PREFACE

With its grounding in the "guiding pillars of Access, Equity, Equality, Affordability and Accountability," the New Education Policy (NEP 2020) envisions flexible curricular structures and creative combinations for studies across disciplines. Accordingly, the UGC has revised the CBCS with a new Curriculum and Credit Framework for Undergraduate Programmes (CCFUP) to further empower the flexible choice based credit system with a multidisciplinary approach and multiple/ lateral entry-exit options. It is held that this entire exercise shall leverage the potential of higher education in three-fold ways – learner's personal enlightenment; her/his constructive public engagement; productive social contribution. Cumulatively therefore, all academic endeavours taken up under the NEP 2020 framework are aimed at synergising individual attainments towards the enhancement of our national goals.

In this epochal moment of a paradigmatic transformation in the higher education scenario, the role of an Open University is crucial, not just in terms of improving the Gross Enrolment Ratio (GER) but also in upholding the qualitative parameters. It is time to acknowledge that the implementation of the National Higher Education Qualifications Framework (NHEQF), National Credit Framework (NCrF) and its syncing with the National Skills Qualification Framework (NSQF) are best optimised in the arena of Open and Distance Learning that is truly seamless in its horizons. As one of the largest Open Universities in Eastern India that has been accredited with 'A' grade by NAAC in 2021, has ranked second among Open Universities in the NIRF in 2024, and attained the much required UGC 12B status, Netaji Subhas Open University is committed to both quantity and quality in its mission to spread higher education. It was therefore imperative upon us to embrace NEP 2020, bring in dynamic revisions to our Undergraduate syllabi, and formulate these Self Learning Materials anew. Our new offering is synchronised with the CCFUP in integrating domain specific knowledge with multidisciplinary fields, honing of skills that are relevant to each domain, enhancement of abilities, and of course deep-diving into Indian Knowledge Systems.

Self Learning Materials (SLM's) are the mainstay of Student Support Services (SSS) of an Open University. It is with a futuristic thought that we now offer our learners the choice of print or e-slm's. From our mandate of offering quality higher education in the mother tongue, and from the logistic viewpoint of balancing scholastic needs, we strive to bring out learning materials in Bengali and English. All our faculty members are constantly engaged in this academic exercise that combines subject specific academic research with educational pedagogy. We are privileged in that the expertise of academics across institutions on a national level also comes together to augment our own faculty strength in developing these learning materials. We look forward to proactive feedback from all stakeholders whose participatory zeal in the teaching-learning process based on these study materials will enable us to only get better. On the whole it has been a very challenging task, and I congratulate everyone in the preparation of these SLM's.

I wish the venture all success.

Professor Indrajit Lahiri
Vice Chancellor

Netaji Subhas Open University

Four Year Undergraduate Degree Programme

Under National Higher Education Qualifications Framework (NHEQF) &
Curriculum and Credit Framework for Under Graduate Programmes

B. Sc. Mathematics (Hons.)

Programme Code : NMT

Course Type : Discipline Specific Core (DSC)

Course Title : Numerical Analysis

Course Code : 6CC-MT-03

Printed in accordance with the regulations of the
Distance Education Bureau of the University Grants Commission.

Netaji Subhas Open University

Four Year Undergraduate Degree Programme

Under National Higher Education Qualifications Framework (NHEQF) &

Curriculum and Credit Framework for Under Graduate Programmes

B. Sc. Mathematics (Hons.)

Programme Code : NMT

Course Type : Discipline Specific Core (DSC)

Course Title : Numerical Analysis

Course Code : 6CC-MT-03

Course Writer

Prof. Mridula Kanoria
Retd. Professor of Mathematics, C.U.
Professor & HOD, Mathematics, Sister Nivedita University

Course Editor

Dr. Diptiman Saha
Associate Professor of Mathematics, St. Xavier's College

Format Editing

Dr. Chandan Kumar Mondal
NSOU

Notification

Ananya Mitra

Registrar (Additional Charge)

Netaji Subhas
Open University

UG : Mathematics
(NMT)

Course Code : 6CC-MT-03
Course : **Numerical Analysis**

CONTENTS

Unit 1 ❐ Error Analysis

Structure

1.0 Objectives

After going through this unit one can able to learn about

- types of errors
- measurment of errors

1.1 Introdution

The process of solving physical or any scientific problems can be roughly divided into three phases. The first consists of constructing a mathematical model for the corresponding problem. This model could be in the form of differential equations or algebraic equations. In most cases, this mathematical model cannot be solved analytically, and hence a numerical solution is required. In which case, the second phase in the solution process usually consists of constructing an appropriate numerical model or approximation to the mathematical model. For example, an integral or a differential equation in the mathematical formulation will have to be approximated for numerical solution appropriately. A numerical model is one where everything in principle can be calculated using a finite number of basic arithmetic operations. The third phase of the solution process is the actual implementation and solution of the numerical model.

1.2 Reason of numerical Errors

It can be the combined effect of two kinds of error in a calculation.

- the first is caused by the finite precision of computations involving floating-point or integer values called **Round off** error
- The second usually called **Truncation** error is the difference between the exact mathematical solution and the approximate solution obtained when simplifications are made to the mathematical equations to make them more amenable to calculation. The term truncation comes from the fact that either these simplifications usually involve the truncation of an infinite series expansion so as to make the computation possible and practical, or because the least significant bits of an arithmetic operation are thrown away.

1.3 Measurement of Errorss

Numerical Errors usually measured in three ways, Absolute Error, Relative Error and Percentage Error.

Absolute Error : Absolute Error is the magnitude of the difference between the true value x and the approximate value x_a. Therefore absolute error is defined as the error between two values is defined as $E_a = |x - x_a|$, where x denotes the exact value and x_a denotes the approximation.

Relative Error: The relative error of x is the absolute error relative to the exact value. Look at it this way: if your measurement has an error of ± 1 inch, this seems to be a huge error when you try to measure something which is 3 inch long but when measuring distances on the order of miles, this error is mostly negligible. The definition of the relative error is $E_r = \frac{|x - x_a|}{|x|}$.

Note : Consider you try to measure a rod of length 10 cm, and found length as 9.98 cm from your scale. Here True value or actual value of the rod 10 cm and approximate value of the length of the rod is 9.98 cm. So, the absolute error will be (10 – 9.98) cm = 0.02 cm and the relative error will be $\frac{10 - 9.98}{10} = 0.002$.

Percentage error : One can express this error in percentage as $E_p = \frac{|x - x_a|}{|x|} \times 100,$ which gives the value 0.002 × 100 = 0.2 for the example taken here. This is called percentage error.

Example 1.3.1 : If $\pi = \frac{22}{7}$ is approximated as 3.14, find the absolute error, relative error and relative percentage error.

Solution: Absolute error $E_a = \left|\frac{22}{7} - 3.14\right|$

$$= \left|\frac{22 - 21.98}{7}\right|$$

$$= \left|\frac{0.02}{7}\right| = 0.002857$$

Relative error $E_r = \left|\frac{0.002857}{22/7}\right|$

$$= 0.0009$$

Relative percentage error = $E_p = E_r \times 100$

$$= 0.0009 \times 100$$

$$= 0.09\%$$

Example 1.3.2 : Compute the percentage error in the time period for $l = 1$ if the error in the measurement of l is 0.01.

Solution : Given the $T = 2\pi\sqrt{\frac{l}{g}}.$

Taking *log* of both sides we have,

$$\log T = \log 2\pi + \frac{1}{2}\log l - \frac{1}{2}\log g$$

$$\therefore \frac{dT}{T} = \frac{1}{2}\frac{dl}{l}$$

$$\frac{dT}{T} \times 100 = \frac{1}{2}\frac{dl}{l} \times 100 = \frac{0.01}{2 \times 1} \times 100 = 0.5\%$$

Now we will discuss some important types of Numerical Errors

- **Loss of significance**
- **Inherent errors**
- **Round-off error**
- **Truncation errors :**

(i) Loss of significance is an undesirable effect in calculations using finite-precision arithmetic such as floating-point arithmetic. It occurs when an operation on two numbers increases relative error substantially more than it increases absolute error, for example in subtracting two nearly equal numbers (known as **catastrophic cancellation**). The effect is that the number of significant digits in the result is reduced unacceptably. Ways to avoid this effect are studied in numerical analysis.

Example: As an example, consider the behavior of $f(x) = \sqrt{x^2 + 1} - 1$ as x approaches to 0. Evaluating this function at $x = 1.89 \times 10^{-9}$ using Matlab incorrectly returns the answer 0, which shows that too many significant digits have cancelled.

(ii) **Inherent errors:** This type of errors is present in the statement of the problem itself, before determining its solution. Inherent errors occur due to the simplified assumptions made in the process of mathematical modelling of a problem. It can also arise when the data is obtained from certain physical measurements of the parameters of the proposed problem.

Inherent errors can be minimized by taking better data on by using high precision computing aids. High precision refers to the number of decimal positions, i.e. the order of magnitude of the last digit in a value. For example the number 46.398 has a precision of 0.001 or 10^{-3}.

Example 1.3.3 : Which of the following numbers have greatest precision?

3.1201, 2.42, 5.320205.

Solution: In 3.1202, the precision is 10^{-4},

In 2.42, the precision is 10^{-2},

In 5.320205, the precision is 10^{-6}.

Hence the 5.320205 has the greatest precision.

(iii) **Round-off errors:** Generally, the numerical methods are carried out using calculator or computer. In numerical computation, all the numbers are represented by decimal fraction. Some numbers such as 1/3, 2/3, 1/7 etc. can not be represented by decimal fraction in finite numbers of digits. Thus, to get the result, the numbers should be rounded-off into some finite number of digits.

Again, most of the numerical computations are carried out using calculator and computer. These machines can store the numbers up to some finite number of digits. So in arithmetic computation, some errors will occur due to the finite representation of the numbers; these errors are called round-off error. Thus, round-off errors occur due to the finite representation of numbers during arithmetic computation. These errors depend on the word length of the computational machine.

Method of rounding off: To round off a number to n significant digits first truncate it to n digits: if truncated part is less than half a unit at last significant place then ignore it, if it is greater than half a unit at last significant place then add one to last significant digit: if it is exactly half a unit at last significant place then add one to it if it is odd. So absolute error is always minimum by this process which is less than or equal to half a unit at last significant figure (s.f) i.e. $\leq \frac{1}{2}\times 10^{-m}$ if approximation is done to m places after decimal. Sign of equality holds in the case when truncated part is exactly half a unit at last s.f .Reader may think that can't we do the reverse in this case i.e. if last s.f is even the we add one to it and ignore the other case? Because in this case also $E_a = \frac{1}{2}\times 10^{-m}$. But on a closure look we can identify that this make the last digit of the approximated number odd which attract more error in further calculation.

Example 1.3.4 : Round off the following numbers, to four significant digits

i) 23.4251 ii) 32.4250 iii) 24.87500 iv) 19.995 v) 437.261 vi) 19.36235

Solution: i) 23.43 ii) 32.42 iii) 24.88 iv) 20.00 v) 437.3 v) 19.36

Example 1.3.5 : Round off the number 54762 to four significant digits and then calculate absolute error, relative error and percentage error.

Solution: i) The given number is 54762 (= N)

After round off to four significant figures,

The given number would be 54760 (= N_1)

Absolute error $E_a = |54762 - 54760| = 2$

Relative error $E_r = \left|\frac{2}{54762}\right| = 3.652\times 10^{-5}$

Relative percentage error = $E_p = E_r \times 100$

$= 3.652 \times 10^{-5} \times 100$

$= 3.652 \times 10^{-3}$ %

Exercise 1.3.6 : Round off the following numbers to four significant digits and then calculate absolute error, relative error and percentage error.

i) 437.261 ii) 19.36235

(iv) **Truncation errors:** These errors occur due to the finite representation of an

Inherently infinite process. For example, the use of a finite number of terms in the infinite series to compute the value of $\cos x, \sin x, e^x$, etc.

The Taylor's series expansion of *sin x* is

$$\sin x = x - \frac{x^3}{3} + \frac{x^5}{5} - ....$$

This is an infinite series expansion. If only first five terms are taken to compute the value of *sin x* for a given *x*, then we obtain an approximate result. Here, the error occurs due to the truncation of the series. Suppose, we retain the first *n* terms, the truncation Error is given by

$$E_{trunc} \le \frac{x^{2n+1}}{(2n+1)!}$$

It may be noted that the truncation error is independent of the computational machine.

Example 1.3.7 : Find the number of terms of the exponential series such that their sum gives the value e^x correct to six decimal places at $x = 1$.

Solution: We know,

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + ... \frac{x^{n-1}}{(n-1)!} + R_n(x)$$

Where $R_n(x) = \frac{x^n}{n!} e^{\theta}$, $0 < \theta < x$.

Maximum absolute error (at $\theta = x) = \frac{x^n}{n!} e^x$ and maximum relative error is $\frac{x^n}{n!}$.

Hence $\left(e^x\right)_{\max}$ at $x = 1$ is $\frac{1}{n!}$.

For a six decimal accuracy at $x = 1$, we have

$$\frac{1}{n!} < \frac{1}{2} \times 10^{-6}$$

or, $$n! > 2 \times 10^6$$

which gives $n = 10$.

1.4 Summary

In this unit, the concept of Numerical errors, measurement of errors like absolute errors, relative errors, percentage error, loss of significant, inherent , round off and truncations errors are discussed with different examples.

1.5 Exercises

1) If 0.333 is the approximate value of $\frac{1}{3}$, find absolute, relative and percentage errors. (Ans: .00033, 0.00099, 0.99)

2) If $u = \frac{5xy^2}{z^3}$ and error in x, y, z be 0.001,0.002 and 0.003. Compute the relative error in u when $x = y = z = 1$. (Ans: .14)

3) Find the difference of $\sqrt{2.01} - \sqrt{2}$ correct to three digits.

(Ans: 3.53×10^{-3})

4) If $\Delta x = 0.005, \Delta y = 0.001$ be the absolution errors in $x = 2.11$ and $y = 4.15$, find the relative error in the computation of $x + y$. (Ans: 0.001 (approx.))

5) Use the series of $\log_e\left(\frac{1+x}{1-x}\right) = 2\left(x + x + \frac{x^3}{3} + \frac{x^5}{5} + ...\right)$ to compute the value of log $\log_e(1.2)$ correct to seven deciamal places and find the number of terms retained.

(Ans : $n \geq 2, 0.1823215$)

6) What do you understand by Inherent errors occurs in numerical computation?

7) Write process of rounding off?

Unit 2 ❐ Transcendental and Polynomial Equations

Structure

2.0 Objectives

After going through this unit one can able to learn about

- how to find the roots of non-linear equation by using different methods.
- the covergence of methods are also discussed.

2.1 Introduction

Determination of roots of algebraic and transcendental is a very important problem in science and engineering.

A function $f(x)$ is called algebraic if, to get the values of the function starting from the given values of x, we have to perform arithmetic operations between some real numbers and rational power of x.On the other hand, transcendental functions include all non-algebraic functions, i.e. $e^x, a^x, \log x, \sin x, \cos x, \sin^{-1} x, \cos^{-1} x$ etc. And others.

An equation $f(x) = 0$ is called algebraic or transcendental as $f(x)$ is algebraic or transcendental.

The equations $x^7 + 3x^2 + 7x + 1 = 0,\ x^3 + 8x + 7 = 0$ etc. are the examples of algebraic equations and on the other hand $e^x + 3\log x + \cos x = 0,\ e^{-4x} + x + \cot x = 0$ etc. are the examples of transcendental equation. Though we know some methods like Cardan's method, Euler's method, Ferrari's method, Descartes' method in algebra to solve algebraic equation up to fourth order. In general there is no closed form formula to evaluate the algebraic equation of degree greater than two.

The definition of roots of an equation can be given in two different ways:

Algebraically, a number *c* is called a root of an equation $f(x) = 0$ iff $f(c) = 0$ and geometrically, the real roots of the equation $f(x) = 0$ are the values of *x* where the graph of $y = f(x)$ meets the *x*-axis.

Throughout our discussion, we assume that

I. The function *f* (*x*) is continuous and continuously differentiable up to a sufficient number of times.

II. $f(x) = 0$ has no multiple root i.e., if α is a real root of $f(x) = 0,$ in a sufficiently small interval (*a, b*), then *f* (*a*) = 0 and either $f'(x) < 0$ or $f'(x) > 0$ in (a, b).

Most of the numerical methods, used to solve an equation are based on iterative techniques. Different numerical methods are available to solve the equation *f* (*x*) = 0. But each method has some advantage and disadvantage over another method. Generally, the following aspects are considered to compare the methods:

Convergence or divergence, rate of convergence, applicability of the method, amount of pre-calculation needed before application of the method. etc.

The process of finding the approximate values of the roots of an equation can be divided into two stages:

I. Location of the roots.

II. Computation of the values of the roots with the specified degree of accuracy.

The interval [*a, b*] is said to be the location of a real root *c* if *f* (*c*) = 0 for $a < c < b$. There are two methods used to locate the real roots of an equation

I. Graphical method

II. Method of tabulation which is an analytic method.

Graphical method

● In this method the graph of $y = f(x)$ is drawn in rectangular co-ordinate system. Then the points at which graph meets the x –axis are the location of the roots of the equation $f(x) = 0$.

As an example, we consider the equation $x^2 + x - 1 = 0.$ We draw the graph of $y = x^2 + x - 1$ with respect to $x'0x, y'oy$ as rectangular axes, which meets the x-axis at A and $A'.$ Thus the equation has two real roots, one is positive and other is negative. From the graph it is clear that the co-ordinate of A is lies between 0.6 and 0.7 and that of A' is between -1.6 to -1.7. Thus 0.6 is an approximate value of the positive root $\alpha'(\text{say}).$ and –1.6 is an approximate value of the negative root $\alpha'(\text{say}).$

● If $f(x)$ is not simple, rather complicated in form, we rewrite the equation $f(x)$ as $\theta_1(x) = \theta_2(x),$ where $\theta_1(x)$ and $\theta_2(x)$ are simple functions such that, we can draw conveniently the graphs of $y = \theta_1(x)$ and $y = \theta_2(x)$ with respect to rectangular axes. Then the x-co-ordinate of the point of intersection of the graphs give the location of the real roots of the equation $f(x) = 0.$

As an example, we consider an equation $x^3 - 4x - 2 = 0,$ we rewrite the equation as $x^3 = 4x + 2.$ The graphs $y = x^3$ and $y = 4x - 2$ are drawn with respect to the rectangular axes. From the graph it is seen that the roots are in [–2, –1], [–1, 0], [2,3].

DISADVANTAGE :

The graphical method to locate the roots is not very useful. Because the drawing of the location of the function $y = f(x)$ is itself complicated. But it makes possible to roughly determine the interval of the roots. Then an analytic method is used to locate the root.

METHOD OF TABULATION

This method depends on the continuity of the function $f(x)$. Before applying the tabulation method, the following nature should be noted.

Theorem 2.1.1 : If $f(x)$ is continuous in the interval (a, b) and if $f(a)$ and $f(b)$

have the opposite signs, then at least one real root of the equation $f(x) = 0$ lies within the interval (a, b).

