of $y'(x)$ which we discussed in the previous Chapter 5. That is

$$y' = \frac{dy}{dx} \approx \frac{y(x_{i+1}) - y(x_i)}{h}, \tag{6.8}$$

where h is the stepsize and it is equal to $x_{i+1} - x_i$. Given that $\dfrac{dy}{dx} = f(x, y)$ and the initial conditions $x = x_0,\ y(x) = y(x_0)$, we have

$$\frac{y(x_1) - y(x_0)}{h} \approx f(x_0, y(x_0)), \qquad \text{or} \qquad y(x_1) \approx y(x_0) + hf(x_0, y(x_0)),$$

which shows that $y(x_1)$ is approximately given by $y(x_0) + hf(x_0, y(x_0))$. We can now use this approximation for $y(x_1)$ to estimate $y(x_2)$, that is

$$y(x_2) \approx y(x_1) + hf(x_1, y(x_1)),$$

and so on. In general,

$$y(x_{i+1)} \approx y(x_i) + hf(x_i, y(x_i)), \quad i = 0, 1, \ldots, n-1.$$

Taking $y_i \approx y(x_i)$, for each $i = 1, 2, \ldots, n$, we have

$$y_{i+1} = y_i + hf(x_i, y_i), \quad i = 0, 1, \ldots, n-1. \tag{6.9}$$

This simple integration strategy is known as the Euler's method, or the *Euler-Cauchy method*. It is called an explicit method because the value of $y(x)$ at the next step is calculated only from the value of $y(x)$ at the previous step. Given the approximate formula, one can solve for y_{i+1} in terms of $x_i,\ y_i$ and $f(x_i, y_i)$, all of which are known. Note that the above formula (6.9) can be derive by using the Taylor series expansion of the unknown solution $y(x)$ to the problem (6.6) about the point $x = x_i$, for each $i = 0, 1, \ldots, n-1$

$$y(x_{i+1}) = y(x_i) + (x_{i+1} - x_i)y'(x_i) + \frac{(x_{i+1} - x_i)^2}{2!}y''(\eta_i) = y(x_i) + hy'(x_i) + \frac{h^2}{2!}y''(\eta_i), \tag{6.10}$$

where unknown point η_i lies in the interval (x_i, x_{i+1}). For the smaller value of stepsize h, the higher power h^2 will be very small and may be neglected. Using $f(x_i, y_i)$ to evaluate $y'(x_i)$ and $y_i \approx y(x_i)$, we have the formula (6.9). Geometrically interpretation of the method is shown by Figure 6.1.

Example 6.1 *Use the Euler's method to find the approximate value of* $y(1)$ *for the given initial-value problem*

$$y' = xy + x, \qquad 0 \le x \le 1, \quad y(0) = 0, \quad \textit{with} \quad h = 0.1,\ 0.2.$$

Compare your approximate solutions with the exact solution $y(x) = -1 + e^{x^2/2}$.

Solution. *Since* $f(x, y) = xy + x$, *and* $x_0 = 0,\ y_0 = 0$, *then*

$$y_{i+1} = y_i + hf(x_i, y_i), \quad \textit{for} \quad i = 0, 1, \ldots, 9.$$

Then for $h = 0.1$ *and taking* $i = 0$, *we have*

$$y_1 = y_0 + hf(x_0, y_0) = y_0 + h(x_0y_0 + x_0) = 0 + (0.1)[(0)(0) + (0)] = 0.0000.$$

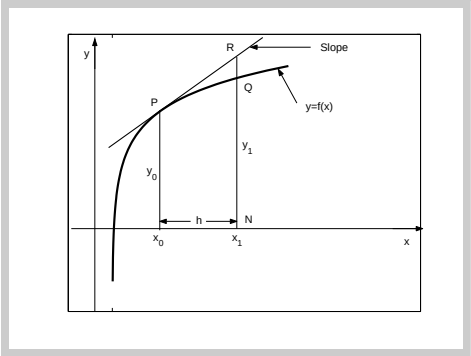


Figure 6.1: Geometrically interpretation of the Euler's method.

Similar way, we have other approximations by taking $x_i = x_{i-1} + h, i = 1, \ldots, 9$, *as follows*

$$y_2 = 0.0100, \quad y_3 = 0.0302, \quad y_4 = 0.0611, \quad y_5 = 0.1036, \quad y_6 = 0.1587,$$

$$y_7 = 0.2283, \quad y_8 = 0.3142, \quad y_9 = 0.4194, \quad y_{10} = 0.5471,$$

with possible absolute error, $|y(1) - y_{10}| = |0.6487 - 0.5471| = 0.1016$. *Similarly, for* $h = 0.2$, *give*

$$y_1 = 0.0000, \quad y_2 = 0.0400, \quad y_3 = 0.1232, \quad y_4 = 0.2580, \quad y_5 = 0.4592,$$

with possible absolute error, $|y(1) - y_5| = |0.6487 - 0.4592| = 0.1895$.
It showed that the result for $h = 0.1$ is better than the $h = 0.2$ and for both cases the approximation is not even correct to 1 decimal place. Clearly, the results using this method are inferior to those we will obtain in the coming methods. However, the accuracy of the Euler's method could be considerable improved by using smaller value of h than 0.1. •
Note that before calling MATLAB function Euler1 which defined below, we must define MATLAB function fun1 as follows:

$$\begin{array}{l} function\ f = fun1(x, y) \\ f = x * y + y; \end{array}$$

Given Euler1.m and fun1.m the results obtained manually in the preceding example are reproduced with following MATLAB command:

$$>> [x', y'] = Euler1('fun1', 0, 1, 0, 10);\ [x', y'] = Euler1('fun1', 0, 1, 0, 5)$$

The same results are obtained with the following statements that define MATLAB command as an inline function object:

$$>> sol = inline('x * y + x', 'x', 'y');\ [x', y'] = Euler1(sol, 0, 1, 0, 10);\ disp([x', y'])$$

Example 6.2 *Use the Euler's method to find the approximate value of* $y(1.4)$ *for the given initial-value problem*

$$\frac{1}{x}y' - y^2 = 0, \quad y(1) = 1, \quad with \quad n = 2.$$

Compare your approximate solutions with the exact solution $y(x) = 2/(3 - x^2)$.

Solution. *Since* $f(x, y) = xy^2$, *and* $x_0 = 1,\ y_0 = 1,\ h = 0.2, x_1 = 1.2, x_2 = 1.4$, *so*

$$y_{i+1} = y_i + hf(x_i, y_i), \quad for \quad i = 0, 1.$$

Then

$$y_1 = y_0 + hf(x_0, y_0) = y_0 + h(x_0 y_0^2) = 1 + (0.2)[(1)(1)] = 1.2.$$
$$y_2 = y_1 + hf(x_1, y_1) = y_1 + h(x_1 y_1^2) = 1.2 + (0.2)[(1.2)(1.44)] = 1.5456,$$

the required approximation of $y(1.4)$ *and* $|y(1.4) - y_2| = |1.9231 - 1.5456| = 0.3775$, *is the possible absolute error.* •

Program 6.1
MATLAB m-file for Euler Method
function sol=Euler1(fun1,a,b,y0,n)
h=(b-a)/n; x=a+(0:n)*h; y(1)=y0;
$for\ k = 1 : n;\ y(k+1) = y(k) + h * feval(fun1, x(k), y(k));$ end; sol=[x',y'];

Example 6.3 *Use the Euler's method to find the values of* $y(0.1)$ *and* $y(0.2)$ *from the initial-value problem* $\dfrac{dy}{dx} = x^2 + y^2, \quad y(0) = 1\ h = 0.05.$

Solution. *Given,* $f(x, y) = x^2 + y^2, \quad x_0 = 0,\ y_0 = 1,\ h = 0.05$, *then*

$$\begin{aligned}
x_1 &= x_0 + h = 0.05 \\
y_1 &= x(0.05) = y_0 + hf(x_0, y_0) = 1 + 0.05(0 + 1) = 1.05 \\
x_2 &= x_1 + h = 0.1 \\
y_2 &= x(0.1) = y_1 + hf(x_1, y_1) = 1.05 + 0.05(0.05^2 + 1.05^2) = 1.1053 \\
x_3 &= x_2 + h = 0.15 \\
y_3 &= y(0.15) = y_2 + hf(x_2, y_2) = 1.1053 + 0.05(0.15^2 + 1.1053^2) = 1.1675 \\
x_4 &= x_3 + h = 0.2 \\
y_4 &= y(0.2) = y_3 + hf(x_3, y_3) = 1.1675 + 0.05(0.2^2 + 1.1675^2) = 1.2377.
\end{aligned}$$

Hence $y(0.1) \approx y_2 = 1.1053$ *and* $y(0.2) \approx y_4 = 1.2377$ *the required approximate solutions.* •

Example 6.4 *Use the Euler's method to find the approximate value of* $y(0.2)$ *for the given initial-value problem*

$$y' = e^{-2x} - 2y, \qquad 0 \le x \le 0.2, \quad y(0) = 0.1, \quad with \quad h = 0.05,\ 0.1.$$

Compare your approximate solutions with the exact solution $y(x) = 0.1e^{-2x} + xe^{-2x}$.

Solution. *Since* $f(x, y) = e^{-2x} - 2y$, *then the Euler's method gets the form*

$$y_{i+1} = y_i + hf(x_i, y_i), \quad for \quad i = 0, 1, \ldots, 3.$$

Now taking $x_0 = 0,\ y_0 = 0.1,\ h = 0.05$ *and for* $i = 0$*, we have*

$$y_1 = y_0 + hf(x_0, y_0) = y_0 + h(e^{-2x_0} - 2y_0) = 0.1 + (0.05)[1 - 2(0.1)] = 0.1400.$$

Similar way, we have other approximations by taking $x_i = x_{i-1} + h, i = 1, 2, \ldots, 3$*, as follows*

$$\begin{array}{rclcrcl} y_2 &=& y_1 + hf(x_1, y_1) &=& y_1 + h(e^{-2x_1} - 2y_1) &=& 0.1712 \\ y_3 &=& y_2 + hf(x_2, y_2) &=& y_2 + h(e^{-2x_2} - 2y_2) &=& 0.1950 \\ y_4 &=& y_3 + hf(x_3, y_3) &=& y_3 + h(e^{-2x_3} - 2y_3) &=& 0.2125, \end{array}$$

Now for $h = 0.1$*, we have the approximations as follows*

$$\begin{array}{rclcrcl} y_1 &=& y_0 + hf(x_0, y_0) &=& y_0 + h(e^{-2x_0} - 2y_0) &=& 0.1800 \\ y_2 &=& y_1 + hf(x_1, y_1) &=& y_1 + h(e^{-2x_1} - 2y_1) &=& 0.2259, \end{array}$$

with possible error $|y(0.2) - y_2| = |0.2011 - 0.2125| = 0.0248$.
Since the Euler's method is an iterative method so, it may be converge or diverge. If divergence occurs, then the procedure should be terminated because there may be no solution.

6.3.1.1 Analysis of the Euler's Method

The preceding Example 6.4 demonstrates that the error in applying the Euler's method is reduced when h is reduced. The question of how well the Euler's method for solving the initial-value problem (6.6) works is closely related to the truncation error of the method. There are two types of such error, *local* and *global* truncation error. In case of local truncation error one consider the size of the error made during one step and for global truncation error, one can consider the errors in the entire interval $x_0 \le x \le x_n$ over which the solution is sought.
We turn to the Taylor series to find an expression that represents the error, we have

$$y(x + h) = y(x) + hy'(x) + \frac{h^2}{2!}y''(\eta(x)), \quad \text{for unknown} \quad \eta(x) \in [x, x + h].$$

If $y(x_{i+1})$ is the true value of $y(x)$, then the Taylor series expression at x_i is

$$y(x_{i+1}) = y(x_i) + hf(x_i, y_i) + \frac{h^2}{2!}y''(\eta(x_i)), \quad \eta(x_i) \in (x_i, x_{i+1})$$

as $y' = f(x_i, y_i)$. The Euler's formula uses the recurrence relation

$$y_{i+1} = y(x_i) + hf(x_i, y_i),$$

to estimate y_{i+1} assuming that $y(x_i)$ is the true solution. The error in y_{i+1} is given by $y_{i+1} - y(x_{i+1})$ which can be written as

$$y_{i+1} - y(x_{i+1}) = (y(x_i) + hf(x_i, y_i) + \frac{h^2}{2!}y''(\eta(x_i))) - (y(x_i) + hf(x_i, y_i)) = \frac{h^2}{2!}y''(\eta(x_i)), \quad (6.11)$$

for $i = 0, 1, \ldots, n - 1$. We call the term $\dfrac{h^2}{2}y''(\eta(x_i))$, the *local* truncation error for the Euler's method. It is of order h^2.