Geometrically we can explain the theorem as:

Let, $f(x) > 0$ and $f(b) < 0$. Then from the graph we can say that there must be a point in (a, b) such that $f(x) = 0$

If the curve $y = f(x)$ touches the x–axis at some point, say at $x = c$ then c is a root of $f(x) = 0$, though $f(a)$ and $f(b)$ may have same sign where $a < c < b$. For example $f(x) = (x - 3)^2$, touches the x–axis at $x = 3$. Although $f(2.5) > 9$ and $f(3.5) > 0$ but $x = 3$ is a root of the equation $f(x) = 0$.

A trial method for tabulation is as follows:

From the table of signs of (x), setting $x =$ $x = 0, \pm, 1, \pm 2, \ldots\ldots\ldots$ If the signs of $f(x)$ changes its signs for two consecutive values of x then at least one root lies between these two values.

Example 2.1.2 : Find the location of the roots of the equation $x^2 + x - 1 = 0.$

Solution: we form a table :

x	0	-1	1	0.5	-0.5	-1.6	-1.7
f (x)	-	-	+	-	-	-	+

Since $\deg f(x) = 2,$ the $f(x)$ has two roots. Since $f(1) > 0$ and $f(0.5) < 0,$ then the location of one root is (0.5, 1). Also $f(-1.6) < 0$ and $f(-1.7) > 0.$ Then the location of the other root is (–1,6, –1.7).

Example 2.1.3 : Find the number of real roots of the equation $3^x - 3x - 2 = 0$ and locate them.

Solution : $f(x) = 3^x - 3x - 2.$ The domain of definition of the function is $(-\infty, \infty).$

we form a table :

x	$-\infty$	0	1	∞
Sign of $f(x)$	+	-	-	+

$f(x) = 0$ has two real roots, since the function has twice changes sign, among them one is negative root and other is greater than one.

A new table with small intervals of the location of the root is constructed in the following:

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

Then the roots are in (–1, 0) and (1, 2).

ORDER OF CONVERGENCE:

Assume that the sequence $\{x_n\}$ of numbers to α and let $\in_n = \alpha - x_n$ for $n \geq 0.$ If there exists two positive constants A & p such that $\lim_{n\to\infty} \frac{\in_{n+1}}{\in_n^p} = A.$ Then the sequence is said to converge to α wth the order of convergence p. The number A is called the asymptotic error constant.

If p = 1, the error of convergence of $\{x_n\}$ is called linear and if p = 2, the error of convergence of $\{x_n\}$ is called quadratic etc.

2.2 Iteration method or Fixed point iteration

Let $f(x)$ be a continuous function on the interval $[a,b]$ and the equation $f(x)=0$ has at least one root on $[a,b]$. The equation $f(x)=0$ can be written in the form $x=\varphi(x).....(1)$

Thus a root ξ of the given equation satisfies $\xi=\varphi(\xi)$. Therefore the point ξ remains fixed under the mapping φ and so a root of the equation is a fixed point of φ.

$\varphi(x)$ is called the iteration function. Here we also assume that $\varphi(x)$ is continuously differentiable in $[a,b]$.

Using graphical or tabulation method, we first find a location or crude approximation $[a_0,b_0]$ of a real root ξ (say) of $f(x)=0$ and let $x=x_0\,[a_0 \leq x_0 \leq b_0]$ be the initial

approximation of ξ. Thus ξ satisfies the equation $\xi = \varphi(\xi).....(2)$.

Putting $x = x_0$ in (1), we get first approximation of ξ as $x_1 = \varphi(x_0)$, and then the successive approximations are calculated as: $x_2 = \varphi(x_1)$, $x_3 = \varphi(x_2)$,....., $x_{n+1} = \varphi(x_n)....(3)$

The above iteration is generated by the formula $x_{n+1} = \varphi(x_n)$ and is called the iteration formula, where x_n is the n-th approximation of the root ξ of $f(x) = 0$.

These successive iterations are repeated till the approximate numbers $x_n's$ converges to the root with desired accuracy, i.e. $|x_{n+1} - x_n| < \in$, where $\in$ is a sufficiently small number.

The sequence $\{x_n\}$ of iterations or the successive better approximations may or may not be converge to a limit. If $\{x_n\}$ converges, then it converges to ξ and the number of iterations required depends upon the desired degree of accuracy of the root ξ.

CONVERGENCE OF METHOD OF ITERATION:

The presentation of $f(x) = 0$ as $x = \varphi(x)$ is not unique, therefore the convergence of $\{x_n\}$ depends upon the nature of $\varphi(x)$. Now we investigate about the nature of $\varphi(x)$ which yields a convergent sequence $\{x_n\}$.

By Lagrange's mean value theorem we get,

$|\xi - x_1| = |\varphi(\xi) - \varphi(x_0)| = |\xi - x_0||\varphi'(\varepsilon_1)|$ where $x_0 < \varepsilon_1 < \xi$

$|\xi - x_2| = |\varphi(\xi) - \varphi(x_1)| = |\xi - x_1||\varphi'(\varepsilon_2)|$ where $x_0 < \varepsilon_2 < \xi$

.....

$|\xi - x_{n+1}| = |\varphi(\xi) - \varphi(x_n)| = |\xi - x_n||\varphi'(\varepsilon_n)|$ where $x_0 < \varepsilon_n < \xi$

Thus, $|\xi - x_{n+1}| = (\xi - x_n)|\varphi'(\varepsilon_n)| = |\xi - x_0|\left|\varphi'^{(\varepsilon_1)}\right|\left|\varphi'^{(\varepsilon_2)}\right|....\left|\varphi'^{(\varepsilon_n)}\right|$

Assuming , $|\varphi'(x)| < \rho$ in $a_0 \le x \le b_0$ we have

$|\xi - x_{n+1}| \le |\xi - x_0|\rho^n$

Thus,

$\lim_{n\to\infty} |\xi - x_{n+1}| \le \lim_{n\to\infty} |\xi - x_0|\rho^n \to 0$ *if* $p < 1$, i.e. $|\varphi(x)| < 1$

$\to \infty$ *if* $\rho > 1, i.e. |\varphi'(x)| > 1$

Therefore the method is convergent for $|\varphi'(x)| \le \rho < 1$ in $[a_0, b_0]$.

ESTIMATION OF ERROR:

Let, ξ be an exact root of the equation $x = \varphi(x)$ and $x_{n+1} = \varphi(x_n)$.

Therefore, $|\xi - x_n| = |\varphi(\xi) - \varphi(x_{n-1})| = |\xi - x_{n-1}||\varphi'(c)|$, , where $x_{n-1} < c < \xi$

$\le l|\xi - x_{n-1}|$, ,[where, $|\varphi'(c)| \le l < 1$]

$\le l\{|\xi - x_n| + |x_n - x_{n-1}|\}$

After rearrangement, this relation becomes

$|\xi - x_n| \le \frac{l}{1-l}|x_n - x_{n-1}| \le \frac{l^n}{1-l}|x_1 - x_0|$

Let the maximum number of iteration needed to achieve the accuracy ε be $N(\varepsilon)$. Then

$$\frac{l^N}{1-l}|x_1 - x_0| \le \varepsilon, i.e. N(\varepsilon) \ge \frac{\log \frac{\varepsilon(1-l)}{|x_1 - x_0|}}{\log l}$$

For $l \le 0.5$, the estimation of the error is given by the following simple form :

$|\xi - x_n| \le |x_n - x_{n-1}|$

ORDER OF CONVERGENCE:

The convergence of an iteration method depends on the suitable choice of the iteration function $\varphi(x)$ and the initial guess x_0.

Let, $\{x_n\}$ converges to the exact root α, so that $\xi = \varphi(\xi)$.

Thus $x_{n+1}-\xi=\varphi(x_n)-\varphi(\xi)$. Let, $\varepsilon_{n+1}=x_{n+1}-\xi$. Note that $\varphi'(x)\neq 0$. Then the above relation becomes

$$\varepsilon_{n+1}=\varphi(\varepsilon_n+\xi)-\varphi(\xi)$$

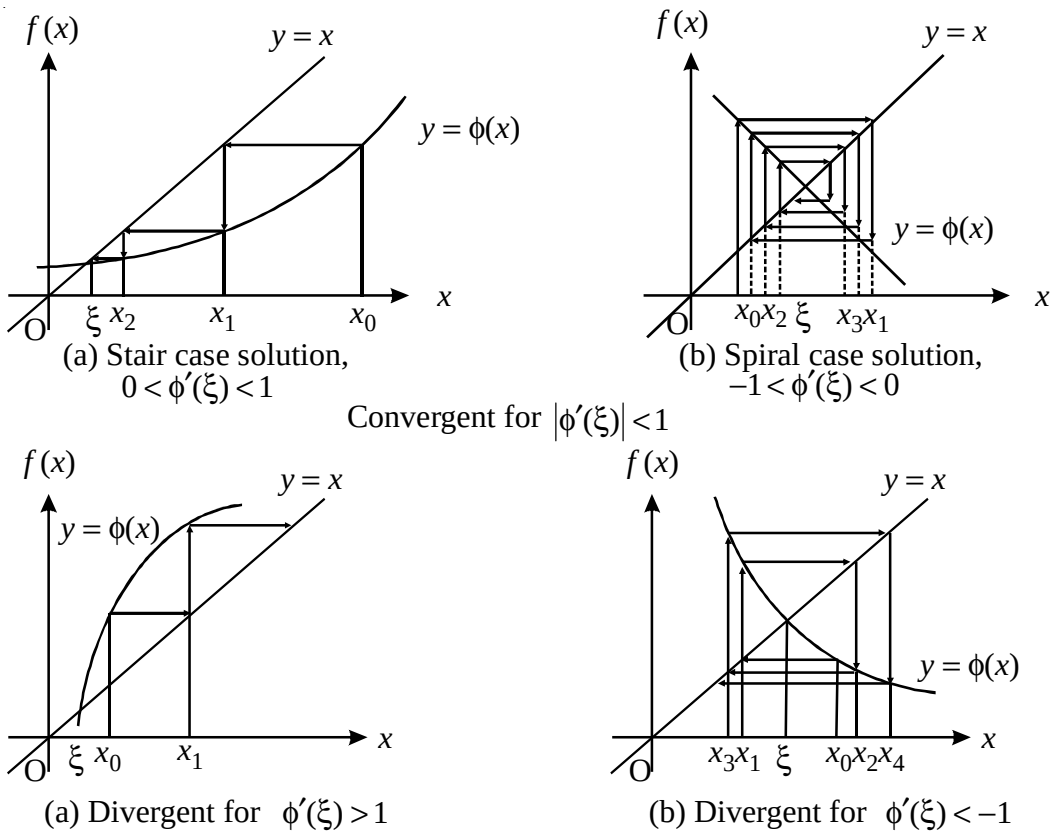
$$=\varepsilon_n\varphi'(\xi)+\frac{1}{2}\varepsilon_n^2\varphi''(\xi)+.....$$

$$=\varepsilon_n\varphi'(\xi)+o\left(\varepsilon_n^2\right)$$

i.e. $\dfrac{\varepsilon_{n+1}}{\varepsilon_n}=\varphi'(\xi)\neq 0$

hence the order of convergence of the iteration method is linear.

GEOMETRICAL INTERPRETATION : The geometrical meaning s of the fixed-point iteration in different cases are illustrated by Figure.



(a) Stair case solution, $0<\phi'(\xi)<1$

(b) Spiral case solution, $-1<\phi'(\xi)<0$

Convergent for $|\phi'(\xi)|<1$

(a) Divergent for $\phi'(\xi)>1$

(b) Divergent for $\phi'(\xi)<-1$

Fig 2.1 : Illustration for Fixed-point iteration

ADVANTAGE AND DISADVANTAGE:

The disadvantage of this method is that a pre-calculation is required to re-write $f(x)=0$ to $x=\varphi(x)$ in such a way that $|\varphi'(x)|<1$.

The advantage of this method is that the operation carried out at each stage are of same kind, and this makes easier to develop computer program.

2.3 BISECTION METHOD

It is an iterative method and is based on a well-known theorem which states that if $f(x)$ be a continuous function in a closed interval $[a,b]$ and $f(a)f(b)<0$, then $\exists$ at least one real root of the equation $f(x)=0$, between *a* and *b*. If further $f'(x)$ exists and $f'(x)$ maintains same sign in $[a,b]$, i.e. $f(x)$ is strictly monotonic, then there is only one real root of $f(x)=0$ in $[a,b]$. This method is nothing but a repeated application of the above theorem.

First we consider a sufficiently small interval $[a_0,b_0]$, by graphical or tabulation method , in which $f(a_0)f(b_0)<0$ and $f'(x)$ maintains same sign in $[a_0,b_0]$, then there is only one real root of $f(x)=0$, in $[a_0,b_0]$. Now divide the interval $[a_0,b_0]$ into two equal intervals $[a_0,c]$ and $[c,b_0]$ where $c=\frac{a_0+b_0}{2}$. If $f(c)=0$, then *c* is an exact root of the equation. If $f(c)\neq 0$ then the root lies either in $[a_0,c]$ or in $[c,b_0]$. If $f(a_0)f(c)<0$ then we take the interval $[a_0,c]$ as the new interval, otherwise we take $[c,b_0]$. Let the new interval be $[a_1,b_1]$ and use the same process to select the next new interval. In the next step, let the new interval be $[a_2,b_2]$. The process of bisection is continued until either the midpoint of the interval is a root, or the length (b_n-a_n) of the interval $[a_n,b_n]$ is sufficiently small. The number a_n and b_n are approximate roots of the equation $f(x)=0$. Finally $x_n=\frac{a_n+b_n}{2}$ is taken as the approximate value of the root α.

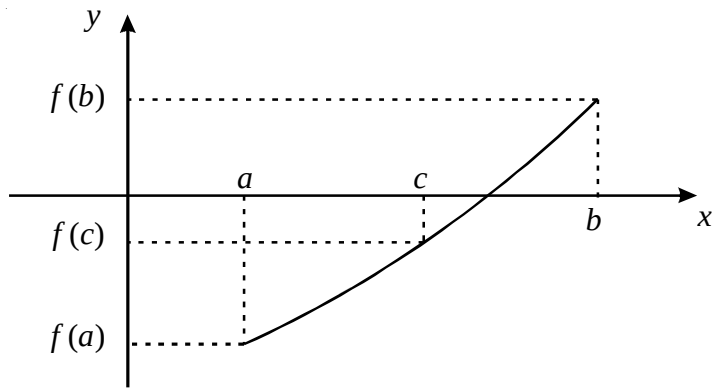


Fig 2.2 : Illustration for Bisection method

Now the length of the interval $[a_1, b_1]$ is $\frac{b_0 - a_0}{2}$ and the length of the interval $[a_2, b_2]$ is $\frac{b_0 - a_0}{2^2}$ and at the n-th step the length of the interval $[a_n, b_n]$ is $\frac{b_0 - a_0}{2^n}$. In the final step $\alpha = \frac{a_n + b_n}{2}$ is chosen as root, then the length of the interval being $\frac{b_0 - a_0}{2^{n+1}}$ and hence the error does not exceed $\frac{b_0 - a_0}{2^{n+1}}$.

Thus, if ε be the error at the n-th step then the lower bound of n is obtained from the following relation

$$\frac{|b_0 - a_0|}{2^{n+1}} \leq \varepsilon.$$

CONVERGENCY: let ε_{n+1} be the error in approximating α by x_{n+1}, then $\varepsilon_{n+1} = |\alpha - x_{n+1}| < |b_n - a_n| = \frac{b_0 - a_0}{2^n} \to 0$ as $n \to \infty$. Thus the iterative method must be convergent. To get a root of $f(x) = 0$ correct up to p-significant figures, we are to go up to q-th iteration so that x_q and x_{q+1} have same p-significant figures.

DISADVANTAGE : This method is very slow, but it is very simple and will converge surely to the exact root. So the method for any function only if the function is continuous within the interval [*a, b*], where the root lies.

Example 2.3.1 : Find a root of the equation $x^2 + x - 7 = 0$ by bisetion method, correct up to two decimal places.

Solution. Let $f(x) = x^2 + x - 7.$

$f(2) = -1 < 0$ and $f(3) = 5 > 0.$ So, a root lies between 2 and 3.

Left end point Right and point Midpoint

n	a_n	b_n	x_{n+1}	$f(x_{n+1})$
0	2	3	2.5	1.750
1	2	2.5	2.250	0.313
2	2	2.250	2.125	-0.359
3	2.125	2.250	2.188	-0.027
4	2.188	2.250	2.219	0.143
5	2.188	2.219	2.204	0.062
6	2.188	2.204	2.196	0.018
7	2.188	2.196	2.192	-0.003
8	2.192	2.196	2.194	0.008
9	2.192	2.194	2.193	0.002
10	2.192	2.193	2.193	0.002

Therefore, the root is 2.19 correct up to two decimal places.

Another popular method is the regula falsi method. This method was developed because the bisection method converges at fairly slow speed. In general regula falsi method is faster than bisection method.

2.4 Regula Flasi Method

This method is also known as *method of false position, Method of chords, method of linear interpolation.*

Let a root of the equation $f(x) = 0$ be lies in the interval $[a,b]$, i.e. $f(a)\, f(b) < 0.$ The idea of this method is that on a sufficiently small $[a,b]$, the arc of the $y = f(x)$ is replaced by the chord joining the points $(a, f(a))$ and $(b, f(b))$. The abscissa of the point of intersection of the chord and the *x*-axis is taken as the approximate value of the root.

Let, $x_0 = a$ and $x_1 = b$. The equation of the chord joining the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ is

$$\frac{y - f(x_0)}{f(x_0) - f(x_1)} = \frac{x - x_0}{x_0 - x_1} \ldots\ldots(1)$$

To find the point of intersection, set $y = 0$ in (1) and let $(x_2, 0)$ be the point. Then,

$$\frac{0 - f(x_0)}{f(x_0) - f(x_1)} = \frac{x_2 - x_0}{x_0 - x_1}$$

Therefore, $x_2 = x_0 - \frac{f(x_0)(x_1 - x_0)}{f(x_1) - f(x_0)} \ldots\ldots(2)$

This is the second approximation of the root. Now if $f(x_2)$ and $f(x_0)$ are opposite signs then the root lies between x_0 and x_2 and replace x_1 by x_2 in (2). Then the next approximation is obtained as :

$$x_3 = x_0 - \frac{f(x_0)(x_2 - x_0)}{f(x_2) - f(x_0)}$$

If $f(x_2)$ and $f(x_1)$ are opposite signs then the root lies between x_1 and x_2 and the new approximation is obtained as:

$$x_3 = x_2 - \frac{f(x_2)(x_1 - x_2)}{f(x_1) - f(x_2)}$$

The procedure is repeated till the root is obtained to the desired accuracy. If the n-th approximate root x_n lies between a_n and b_n, then the approximate root is thus obtained as :

$$x_{n+1} = a_n - \frac{f(a_n)(b_n - a_n)}{f(b_n) - f(a_n)} \ldots\ldots(3)$$

GEOMETRICAL INTERPRETATION :

The illustration of the method is shown Figure where ξ is the root of the equation $f(x) = 0$.

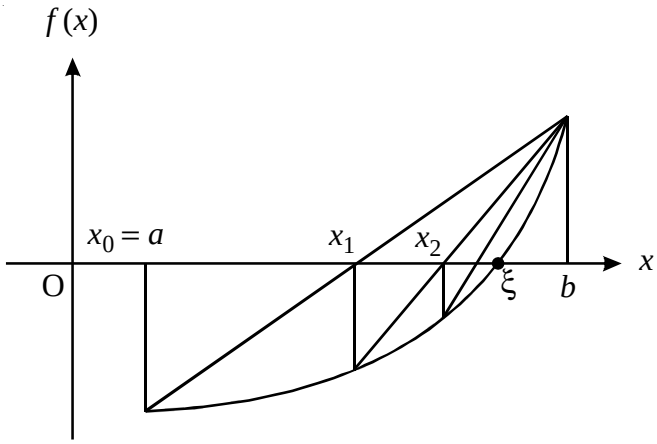


Fig 2.3 : Illustration for Regula-falsi method

CONVERGENCE OF REGULA FALSI METHOD:

As $f(a_n) f(b_n) < 0$, considering the proper sign of $f(a_n)$ and $f(b_n)$ we can write the equation (3) as follows:

$$x_{n+1} = a_n - \frac{f(a_n)(b_n - a_n)}{f(b_n) - f(a_n)}$$

$$\text{or } x_{n+1} = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)} \ldots\ldots(4)$$

Since, $x_n = a_n$ or b_n, we have for both relation of (4) as

$$x_{n+1} = x_n - \frac{f(x_n)(b_n - a_n)}{f(b_n) - f(a_n)}$$

Or, $(x_n - x_{n+1})\big(f(b_n) - f(a_n)\big) = f(x_n)(b_n - a_n)$

Or, $(x_n - x_{n+1})(b_n - a_n) f'(\alpha_n) = f(x_n)(b_n - a_n)$ when $a_n < \alpha_n < b_n$

Or, $\big[(\alpha - x_{n+1}) - (\alpha - x_n)\big] f'(\alpha_n) = f(x_n) - f(\alpha) = (x_n - \alpha) f'(\alpha'_n)$,

$\big[\text{since, } f(\alpha) = 0\big]$, where $\text{Min}\{x_n, \alpha\} < \alpha'_n < Max\{x_n, \alpha\}$

Or, $(\alpha - x_{n+1}) = (\alpha - x_n)\dfrac{f'(\alpha_n) - f'(\alpha'_n)}{f'(\alpha)}$, where $a_0 < \alpha_n, \alpha'_n < b_0 \ldots\ldots (5)$

The approximation lies in $[a_0, b_0]$ and $f'(x)$ is continuous, then there exist two numbers *m, M* such that

$0 < m \le |f'(x_n)| \le M$ for all $x \in [a_0, b_0]$.

Then from (5) we get, $|(\alpha - x_{n+1})| \le \dfrac{M-m}{m}|(\alpha - x_n)|$

Now putting $n = n-1,\ n-2, \ldots, 2, 1, 0$ for n successively and multiplying $(n+1)$ relations we get :

$$\varepsilon_{n+1} = |(\alpha - x_{n+1})| \le \left(\frac{M-m}{m}\right)^{n+1} |(\alpha - x_0)|$$

If we choose the interval $[a_0, b_0]$ such that $\left|\dfrac{M-m}{m}\right| < 1, i.e. M < 2m,$

Then $\lim\limits_{x\to\infty} \varepsilon_{n+1} = \lim\limits_{x\to\infty} |(\alpha - x_{n+1})| = 0$

Therefore the method is convergent. Thus for the convergence of the Regula Falsi Method, the interval $[a_0, b_0]$ must be very small.

ADVANTAGE:

The advantage of this method is that it is very simple and the sequence $\{x_n\}$ is sure to converge. The another advantage of this method is that it does not require the evaluation of derivatives and pre-calculation.

DISADVANTAGE:

The method is very slow and not suitable for hand calculation.

Example 2.4.1 : Find a root of the equation $x^3 + 2x - 2 = 0$ using Regula-Falsi method, correct up to three decimal places.

Solution. Let $f(x) = x^3 + 2x - 2$. $f(0) = -2 < 0$ and $f(1) = 1 > 0$. Thus, one root lies between 0 and 1. The calculations are shown in the following table.

n	left end point a_n	right end point b_n	$f(a_n)$	$f(b_n)$	x_{n+1}	$f(x_{n+1})$
0	0.0000	1.0	–2.0000	1.0	0.6700	–0.3600
1	0.6700	1.0	–0.3600	1.0	0.7570	–0.0520
2	0.7570	1.0	–0.0520	1.0	0.7696	–0.0072
3	0.7696	1.0	–0.0072	1.0	0.7707	–0.0010
4	0.7707	1.0	–0.0010	1.0	0.7709	–0.0001

Therefore, a root of the equation is 0.771 correct up to three decimal places.

2.5 Netwon-Raphson Method

This is also an iterative method and is used to find isolated roots of an equation $f(x)=0$. The object of this method is to correct the approximate root x_0 (say) successively to the exact root *a*. Initially, a crude approximation of a small interval $[a_0, b_0]$ is found out in which only one root *a* (say) of $f(x)=0$.

Let, $x = x_0 \left(a_0 \le x_0 \le b_0\right)$ is an approximation of the root *a* of the equation $f(x)=0$. Let, *h* be a small correction on x_0, then $x_1 = x_0 + h$ is the correct root.

Using Taylor's series expansion,

$f(x_1) = f(x_0 + h) = f(x_0) + hf'(x_0) + \ldots = 0$, since x_1 is a root of $f(x)=0$

Neglecting the second and the higher order derivatives, the above equation reduces to-

$$f(x_0) + hf'(x_0) = 0$$

Or, $h = -\dfrac{f(x_0)}{f'(x_0)}$

Therefore, $x_1 = x_0 + h = x_0 - \dfrac{f(x_0)}{f'(x_0)} \ldots\ldots(1)$

Further if h_1 be the correction on x_1, then $x_2 = x_1 + h_1$ is the correct root of $f(x)=0$.

Then using the previous process we get,

$$h_1 = -\frac{f(x_1)}{f'(x_1)}$$

Therefore, $x_2 = x_1 + h_1 = x_1 - \frac{f(x_1)}{f'(x_1)}$

Processing in this way, we get $(n+1)$ th corrected root as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \ldots\ldots(2)$$

This expression generates a sequence of approximate values $x_1, x_2, x_3, \ldots\ldots, x_n, \ldots.$ each successive term of which is closer to the exact value of the root *a*. The method will terminate when $|x_{n+1} - x_n|$ becomes very small.

In this method the arc of the curve is replaced by the tangent to the curve, hence this method is sometimes called method of tangent.

Note : the Newton Raphson method may also used to find a complex root of an equation when the initial guess is taken as a complex number.

GEOMETRICAL INTERPRETATION:

The geometrical interpretation of this method is shown in the figure 1. In this method, a tangent is drawn at $(x_0, f(x_0))$ to the curve $y = f(x)$. The tangent cuts the x-axis at $(x_1, 0)$. Again the tangent is drawn at $(x_1, f(x_1))$, which cuts the x-axis at $(x_2, 0)$. This process is continued until $x_n = \xi$ as $n \to \infty$. .

The choice of initial guess of this method is very important. If the initial guess is near the root then the method

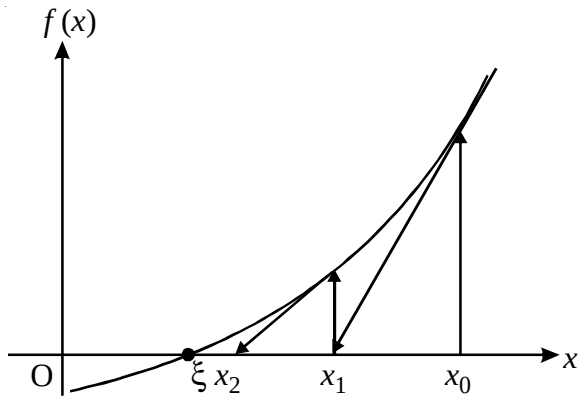


Fig 2.4 : Geometrical interpretation of Newton-Raphson method

converges very fast. If it is not so near the root or if the starting point is wrong, then the method may lead to an endless cycle.

This illustrated in figure2. In this figure the initial guess x_0 gives the fast convergence to the root, the initial guess y_0 leads to an endless cycle and the initial guess z_0 gives a divergent solution.

Even if the initial guess is not close to the exact root, the method may diverge. To chose the initial guess the following rule may be followed. If $f(b)\,f''(x) < 0$ the initial guess be $x_0 = b$ and if $f(a)\,f''(x) < 0$ then $x_0 = a$ be the initial guess.

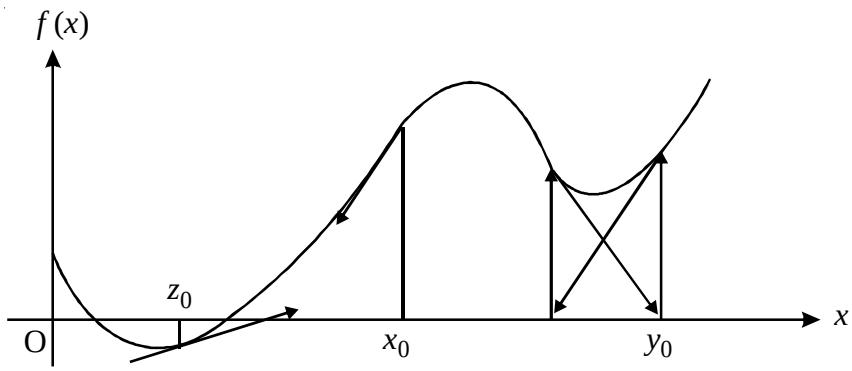


Fig: Illustration of the choice of the initial guess
of the Newton-Raphson method.

CONVERGENCE OF NEWTON RAPHSON METHOD:

Comparing with the iteration method, we may assume the iteration function as:

$$\varphi(x) = x - \frac{f(x)}{f'(x)}$$

Thus the above sequence will be convergent, if and only if

$$|\varphi'(x)| = \left|1 - \frac{f'(x)^2 - f(x)\,f''(x)}{f'(x)^2}\right|$$

i.e. $\left|\frac{f(x)\,f''(x)}{f'(x)^2}\right|<1, i.e. \left|f'(x)^2\right|>\left|f(x)\,f''(x)\right|$

RATE OF CONVERGENCE OF N-R METHOD:

Let, ξ be a root of the equation $f(x)=0.$ Then, $f(\xi)=0.$ The iteration scheme for NR-method is

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

Let, $x_n=\varepsilon_n+\xi.$ Then from the above relation we get-

$$\varepsilon_{n+1}+\xi=\varepsilon_n+\xi-\frac{f(\varepsilon_n+\xi)}{f'(\varepsilon_n+\xi)}$$

$$\text{Or, } \varepsilon_{n+1}=\varepsilon_n-\frac{f(\xi)+\varepsilon_n f'(\xi)+\frac{\varepsilon_n^2}{2}f''(\xi)+.....}{f'(\xi)+\varepsilon_n f''(\xi)+\frac{\varepsilon_n^2}{2}f''(\xi)+.....}$$

$$\text{Or, } \varepsilon_{n+1}=\varepsilon_n-\frac{\varepsilon_n+\frac{\varepsilon_n^2 f''(\xi)}{2f'(\xi)}+....}{1+\varepsilon_n\frac{f''(\xi)}{f'(\xi)}+.....}$$

$$\text{Or, } \varepsilon_{n+1}=\varepsilon_n-\left(\varepsilon_n+\frac{\varepsilon_n^2}{2}\frac{f''(\xi)}{f'(\xi)}+.....\right)\left(1-\varepsilon_n\frac{f''(\xi)}{f'(\xi)}+.....\right)$$

$$\text{Or , } \varepsilon_{n+1}=\frac{\varepsilon_n^2}{2}\frac{f''(\xi)}{f'(\xi)}+0\left(\varepsilon_n^3\right)$$

Neglecting the terms of order $\varepsilon_n{}^3$ and higher power the expression becomes

$$\varepsilon_{n+1}=A\varepsilon_n^2, \text{ where } A=\frac{f''(\xi)}{2f'(\xi)}$$

This relation shows that NR method has quadratic convergence or second order convergence.

Example 2.5.1 : Use Newtwon-Raphson method of find a root of the equation $x^3 + x - 1 = 0.$

Solution. Let $f(x) = x^3 + x - 1.$ Then $f(0) = -1 < 0$ and $f(1) = 1 > 0.$ So one root lies between 0 and 1. Let $x_0 = 0$ be the initial root.

The iteration scheme is

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

$$= x_n - \frac{x_n^3 + x_n - 1}{3x_n^2 + 1} = \frac{2x_n^3 + 1}{3x_n^2 + 1}.$$

The sequence $\{x_n\}$ for different values of *n* is shown below.

n	x_n	x_{n+1}
0	0	1
1	1	0.7500
2	0.7500	0.6861
3	0.6861	0.6823
4	0.6823	0.6823

Therefore, a root of the equation is 0.682 correct upo to three decrimal places.

Example 2.5.2 : Find an iteration scheme to find the *k*th root of a number *a*.

Solution. Let *x* be the *k*th root of *a*. That is $x = a^{1/k}$ or $x^k - a = 0.$

Let $f(x) = x^k - a.$ The iteration scheme is

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

or, $x_{n+1} = x_n - \frac{x_n^k - a}{kx_n^{k-1}} = \frac{kx_n^k - x_n^k + a}{kx_n^{k-1}}$

$$= \frac{1}{k}\left[(k-1)\, x_n + \frac{a}{x_n^{k-1}} \right].$$

2.6 Summary

In this unit we have studied how to calculate the roots of a transcendental equations and polynomial equations by the methods of tabulation, graphical, fixed point iteration, bisection ,Regula Falsi and Newton-Raphson. Their convergence analysis have also been studied.

2.7 Exercises

1. Solve the equation $x \tan x = -1$ by Regula falsi method starting with $x_0 = 2.5$ and $x_1 = 3.0$ correct upto three decimal places.

2. Obtain the a root for each of the following equations using bisection method, regula-falsi method and Newto-Raphson method

i) $x^3 + 2x^2 - x + 7 = 0$

ii) $\sin x = 10(x-1)$

iii) $x - \cos x = 0$

3. Describe Newton-Raphson method for computing a simple real root of an equation $f(x) = 0$. Give a geometrical interpretation of the method. Prove that the Newton-Raphson method converges quadratically.

4. Use Newton-Raphson method to find the value of the following terms

i) $\sqrt{35}$ ii) $\sqrt[3]{24}$

Ans. i) 5.916080, ii) 2.884499

Unit 3 ❐ System of linear algebratic equations

Strucure

3.0 Objectives

After studying this unit one can

- get an idea of finding the solutions of system of linear equations by using direct methods and iterative methods.

3.1 Introduction

A **linear equation** in variables $x_1, x_2, ..., x_n$ is an equation of the form

$a_1x_1 + a_2x_2 + ... + a_nx_n = b$

where $a_1, a_2, ..., a_n$ and $b$are constant real or complex numbers. The constant a_1 is called the c**oefficient** of x_i; and b is called the **constant term** of the equation.

A **system of linear equations** (or **linear system**) is a finite collection of linear equations in same variables. For instance, a linear system of n equations in n variables $x_1, x_2, ..., x_n$ can be written as

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ ..... \\ ..... \\ a_{n_1}x_1 + a_{22}x_2 + ... + a_{nn}x_n = b_n \end{cases} \qquad (3.1.1)$$

The above system can be written in the form $AX = B$

where $A = \left[a_{ij} \right]_{n \times n} \left(i, j = 1, 2, 3.....n \right)$ is a non-singular matrix and $B = \left[b_i \right]' = \left(i = 1, 2, 3, ..., n \right)$

Two types of methods are availavle.

i) Exact methods or Direct method

ii) Iterative methods

When A is of moderate order with co-efficients most non-zero, then usually exact or direct methods are used. Order of A is usually < 200 and the linear system is called ***dense***.

When A is of large order and most co-efficients zero, then iterative methods are used. A is sparse and order of A is sometimes as large as 10^6.

Exact or direct methods : Cramer's rules, Gaussian elimination method,

Gauss Jordan Method etc

Iterative methods : Method of simple iteration, Gauss-Seidal iteration method

Theorem 3.1.1 : *Any system of linear equations has one of the following exclusive conclusions.*

(a) No solution.

(b) Unique solution.

(c) Infinitely many solutions.

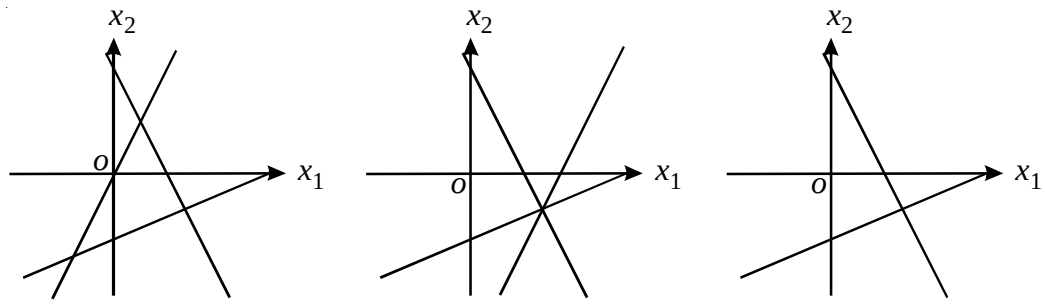
A linear system is said to be **consistent** if it has at least one solution; and is said to be **inconsistent** if it has no solution.

Geometric interpretation

The following three linear systems

(a) $\begin{cases} 2x_1 + x_2 = 3 \\ 2x_1 - x_2 = 0 \\ x_1 - 2x_2 = 4 \end{cases}$ (b) $\begin{cases} 2x_1 + x_2 = 3 \\ 2x_1 - x_2 = 5 \\ x_1 - 2x_2 = 4 \end{cases}$ (c) $\begin{cases} 2x_1 + x_2 = 3 \\ 4x_1 - 2x_2 = 6 \\ 6x_1 - 3x_2 = 9 \end{cases}$

have no solution, a unique solution, and infinitely many solutions, respectively. See Figure1.



(a) No solution (b) a unique solution (c) infinitely many solutions

Figure : 3.1

Note : A linear equation of two variables represents a straight line in R^2. A linear equation of three variables represents a plane in R^3. In general, a linear equation of *n* variables represents a hyperplane in the *n*-dimensional Euclidean space R^n.

Matrices of a linearsystem

Definition 3.1.2 The **augmented matrix** of the general linear system (3.1.1) is the table

$$\begin{pmatrix} a_{11} & \dots & a_{1n} & b_1 \\ \dots & \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} & b_m \end{pmatrix} \tag{3.1.2}$$

and the **coefficient matrix** of (3.1.1) is $\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$ (3.1.3)

Systems of linear equations can be represented by matrices. Operations on equations (for eliminating variables) can be represented by appropriate row operations on the corresponding matrices. For example,

$$\begin{cases} x_1 + x_2 - 2x_3 = 1 \\ 2x_1 - 3x_2 + x_3 = -8 \\ 3x_1 + x_2 + 4x_3 = 7 \end{cases}$$

The corresponding **augmented matrix is**

$$\begin{pmatrix} 1 & 1 & -2 & 1 \\ 2 & -3 & 1 & -8 \\ 3 & 1 & 4 & 7 \end{pmatrix}$$

Now we will do the needful row operations.