Note that this error term only applies in the region (x_i, x_{i+1}), hence it is only the error in estimating y_{i+1} from $y(x_i)$. It does not take into account the compounded error from previous estimates. If we assume that the error is increasing linearly with n, then the error will be proportional to nh^2, but n is dependent on h as $h = \dfrac{x_n - x_0}{n}$, so the error will be proportional to

$$\frac{(x_n - x_0)}{h} h^2 = (x_n - x_0)h,$$

which is order h. This error is called the *global* truncation error. We give an error bound on the global error for Euler's method.

Theorem 6.2 (Euler's Method Error Bound Formula)

Let $y(x)$ denotes the unique solution to the initial value problem

$$\frac{dy}{dx} = f(x, y), \qquad for \quad a = x_0 \le x \le x_n = b, \quad with \quad y(x_0) = y_0,$$

and $y_0, y_1, \ldots, y_n$ be the approximations generated by Euler's method for some positive integer n. Suppose that f is continuous for all x in $[a, b]$ and all y in $(-\infty, \infty)$ and constants L and M exist with

$$\left|\frac{\partial f}{\partial y}\right| \le L \qquad and \qquad |y''(x)| \le M,$$

then, for each $i = 0, 1, \ldots, n$,

$$|y(x_i) - y_i| \le \frac{hM}{2L}\left(e^{L(x_i - x_0)} - 1\right). \tag{6.12}$$

This shows that the global error for Euler's method is $O(h)$. In other words, this means that error bound is proportional to the step size h. •

Example 6.5 *Use the Euler's method to find the approximate value of $y(0.4)$ for the given initial-value problem*

$$y' - y + 1 = 0, \quad y(0) = 2, \quad with \quad n = 4.$$

Compare your approximate solutions with the exact solution $y(x) = e^x + 1$ along with the error bound.

Solution. *Since $f(x, y) = y - 1$ and $\left|\dfrac{\partial f}{\partial y}\right| \le 1 = L$, and the exact solution is $y(x) = e^x + 1$, we have $|y''(x)| \le e^{0.4} = M$ for all x in $[0, 0.4]$. Thus using $x_0 = 0,\ y_0 = 2,\ h = 0.1$, the Euler's method*

$$y_{i+1} = y_i + hf(x_i, y_i) = y_i + h[y_i - 1], \quad for \quad i = 0, 1, 2, 3, 4,$$

and the error bound formula

$$|y(x_i) - y_i| \le \frac{(0.1)(e^{0.4})}{2(1)}\left(e^{1(x_i - 0)} - 1\right),$$

Table 6.1 gives the approximate solutions, together with actual error and the error bound. In this example, the error bounds turn out to be reasonably sharp. However, the global error bounds are usually hopelessly conservative. •

Table 6.1: Solution of the Example 6.5.

x_i	y_i	Actual Error	Error Bound
0	2.0000	0.000000	0.000000
0.1	2.1000	0.005171	0.007845
0.2	2.2100	0.011403	0.016515
0.3	2.3310	0.018886	0.026096
0.4	2.4641	0.027725	0.036686

Example 6.6 *Consider the initial value problem*

$$y' - \frac{y}{x} - 1 = 0, \quad 1 \le x \le 2, \quad y(1) = 2.$$

The exact solution to this problem is $y(x) = x \ln x + 2x$. *If Euler's method is used to solve this problem and an accuracy of* 10^{-4} *is desired for the final value* $y(2)$, *what stepsize* h *should be used approximately ?*

Solution. *We will use the following formula to obtain the stepsize* h*:*

$$|y(x_i) - y_i| \le \frac{hM}{2L}\left(e^{L(x_i - x_0)} - 1\right).$$

Since $f(x, y) = \dfrac{y}{x} + 1$, *we have* $y'(x) = \ln x + 3$ *and* $y''(x) = \dfrac{1}{x}$. *The bound* M *is obtained from* $|y''(x)| = \left|\dfrac{1}{x}\right| \le 1 = M$, *on* $[1, 2]$. *Also, since* $f(x, y) = \dfrac{y}{x} + 1$, *we have*

$$\left|\frac{\partial f}{\partial y}\right| = \left|\frac{1}{x}\right| \le 1 = L, \quad on \quad [1, 2].$$

Thus using these into the error bound formula

$$\frac{h}{2}\left(e^1 - 1\right) \le 10^{-4}, \quad gives \quad h = 1.164 \times 10^{-4}.$$

Taking $h = 1.164 \times 10^{-4}$, $n = 8591$, $x = 2$, *using Euler's method, we get approximate solution* $y = 5.3862361624$ *of the exact solution* $y = 5.3862943611$ *with possible error* $5.8198752389 \times 10^{-5}$.•

Example 6.7 *Assume that* $y(x) = 4\cos x - 7(x - 4)^3$ *is the solution to some initial value problem on* $[-2, 3]$. *Then find the Euler's formula for the initial value problem. Find upper bounds for local and global truncation errors at* $x = 3$ *by assuming* $L = 2$ *and take* $h = 0.001$.

Solution. *Since* $y(x) = 4\cos x - 7(t - 4)^3$, *then* $y'(x) = -4\sin x - 21(x - 4)^2$, *and* $y''(x) = -4\cos x - 42(x - 4)$. *Thus* $f(x, y) = y'(x) = -4\sin x - 21(x - 4)^2$ *and the Euler's formula gets the form*

$$y_{i+1} = y_i + hf(x_i, y_i) = y_i + h[-4\sin x_i - 21(x_i - 4)^2], \quad y(x_0) = y_0, \quad for \quad i = 0, 1, \ldots, n - 1.$$

The local truncation error for Euler's method is:

$$E = \frac{h^2}{2!}y''(\eta(x_i)), \qquad \text{for some} \quad \eta(x_i) \in (x_i, x_{i+1}).$$

The upper bound for the local truncation error is

$$|E| \leq \frac{h^2}{2}M,$$

and at $x = 3$, $|y''(3)| = |-4\cos 3 - 42(3-4)| = 45.9600 = M$. *Hence, for* $h = 0.001$ *and* $x = 3$, *we have*

$$|E| \leq \frac{(0.001)^2}{2}(45.9600) = 2.2980 \times 10^{-5}.$$

To calculate the upper bound for global truncation error, we use the formula:

$$|E| \leq \frac{hM}{2L}\left(e^{L(x_i - x_0)} - 1\right).$$

Since given $L = 2$, *therefore*

$$|E| \leq \frac{(0.001)(45.9600)}{2(2)}\left(e^{2(3+2)} - 1\right) = 253.0726.$$

We likely got a bigger number than the local error. •

6.3.2 Higher-Order Taylor Methods

The basis for many numerical techniques finding the approximate solution of the initial-value problem can be depend to the Taylor's series, as we used this series in the previous section in finding the Euler's method which also called the Taylor's method of order one. One can, of course, develop the Taylor's method for higher-order to obtain better accuracy, and in general, one expect that higher the order of the method, greater the accuracy for a given stepsize. The Taylor's method is relatively easy to use, however, the necessity of calculating the higher derivatives makes the Taylor's method completely unsuitable. Nevertheless, it is of great theoretical interest because the most of the practical methods attempt to achieve the same accuracy as the Taylor's method of a given order without the disadvantage of having to calculate the higher derivatives. Assuming that the solution $y(x)$ of the initial-value problem (6.6) has $(n+1)$ continuous derivatives and expanding $y(x)$ in terms of its *nth* degree Taylor polynomial about x_i, we get

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2!}y''(x_i) + \cdots + \frac{h^n}{n!}y^{(n)}(x_i) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\eta(x_i)), \tag{6.13}$$

for some $\eta(x_i) \in (x_i, x_{i+1})$. The derivatives in this expansion are not known explicitly since the solution is not known. However, if f is sufficiently differentiable, they can be obtained by taking the total derivative of (6.6) with respect to x, keeping in mind that f is an implicit function of y. Thus

$$y' = f(x,y) = f,\ y'' = f' = f_x + f_y f,\ y''' = f'' = f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_x f_y + f_y^2 f,\ \cdots \tag{6.14}$$

Continuing in this manner, we can express any derivative of y in terms of $f(x,y)$ and its partial derivatives. It is already clear, however, that unless $f(x,y)$ is a very simple function, the higher total derivatives become increasingly complex. Now substituting these results into (6.13), gives

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + hf(x_i, y(x_i)) + \frac{h^2}{2!}f'(x_i, y(x_i)) + \cdots \\ &+ \frac{h^n}{n!}f^{(n-1)}(x_i, y(x_i)) + \frac{h^{n+1}}{(n+1)!}f^{(n)}(\eta(x_i), y(\eta(x_i))). \end{aligned} \tag{6.15}$$

By taking $y_i \approx y(x_i)$, that the approximation to the exact solution at x_i, for each $i = 0, 1, \ldots, n-1$,

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2!}f'(x_i, y_i) + \cdots + \frac{h^n}{n!}f^{(n-1)}(x_i, y_i). \tag{6.16}$$

Then this formula is called the Taylor's method of order n. The last term of (6.15), called remainder, shows that the local error of Taylor's method of order n is

$$E = \frac{h^{n+1}}{(n+1)!}f^{(n)}(\eta_i, y(\eta(x_i))) = \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\eta(x_i)), \quad \text{for some} \quad x_i < \eta(x_i) < x_{i+1}. \tag{6.17}$$

Example 6.8 *Use the Taylor's method of order 2 to find the approximate value of $y(1)$ for the given initial-value problem.*

$$y' = xy + x, \quad 0 \le x \le 1, \quad y(0) = 0, \quad \textit{with} \quad h = 0.2$$

Compare your approximate solution with the exact solution $y(x) = -1 + e^{x^2/2}$.

Solution. *Since $f(x,y) = xy + x$, and $x_0 = 0$, $y_0 = 0$, then*

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2}f'(x_i, y_i), \quad \textit{for} \quad i = 0, 1, 2, 3, 4$$

where $f'(x_i, y_i) = y_i + x_i^2 y_i + x_i^2 + 1$. Then for $i = 0, 1, 2, 3, 4$, we have

$$y_1 = y_0 + h(x_0 y_0 + x_0) + \frac{h^2}{2}(y_0 + x_0^2 y_0 + x_0^2 + 1) = 0 + (0.2)(0) + (0.02)(1) = 0.0200,$$

$$y_2 = 0.0820, \quad y_3 = 0.1937, \quad y_4 = 0.3694, \quad y_5 = 0.6334,$$

with absolute possible error $|y(1) - y_5| = |0.6487 - 0.6334| = 0.0153$.
The result is entirely correct to 1 decimal place. Clearly, the result using this method is better than the Euler's method and it could be considerable improved by using smaller value of h than 0.2.•

Note that before calling MATLAB functions tayl1 and fun1, we must define MATLAB function dfun1 as follows:

```
function f = dfun1(x,y)
f = y + x.^2*y + x.^2 + 1;
```

Given tayl1.m, fun1.m and dfun1.m, the results obtained manually in the preceding example are reproduced with following MATLAB commands:

```
>> [x',y'] = tayl1('fun1','dfun1',0,1,0,5); disp([x',y'])
```

Example 6.9 *Use the Taylor's method of order 2 to find the approximate value of $y(0.2)$ for the given initial-value problem*

$$y' = e^{-2x} - 2y, \qquad 0 \le x \le 0.2, \quad y(0) = 0.1, \quad \textit{with} \quad h = 0.05,\ 0.1.$$

Compare your approximate solutions with the exact solution $y(x) = 0.1e^{-2x} + xe^{-2x}$.