Operating $R_2 - 2R_1$ and $R_3 - 3R_1$ on the above, we get

$$\begin{pmatrix} 1 & 1 & -2 & 1 \\ 0 & -5 & 5 & -10 \\ 0 & -2 & 10 & 4 \end{pmatrix}$$

Operating $(-1/5)R_2$ and $(-1/2)R_3$ on the above, we get

$$\begin{pmatrix} 1 & 1 & -2 & 1 \\ 0 & 1 & -1 & 2 \\ 0 & 1 & -5 & -2 \end{pmatrix}$$

Operating $R_3 - R_2$ on the above, we get

$$\begin{pmatrix} 1 & 1 & -2 & 1 \\ 0 & 1 & -1 & 2 \\ 0 & 0 & -4 & -4 \end{pmatrix}$$

Operating $(-1/4)R_3$ on the above, we get

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

Operating $R_1 + 2R_3$ and $R_2 + R_3$ on the above, we get

$$\begin{pmatrix} 1 & 1 & 0 & 3 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Operating $R_1 - R_2$ on the above, we get

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

That is, we get the solution as $x_3 = 1, x_2 = 3$ and $x_1 = 0.$

Elementary row operations

Definition 3.1.3 : There are three kinds of elementary row operations on matrices:

(a) Adding a multiple of one row to another row;

(b) Multiplying all entries of one row by a non zero constant;

(c) Interchanging two rows.

Another method for solving system of linear algebraic equations is **Cramer's Rule.**

Cramer's Rule :

To solve a system of linear equations, a simple method (but, not efficient) was discovered by Gabriel Cramer in 1750.

Let the system of linear algebraic equations are

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, i = 1, 2, \ldots, n \qquad (3.2.1)$$

Let the determinant of the coefficients of the system (3.2.1) be of oder *n* i.e., $D = \left|a_{ij}\right|, i, j = 1, 2, \cdots, n$. In this method, it is assumed that $D \neq 0.$ The Cramer's rule is described in the following. From the properties of determinant

$$x_1 D = x_1 \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{vmatrix} = \begin{vmatrix} x_1 a_{11} & a_{12} & \ldots & a_{1n} \\ x_1 a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_1 a_{n1} & a_{n2} & \ldots & a_{nn} \end{vmatrix}$$

$$= \begin{vmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n & a_{12} & \ldots & a_{1n} \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & a_{22} & \ldots & a_{2n} \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n & an2 & \ldots & a_{nn} \end{vmatrix}$$

[Using operation $C_1' = C_1 + x_2 C_2 + \cdots + x_n C_n$]

$$= \begin{vmatrix} b_1 & a_{12} & \ldots & a_{1n} \\ b_2 & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ b_n & a_{n2} & \ldots & a_{nn} \end{vmatrix}$$ [Using (3.1.1)]

Therefore, $x_1 = \frac{D_{x_1}}{D}$.

Similarly, $x_2 = \frac{D_{x_2}}{D}, \ldots\ldots x_n = \frac{D_{x_n}}{D}$.

Ingeneral, $x_i = \frac{D_{x_i}}{D}$

where $D_{x_i} = \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1i-1} & b_1 & a_{1i+1} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2i-1} & b_2 & a_{2i+1} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{ni-1} & b_n & a_{ni+1} & \ldots & a_{nn} \end{vmatrix} (i = 1, 2, \ldots., n)$

Inverse of a Matrix

From the theory of matrices, it is well known that every square non-singular matrix has unique inverse. The inverse of a matrix **A** is defined by

$$A^{-1} = \frac{adjA}{|A|}.$$

The matrix *adj A* is called adjoint of A and defined as

$adjA = \begin{pmatrix} A_{11} & \ldots & A_{n1} \\ \ldots & \ldots & \ldots \\ A_{n1} & \ldots & A_{nn} \end{pmatrix}$, where A_{ij} being the cofactor of a_{ij} in $|A|$.

The main difficulty of this method is to compute the inverse of the matrix *A*. From the definition of $adj\ A$ it is easy to observe that to compute the matrix $adj\ A$, we have to determine n^2 determinants each of order $(n-1)$. So, it is very much time consuming. Many efficient methods are available to find the inverse of a matrix, among them **Gauss-Jordan** is most popular.

3.2 Gaussian elimination method

We assume that the set of linear equations given by

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{n1}x_1 + a_{22}x_2 + \ldots + a_{nn}x_n = b_n \end{aligned} \qquad (3.2.1)$$

has a unique solution and we proceed as follows.

$a_{ij}^{(1)} = a_{ij}, b_i^{(1)} = b_i, (i, j = 1, 2, 3, \ldots\ldots, n)$

Let $a_{11}^{(1)} \neq 0.$ Multiply the 1st equation of (1) by $m_{i1} = -a_{i1}^{(1)} / a_{11}^{(1)}$ and add to the ith equation when x_1 is eliminated from that equation $(i = 2, 3, \ldots., n)$ giving the following equivalent equations

$$\begin{aligned} &a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \ldots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ &\qquad a_{22}^{(2)}x_2 + \ldots. + a_{2n}^{(2)}x_n = b_2^{(2)} \\ &\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\qquad a_{n2}^{(2)}x_2 + \ldots + a_{nn}^{(2)}x_n = b_n^{(2)} \end{aligned} \qquad (3.2.2)$$

where $m_{i1} = -a_{i1}^{(1)} \big/ a_{11}^{(1)}$ and

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)},\ b_i^{(2)} = b_i^{(1)} - m_{i1}b_1^{(1)}\ (i, j = 2,3,.....,n) \tag{3.2.3}$$

Assuming again $a_{22}^{(2)} \neq 0.$ We note that the set of equations (3.2.2) except the 1st is a system of $n-1$ linear equations in the $n-1$ unknowns $x_2, x_3, ...x_n$ and applying the above eliminations procedure to this system x_2 is eliminated from the last $n-2$ equations of the set giving the equivalent system

$$a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 ..... + a_{1n}^{(1)}x_n = b_1^{(1)} \tag{3.2.4}$$

$$a_{22}^{(2)}x_2 + ... + a_{2n}^{(2)}x_n = b_2^{(2)}$$

$$a_{33}^{(3)}x_2 + ... + a_{3n}^{(3)}x_n = b_3^{(3)}$$

$$.....$$

$$.....$$

$$a_{3n}^{(3)}x_2 + ... + a_{nn}^{(3)}x_n = b_n^{(3)}$$

where $m_{i2} = -a_{i2}^{(2)} \big/ a_{22}^{(2)}$ and

$$a_{ij}^{(3)} = a_{ij}^{(2)} - m_{i2}a_{2j}^{(2)},\ b_i^{(3)} = b_i^{(2)} - m_{i2}b_2^{(2)}\ (i, j = 3,4,.....,n) \tag{3.2.5}$$

Continuing this process, we finally obtain equivalent system of equations at the $(n-1)th\ step$

$$a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 ..... + a_{1n}^{(1)}x_n = b_1^{(1)} \tag{3.2.6}$$

$$a_{22}^{(2)}x_2 + ... + a_{2n}^{(2)}x_n = b_2^{(2)}$$

$$a_{33}^{(3)}x_2 + ... + a_{3n}^{(3)}x_n = b_3^{(3)}$$

$$.....$$

$$.....$$

$$a_{nn}^{(n)}x_n = b_n^{(n)}$$

where $m_{ik} = a_{ik}^{(k)} \Big/ a_{kk}^{(k)}$ and

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)},\ b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)} \left(i, j = k+1....., n, k.1, 2, 3, ...n\right) \tag{3.2.7}$$

The upper triangular system (6) may easily be solved as follows. From the last equation $x_n = b_n^{(n)} \Big/ a_{nn}^{(n)}$; then substituting this value of x_n in the last but one equation we get the value of x_{n-1}, and then again substituting the values of x_n, x_{n-1} in the last but two equation we compute x_{n-2} and so on. Finally we get x_1. This process of solving an upper triangular system of linear system of equations is often called ***back substitution***.

When the diagonal coefficient there is unity, the last term of the constant vector contains the value of x_n. This can be used in the $(n-1)$ th equation represented by the second to the last line to obtain x_{n-1} and so on right up to the first line which will yield the value of x_1. The name of this method simply derives from the elimination of each unknown from the equations below it producing a triangular system of equations represented by

$$\begin{pmatrix} 1 & a'_{12} & ... & a'_{1n} \\ 0 & 1 & ... & a'_{2n} \\ ... & ... & ... & ... \\ 0 & 0 & ... & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix} = \begin{pmatrix} c'_1 \\ c'_2 \\ ... \\ c'_n \end{pmatrix} \tag{3.2.8}$$

which can then be easily solved by back substitution where

$$\begin{cases} x_n = c'_n \\ x_j = c'_1 - \sum_{j=i+1}^{n} a'_{ij} x_j \end{cases}$$

One of the disadvantages of this approach is that errors (principally round off errors) from the successive subtractions build up through the process and accumulate in the last equation for x_n. The errors thus incurred are further magnified by the

process of back substitution forcing the maximum effects of the round-off error into x_i. A simple modification to this process allows us to more evenly distribute the effects of round off error yielding a solution of more uniform accuracy. In addition, it will provide us with an efficient mechanism for calculation of the inverse of the matrix *A*.

Example 3.2.1 : Solve the eqations by Gauss elimination method.

$2x_1 + x_2 + x_3 = 4,\ x_1 - x_2 + 2x_3 = 2,\ \ 2x_1 + 2x_2 - x_3 = 3.$

Solution. Multiplying the second and third equations by 2 and 1 respectively and subtracting them from first equation we get

$$2x_1 + x_2 + x_3 = 4$$
$$3x_2 - 3x_3 = 0$$
$$-x_2 + 2x_3 = 1.$$

Multiplying third equation by –3 and subtracting from seond equation we obtain

$$2x_1 + x_2 + x_3 = 4$$
$$3x_2 - 3x_3 = 0$$
$$3x_3 = 3.$$

From the third equation $x_3 = 1,$ from the second equations $x_2 = x_3 = 1$ and from the first equation $2x_1 = 4 - x_2 - x_3 = 2$ or, $x_1 = 1.$

Therefore the solution is $x_1 = 1,\ x_2 = 1,\ x_3 = 1.$

3.3 Gauss-Jordan method

Let us begin by writing the system of linear equations as we did in Gauss elimination method but now include a unit matrix on the right hand side of the expression. Thus,

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix} \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 1 \end{pmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.3.1)$$

We will treat the elements of this matrix as we do the elements of the constant vector b_i. Now proceed as we did with the Gauss elimination method producing zeros in the columns below and to the left of the diagonal element. However, in addition to subtracting the line whose diagonal element has been made unity from all those below it, also subtract from the equations above it as well. This will require that these equations be normalized so that the corresponding elements are made equal to one and the diagonal element will no longer be unity. In addition to operating on the rows of the matrix **A** and the elements of , we will operate on the elements of the additional matrix which is initially a unit matrix. Carrying out these operations row by row until the last row is completed will leave us with a system of equations that resemble

$$\begin{pmatrix} a'_{11} & 0 & ... & 0 \\ 0 & a'_{22} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & a'_{nn} \end{pmatrix} \begin{pmatrix} b'_1 \\ b'_2 \\ ... \\ b'_n \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & ... & b_{1n} \\ b_{21} & b_{22} & ... & b_{2n} \\ ... & ... & ... & ... \\ b_{n1} & b_{n2} & ... & b_{nn} \end{pmatrix} \tag{3.3.2}$$

If one examines the, it is clear that so far we have done nothing to change the determinant of the original matrix **A** so that expansion by minors of the modified matrix represent by the elements a a'_{ij} is simply accomplished by multiplying the diagonal elements a_{ii} together. A final step of dividing each row by a'_{ij} will yield the unit matrix on the left hand side and elements of the solution vector x_i will be found . The final elements of **B** will be the elements of the inverse matrix of **A**. Thus we have both solved the system of equations and found the inverse of the original matrix by performing the same steps on the constant vector as well as an additional unit matrix. Perhaps the simplest way to see why this works is to consider the system of linear equations and what the operations mean to them. Since all the operations are performed on entire rows including the constant vector, it is clear that they constitute legal algebraic operations that won't change the nature of the solution in any way. Indeed these are nothing more than the operations that one would perform by hand if he/she were solving the system by eliminating the appropriate variables. We have simply formalized that procedure so that it may be carried out in a systematic fashion. Such a procedure lends itself to computation by machine and may be relatively easily programmed. The reason for the algorithm yielding the matrix inverse is somewhat less easy to see. However, the product of **A** and **B** will be the unit matrix **I**, and the operations that go into that matrix-multiply are the inverse of those used to generate **B**.

Example 3.3.1 : To see specifically how the Gauss-Jordan method works, consider the following system of equations:

$$\left.\begin{aligned} x_1 + 2x_2 + 3x_3 &= 12 \\ 3x_1 + 2x_2 + x_3 &= 24 \\ 2x_1 + x_2 + 3x_3 &= 36 \end{aligned}\right\} \tag{3.3.3}$$

If we put this in the form required by expression (3.3.1) we have

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 2 & 1 & 3 \end{pmatrix}\begin{pmatrix} 12 \\ 24 \\ 36 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3.4}$$

Now normalize the all rows by factoring out the lead elements of the first column so that

$$(1)(2)(3)\begin{pmatrix} 1 & 2 & 3 \\ 1 & \frac{2}{3} & \frac{1}{3} \\ 1 & \frac{1}{2} & \frac{3}{2} \end{pmatrix}\begin{pmatrix} 12 \\ 8 \\ 18 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} \tag{3.3.5}$$

The first row can then be subtracted from the remaining rows (i.e. rows 2 and 3) to yield

$$(6)\begin{pmatrix} 1 & 2 & 3 \\ 0 & \frac{-4}{3} & \frac{-8}{3} \\ 0 & \frac{-3}{2} & \frac{-3}{2} \end{pmatrix}\begin{pmatrix} 12 \\ -4 \\ 6 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ -1 & \frac{1}{3} & 0 \\ -1 & 0 & \frac{1}{2} \end{pmatrix} \tag{3.3.6}$$

Now repeat the cycle normalizing by factoring out the elements of the second column getting

$$(6)\left(-\frac{4}{3}\right)\left(-\frac{3}{2}\right)(2)\begin{pmatrix} \frac{1}{2} & 1 & \frac{3}{2} \\ 0 & 1 & 2 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} 6 \\ 3 \\ -4 \end{pmatrix}\begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{3}{4} & \frac{-1}{4} & 0 \\ \frac{2}{3} & 0 & -\frac{1}{3} \end{pmatrix} \tag{3.3.7}$$

Subtracting the second row from the remaining rows (i.e. rows 1 and 3) gives

$$(24)\begin{pmatrix} \frac{1}{2} & 0 & \frac{-1}{2} \\ 0 & 1 & 2 \\ 0 & 0 & -1 \end{pmatrix}\begin{pmatrix} 3 \\ 3 \\ -7 \end{pmatrix}\begin{pmatrix} -\frac{1}{4} & \frac{1}{4} & 0 \\ \frac{3}{4} & -\frac{1}{4} & 0 \\ -\frac{1}{12} & \frac{1}{4} & -\frac{1}{3} \end{pmatrix} \tag{3.3.8}$$

Again repeat the cycle normalizing by the elements of the third column so

$$(24)\left(-\frac{1}{2}\right)(2)(-1)\begin{pmatrix} -1 & 0 & 1 \\ 0 & \frac{1}{2} & 1 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} -6 \\ \frac{3}{2} \\ 7 \end{pmatrix}\begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 \\ \frac{3}{8} & -\frac{1}{8} & 0 \\ \frac{1}{12} & -\frac{1}{4} & \frac{1}{3} \end{pmatrix} \tag{3.3.9}$$

and subtract from the remaining rows to yield

$$(24)\begin{pmatrix} -1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} -13 \\ -\frac{11}{2} \\ 7 \end{pmatrix}\begin{pmatrix} \frac{5}{12} & -\frac{1}{4} & -\frac{1}{3} \\ \frac{7}{24} & \frac{1}{8} & -\frac{1}{3} \\ \frac{1}{12} & -\frac{1}{4} & \frac{1}{3} \end{pmatrix} \tag{3.3.10}$$

Finally normalize by the remaining elements so as to produce the unit matrix on the left hand side so that

$$(24)(-1)\left(\frac{1}{2}\right)(1)\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 13 \\ -11 \\ 7 \end{pmatrix}\begin{pmatrix} -\frac{5}{12} & \frac{1}{4} & \frac{1}{3} \\ \frac{7}{24} & \frac{1}{4} & -\frac{2}{3} \\ \frac{1}{12} & -\frac{1}{4} & \frac{1}{3} \end{pmatrix} \tag{3.3.11}$$

The solution to the equations is now contained in the center vector while the right hand matrix contains the inverse of the original matrix that was on the left hand side of expression (3.3.4). The scalar quantity accumulating at the front of the matrix is the determinant as it represents factors of individual rows of the original matrix. The

row subtraction shown in expressions (3.3.6), (3.3.8), and (3.3.10) will not change the value of the determinant. Since the determinant of the unit matrix on left side of expression (3.3.11) is one, the determinant of the original matrix is just the product of the factored elements. Thus our complete solution is $x=[13-11\ 7]$, where $Det(A)=-12$ and

$$A^{-1}=\begin{pmatrix} -\frac{5}{12} & \frac{1}{4} & \frac{1}{3} \\ \frac{7}{12} & \frac{1}{4} & -\frac{2}{3} \\ \frac{1}{12} & \frac{-1}{4} & \frac{1}{3} \end{pmatrix} \qquad (3.3.12)$$

Pivoting : We have assumed in each step k fo the Gaussian elimination that $a_{kk}^{(k)} \neq 0.$

To remove this restriction, begin each step of elimination process by switching rows to put a non-zero elemnt in the pivot posion. Since *A* is non-singular, this is always possible. Sometimes it may happen that the pivot element is small (actually zero, but due to roundoff it becomes vary small). To guard against this, pivoting is used.

Let at stage $k\left(1\le k\le n-1\right)$

$$c_k=\max\left|a_{ij}{}^{(k)}\right|$$

Let i_0 be smallest row index $i>k$ for which the maximum is attain. If $i_0>k,$ then switch rows *k* and i_0 in $A^{(k)}$ and $b^{(k)}$; and proceed with step *k* of the elimination process.

All multipliesrs will now satisfy

$\left|m_{ik}\right|\le 1, i=k+1,....,n$ (remember $m_{ik}=a_{ik}^{(k)}\Big/a_{kk}^{(k)}$)

And this ensures the groth in the elements of $A^{(k)}$ and thus eliminating the possibility of loss of significant errors. The pivoting is used in the solving in the linear system of equation is shown in the example given below.

Example 3.3.2 : Solve the following system of equations by Gauss elimination method (use partical pivoting).