Solution. *Since $f(x,y) = e^{-2x} - 2y$, so $f'(x,y) = -4e^{-2x} + 4y$. Thus the Taylor's method of order 2 gets the form*

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2} f'(x_i, y_i), \quad \textit{for} \quad i = 0, 1, \ldots, 3.$$

Now taking $x_0 = 0,\ y_0 = 0.1,\ h = 0.05$, then for $i = 0$, we have

$$y_1 = y_0 + hf(x_0, y_0) + \frac{h^2}{2} f'(x_0, y_0) = y_0 + h(e^{-2x_0} - 2y_0) + 2h^2(-e^{-2x_0} + y_0) = 0.1355.$$

Similar way, we have other approximations by taking $x_i = x_{i-1} + h, i = 1, 2, \ldots, 3$, as follows

$$\begin{array}{rclcrcl} y_2 &=& x_1 + hf(x_1,y_1) + h^2/2f'(x_1,y_1) &=& y_1 + h(e^{-2x_1} - 2y_1) + 2h^2(-e^{-2x_1} + y_1) &=& 0.1633 \\ y_3 &=& x_2 + hf(x_2,y_2) + h^2/2f'(x_2,y_2) &=& y_2 + h(e^{-2x_2} - 2y_2) + 2h^2(-e^{-2x_2} + y_2) &=& 0.1846 \\ y_4 &=& x_3 + hf(x_3,y_3) + h^2/2f'(x_3,y_3) &=& y_3 + h(e^{-2x_3} - 2y_3) + 2h^2(-e^{-2x_3} + y_3) &=& 0.2004, \end{array}$$

with possible error

$$|y(0.2) - y_4| = |0.2011 - 0.2004| = 7 \times 10^{-4}.$$

Now for $h = 0.1$, we have the approximations as follows

$$\begin{array}{rclcrcl} y_1 &=& y_0 + hf(x_0,y_0) + h^2/2f'(x_0,y_0) = y_0 + h(e^{-2x_0} - 2y_0) + 2h^2(-e^{-2x_0} + y_0) &=& 0.1620 \\ y_2 &=& y_1 + hf(x_1,y_1) + h^2/2f'(x_1,y_1) = y_1 + h(e^{-2x_1} - 2y_1) + 2h^2(-e^{-2x_1} + y_1) &=& 0.1983, \end{array}$$

with possible error

$$|y(0.2) - y_2| = |0.2011 - 0.1983| = 2.8 \times 10^{-3}.$$

It showed that the result is entirely correct to 1 decimal place. Clearly, the result using this method is better than the Euler's method and it could be considerable improved by using smaller value of h than 0.2. •

Example 6.10 *Show that Taylor's method of order 2 for the initial-value problem*

$$e^y y' - e^x = 0, \quad 0 \le x \le 1, \quad y(0) = 1, \quad \textit{with} \quad h = 0.5,$$

is

$$y_{i+1} = y_i + he^{(x_i - y_i)} \left[1 + \frac{h}{2}\left(1 - e^{(x_i - y_i)}\right)\right], \quad i \ge 0.$$

What are the values of y_0, y_1, y_2. Find absolute error if the exact solution is $y(x) = \ln(e^x + e - 1)$.

Solution. *Since the Taylor's method of order 2 is*

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2} f'(x_i, y_i), \quad \textit{for} \quad i \ge 0,$$

and the given function is $f(x,y) = e^{x-y}$ *with its first derivative* $f'(x,y) = e^{x-y}[1 - e^{x-y}]$. *So using these values, we have*

$$y_{i+1} = y_i + he^{x_i - y_i} + \frac{h^2}{2}\left[e^{x-y}\left(1 - e^{x-y}\right)\right], \quad for \quad i \geq 0,$$

or

$$y_{i+1} = y_i + he^{(x_i - y_i)}\left[1 + \frac{h}{2}\left(1 - e^{(x_i - y_i)}\right)\right], \quad for \quad i \geq 0.$$

Now for $i = 0, 1$ *and using* $x_0 = 0, y_0 = 1$, $h = 0.5$, *we get*

$$y(0.5) \approx y_1 = 1 + (0.5)e^{(0-1)}\left[1 + \frac{0.5}{2}\left(1 - e^{(0-1)}\right)\right] = 1.2130.$$

$$y(1) \approx y_2 = 1.2130 + (0.5)e^{(0.5-1.2130)}\left[1 + \frac{0.5}{2}\left(1 - e^{(0.5-1.2130)}\right)\right] = 1.4893,$$

Thus $|y(1) - y_2| = |1.4899 - 1.4893| = 0.0006$, *is the absolute error in our approximation.* •

Program 6.2
MATLAB m-file for Taylor's Method of order 2
function sol=tayl1(fun1,dfun1,a,b,y0,n)
h=(b-a)/n; $x = a + (0:n) * h$; y(1)=y0; for k=1:n
$y(k+1) = y(k) + h * feval(fun1, x(k), y(k)) + (\text{h.}\hat{}\ 2 * feval(dfun1, x(k), y(k)))/2$; end;
$sol = [x', y']$;

Example 6.11 *Use the Taylor's method of order 3 to find the approximate value of* $y(1)$ *for the given initial-value problem*

$$4y' - y = 0, \quad 0 \leq x \leq 1, \quad y(0) = 1, \quad with \quad n = 2.$$

Compare your approximate solution with the exact solution $y(x) = e^{x/4}$.

Solution. *Since the Taylor's method of order 3 is*

$$y_{i+1} = y_i + hf(x_i, y_i) + \frac{h^2}{2!}f'(x_i, y_i) + \frac{h^3}{3!}f''(x_i, y_i),$$

for $i = 0, 1, \ldots, n-1$, *and using the given values* $x_0 = 0$, $y_0 = 1$ *and* $f(x,y) = 1/4y$, *we get,* $f'(x,y) = 1/16y$ *and* $f''(x,y) = 1/64y$. *So using these values we obtain Taylor's method of order 3 of the form*

$$y_{i+1} = y_i + h(1/4y_i) + \frac{h^2}{2}(1/16y_i) + \frac{h^3}{6}(1/64y_i).$$

Then for $i = 0, 1$ *and by taking* $y_0 = 1$, $h = 0.5$, *we get*

$$y(0.5) \approx y_1 = 1(1 + 0.125 + 0.0078 + 0.0003) = 1.1331,$$

$$y(1) \approx y_2 = y_1(1 + 0.125 + 0.0078 + 0.0003) = 1.1331(1.1331) = 1.2839.$$

Thus $|y(1) - y_2| = |1.2840 - 1.2839| = 0.0001$, *is the absolute error.* •

Example 6.12 *Show that third order Taylor's method for the given initial-value problem*

$$y' - 2x + y = 0, \qquad y(0) = -1,$$

is

$$y_{i+1} = y_i + (2x_- y_i)h + \left(\frac{h^2}{2} - \frac{h^3}{6}\right)[2(1 - x_i) + y_i], \quad i = 0, 1, \ldots, n-1.$$

Use it to find approximation of $y(0.1)$.