$x_2 + 2x_3 = 5$

$x_1 + 2x_2 + 4x_3 = 11$

$-3x_1 + x_2 - 5x_3 = -12.$

Solution. The largest element (the pivot) in the coefficients of the variable x_1 is –3, attained the third equation. So we interchange first and third equations

$-3x_1 + x_2 - 5x_3 = -12$

$x_1 + 2x_2 + 4x_3 = 11$

$x_2 + 2x_3 = 5.$

Multiplying the second equation by 3 and adding with the first equation we get,

$-3x_1 + x_2 - 5x_3 = -12$

$x_2 + x_3 = 3$

$x_2 + 2x_3 = 5$

The seond pivot is 1, which is at the position a_{22} and a_{32}. Taking $a_{22} = 1$ as pivot to avoid interchange of rows. Now, subtracting and third equation from second equation, we obtain

$-3x_1 + x_2 - 5x_3 = -12$

$x_2 + x_3 = 3$

$-x_3 = -2.$

Now by back substitution, the values of x_3, x_2, x_1 are obtained as

$x_3 = 2, x_2 = 3 - x_3 = 1, x_1 = -\frac{1}{3}(-12 - x_2 + 5x_3) = 1.$

Hence the solution is $x_1 = 1, x_2 = 1, x_3 = 2.$

Some prelimary concepts

Let V be the vector space.

Norm of a Vector is defined as a real valued function $N(x)$ satisfying the conditions

i) $N(x) \geq 0 \forall x, \in V, \|x\| = 0$ if $f\ x = 0$

ii) $N(\alpha.x) = |\alpha| N(x) (\alpha\ is\ a\ scalar) \forall x \in V$

iii) $N(x+y) \leq N(x) + N(y)$

(1) $N(x) = \|x\|_1 \underline{\underline{\text{def}}} \sum_{i=1}^{n} |x_i|\ where\ x = (x_1, x_2, ..., x_n)'$

(2) $N = (x) = \|x\|_2 \underline{\underline{\text{def}}} \left\{ \sum_{i=1}^{n} |x_i|^2 \right\}^{1/2}$

(3) $N = (x) = \|x\|_\infty \underline{\underline{\text{def}}} \max_{k \leq i \leq n} |x_i|$

Example 3.3.3 : $x = (1, 0, -1, 2)'$

Then $\|x\|_1 = 4, \quad \|x\|_2 = \sqrt{6}, \quad \|x\|_\infty = 2$

Norm of a Matrix : By a norm of a matrix $A = \left[a_{ij} \right]_{n \times n} (i, j = 1, 2, 3.....n)$ is defined as a real number $\|A\|$ which satisfies the following conditions

i) $\|A\| \geq 0, \|A\| = 0 iff\ A\ is\ a\ null\ matrix$

ii) $\|\alpha A\| = |\alpha| \|A\| (\alpha\ is\ a\ scalar)$

iii) $\|A + B\| \leq \|A\| + \|B\|$

iv) $\|AB\| \leq \|A\| \|B\|$

$\therefore \|A^n\| \leq \|A\|^n$

(1) $\|A\| = \|A\|_1 \underline{\underline{\text{def}}} \max_j \sum_i |a_{ij}|$

(2) $\|A\| = \|A\|_2 \underline{\underline{\text{def}}} \left\{ \sum_{i.j} |a_{ij}|^2 \right\}^{1/2}$

(3) $\|A\| = \|A\|_\infty \underline{\underline{\text{def}}} \max_i \sum_j |a_{ij}|$

Example 3.3.4 : $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

Then $\|A\|_1 = \max(12,15,18) = 18$

$\|A\|_2 = \left(1^2 + 2^2 + ...9^2\right)^{1/2} = \sqrt{285} = 16.88$

$\|A\|_\infty = \max(6,15,24) = 24$

3.4 Gauss-Jacobi interation method

Consider the system of linear equations

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ &a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ &..... \\ &..... \\ &a_{n1}x_1 + a_{22}x_2 + ... + a_{nn}x_n = b_n \end{aligned} \qquad (3.4.1)$$

Intially the given equations of the systems are so arranged the $a_{ii} \neq 0$ for $i = 1, 2, ..., n,$ and suppose that this rearrangement is (3.4.1). Now (3.4.1) is reset in the form

$$x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2 - ... - \frac{a_{1n}}{a_{11}}x_n$$

$$x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_2 - ... - \frac{a_{2n}}{a_{22}}x_n$$

….. …… …..

$$x_n = \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}} x_2 - \ldots - \frac{a_{n.n-1}}{a_{nn}} x_{n-1}$$

Or in brief

$$x_i = \frac{1}{a_{ij}}\left[b_i - \sum\nolimits_{j \neq i} a_{ij} x_j \right] (i = 1, 2, \ldots, n) \tag{3.4.2}$$

In the Gauss-Jacobi method the iteration is generated by the formula

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum\nolimits_{j \neq i} a_{ij} x_j^{(k)} \right] (i = 1, 2, \ldots, n) \tag{3.4.3}$$

The initial guess $x_i^{(0)} (i = 1, 2, \ldots, n)$ being chosen arbitrarily.

To examine the convergence of the process, set

$$K = \max_i \frac{\sum_{j \neq i} |a_{ij}|}{|a_{ii}|} \tag{3.4.4}$$

From (3.4.3) for every i, $\varepsilon_i^{(k+1)} = \left[-\frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} \varepsilon_j^{(k)} \right]$ and so

$$\left|\varepsilon_i^{(k+1)}\right| \leq \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| \left|\varepsilon_j^{(k)}\right| \leq \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| \left\|\varepsilon^{(k)}\right\| \leq K \left\|\varepsilon^{(k)}\right\|$$

And so

$$\left\|\varepsilon^{(k+1)}\right\| \leq K \left\|\varepsilon^{(k)}\right\| \tag{3.4.5}$$

Hence for every k

$$\left\|\varepsilon^{(k)}\right\| \leq K^k \left\|\varepsilon^{(0)}\right\| \tag{3.4.6}$$

This shows that if $K < 1, \left\|\varepsilon^{(k)}\right\| \to 0$ as $k \to \infty$, i.e., the iteration converges.

The system of linear equations (1) is said to be srtictly diagonally dominant if

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| (i = 1, 2, \ldots, n)$$

i.e. if $K < 1.$

Thus the Gauss-Jacobi iteration converges if the given system of linear equations is strictly diagonally dominant.

Let $K < 1.$ By (3.4.5)

$$\left\|\varepsilon^{(k+1)}\right\| \le K\left\|\varepsilon^{(k+1)} + h^{(k)}\right\| \le K\left\{\left\|\varepsilon^{(k+1)}\right\| + \left\|h^{(k)}\right\|\right\}$$

where $h^{(k)} = x^{(k+1)} - x^{(k)} = \varepsilon^{(k+1)} - \varepsilon^{(k)}$

Or $\left\|\varepsilon^{(k+1)}\right\| \le \frac{K}{K-1}\left\|h^{(k)}\right\|$ which gives the estimation of error.

Smaller the value of *K*, more rapid will be the convergence. Also note that the above condition of convergence is sufficient but not necessary.

Example 3.4.1 : Solve the following system of linear equations by Gauss-Jacobi's method correct up to four decimal places and calculate the upper bound of absolute errors.

$$27x + 6y - z = 54$$

$$6x + 15y + 2z = 72$$

$$x + y + 54z = 110.$$

Solution. Obviously, the system is diagonally dominant as

$$|6| + |-1| < |27|, \quad |6| + |2| < |15|, \quad |1| + |1| < |54|.$$

The Gauss-Jacobi's iteration scheme is

$$x^{(x+1)} = \frac{1}{27}\left(54 - 6y^{(k)} + z^{(k)}\right)$$

$$x^{(k+1)} = \frac{1}{15}\left(27 - 6x^{(k)} - 2z^{(k)}\right)$$

$$x^{(k+1)} = \frac{1}{54}\left(110 - x^{(k)} - y^{(k)}\right).$$

Let the initial solution be (0, 0, 0). The next iterations are shown in the following table.

k	x	y	z
0	0	0	0
1	2.00000	4.80000	2.03704
2	1.00878	3.72839	1.91111
3	1.24225	4.14167	1.94931
4	1.15183	4.04319	1.93733
5	1.17327	4.08096	1.94083
6	1.16500	4.07191	1.93974
7	1.16697	4.07537	1.94006
8	1.16614	4.07488	1.93999
9	1.16632	4.07477	1.93998
10	1.16632	4.07477	1.93998
11	1.16635	4.07481	1.93998

Fig. : 3.1

The solution correct up to four decimal places is

$x = 1.1664,\ y = 4.0748,\ z = 1.9400.$

Here

$$A = \max_i \left\{ \frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \right\} = \max\left\{ \frac{7}{27}, \frac{8}{15}, \frac{2}{54} \right\} = \frac{8}{15}.$$

$e^{(0)} = \left(3\times10^{-5}, 4\times10^{-5}, 0\right)$. Therefore the upper bound of absolute error is

$$\left\| e^{(0)} \right\| \leq \frac{A}{1-A} \left\| e^{(0)} \right\| = 5.71\times10^{-5}.$$

3.5 Gauss-Seidel iteration method

A slight variant of the Gauss-Jacobi iteration is the Gauss-siedel method in which the system is also written in the form (2) with $a_{ii} \neq 0 \; for \; i = 1, 2, 3, ...n,$ but the iteration is carried out successively by the formulae

$$x_1^{(k+1)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(k)} - \ldots - a_{1n}x_n^{(k)}\right)$$

$$x_2^{(k+1)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k+1)} - \ldots - a_{2n}x_n^{(k)}\right)$$

….. …… …..

$$x_n^{(k+1)} = \frac{1}{a_{nn}}\left(b_n - a_{n_1}x_1^{(k+1)} - \ldots - a_{n,n-1}x_{n-1}^{(k+1)}\right)$$

$$\left(i = 1,2,3,\ldots.n\right) \qquad (3.5.1)$$

The initial guess $x_i^{(0)}\left(i = 1,2,\ldots,n\right)$ being chosen arbitrarily.

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j<i} a_{ij}x_j^{(k+1)} - \sum_{j>i} a_{ij}x_j^{(k)}\right]\left(i = 1,2,3,\ldots.n\right)$$

We Assert that Gauss-Seidel iteration also converges if $K < 1$ where K is defined in (3.4.4). Assume the $K < 1$. For every i

$$\varepsilon_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum\nolimits_{j<i} a_{ij}\varepsilon_j^{(k+1)} - \sum\nolimits_{j>i} a_{ij}\varepsilon_j^{(k)}\right] \qquad (3.5.2)$$

Define temporarily

$$K_i = \frac{\sum_{j<i}\left|a_{ij}\right|}{\left|a_{ii}\right|} \text{ for } \left(i = 1,2,3,\ldots..n\right) \qquad (3.5.3)$$

$0 \le K_i < K < 1$ and

$$\left|\varepsilon_i^{(k+1)}\right| \le \frac{1}{\left|a_{ii}\right|}\left[\sum_{j<i}\left|a_{ij}\right|\left|\varepsilon_j^{(k+1)}\right| + \sum_{j<i}\left|a_{ij}\right|\left|\varepsilon_j^{(k)}\right|\right]$$

$$\le \frac{1}{\left|a_{ii}\right|}\sum\nolimits_{j<i}\left|a_{ij}\right|\left\|\varepsilon^{(k+1)}\right\| + \sum\nolimits_{j>i}\left|a_{ij}\right|\left\|\varepsilon^{(k)}\right\|$$

$$\le K_i\left\|\varepsilon^{(k+1)}\right\| + \left(K - K_i\right)\left\|\varepsilon^{(k)}\right\|$$

So that for some i,

And so

$$\left\|\varepsilon^{(k+1)}\right\| \le K_i \left\|\varepsilon^{(k+1)}\right\| + \left(K - K_i\right)\left\|\varepsilon^{(k)}\right\|$$

Or

$$\left\|\varepsilon^{(k+1)}\right\| \le \frac{\left(K - K_i\right)}{1 - K_i}\left\|\varepsilon^{(k)}\right\| \tag{3.5.4}$$

Since $\frac{\left(K - K_i\right)}{1 - K_i} \le K \; as \; K < 1,$ we have

Which leads to

$$\left\|\varepsilon^{(k+1)}\right\| \le K\left\|\varepsilon^{(k)}\right\| \tag{3.5.5}$$

Hence for every k

$$\left\|\varepsilon^{(k)}\right\| \le K^k \left\|\varepsilon^{(0)}\right\| \tag{3.5.6}$$

So that $\left\|\varepsilon^{(k)}\right\| \to 0 \; as \; k \to \infty \sin ce \; K < 1.$

If $K < 1$, an estimate of the error is given by

$$\left\|\varepsilon^{(k+1)}\right\| \le \frac{K}{K-1}\left\|h^{(k)}\right\| \text{ where } h^{(k)} = x^{(k+1)} - x^{(k)} = \varepsilon^{(k+1)} - \varepsilon^{(k)}.$$

It may appear the Gauss-Seidrel method is more rapidly convergent than the Gauss-Jacobi method.

Here also the condition that the given system is strictly diagonally dominant is sufficient for the convergence of the method but not necessary.

3.6 Successive Overrelaxation (S.O.R) Method

We have to sove the linear system AX = b

where $A = \left[a_{ij}\right]_{n \times n} \left(i, j = 1, 2, 3...n\right)$ is a non-singular matrix and $b = \left[b_i\right]' \left(i = 1, 2, 3, ...n\right).$

Assume that the diagonal elements of matrix *A* are non-zero. If some $a_{ii} = 0,$ then by interchanging some rows , we can make all $a_{ii} \neq 0.$ This is possible as A is non-singular.

The matrix *A* can always be written as

$A = D + L + U$

Where $D = \left[a_{ij}\delta_{ij} \right]$

$L \rightarrow$ Lower triangular matrix with diagonal elements zero

$U \rightarrow$ Upper triangular matrix with diagonal elements zero

So , $AX = \boldsymbol{b}$ (3.6.1)

becomes $\left(D + L + U\right) X = \boldsymbol{b}$ (3.6.2)

Now multiplying by some non-zero scalar ω on bothside of equation (3.6.2) we have

$\omega\left(D + L + U\right) = \omega\boldsymbol{b}$

or, $\omega LX = \omega\boldsymbol{b} - \omega\left(D + U\right) X$ (3.6.3)

$DX = DX$ (3.6.4)

Adding (3.6.3) and (3.6.4) we get,

$\left(D + \omega L\right) X = \omega\boldsymbol{b} + \left(1 - \omega\right) DX - \omega UX$ (3.6.5)

The iteration scheme is

$\left(D + \omega L\right) X^{(i+1)} = \omega\boldsymbol{b} + \left(1 - \omega\right) DX^{i} - \omega UX^{i},\ i = 0\left(1\right)\infty$ (3.6.6)

(3..6.6) – (3.6.5) gives,

$\left(D + \omega L\right) e^{(i+1)} = \left(1 - \omega\right) De^{(i)} - \omega Ue^{(i)},\ i = \left(1\right)\infty$

Where $e^{(i+1)} = X^{(i+1)} - X$ where $e^{(i+1)}$ iis the error in the $\left(i+1\right)th$ stage of approximation.

Or, $e^{(i+1)} = \left(D + \omega L\right)^{-1}\left[\left(1 - \omega\right) D - \omega U\right] e^{(i)}$

$= Me^{(i)} = M^{2}e^{(i-1)} = \ldots = M^{i+1}e^{(0)}$

where $M = \left(D + \omega L\right)^{-1}\left[\left(1 - \omega\right) D - \omega U\right]$

Suppose $\lambda_1, \lambda_2, ..., \lambda_n$ are eigen values of the matrix M and $X_1, X_2, ..., X_n$ are corresponding eigen-vectors such that they are linearly independent.

Let $e^{(0)} = \alpha_1 X_1 + \alpha_2 X_2, + ... + \alpha_n X_n$

$e^{(i+1)} = \alpha_1 \lambda_1^{(i+1)} X_1 + \alpha_2 \lambda_2^{(i+1)} X_2, + ... + \alpha_n \lambda_n^{(i+1)} X_n$

$\to 0 \ as \ i \to \infty,$

(if all eigen values are < 1 numerically or spectral radius

$< 1 i.e \max_{1 \le j \le n} |\lambda_j| < 1$

$\therefore X^{(i+1)} - X \to 0 \ as \ i \to \infty$

Now, $\det M = \det (D + \omega L)^{-1} . \det [(1-\omega) D - \omega U]$

$= \det D^{-1} \det (1-\omega) D$

$= \det D^{-1} \det D \det (1-\omega) I$

$= (1-\omega)^n$

Now, $\det M = \lambda_1, \lambda_2, ..., \lambda_n$

$\therefore \lambda_1, \lambda_2, ..., \lambda_n = (1-\omega)^n$

i.e. $\max_i |\lambda_i| \ge |1-\omega|$

or, $|1-\omega| \le \max_i |\lambda_i| < 1$

therefore, equation (3.6.6) will converge if $(0 < \omega < 2)$ where ω is real. This method is called **overrelaxation method** when $1 < \omega < 2,$ and is called the underrelaxation method when $0 < \omega < 1.$ When $\omega = 1,$ the method becomes Gauss – Seidel's method.

Example 3.6.1 : Solve the following system of equations

$3x_1 + x_2 + 2x_3 = 6$

$-x_1 + 4x_2 + 2x_3 = 5$

$2x_1 + x_2 + 4x_3 = 7$

by SOR method taken $w = 1.01$

Solution. The iteration scheme for SOR method is

$$a_{11}x_1^{(k+1)} = a_{11}x_1^{(k)} - w\left[a_{11}x_1^{(k)} + a_{12}x_2^{(k)} + a_{13}x_3^{(k)} - b_1\right]$$

$$a_{22}x_2^{(k+1)} = a_{22}x_2^{(k)} - w\left[a_{21}x_1^{(k+1)} + a_{22}x_2^{(k)} + a_{23}x_3^{(k)} - b_2\right]$$

$$a_{33}x_3^{(k+1)} = a_{33}x_3^{(k)} - w\left[a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k)} - b_3\right]$$

or

$$3x_1^{(k+1)} = 3x_1^{(k)} - 1.01\left[3x_1^{(k)} + x_2^{(k)} + 2x_3^{(k)} - 6\right]$$

$$4x_2^{(k+1)} = 4x_2^{(k)} - 1.01\left[-x_1^{(k+1)} + 4x_2^{(k)} + 2x_3^{(k)} - 5\right]$$

$$4x_3^{(k+1)} = 4x_3^{(k)} - 1.01\left[2x_1^{(k+1)} + x_2^{(k+1)} + 4x_3^{(k)} - 7\right]$$

Let $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0.$

The detail calculatios are shown in the following table.

k	x_1	x_2	x_3
0	0	0	0
1	2.02000	1.77255	0.29983
2	1.20116	1.39665	0.80526
3	0.99557	1.09326	0.98064
4	0.98169	1.00422	1.00838
5	0.99312	0.99399	1.00491
6	0.99879	0.99728	1.00125
7	1.00009	0.99942	1.00009
8	1.00013	0.99999	0.99993
9	1.00005	1.00005	0.99997

Therefore the required solution is

$x_1 = 1,0000,\ x_2 = 1,0000,\ x_3 = 1,0000$

correct up to four decimal places.

Example : 3.6.2

Consider a linear system $Ax = b$, where

$$A = \begin{bmatrix} 3 & -1 & 1 \\ -1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix},\ b = \begin{bmatrix} -1 \\ 7 \\ -7 \end{bmatrix}$$

(a) Check, that the SOR method with value $\omega = 1.25$ of the relaxation parameter can be used to solve this system.

(b) Compute the first iteration by the SOR method starting at the point $x^{(0)} = (0,0,0)^T$.