Solution. *Using* $f(x, y) = 2x - y$, $f'(x, y) = 2(1-x) + y$, $f''(x, y) = 2(x-1) - y$, *then the Taylor's method of order* 3 *gets the form*

$$y_{i+1} = y_i + h[2x_i - y_i] + \frac{h^2}{2}[2(1 - x_i) + y_i] + \frac{h^3}{6}[2(x_i - 1) - y_i],$$

and after simplifying, we get

$$y_{i+1} = y_i + (2x_i - y_i)h + \left(\frac{h^2}{2} - \frac{h^3}{6}\right)[2(1 - x_i) + y_i].$$

Now by taking $i = 0$ *and by using* $x_0 = 0, y_0 = 1$, $h = 0.1$, *we obtain*

$$y(x_1) \approx y_1 = y_0 + (2x_0 - y_0)h + \left(\frac{h^2}{2} - \frac{h^3}{6}\right)[2(1 - x_0) + y_0],$$

$$y(0.1) \approx y_1 = -1 + 0.1 + (0.005 - 0.0002)(2 - 1) = -0.8952.$$

In using the Taylor's method, we replace the infinite Taylor series for $f(x + h)$ by a partial sum. The *local truncation* error is inherent in any algorithm that we might choose.
If we retain term up to and including h^n in the series, then the local truncation error is the sum of all the remaining terms that we do not include by Taylor's method. These terms can be compressed into a single term of the form

$$\frac{h^{n+1}}{(n+1)!}\, f^{(n+1)}(\eta(x), y(\eta(x))),$$

for some unknown point $\eta(x)$. We say that the local truncation error is order h^{n+1}. An error of this sort is present in each step of the numerical solution. The accumulation of all the many local truncation error gives rise to the *global truncation* error which must be of order h^n because the number of steps necessary to reach on arbitrary point x, having started at x_0, is $\dfrac{x - x_0}{h}$. Choosing n large so that this error is small. •

6.3.3 Second-Order Runge-Kutta Method

Since we studied that the Euler's method is not very useful in practical problems because it requires a very small stepsize for reasonable accuracy. the Taylor's method of higher-order is difficult to use because it needs to obtain higher total derivatives of $y(x)$. An important group of methods which

allow us to obtain greater accuracy at each step and yet require only initial value of $y(x)$ to be given with the differential equation are called the Runge-Kutta methods. The Runge-Kutta methods attempt to obtain greater accuracy, and at the same time avoid the need of higher derivatives by evaluating the function $f(x, y)$ at selected points on each subintervals. These methods can be used to generate not only starting values but, in fact, in whole solution. They are self-starting and easy to program for a digital computer. We shall begin by showing how to derive the simplest formulas in this class. These are of the form

$$y_{i+1} = y_i + (w_1 k_1 + w_2 k_2), \tag{6.18}$$

where

$$k_1 = hf(x_i, y_i) \qquad \text{and} \qquad k_2 = hf(x_i + ah, y_i + bk_1).$$

The parameters $w_1,\ w_2,\ a$, and b are chosen in order to make the formula (6.18) as accurate as possible, that is, to make the order of accuracy as large as possible. To this end, we substitute the exact value $y(x), y(x_{i+1})$ by the local solution into the formula (6.18) and expand about the point x_i. The parameters are then chosen to make the resulting expansion agree as much as possible with the Taylor series for $y(x_{i+1})$ about x_i. Upon substituting into (6.18), we first expanding $y(x_{i+1})$ in the Taylor series through terms of order h^3, we obtain

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2!}y''(x_i) + \frac{h^3}{3!}y'''(x_i) + \cdots \tag{6.19}$$

Since

$$\begin{aligned} y' &= f(x, y) \\ y'' &= f'(x_i, y_i) = (f_x + f_y f)_i \\ y''' &= f''(x_i, y_i) = (f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_x f_y + f_y^2 f)_i + O(h^4). \end{aligned} \tag{6.20}$$

So

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + hf(x_i, y_i) + \frac{h^2}{2!}(f_x + f f_y)_i \\ &+ \frac{h^3}{3!}(f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_x f_y + f_y^2 f)_i + O(h^4), \end{aligned} \tag{6.21}$$

where the subscripts on f denote partial derivatives with respect to the indicated variables, and the subscript i means that all functions involved are to be evaluated at (x_i, y_i). Now using the Taylor's expansion for functions of two variables, we find that

$$\begin{aligned} k_2 &= hf(x_i + ah, y_i + bk_1) = h[f + h(af_x + bff_y) \\ &+ \frac{h^2}{2}(a^2 f_{xx} + 2abff_{xy} + b^2 f^2 f_{yy}) + O(h^4)]_i. \end{aligned} \tag{6.22}$$

Now we substitute this expression for k_2 into (6.18), gives

$$\begin{aligned} y_{i+1} &= y_i + h[w_1 f(x_i, y_i) + w_2 f(x_i + ah, y_i + bk_1)] \\ &= y_i + h[(w_1 + w_2)f]_i + h^2 w_2[(af_x + bff_y)]_i \\ &+ \frac{h^3}{2}w_2[a^2 f_{xx} + 2abff_{xy} + b^2 f^2 f_{yy}]_i + O(h^4). \end{aligned} \tag{6.23}$$

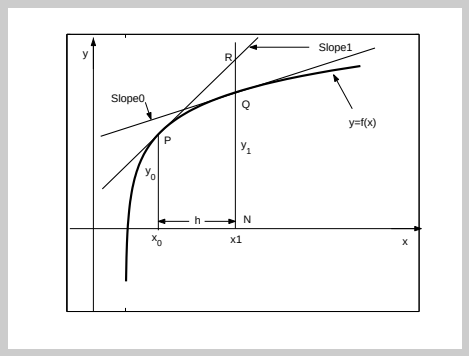


Figure 6.2: Geometrically interpretation of the Modified Euler method.

On comparing (6.21) and (6.23), we see that to make the corresponding powers of h and h^2 agree, we must have

$$w_1 + w_2 = 1 \qquad \text{and} \qquad a = \frac{1}{2w_2} = b.$$

This is a system of two nonlinear equations in the four unknowns a, b, w_1, and w_2 and its solution can be written in the form

$$b = a = \frac{1}{2w_2}, \quad w_1 = 1 - w_2. \tag{6.24}$$

There are many solutions to (6.24) depending on the choices of w_2. These choices leads to the numerical method which has order 2 and some of them do correspond to any of the standard numerical integration formulas. Taking the first choice when $w_2 = 1/2$, we have

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))]. \tag{6.25}$$

6.3.3.1 Modified Euler's Method

The equation (6.25) can be written in a standard form as

$$y_{i+1} = y_i + \frac{h}{2}[k_1 + k_2], \tag{6.26}$$

where

$$k_1 = f(x_i, y_i) \qquad \text{and} \qquad k_2 = f(x_{i+1}, y_i + hk_1),$$

for each $i = 0, 1, \ldots n-1$. Then the relation (6.26) is called the *Runge-Kutta method of order 2* which is also known as the *Modified Euler's method*. This method corresponds to using the Trapezoidal rule to estimate the integral where a preliminary (full) Euler step is taken to obtain the (approximate) value at x_{i+1}. Geometrically interpretation of the method is shown by Figure 6.2. The local error of this formula is $-\frac{h^3}{12}y''(\eta)$, however, of order h^3, whereas that of the Euler's method is h^2. We can therefore expect to be able to use a large stepsize with this formula. After n steps, the local accumulated error of this method is, $-\frac{(x_n - x_0)h^2}{12}y''(\eta)$, called the global truncation error of order h^2.

Example 6.13 *Use Runge-Kutta method of order two (Modified Euler's method) to find the approximate value of* $y(1)$ *for the given initial-value problem*

$$y' = xy + x, \quad 0 \le x \le 1, \quad y(0) = 0, \quad \textit{with} \quad h = 0.2.$$

Compare your approximate solution with the exact solution $y(x) = -1 + e^{x^2/2}$.

Solution. *Since* $f(x, y) = xy + x$, *and* $x_0 = 0,\ y_0 = 0$, *then for* $i = 0$, *we have*

$$\begin{aligned} k_1 &= f(x_0, y_0) = (x_0 y_0 + x_0) = 0.0000 \\ k_2 &= f(x_1, y_0 + hk_1) = (x_1(y_0 + hk_1) + x_1) = (0 + 0.2) = 0.2000, \end{aligned}$$

and using these values, we have

$$y_1 = y_0 + \frac{h}{2}\left[k_1 + k_2\right] = 0 + 0.1(0 + 0.2000) = 0.0200.$$

Continuing in this manner, we have

$$\begin{array}{llllll} k_1 = & 0.204, & k_2 = & 0.4243, & \textit{then} & y_2 = \ 0.0828, \\ k_1 = & 0.4331, & k_2 = & 0.7017, & \textit{then} & y_3 = \ 0.1963, \\ k_1 = & 0.7178, & k_2 = & 1.0719, & \textit{then} & y_4 = \ 0.3753, \\ k_1 = & 1.1002, & k_2 = & 1.5953, & \textit{then} & y_5 = \ 0.6449. \end{array}$$

The absolute error is $|y(1) - y_5| = |0.6487 - 0.6449| = 0.0039$. The results of the Example 6.13 can be obtained by using the following MATLAB command as follows:

```
>> sol = mod1('fun1', 0, 1, 0, 5);
```

So there is a significant improvement in accuracy of this method as compared with the Euler's method but the problem of accuracy still remains however since error will accurate from step to step. In particular, since the function $f(x, y)$ calculated repeatedly from values of $y(x)$ which include the accumulated error, these errors may grow in an unpredictable way.

Example 6.14 *Use the Runge-Kutta method of order two (the Modified Euler's method) to find the approximate value of* $y(1.4)$ *for the given initial-value problem*

$$xy' + y' - 2y = 0, \quad y(1) = 4, \quad \textit{with} \quad n = 2.$$

and compare your approximate solution with the exact solution $y(x) = (x + 1)^2$.

Solution. *Since* $f(x, y) = \dfrac{2y}{x + 1}$, *and* $x_0 = 1,\ y_0 = 4,\ h = (1.4 - 1)/2 = 0.2$, *then for* $i = 0$, *we have*

$$k_1 = f(x_0, y_0) = \frac{2y_0}{x_0 + 1} = 4, \quad k_2 = f(x_1, y_0 + hk_1) = \frac{2(y_0 + hk_1)}{x_1 + 1} = 4.3636.$$

By using these values, we have

$$y(1.2) \approx y_1 = y_0 + \frac{h}{2}\left[k_1 + k_2\right] = 4 + 0.1(4 + 4.3636) = 4.8364.$$

Similar manner, we have the other approximation for taking $i = 1$, as follows

$$k_1 = f(x_1, y_1) = \frac{2y_1}{x_1 + 1} = 4.3967 \quad \textit{and} \quad k_2 = f(x_2, y_1 + hk_1) = \frac{2(y_1 + hk_1)}{x_2 + 1} = 4.7631,$$

and by using these values, we have

$$y(1.4) \approx y_2 = y_1 + \frac{h}{2}[k_1 + k_2] = 4.8364 + 0.1(4.3967 + 4.7631) = 5.7523,$$

the approximation of $y(1.4)$ and $|y(1.4) - y_2| = |5.7600 - 5.7523| = 0.0077$, is the absolute error. •

Example 6.15 *Use the Runge-Kutta method of order two (the Modified Euler's method) to find the approximate value of $y(1.2)$ for the given initial-value problem*

$$x^2y' - y = 0, \quad y(1) = 2, \quad \textit{with} \quad n = 2.$$

and compare your approximate solution with the exact solution $y(x) = e^{(x-1)/x}$.