Solution :

(a) Let us verify the sufficient condition for using the SOR method. We have to cheek, if matrix A is sysmmetric, positive definite (spd) : A is symmetri, so let us check positive definitness :

$$\det\ (3) = \ 3 > 0,\ \det \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} = 8 > 0,\ \det \begin{bmatrix} 3 & -1 & 1 \\ -1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix} = 20 > 0$$

All leading principal minors are positive and so the matrix A is positive definite.

We know, that for spd matrices the SOR method converges for values of the relaxation parameter *w* from the interval 0 < w < 2.

Conclusion : the SOR method with value $w = 1.25$ can be used to solve this system.

(b) The iterations of the SOR method are easier to compute by elements than in the voctor form :

1. Write the system as equations :

$$3x_1 - x_2 + x_3 = -1$$

$$-x_1 + 3x_2 - x_3 = 7$$

$$x_1 - x_2 + 3x_3 = -7$$

2. First, write down the equations for the GS interations :

$$x_1^{(k+1)} = \left(-1 + x_2^{(k)} - x_3^{(k)}\right)/3$$

$$x_2^{(k+1)} = \left(7 + x_1^{(k+1)} + x_3^{(k)}\right)/3$$

$$x_3^{(k+1)} = \left(-7 - x_1^{(k+1)} + x_2^{(k+1)}\right)/3$$

3. Now multiply the right hand side by the parameter *w* and add to it the vector $x^{(k)}$ from the previous interation multiplie by the factor of $(1-w)$:

$$x_1^{(k+1)} = (1-w)\,x_1^{(k)} + w\left(-1 + x_2^{(k)} - x_3^{(k)}\right)/3$$

$$x_2^{(k+1)} = (1-w)\,x_2^{(k)} + w\left(7 + x_1^{(k+1)} - x_3^{(k)}\right)/3$$

$$x_3^{(k+1)} = (1-w)\,x_3^{(k)} + w\left(-7 - x_1^{(k+1)} + x_3^{(k+1)}\right)/3$$

4. For *k* = 0, 1, 2,.... compute $x^{(k+1)}$ from these equations, starting by the first one. Computation for *k* = 0.

$$x_1^{(1)} = (1-w)\,x_1^{(0)} + w\left(-1 + x_2^{(0)} - x_3^{(0)}\right)\Big/3 = (1-1.25)\cdot 0 + 1.25(-1 + 0 - 0)\Big/3 = -0.41667$$

$$x_2^{(1)} = (1-w)\,x_2^{(0)} + w\left(7 + x_1^{(1)} - x_3^{(0)}\right)\Big/3 = -0.25\cdot 0 + 1.25(7 - 0.41667 + 0)\Big/3 = 2.7431$$

$$x_3^{(1)} = (1-w)\,x_3^{(0)} + w\left(-7 - x_1^{(1)} + x_2^{(1)}\right)$$

$$/3 = 0.25\cdot 0 + 1.25(-7 + 0.41667 + 2.7431)/3 = -1.6001$$

The next three interations are

$$x^{(2)} = (1.4972, 2.1880, -2.2288)^T,$$

$$x^{(3)} = (1.0494, 1.8782, -2.0141)^T,$$

$x^{(4)} = (0.9428, 2.0007, -1.9723)^T$,

the exact solution is equal to $x = (1, 2, -2)^T$.

3.7 Summary

The system of linear equations has been solved by using direct approach and iterative approach. In the direct approach Gauss elimination method and Gauss-Jordan method have been studied in detail where as the iterative approach Gauss Jacobi, Gauss Seidal methods are studied and their convergence are also studied. In SOR method also the convergence analysis has been studied.

3.8 Exercises

1. Using Gauss elimination method with pivoting, solve the system of linear equations

$2x_1 + 4x_2 + x_3 = 3,$

$3x_1 + 2x_2 - 2x_3 = 2,$

$x_1 - x_2 + x_3 = 6.$

(Ans: $x_1 = 2.8, = -1.16, x_3 = 2.04$)

2. Solve the following system of equations with and without pivoting and compare the result with exact solution (1, 1, 1).

3. Solve the following system of equations by Gauss-Jacobi methos:

i) $10x_1 + x_2 + x_3 = 12,$

$2x_1 + 210 + x_3 = 13,$

$2x_1 + 2x_2 + 10x_3 = 14.$

(Ans: $x_1 = 1, x_2 = 1, x_3 = 1$)

ii) $8x_1 - 3x_2 + 2x_3 = 20,$

$4x_1 + 11x_2 - x_3 = 33,$

$6x_1 + 3x_2 + 12x_3 = 35.$

(Ans: $x_1 = 3.-168,\ x_2 = 1.9858, x_3 = .9117$)

4. Solve the following system of equations by Gauss-Seidel method correct upto four decimal places:

i) $12x + y + 6z = 9,\ 8x + 3y + 2z + 13,\ x + 5y + z = 7$ (Ans : x = 1, y = 1, z = 1)

ii) $8x - y + z = 18,\ 2x + 5y - 2z = 3,\ x + y - 3z = -16$

(Ans : x = 2, y = 0.9998, z = 2.9999)

5. Solve the following system of equations by S.O.R method correct upto four decimal places:

$x + y + z = 6,\ \ x - y - z = -4,\ x + 2y - 2z = -1.$ (Ans: x = 1, y = 2, z = 3)

Unit 4 ❐ Interpolation

Structure

4.0 Objectives

After studying this unit one can be able to

- construct different forms of interpolation polynomial
- some knowledge of finite difference operators are also discussed.

4.1 Introduction

The method of obtaining the value of the function for any intermediate value of the argument when the values of a functions are known for a set of values of the arguments is known as interpolation. Mathematically, if the values of the function $y = f(x)$ at $x = a, a+h, a+2h, ..., a+nh$ be known then finding the value of the function at $x = b$ where $a < b < a+nh$ is known as interpolation. If *x* lies outside the above said range, then the corresponding process is called extrapolation.

4.2 Polynomial Interpolation

Let $f(x) \in C^{\infty}(-\infty, \infty)$. The principle of interpolating polynomial is "the selection of a function $\varphi(x)$ from a given class of functions such that the graph $y = \varphi(x)$ passes through a finite set of given points". When the function $y = \varphi(x)$ is a polynomial, the process of representing $f(x)$ by $\varphi(x)$ is called polynomial interpolation. The polynomial interpolation is based on the following theorem known as Weierstrass theorem:

Theorem 4.2.1 : Let a function $f(x) \in C[a,b]$ and let $\varepsilon > 0$ be any preassigned small number. Then, $\exists$ a polynomial $\varphi(x)$ for which $|f(x) - \varphi(x)| < \varepsilon;\ x \in [a,b]$ i.e. any continuous function can be uniformly approximated by a polynomial of sufficiently high degree within any prescribed tolerance on the finite interval.

Theorem 4.2.2 : Given any real valued function $f(x)$ and $(n+1)$ distinct points $x_0, x_1, x_2, x_3, ...x_n$ there exist unique polynomial of maximum degree n which interpolates $f(x)$ at the points $x_0, x_1, x_2, x_3, ...x_n$.

Exersise: Prove the above theorem.

In a polynomial interpolation the approximation function $\varphi(x)$ is taken to be a polynomial $y_n(x)$ of degree $\leq n$ given by

$$y_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n \qquad (4.1)$$

and it is given $y_n(x_i) = f(x_i)\ \ (i = 0,1,2,....,n)$ (4.2)

i.e. $a_0 + a_1 x_i + a_2 x_i^2 + ... + a_n x_i^n = f(x_i)(i = 0,1,2,....., n)$

Now (4.2) is a system of $(n+1)$ linear equation with $(n+1)$ unknowns $a_0, a_1, a_2, ..., a_n$. Since the

co-efficients determinant

$$\begin{vmatrix} 1 & x_0 & ..... & x_0^n \\ 1 & x_1 & ..... & x_1^n \\ 1 & x_n & ..... & x_n^n \end{vmatrix} = \prod_{i>j}\left(x_i - x_j\right) \neq 0$$ by Vandermonde's determinant

as the points $x_0, x_1, x_2, ..., x_n$ are distinct the values of $a_0, a_1, a_2, ..., a_n$ can be uniquely determined so that $y_n(x)$ exists and is called interpolating polynomial. The given points $x_0, x_1, x_2, ..., x_n$ are called interpolating points or nodes such that $x_0 < x_1 < x_2, ..., < x_n$ and also we shall write $y_i = f(x_i)(i = 0,1,2,...., n)$

4.3 Newton's Forward Interpolation Formula

Let $y = f(x)$ be a continuously differentiable function. Given set of $(n+1)$ values $(x_0, y_0), (x_1, y_1), ...(x_n, y_n)$ of *x* and *y,* it is required to find $y_n(x)$, a polynomial of degree *n,* so that *y* and $y_n(x)$ coincide at tabulated points. Let the values of *x* be equidistant so that $x_i = x_0 + ih$, ($h > 0$ is the step length, $i = 0,1,2,....n$). Since $y_n(x)$ is a polynomial of degree *n*, this can be written in the form

$$y_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... + a_n(x - x_0)(x - x_1)...(x - x_{n-1}) \quad (4.3.1)$$

We now determine the coefficient $a_0, a_1, a_2, ...a_n$ using the notation $y_n(x_i) = y_i(i = 0,1,2,...., n)$

We have $a_0 = y_0$. $a_1 = \frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y_0}{x_1 - x_0} = \frac{\Delta y_0}{h}, a_2 = \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{\Delta^2 y_0}{2!h^2}$

By continuing this method of calculating the coefficients we shall find that

$$a_3 = \frac{\Delta^3 y_0}{3!h^3}, a_4 = \frac{\Delta^4 y_0}{4!h^4}, ...a_n = \frac{\Delta^n y_0}{n!h^n}.$$

Substituting these values of $a_0, a_1, a_2, ..., a_n$ in equation (4.3.1), we get

$$y_n(x) = y_0 + (x - x_0)\frac{\Delta y_0}{h} + (x - x_0)(x - x_1)\frac{\Delta^2 y_0}{2!h^2} + ... + (x - x_0)$$

$$(x-x_1)....(x-x_{n-1})\frac{\Delta^n y_0}{n!h^n} \qquad (4.3.2)$$

Setting $u=\frac{x-x_0}{h}$, we have from equation (4.3.2)

$$y_n(x)=y_0+u\Delta y_0+\frac{u(u-1)}{2!}\Delta^2 y_0+\frac{u(u-1)(u-2)}{3!}\Delta^3 y_0+...+$$

$$\frac{u(u-1)(u-2)...(u-u+1)}{n!}\Delta^n y_0 \qquad (4.3.3)$$

Equation (4.3.3) is **Newton's forward interpolation formula**.

The **error term** is given by

$$R_{n+1}(x)=\frac{u(u-1)(u-2)...(u-n)}{(n+1)!}h^{n+1}f^{(n+1)}(\xi)$$

$$mim\{x,x_0,x_n\}<\xi<\max\{x,x_0,x_n\}$$

Note: Newton's forward interpolation formula is used to interpolate the values of y near the beginning of a set of tabulator values.

The difference table used in Newton's forward formula is as follows :

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^n y$
x_0	y_0				
		Δy_0			
x_1	y_1		$\Delta^2 y_0$		
		Δy_1		$\Delta^3 y_0$	
x_2	y_2		$\Delta^2 y_1$		
....	....				
			....		$\Delta^n y_0$
		.....	$\Delta^2 y_{n-2}$		
		Δy_{n-1}			
x_n	y_n				

Example 4.3.1 : The following table gives the values of e^x for certain equidistant values of x. Find the value of e^x when $x = 0.612$ using Newton's forward differene formulae.

x	:	0.61	0.62	0.63	0.64	0.65
y	:	1.840431	1.858928	1.877610	1.896481	1.915541

Solution. The forward difference difference table is

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
0.61	1.840431			
		0.01897		
0.62	1.858928		0.000185	
		0.018682		0.000004
0.63	1.877610		0.000189	
		0.018871		0.0
0.64	1.896481		0.000189	
		0.019060		
0.65	1.915541			

Here, $x_0 = 0.61,\ x = 0.612,\ h = 0.01,\ u = \dfrac{x - x_0}{h} = \dfrac{0.612 - 0.61}{0.01} = 0.2.$

Then,

$$y(0.612) = y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!} = 0.2.$$

$$= 1.840431 + 0.2 \times 0.018497 + \frac{0.2(0.2-1)}{2} \times 0.000185$$

$$+ \frac{0.2(0.2-1)(0.2-2)}{6} \times 0.000004$$

$$= 1.840431 + 0.003699 - 0.000015 + 0.0000019$$

$$= 1.844115.$$

4.4 Newton's Backward Interpolation Formual

Let $y = f(x)$ be a continuously differentiable function. Given set of $(n+1)$ values $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$ of *x* and *y*, it is required to find $y_n(x)$, a polynomial of degree *n*, so that *y* and $y_n(x)$ coincide at tabulated points. Let the values of *x* be equidistant so that $x_i = x_0 + ih$, ($h > 0$ is the step length, $i = 0, 1, 2, ...n$). Since $y_n(x)$ is a polynomial of degree *n*, this can be written in the form

$$y_n(x) = a_0 + a_1(x - x_n) + a_2(x - x_n)(x - x_{n-1}) + ... + a_n(x - x_n)$$
$$(x - x_{n-1})....(x - x_0) \qquad (4.4.1)$$

We now determine the coefficient $a_0, a_1, a_2, ..., a_n$ using the notation $y_n(x_i) = y_i \; (i = 0, 1, 2, ..., n)$

We have $a_0 = y_n, a_1 = \dfrac{y_n - y_{n-1}}{x_n - x_{n-1}} = \dfrac{\nabla y_n}{h}, a_2 = \dfrac{y_n - 2y_{n-1} + y_{n-2}}{2h^2} = \dfrac{\nabla^2 y_n}{2!h^2}$

By continuing this method of calculating the coefficients we shall find that

$a_3 = \dfrac{\nabla^3 y_n}{3!h^3}$, $a_4 = \dfrac{\nabla^4 y_n}{4!h^4}, ..., a_n = \dfrac{\nabla^4 y_n}{n!h^4}$.

Substituting these values of $a_0, a_1, a_2, ..., a_n$ in equation (4.4.1), we get

$$y_n(x) = y_0 + (x - x_n)\frac{\nabla y_n}{h} + (x - x_n)(x - x_{n-1})\frac{\nabla^2 y_n}{2!h^2} + ... + (x - x_n)$$

$$(x - x_{n-1})...(x - x_1)\frac{\nabla^n y_n}{n!h^n} \qquad -1 \qquad (4.3.2)$$

Setting $v = \dfrac{x - x_n}{h}$, we have from equation (4.3.2)

$$y_n(x) = y_0 + v\nabla y_n + \frac{v(v+1)}{2!}\nabla^2 y_n + \frac{v(v+1)(v+2)}{3!}\nabla^3 y_n + ... +$$

$$\frac{v(v+1)(v+2)...(v+n-1)}{n!}\nabla^n y_n \quad (4.3.3)$$

Equation (4.3.3) is **Newton's backward interpolation formula**.

The **error term** is given by

$$R_{n+1}(x) = \frac{v(v+1)(v+2)....(v+n)}{(n+1)!} h^{n+1} f^{(n+1)}(\xi)$$

$$\min\{x, x_0, x_1, \cdots, x_n\} < \xi < \max\{x_1, x_0, \cdots x_n\}$$

Note : Newton's backward interpolation formula is used to interpolate the values of y near the end of a set of tabulator values.

The difference table used in Newton's backward formula is as follows

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^n y$
x_0	y_0				
		∇y_1			
x_1	y_1		$\nabla^2 y_2$		
		∇y_2		$\nabla^3 y_3$	
x_2	y_2		$\nabla^2 y_3$		
....	....	∇y_3			
			....		$\nabla^n y_n$
		.....	$\nabla^2 y_n$		
		∇y_n			
x_n	y_n				

Example 4.4.1 : From the following table of values of x and $f(x)$ determine the value of $f(0.29)$ using Netwon's backward interpolation formula.

x	:	0.20	0.22	0.24	0.26	0.28	0.30
$f(x)$	:	1.6596	1.6698	1.6804	1.6912	1.7024	1.7139

Solution. The difference table is

x	$f(x)$	$\nabla f(x)$	$\nabla^2 f(x)$	$\nabla^3 f(x)$
0.20	1.6596			
0.22	1.6698	0.0102		
0.24	1.6804	0.0106	0.004	
0.26	1.6912	0.0108	0.0002	–0.0002
0.28	1.7024	0.0112	0.0004	0.0002
0.30	1.7139	0.0115	0.0003	–0.0001

Here, $x_n = 0.30,\ x = 0.30,\ h = 0.02,\ v = \frac{x - x_n}{h} = \frac{0.29 - 0.30}{0.02} = -0.5.$

Then,

$$f(0.29) = f(x_n) + u\nabla f(x_n) + \frac{\upsilon(u+1)}{2!}\nabla^2 f(x_n) + \frac{u(u+1)}{3!}\nabla^3 f(x_n) + \ldots$$

$$= 1.7139 - 0.5 \times 0.0115 + \frac{-0.5(-0.5+1)}{2} \times 0.0003$$

$$+ \frac{-0.5(-0.5+1)(-0.5+2)}{6} \times (0.0001)$$

$$= 1.7139 - 0.00575 - 0.0000375 + 0.00000625$$

$$= 1.70811875 \simeq 1.7081.$$

4.5 Central Interpolation formula

Stirling's Interpolation formula :

For this formula the number of nodes will be taken to be odd, i.e. $n = 2m,$ The nodes being $x_0, x_{\pm 1}, x_{\pm 2}, ..., x_{\pm m}.$

The Gauss forward interpolation formula is given by

$$y_n(x) = y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\left(\Delta^2 y_{-1} + \Delta^3 y_{-1}\right) + \frac{u(u-1)(u-2)}{3!}\left(\Delta^3 y_{-1} + \Delta^4 y_{-1}\right).....$$

where u lies 0 and 1

And Gauss Backward formula is given by

$$y_n(x) = y_0 + u\left(\frac{\Delta y_{-1} + \Delta y_0}{2}\right) + \frac{u^2}{2!}\left(\Delta^2 y_{-1}\right) + \frac{u\left(u^2-1\right)\left(\Delta^3 y_{-1} + \Delta^3 y_{-2}\right)}{3!} + \ldots$$

where u lies between -1 and 0

Taking mean of the above two Gauss's formulas, we get

$$y_n(x) = y_0 + u\left(\Delta y_{-1} + \Delta^2 y_{-1}\right) + \frac{u\left(u^2-1\right)}{2!}\left(\Delta^2 y_{-1} + \Delta^3 y_{-1}\right) + \frac{u(u-1)(u-2)}{3!}$$

$$\left(\Delta^3 y_{-1} + \Delta^4 y_{-1}\right)\ldots$$

The above equation is called **Stirling's interpolation** formula.

4.5.2 Bessel's formula is for n is odd and is given by

$$y_n(x) = \frac{1}{2}(y_0 + y_1) + \left(u - \frac{1}{2}\right)\Delta y_0 + \frac{u(u-1)}{2!}\left(\frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2}\right)$$

$$+ \frac{\left(u - \frac{1}{2}\right)u(u-1)}{3!}\Delta^3 y_{-1} + \ldots.$$

The above relation is **Bessel's formula**.