Solution. *Since $f(x, y) = x^{-2}y$ and $x_0 = 1$, $y_0 = 2$, $h = (1.2 - 1)/2 = 0.1$, for $i = 0$, $x_1 = 1.1, x_2 = 1.2$ we have*

$$\begin{aligned} k_1 &= f(x_0, y_0) = f(1, 2) = (1)^{-2}(2) = 2, \\ k_2 &= f(x_1, y_0 + hk_1) = f(1.1, 2.2) = (1.1)^{-2}(2.2) = 1.8182, \end{aligned}$$

and by using these values, we have

$$y(1.1) \approx y_1 = y_0 + \frac{h}{2}[k_1 + k_2] = 2 + 0.05(2 + 1.8182) = 2.1909.$$

Similar manner, we have the other approximation for taking $i = 1$, as follows

$k_1 = f(x_1, y_1) = f(1.1, 2.1909) = (1.1)^{-2}(2.1909) == 1.8107 \quad \textit{and} \quad k_2 = f(x_2, y_1 + hk_1) = f(1.2, 2.3720) = (1.2)^{-2}(2.3$

and by using these values, we have

$$y(1.2) \approx y_2 = y_1 + \frac{h}{2}[k_1 + k_2] = 2.1909 + 0.05(1.8107 + 1.6472) = 2.3638,$$

the approximation of $y(1.2)$ and $|y(1.2) - y_2| = |2.3627 - 2.3638| = 0.0011$, is the absolute error. •

Example 6.16 *Use the Runge-Kutta method of order two (the Modified Euler's method) to find the approximate value of $y(0.2)$ for the given initial-value problem*

$$2\sin xy' - y\sin 2x = 0, \quad y(0) = 1, \quad \textit{with} \quad n = 2.$$

and compare your approximate solution with the exact solution $y(x) = e^{\sin x}$.

Solution. *Since* $f(x,y) = y\cos x$ *and* $x_0 = 0,\ y_0 = 1,\ h = (0.2-0)/2 = 0.1$,$x_1 = 0.1, x_2 = 0.2$ *then for* $i = 0$*, we have*

$$\begin{aligned} k_1 &= f(x_0, y_0) = f(0,1) = 1\cos 0 = 1, \\ k_2 &= f(x_1, y_0 + hk_1) = f(0.1, 1.1) = (1.1)\cos 0.1 = 1.0945, \end{aligned}$$

and by using these values, we have

$$y(0.1) \approx y_1 = y_0 + \frac{h}{2}\left[k_1 + k_2\right] = 1 + 0.05(1 + 1.0945) = 1.1047.$$

Similar manner, we have the other approximation for taking $i = 1$*, as follows*

$k_1 = f(x_1, y_1) = f(0.1, 1.1047) = 1.1047\cos(0.1) = 1.0992$ *and* $k_2 = f(x_2, y_1 + hk_1) = f(0.2, 1.2146) = (1.2146)\,c$

and by using these values, we have

$$y(0.2) \approx y_2 = y_1 + \frac{h}{2}\left[k_1 + k_2\right] = 1.1047 + 0.05(1.0992 + 1.1904) = 1.2192,$$

the approximation of $y(1.2)$ *and* $|y(0.2) - y_2| = |1.2198 - 1.2192| = 0.0006$*, is the absolute error.* •

The modified Euler's method is classified as a predictor-corrector method. This means that in the case of the modified Euler's method the initial-value problem is given by the formula

$$y_{i+1}^{(k)} = y_i + hf(x_i, y_i), \tag{6.27}$$

which is called the predictor and this is corrected by the repeated application of the formula

$$y_{i+1}^{(k+1)} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k)})\right], \quad k = 0, 1, 2, \ldots \tag{6.28}$$

for each $i = 0, 1, \ldots, n-1$. This is called the corrector. There are many predictor-corrector formulas and some provide much greater accuracy than the relatively the modified Euler's method. These methods however, require accurate estimates for a number of initial values of $y(x)$ before they can be used.

Program 6.3

MATLAB m-file for the Modified Euler's Method
function sol=mod1(fun1,a,b,y0,n)
$h = (b-a)/n; x = a + (0:n) * h; y(1) = y0;\ for\ k = 1:n$
$k1 = feval(fun1, x(k), y(k)); k2 = feval(fun1, x(k) + h, y(k) + h * k1);$
$y(k+1) = y(k) + h * (k1 + k2)/2;$ end; $sol = [x', y'];$

6.4 Exercises

1. Solve the following initial-value problems using the Euler's method.

(a) $y' = y + x^2, \quad x = 0(0.2)1, \quad y(0) = 1.$

(b) $y' = (x-1)(x+y+1), \quad x = 0(0.2)1, \quad y(0) = 1.$

(c) $x' = x + \sin(t), \quad t = 0.2(0.01)0.25, \quad x(0.2) = 1.5.$

2. Solve the following initial-value problems and compare the numerical solutions obtained with the Euler's method using the values of $h = 0.1$ and $h = 0.2$. Compare the results to the actual values.

(a) $y' = 1 + x^2, \quad 0 \le x \le 1, \quad y(0) = 0, \quad y(x) = \tan x.$

(b) $y' = 2(y+1), \quad 0 \le x \le 1, \quad y(0) = 2, \quad y(x) = e^{2x} - 1.$

(c) $y' = 2(y-1)^2, \ 0 \le x \le 1, \ y(0) = 2, \ y(x) = 2(1-x)/(1-2x).$

3. Solve the following initial-value problems using the Euler's method.

(a) $y' = y + x^2, \quad x = 0(0.2)1, \quad y(0) = 1.$

(b) $y' = (x-1)(x+y+1), \quad x = 0(0.2)1, \quad y(0) = 1.$

(c) $y' = y + \sin(x), \quad x = 0.2(0.01)0.25, \quad y(0.2) = 1.5.$

4. Solve the following initial-value problems and compare the numerical solutions obtained with the Euler's method using the values of $h = 0.1$ and $h = 0.2$. Compare the results to the actual values.

(a) $y' = 1 + x^2, \quad 0 \le x \le 1, \quad y(0) = 0, \quad y(x) = \tan x.$

(b) $y' = 2(y+1), \quad 0 \le x \le 1, \quad y(0) = 2, \quad y(x) = e^{2x} - 1.$

(c) $y' = 2(y-1)^2, \ 0 \le x \le 1, \ y(0) = 2, \ y(x) = 2(1-x)/(1-2x).$

5. Solve the following initial-value problems using the Taylor's method of order two.

(a) $y' = 2x^2 - y, \quad x = 0(0.2)1, \quad y(0) = -1.$

(b) $y' = 3x^2 y, \quad x = 0(0.2)1, \quad y(0) = 1.$

(c) $y' = x/y - x, \quad x = 0(0.2)1, \ y(0) = 2.$

Index