Exercise: Obtain the difference table for Stirling's and Bessel's formula.

Example 4.5.1 : Use the central difference interpolation formula of Stirling of Bessel to find the values of y at (i) x = 1.40 and (ii) x = 1.60 from the following table

x	:	1.0	1.25	1.50	1.75	2.00
y	:	1.0000	1.0772	1.1447	1.2051	2.2599

Solution. The central difference table is

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$
–2	1.00	1.0000			
			0.772		
–1	1.25	1.0772		–0.0097	
			0.0675		0.0026
0	1.50	1.1447		–0.0071	
			0.0604		0.0015
1	1.75	1.2051		–0.0056	
			0.0548		
2	2.00	1.2599			

(i) For $x = 1.40,$ we take $x_0 = 1.50,$ then $u = (1.40 - 1.50)/0.25 = -0.4.$

The Bessel's formula gives

$$y(1.40) = \frac{y_0 + y_1}{2} + \left(u - \frac{1}{2}\right)\Delta y_0 \frac{u(u-1) + \Delta^2 y_0 + \Delta^2 y_{-1}}{2!} \frac{}{2}$$

$$+\frac{1}{3!}\left(u-\frac{1}{2}\right)u(u-1)\Delta^3 y_{-1}$$

$$=\frac{1.1447+1.2051}{2}+(-0.4-0.5)\times 0.0604$$

$$+\frac{-0.4(-0.4-1)}{2!}\frac{-0.0071-0.0056}{2}$$

$$+\frac{1}{6}(-0.4-0.5)(-0.4)(-0.4-1)\times 0.0015$$

$$=1.118636.$$

(ii) For $x=1.60$, we take $x_0=1.50$, then $u=(1.60-1.50)/0.25=0.4.$

Using Stirling's formula

$$y(1.60)=y_0+s\frac{\Delta y_{-1}+\Delta y_0}{2}+\frac{s^2}{2!}\Delta^2 y_{-1}+\frac{s\left(s^2-1^2\right)}{3!}\frac{\Delta^3 y_{-2}+\Delta^3 y_{-1}}{2}$$

$$=1.1447+0.4\frac{0.675+0.0604}{2}+\frac{(0.4)^2}{2}\times(-0.0071)$$

$$+\frac{0.4(0.16-1)}{6}\frac{0.0026+0.0015}{2}$$

$$=1.1447+0.02558-0.000568-0.0001148=1.1695972.$$

4.6 Lagrange's Interpolation

Let $y=f(x)$ be a continuously differentiable function. Given set of $(n+1)$ values $(x_0,y_0),(x_1,y_1),....(x_n,y_n)$ of *x* and *y*, it is required to find $y_n(x)$, a polynomial of degree *n*, so that *y* and $y_n(x)$ coincide at tabulated points. Here the values of $x_i\,(i=0,1,2,....n)$ are not equispaced. Since $y_n(x)$ is a polynomial of degree *n*, this can be written in the form

$$y_n(x)=a_0(x-x_1)(x-x_2)....(x-x_n)+a_1(x-x_0)(x-x_2)....(x-x_n)$$

$$+a_2(x-x_0)(x-x_1)....(x-x_n)+....+a_n(x-x_0)(x-x_1)...(x-x_{n-1}) \quad (4.5.1)$$

where $a_0, a_1, a_2, ..., a_n$ are coefficient to be determined from the relation $y_n(x_i) = y_i = f(x_i)$ $(i = 0,1,2,...., n)$.

Putting $x = x_0$ in equation (4.5.1), we get

$$a_0 = \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)....(x_0 - x_n)}$$

Putting $x = x_1$ in equation (4.5.1), we get

$$a_1 = \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)....(x_1 - x_n)}$$

Similarly putting $x = x_2, x_3, ...x_n$ in equation (4.5.1), we get

$$a_2 = \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)....(x_2 - x_n)}$$

.....

.....

$$a_n = \frac{f(x_n)}{(x_n - x_0)(x_n - x_0)(x_n - x_1)....(x_n - x_{n-1})}$$

Substituting the values of $a_0, a_1, a_2...., a_n$ in (4.5.1) we get

$$y_n(x) = \frac{(x - x_1)(x - x_2)....(x - x_n)}{(x_0 - x_1)(x_0 - x_2).....(x_0 - x_n)} f(x_0)$$

$$+ \frac{(x - x_0)(x - x_2).....(x - x_n)}{(x_1 - x_0)(x_1 - x_2)....(x_1 - x_n)} f(x_1) +$$

$$\frac{(x - x_0)(x - x_1)....(x - x_n)}{(x_2 - x_0)(x_2 - x_1)....(x_2 - x_n)} f(x_2) + ...$$

$$+ \frac{(x - x_0)(x - x_1)....(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)....(x_n - x_{n-1})} f(x_n)$$

which is Lagrange's interpolation formula. The above formula may be written in the following way as

$$f(x) \approx y_n(x) = \omega(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)}$$

where $\omega(x) = (x-x_0)(x-x_1).....(x-x_n)$

$$f(x) = \omega(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)} + R_{n+1}(x)$$

Where $R_{n+1}(x) = \omega(x)\frac{f^{(n+1)}(\xi)}{(n+1)!}$ $\min\{x, x_0, ..., x_n\} < \xi < \max\{x, x_0, ..., x_n\}$

Example 4.6.1 : A function $f(x)$ defined on the interval (0, 1) is such that $f(0) = 0$, $f(1/2) = -1$, $f(1) = 0$. Find the quadratic polynomial $p(x)$ which agrees with *f* for $x = 0, 1/2, 1$.

If $\left|\frac{d^3 f}{dx^3}\right| \le 1$ for $0 \le x \le 1$, show that $|f(x) - p(x)| \le \frac{1}{12}$ for $0 \le x \le 1$.

Solution. Given $x_0 = 0$, $x_1 = 1/2$, $x_2 = 1$ and $f(0) = 0$, f $f(1/2) = -1$, $f(1) = 0$, From Lagrange's interpolating formula, the required quadratic polynomial i s

$$p(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)$$

$$+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)$$

$$= \frac{(x-1/2)(x-1)}{(0-1/2)(0-1)}\times 0 + \frac{(x-0)(x-1)}{(1/2-0)(1/2-1)}\times(-1) + \frac{(x-0)(x-1/2)}{(1-0)(1-1/2)}\times 0$$

$$= 4x(x-1).$$

The error $E(x) = f(x) - p(x)$ is given by

$$E(x) = (x-x_0)(x-x_1)(x-x_2)\frac{f'''(\xi)}{3!}$$

or, $|E(x)| = |x-x_0||x-x_1||x-x_2|\left|\frac{f'''(\xi)}{3!}\right|$

$\leq |x-0||x-1/2||x-1|1.\frac{1}{3!}\left[as\left|\frac{d^3f}{dx^3}\right| \leq 1 \text{ in } 0 \leq x \leq 1\right].$

Now, $|x-0| \leq 1, |x-1/2| \leq 1/2$ and $|x-1| \leq 1$ in $0 \leq x \leq 1.$

Hence, $|E(x)| \leq 1.\frac{1}{2}.\frac{1}{6} = \frac{1}{12}.$

That is, $|f(x)-p(x)| \leq \frac{1}{12}.$

Example 4.6.2 : Find the missing term in the following table

x	:	0	1	2	3	4
y	:	1	2	4	?	16

Solution.

Using Lagrange's formula

$$L_0(x) = \frac{(x-1)(x-2)(x-4)}{(0-1)(0-2)(0-4)} = \frac{x^3-7x^2+14x-8}{-8}$$

$$L_1(x) = \frac{(x-0)(x-2)(x-4)}{(1-0)(1-2)(1-4)} = \frac{x^3-6x^2+8x}{3}.$$

$$L_2(x) = \frac{(x-0)(x-1)(x-4)}{(2-0)(2-1)(2-4)} = \frac{x^3-5x^2+4x}{-4}.$$

$$L_3(x) = \frac{(x-0)(x-1)(x-2)}{(4-0)(4-1)(4-2)} = \frac{x^3-3x^2+2x}{24}.$$

Therefore,

$$y(x) \simeq y_0L_0(x) + y_1L_1(x) + y_2L_2(x) + y_3L_3(x)$$

$$= \frac{x^3-7x^2+14x-8}{-8} \times 1 + \frac{x^3-6x^2+8x}{3} \times 2$$

$$+\frac{x^3-5x^2+4x}{-4}\times 4+\frac{x^3-3x^2+2x}{24}\times 16$$

$$=\frac{5}{24}x^3-\frac{1}{8}x^2+\frac{11}{12}x+1.$$

Thus, $y(3)=8.25.$

Hence the missing tern is 8.25.

Example 4.6.3 : Using the following data, find by Lagrange's formula, the value of $f(x)\,at\,x=10$

i	0	1	2	3	4
x_i	9.3	9.6	10.2	10.4	10.8
$y_i = f(x_i)$	11.40	12.80	14.70	17.00	19.80

Also find the value of x where $f(x)=16.00.$

Soluion : To compute $f(10),$ we first calculate the following products :

$$\prod_{j=0}^{4}(x-x_j)=\prod_{j=0}^{4}(10-x_j)$$

$$=(10-9.3)(10-9.6)(10-10.2)(10-10.4)(10-10.8)=-0.01792,$$

$$\prod_{j=1}^{4}(x_0-x_j)=0.4455,\ \prod_{j=0,j\neq 1}^{4}(x_1-x_j)=-0.1728,\ \prod_{j=0,j\neq 2}^{4}(x_2-x_j)=+0.0648,$$

$$\prod_{j=0,j\neq 3}^{4}(x_3-x_j)=-0.0704,\ \text{and}\ \prod_{j=0,j\neq 4}^{4}(x_4-x_j)=+0.4320.$$

Thus,

$$f(10)\approx -0.01792\times\left[\frac{11.40}{0.7\times 0.4455}+\frac{12.80}{0.4\times(-0.1728)}+\frac{14.70}{(-0.2)\times 0.0648}\right.$$

$$\left.+\frac{17.00}{(-0.4)\times(-0.0704)}+\frac{19.80}{(-0.8)\times 0.4320}\right]$$

$$=13.197845.$$

4.7 Finite difference operator

Shift Operator *E :* Let *h* be a non-zero constant is the step length. The shift operator *E* for any arbitrary function $f(x)$ defined in $(-\infty, \infty)$ is represented by $Ef(x) = f(x+h)$.

Now $E^2 f(x) = E.Ef(x) = Ef(x+h) = f(x+2h)$ and in general $E^n f(x) = f(x+nh)$.

Forward difference operator Δ: It is defined by $\Delta f(x) = f(x+h) - f(x)$ where *h* is the step length

Δ is a linear operator and $\Delta = E - 1$, $E = \Delta + 1$.

Putting $x = x_0$ we get

$\Delta y_0 = f(x_0 + h) - f(x_0) = y_1 - y_0$, The second order difference is given by

$\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = y_2 - y_1 - (y_1 - y_0) = y_2 - 2y_1 + y_0$

Similarly the 3rd order difference is represented by

$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0 = y_3 - 3y_2 + 3y_1 - y_0$

and k-th order difference is given by

$$\Delta^k y_0 = \sum_{i=0}^{k} (-1)^i \binom{k}{i} y_{k-i}$$

Exercise: i) Prove that first order difference of a constant is 0.

ii) The first order difference of a polynomial of degree *n* is a polynomial of degree $n-1$.

Backward difference operator ∇ : The first order backward difference operator is defined by

$\nabla f(x) = f(x) - f(x-h)$

The central difference operator δ : The central difference operator δ is defined by

$$\delta f(x) = f\left(x+\frac{1}{2}h\right) - f\left(x-\frac{1}{2}h\right) = \left(E^{\frac{1}{2}} - E^{-\frac{1}{2}}\right) f(x)$$

$$\delta f\left(x+\frac{1}{2}h\right) = f(x+h) - f(x) = \Delta f(x)$$

$$\delta f(x) = f\left(x+\frac{1}{2}h\right) - f\left(x-\frac{1}{2}h\right) = \Delta f\left(x-\frac{1}{2}h\right)$$

Thus we have the result $\delta \equiv E^{\frac{1}{2}} - E^{-\frac{1}{2}}$

Example: i) Show that $E^{-1} \equiv 1-\nabla$.

Proof : We know that

$$\nabla f(x) = f(x) - f(x-h) = f(x) - E^{-1} f(x) = \left(1-E^{-1}\right) f(x)$$

$\Rightarrow E^{-1} \equiv 1-\nabla$ (proved)

(ii) Show that $\Delta - \nabla \equiv \delta^2$.

Proof : We know that

$$\delta f(x) = f\left(x+\frac{1}{2}h\right) - f\left(x-\frac{1}{2}h\right) = \left(E^{\frac{1}{2}} - E^{-\frac{1}{2}}\right) f(x)$$

$\Rightarrow \delta \equiv E^{\frac{1}{2}} - E^{-\frac{1}{2}}$

$\Rightarrow \delta^2 \equiv E - 2 + E^{-1} = (1+\Delta) - 2 + (1-\nabla) = \Delta - \nabla$ (proved)

4.8 Summary

In this Unit we have studied Newton's forward, backward interpolations, Central Interpolation, Bessel's and Striling's interpolation, Lagrange's interpolation and the related problems. We have also studied the some operators like shift, forward difference, backward difference and central difference and relations between them.

4.8 Exercise

1. Determine $f(x)$ as a polynomial in x for the following data :

x :	-4	-1	0	2	4
$f(x)$	1245	33	5	9	1335

Ans : $f(x) = 3x^4 - 5x^3 + 6x^2 = -4x + 5 - 5$

2. Given the values :

x :	5	7	11	13	17
$f(x)$	150	392	1452	2366	5202

Evaluate $f(9)$ using Lagrane's interpolation forula. (Ans : 810)

3. The following table gives the sales of a concern for five years. Estimate the sales for the year (i) 1986 (ii) 1992 :

Year	1985	1987	1989	1991	1993
Sales	40	43	48	52	57

Ans : (i) 41.02 (ii) 54.46

4. Find the seventh and the general terms of the series 3, 9, 20, 38, 65,....

Ans : (i) $f(7) = 154$ (ii) $f(x) = \frac{1}{6}\left(2x^3 + 3x^2 + 13x\right)$

5. Using the Stirling's formula to find u_{32} from the following table

x_i	20	25	30	35	40	45
u_{xi}	14.035	13.674	13.257	12.734	12.089	11.309

Ans : $u_{32} = 13.059$

6. Prove that

(i) $E\Delta = \Delta.E$

(ii) $E = e^{hD}$

(iii) $\nabla = E^{-1}.\Delta$

(iv) $\Delta^2 = (1+\Delta)\delta^2$

Unit 5 ❐ Numerical differentiation

Structure

5.0 Objectives

After studying this unit one can be able to

- find numerical differentiation of a function by using different methods.

5.1 Introduction

Numerical differentiation is connected with the computation of derivatives of a function whose values are known at a tabular points. The fundamental operation of differentiation is applied to the interpolating polynomial to evaluate the derivatives of the given of the given function whose values are known at some tabular points.

5.2 Netwon's Forward Differentiation Formula

Let $y = f(x)$ denote a continuously differential function which takes the values $y_0, y_1, y_2, y_3, ...y_n$ for the equidistant values $x_0, x_1, x_2, x_3, ...x_n$ of the independent variables x, then we have from Newton's Forward Interpolation formula as

$$f(x) \approx y_0 + u\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0 + ...$$

$$+\frac{u(u-1)(u-2)...(u-n+1)}{n!}\Delta^n y_0$$

Where $y_i = f(x_i), x_i = x_0 + ih,$ $(h > 0$ is the step length, $i = 0,1,2,....n)$ and $u = \frac{x - x_0}{h}$ so that

$$\frac{df}{dx} = \frac{df}{du}.\frac{du}{dx} = \frac{1}{h}\frac{df}{du}$$

$$\therefore \frac{dy}{dx} = f'(x) \approx \frac{1}{h}\left[u\Delta y_0 + \frac{2u-1}{2!}\Delta^2 y_0 + \frac{3u^2 - 6u + 2}{3!}\Delta^3 y_0 + ...\right]$$

$$\frac{d^2y}{dx^2} = f''(x) \approx \frac{1}{h^2}\left[\Delta^2 y_0 + \frac{6u-6}{3!}\Delta^3 y_0 + ...\right]$$

And so on

In particular for $x = x_0$ i.e. for $u = 0,$ them

$$\left(\frac{dy}{dx}\right)_{x=x_0} \approx \frac{1}{h}\left[\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 + ...\right]$$

$$\left(\frac{d^2y}{dx^2}\right)_{x=x_0} \approx \frac{1}{h^2}\left[\Delta^2 y_0 - \Delta^3 y_0 + ...\right]$$

The above formulae are applicable for numerical differentiation at a point *x* near the beginning of the tabulated values.

5.3 Netwon's Backward Differentiation Formula

Let $y = f(x)$ denote a continuously differential function which takes the values $y_0, y_1, y_2, y_3, ...y_n$ for the equidistant values $x_0, x_1, x_2, x_3, ...x_n$ of the independent variables x, then we have from Newton's Forward Interpolation formula as

$$f(x) \approx y_n + u\Delta y_{n-1} + \frac{u(u+1)}{2!}\Delta^2 y_{n-2} + \frac{u(u+1)(u+2)}{3!}\Delta^3 y_{n-3} + ...$$

$$+\frac{u(u+1)(u+2)...(u+n-1)}{n!}\Delta^n y_0$$

where $y_i = f(x_i), x_i = x_0 + ih,$ ($h > 0$ is the step length, $i = 0,1,2,.....n$) and $u = \frac{x - x_n}{h}$ so that $\frac{df}{dx} = \frac{df}{du}.\frac{du}{dx} = \frac{1}{h}\frac{df}{du}$

$$\therefore \frac{dy}{dx} = f'(x) \approx \frac{1}{h}\left[\Delta y_{n-1} + \frac{2u+1}{2!}\Delta^2 y_{n-2} + \frac{3u^2+6u+2}{3!}\Delta^3 y_{n-3} + ...\right]$$

$$\frac{d^2y}{dx^2} = f''(x) \approx \frac{1}{h^2}\left[\Delta^2 y_{n-2} + \frac{6u+6}{3!}\Delta^3 y_0 + ...\right]$$

and so on

In particular for $x = x_n$ i.e. for $u = 0,$ them

$$\left(\frac{dy}{dx}\right)_{x=x_n} \approx \frac{1}{h}\left[\Delta y_{n-1} + \frac{1}{2}\Delta^2 y_{n-2} + \frac{1}{3}\Delta^3 y_{n-3} + ...\right]$$

$$\left(\frac{d^2y}{dx^2}\right)_{x=x_n} \approx \frac{1}{h^2}\left[\Delta^2 y_{n-2} - \Delta^3 y_{n-3} + ...\right]$$

The above formulae are applicable for numerical differentiation at a point *x* near the end of the tabulated values.

5.4 Lagrange's Differentiation Formula

Let $y = f(x)$ denote a continuously differential function which takes the values $f(x_0), f(x_1),...., f(x_n)$ corresponding to (n+1) non-equidistant values $x_0, x_1, x_2, x_3, ...x_n$. Since the (n+1) values of the function are given corresponding to (n+1) values of the independent variable x, we can represent the function $y = f(x)$ to be a polynomial in x of degree n. Then we have Lagrange's Interpolation formula as

$$f(x) \approx L_n(x) = \omega(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x - x_i)\omega'(x_i)}$$

where $\omega(x) = (x - x_0)(x - x_1).....(x - x_n)$

Now

$$f'(x) \approx L_n'(x) = \omega'(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)} - \omega\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)^2\omega'(x_i)} \quad (5.1)$$

For non tabular points we use the above formula but for the tabular points $x = x_k$ equation (5.1) is indeterminate. Hence we proceed as

$$L_n(x) = \omega(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)}$$

$$= \omega(x)\sum_{\substack{i=0\\ i\neq k}}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)} + \omega_k(x)\,f(x_k)$$

$$L_n'(x) = \omega'(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)} + \omega_k'(x)\,f(x_k) - \omega(x)\sum_{\substack{i=0\\ i\neq k}}^{n}\frac{f(x_i)}{(x-x_i)^2\omega'(x_i)}$$

$$L_n'(x_k) = \omega'(x_k)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)} + \omega_k'(x_k)\,f(x_k)$$

where

$$\omega_k'(x_k) = \frac{1}{x_k-x_0} + \frac{1}{x_k-x_1} + ... + \frac{1}{x_k-x_n} = \sum_{i\neq k}\frac{1}{(x_k-x_i)}$$

$$L_n'(x_k) = \omega'(x_k)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)} + f(x_k)\sum_{i\neq k}\frac{1}{(x_k-x_i)}$$

Example 5.4.1 : Compute $\frac{dy}{dx}$ and $\frac{d^2y}{dx^2}$ for $x=1$, using following table

x	1	2	3	4	5	6
y	1	8	27	64	125	216

Solution: The difference table is

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1	1				
		7			
2	8		12		
		19		6	
3	27		18		0
		37		6	
4	64		24		0
		61		6	
5	125		30		
		91			
6	216				

We have $x_0 = 1,\ h = 1,\ x = 1$ so $u = \dfrac{x - x_0}{h} = 0.$

$$\left(\frac{dy}{dx}\right)_{x=x_0} \approx \frac{1}{h}\left[\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 - \frac{1}{4}\Delta^4 y_0 + \ldots\right]$$

$$\left(\frac{dy}{dx}\right)_{x=1} = \frac{1}{1}\left[7 - \frac{1}{2}\times 12 + \frac{1}{3}\times 6 - 0 + \ldots\right] = [7 - 6 + 2] = 3$$

and

$$\left(\frac{d^2y}{dx^2}\right)_{x=x_0} \approx \frac{1}{h^2}\left[\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12}\Delta^4 y_0 - \ldots\right]$$

$$\left(\frac{d^2y}{dx^2}\right)_{x=1} \approx \frac{1}{1^2}[12 - 16] = 6$$

$$\therefore \left(\frac{dy}{dx}\right)_{x=1} = 3 \text{ and } \left(\frac{d^2y}{dx^2}\right)_{x=1} = 6.$$

Example 5.4.2 : Find the value of x for which y is minimum and find the minimum value from the table:

x	0.60	0.65	0.70	0.75
$y\ (x)$	0.6221	0.6155	0.6138	0.6174

Solution: Taking 0.60 as origin, we have

$$y(x) = y_0 + y\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0$$

We have the difference table as follows:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
0.60	0.6221			
		-0.0066		
0.65	0.6155		0.0049	
		-0.0017		0
0.70	0.6138		0.0049	
		0.0032		
0.75	0.6170			

Putting the values, we have

$$y(x) = 0.6221 + u(-0.0066) + \frac{u(u-1)}{2!}(0.0049)$$

where $u = \dfrac{x - x_0}{h} = \dfrac{x - 0.60}{0.05}$

Also

$$\frac{dy}{dx} = 0,$$

i.e. $\dfrac{1}{h}\left[-0.0066 + \dfrac{2u-1}{2}(0.0049)\right] = 0$

$$u = 1.8469$$

$$x = x_0 + uh = 0.60 + 0.05 \times 1.8469 = .6923$$

$$\therefore (y)_{\min} = 0.6221 + -0.0066 \times 1.8469 + 0.00245(1.8469)(0.0049) = 0.6137426$$

5.5 Summary

In this unit numerical differentiation has been done by Using Newton' Forward, backward, Lagrange's differentiation formulae. Using this maximum and minimum values are also calculated.

5.6 Exercises

1. Find $f'(93)$ from the folloing table :

x	60	75	90	105	120
f(x)	28.2	38.2	43.2	40.9	37.7

Ans : -0.03627

2. Find the first and second order derivative of $\sqrt{x} = \text{at } x = 15$ from the following table:

x	15	17	19	21	23	25
$y = \sqrt{x}$	3.873	4.123	4.359	4.583	4.796	5.000

Ans: 0.1289, -0.004

3. Find the minimum values of $f(x)$ from the table:

x	0	2	4	6
$f(x)$	3	3	11	27

Ans: 2.25

4. Find the maximum values of $f(x)$ from the table:

x	1.2	1.3	1.4	1.5	1.6
$f(x)$	0.9320	0.9636	0.9855	0.9975	0.9996

Ans: 1.58

5. The population of a certain town is given below. Find the rate of growth of the population in 1931, 1971

Year (x)	1931	1941	1951	1961	1971
Population on thousands(y)	40.62	60.80	79.95	103.56	132.65

Ans: 2.36425, 3.10525

Unit 6 ❐ Numerical Integration

Structure

6.0 Objectives

After studying this unit one will be able to learn about

- the numerical integration of a function by using different rules and also the corresponding error terms.

6.1 Introduction

The well-known method of evaluating a definite integral $\int_a^b f(x)dx$ is to find an indefinite integral or a primitive of $f(x)$, i.e. a function $\varphi(x)$ such that $\varphi'(x) = f(x)$ and then calculate the values of $\varphi(a), \varphi(b)$ and take the value of the integral to be $\varphi(b) - \varphi(a)$ But if the function $f(x)$ is such that its indefinite integral cannot be obtained in terms of known functions, as is very often the case, then the above method fails. In such cases we may try to compute an approximate numerical value of the definite integral up to a desired degree of accuracy. This is the problem of ***numerical integration*** which is also called ***mechanical quadrature***.

Again, if the integrand $f(x)$ is not known in its analytic form but is represented by table of values, then the formal method becomes meaningless, and we are turned to numerical integration.

Closed and open type quadrature formula: A mechanical quadrature formula is called closed or open type according as the limits of integration are used as interpolating points or not.

Degree of Precision: A mechanical quadrature formula is said have a degree of precision *k*, (*k* being a positive integer), if it is exact, i.e. the error is zero for an arbitrary polynomial of degree $k \le n,$ but there exist a polynomial of degree $k+1$ for which it is not exact, i.e., the error is not zero.

Composite rule: Sometimes it is more convenient to break up the interval of integration $[a,b]$ into *m* sub-intervals $\left[a_{j-1}, a_j\right](j=1,2,3,....m)$ by the points $a_0, a_1, a_2, ..., a_m$ such that $a = a_0 < a_1 < a_2 ... < a_m = b,$ apply a given quadrature formula separately to each interval $\left[a_{j-1}, a_j\right]$ and add the result. The formula thus obtained will be called *composite rule corresponding* to given quadrature formula.

6.2 Newton-Cotes Formula (closed type)

Let the integral to be evaluated be $I(f) = \int_a^b f(x)\,dx.$ The interval $[a,b]$ is sub-divided into *n* equal subinterval, each of length h. The nodes are $x_0, x_1, x_2, ..., x_n$. such that $x_0 = a, x_n = b, x_i = x_0 + ih, h = \frac{b-a}{n}(i = 0,1,2,3,....,n).$

The corresponding entries $f(x_i), i = 0,1,2,.....n$ are also available. Let us use Lagrange's interpolation formula to approximate $f(x)$ by the interpolating polynomial $y_n(x)$

$$f(x) \approx y_n(x) = \omega(x)\sum_{i=0}^{n} \frac{f(x_i)}{(x-x_i)\omega'(x_i)}$$

where $\omega(x) = (x-x_0)(x-x_1)....(x-x_n).$

Integrating the interpolating polynomial $y_n(x)$ we have the approximate value of the given interval as

$$I_n(f)=\int_a^b \omega(x)\sum_{i=0}^{n}\frac{f(x_i)}{(x-x_i)\omega'(x_i)}dx=\sum_{i=0}^{n}H_i^n f(x_i) \tag{6.2.1}$$

where

$$H_i^n=\int_a^b \frac{\omega(x)}{(x-x_i)\omega'(x_i)}dx \quad (i=0,1,2,.....n) \tag{6.2.2}$$

Setting $u=\frac{x-x_0}{h}$, so that , $dx=h\,du$ (6.2.3)

So $\omega(x)=h^{n+1}u(u-1)(u-2)...(u-n)$ (6.2.4)

Again,

$$\omega'(x_i)=(x_i-x_0)(x_i-x_1)...(x_i-x_{i-1})(x_i-x_{i+1})....(x_i-x_n)$$

$$=ih\{(i-1)h\}....(1h)(-1h)(-2h)....\{-(n-i)h\}$$

$$=\{i(i-1)(i-2)...1\}h^i(-1)^{n-i}h^{n-i}(n-i)!$$

$$=(-1)^{n-i}h^n i!(n-i)! \tag{6.2.5}$$

Now using (6.2.3), (6.2.4), (6.2.5) in (6.2.2) we have

$$H_i^n=h\int_0^n \frac{h^{n+1}u(u-1)(u-2)...(u-n)}{(-1)^{n-i}h^n i!(n-1)!\{(u-i)h\}}\,du$$

$$=\frac{(-1)^{n-i}(b-a)}{n.i!(n-i)!}\int_0^n \frac{u(u-1)(u-2)....(u-n)}{(u-i)}du\,(i=0,1,2,.....,n)$$

$$\therefore H_i^n=(b-a)K_i^n;$$

where $K_i^n=\frac{(-1)^{n-i}}{n.i!(n-i)!}\int_0^n \frac{u(u-1)(u-2)...(u-n)}{(u-i)}(i=0,1,2,.....,n)$ (6.2.6)

Thus we have $I(f) \approx \sum_{i=0}^{n} H_i^n f(x_i) = (b-a)\sum_{i=0}^{n} K_i^n f(x_i)$ (6.2.7)

Where K_i^n is given in equation (6.2.6). This is called the $(n+1)$ – points ***Newton-Cotes Numerical Integration formula of the closed type***.

6.3 Trapezoidal Rule

For $n=1,$ we have from Newton-Cotes Formula

$$I(f) = I_T \approx (b-a)\sum_{i=0}^{1} K_i^n f(x_i) = (b-a)\left[K_0^1 f(x_0) + K_1^1 f(x_1)\right]$$

where $K_0^1 = \frac{(-1)^{1-0}}{1.0!(1-0)!}\int_0^1 (u-1)\,du = \frac{1}{2}$

and $K_1^1 = \frac{(-1)^{1-1}}{1.1!(1-1)!}\int_0^1 u\,du = \frac{1}{2}$

$$I(f) = I_T \approx \frac{(b-a)}{2}\left[f(x_0) + f(x_1)\right]$$

Error in Trapezoidal rule is $E_T = -\frac{h^3}{12} f''(\xi) = -\frac{(b-a)^3}{12} f''(\xi)(a < \xi < b)$

Geometrically, the curve $y = f(x)$ is replaced by the straight line passing through the point $(a, f(a))$ and $(b, f(b)),$ and the integral $\int_a^b f(x)\,dx$ is approximated by the area of the trapezium bounded by the straight line, the ordinates at $x = a, b$ and the name trapezoidal rule.

The degree of precision is 1

Composite trapezoidal rule: Suppose the interval $[a,b]$ is sub-divided into n equal subinterval, each of length *h*. The nodes are $x_0, x_1, x_2, ..., x_n,$ such that $x_0 = a,\ x_n = b,\ x_i = x_0 + ih, h = \frac{b-a}{n}(i = 0,1,2,3,....,n).$ then applying the above

Trapezoidal rule to each subintervals $[x_{i-1}, x_i](i=,1,2,3,....,n)$ and summing over *i* we can obtain the composite Trapezoidal rule given as

$$I(f)=\int_{x_0}^{x_1} f(x)\,dx+\int_{x_1}^{x_2} f(x)\,dx+....+\int_{x_{n-1}}^{x_n} f(x)\,dx$$

$$I(f)=\int_{x_0}^{x_1} f(x)\,dx+\int_{x_1}^{x_2} f(x)\,dx+....+\int_{x_{n-1}}^{x_n} f(x)\,dx$$

$$=\frac{h}{2}\left[\frac{1}{2}f(x_0)+f(x_1)+....+\frac{1}{2}f(x_n)\right]-\frac{h^3}{12}\sum_{i=1}^{n} f''(\xi)$$

(by using Intermediate-value theorem)

6.4 Simpson's Rule

For $n=2,$ we have from Newton-Cotes Formula

$$I(f)=I_s\approx(b-a)\sum_{i=0}^{2}K_i^n f(x_i)=(b-a)\left[K_0^2 f(x_0)+K_1^2 f(x_1)+K_2^2 f(x_2)\right]$$

where $K_0^2=\frac{(-1)^{2-0}}{2.0!(2-0)!}\int_0^2 u(u-2)\,du=\frac{1}{6}$

$$K_1^2=\frac{(-1)^{2-1}}{2.1!(2-1)!}\int_0^1 u(u-2)\,du=\frac{2}{3}$$

$$K_2^2=\frac{2}{2.2!(2-2)!}\int_0^1 u(u-1)\,du=\frac{1}{6}$$

$$I(f)=I_s\approx\frac{(b-a)}{6}\left[f(x_0)+4f(x_1)+f(x_2)\right]$$

Error in Trapezoidal rule is $E_s=-\frac{h^5}{90}f^{iv}(\xi)=-\frac{(b-a)^5}{2880}f''(\xi)(a<\xi<b)$

The degree of precision is 3

Composite Simpson'1/3rd rule: Suppose the interval $[a,b]$ is sub-divided into

$n(=2m)$ of equal subinterval, each of length *h*. The nodes are $x_0, x_1, x_2, ..., x_n$, such that $x_0 = a,\ x_n = b,\ x_0 + ih,\ h = \frac{b-a}{n}\ (i = 0,1,2,3,....,n)$. This divides the range of integration $[a,b]$ into $m = n/2$ subrange then applying the above Simpson's rule to each subintervals $[x_0, x_2], [x_2, x_4], ...[x_{n-2}, x_n]$ and applying Simpson's rule to the subrange $\left[x_{2j-2}, x_{2j}\right]$

$$\int_{x_{2j-2}}^{x_{2j}} f(x)\,dx = \frac{h}{3}\left[f\left(x_{2j-2}\right) + 4f\left(x_{2j-1}\right) + f\left(x_{2j}\right)\right] - \frac{h^5}{90} f^{iv}\left(\xi_j\right)$$

$\left(\left(x_{2j-2} < \xi_j < x_{2j};\ j = 1,2,...,m\right)\right.$

Summing over all the sub-ranges, we have

$$I(f) = \sum_{j=1}^{m} \int_{x_{2j-2}}^{x_{2j}} f(x)\,dx$$

$$= \frac{h}{3}\sum_{j=1}^{m}\left[f\left(x_{2j-2}\right) + 4f\left(x_{2j-1}\right) + f\left(x_{2j}\right)\right] - \frac{h^5}{90}\sum_{j=1}^{m} f^{iv}\left(\xi_j\right)$$

$$= I_s^c + E_s^c$$

$$I_s^c = \frac{h}{3}\left[f(x_0) + f(x_n) + 4\left\{f(x_1) + f(x_3) + ... + f\left(x_{n-1}\right)\right\}\right.$$

$$+2\left\{f(x_2) + f(x_4) + ... + f\left(x_{n-2}\right)\right\}]$$

$$E_s^c = -\frac{nh^5}{90} f^{iv}(\xi)(a < \xi < b)$$

(by using Intermediate-value theorem)

For $n = 1$, 2, 3, 4, 5, 6 the calculated values of K_i^n are given in table 6.4.1

Table for K_i^n

i / n	0	1	2	3	4	5	6
1	$\frac{1}{2}$	$\frac{1}{2}$					
2	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$				
3	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$			
4	$\frac{7}{90}$	$\frac{32}{90}$	$\frac{12}{90}$	$\frac{32}{90}$	$\frac{7}{90}$		
5	$\frac{19}{288}$	$\frac{75}{288}$	$\frac{50}{288}$	$\frac{50}{288}$	$\frac{75}{288}$	$\frac{19}{288}$	
6	$\frac{41}{840}$	$\frac{216}{840}$	$\frac{27}{840}$	$\frac{272}{840}$	$\frac{27}{840}$	$\frac{41}{840}$	

Table: 6.4.1 Newton-Cotes quadrature coefficients (closed type)

6.5 Weddle's Rule

The seven-point Newton-Cotes closed type formula with error is

$$I(f)=\int_a^b f(x)\,dx=\frac{h}{140}\left[41f(x_0)+216f(x_1)+27f(x_2)+272f(x_3)+27f(x_4)+\right.$$

$$\left.216f(x_5)+41f(x_6)\right]-\frac{9h^9}{140}f^{viii}(\xi)(a<\xi<b);h=\frac{b-a}{6} \qquad (6.5.1)$$

The coefficient of the ordinate s are extremely cumbrous which makes the formula unworthy of practical computation. Accordingly, we seek to modify the above formula so that the coefficients are simplified by proceeding as follows. We know

$$\Delta^6 f(x_0)=f(x_0)-6f(x_1)+15f(x_2)-20f(x_3)+15f(x_4)-6f(x_5)+f(x_6) \qquad (6.5.2)$$

(6.5.1) + $+\frac{h}{140}\times$ (6.5.2) gives on writing $\Delta^6 f(x_0)=h^6 f^{vi}(\xi')(a<\xi'<b)$

$$\int_a^b f(x)\,dx = I_W + E_W$$

Where

$$I_W = \int_a^b f(x)\,dx = \frac{3h}{10}\left[f(x_0) + 5f(x_1) + f(x_2) + 6f(x_3) + f(x_4) + 5f(x_5) + f(x_6)\right]$$

(6.5.3)

and

$$E_W = -\frac{h^7}{140} f^{vi}(\xi') - \frac{9h^9}{140} f^{viii}(\xi)(a < \xi, \xi' < b) \qquad (6.5.4)$$

This is called ***Weddle's rule*** inwhich the coefficients of the ordinaltes are fairly simple**.**

Composite Weddle's rule: Suppose the interval $[a,b]$ is sub-divided into $n(=6m)$ of equal subinterval, each of length *h*. The nodes are $x_0, x_1, x_2, ..., x_n,$ such that $x_0 = a,\ x_n = b,\ \ x_i = x_0 + ih,\ \ h = \frac{b-a}{n}(i = 0,1,2,3,....,n)$. This divides the range of integration $[a,b]$ into $m = n/6$ subrange then applying the above Weddle's rule to each subintervals $[x_0, x_6], [x_6, x_{12}], ...[x_{n-6}, x_n]$ and applying Weddle's rule to the subrange $\left[x_{2j-6}, x_{6j}\right]$ and summing over $j = 1,2,3,....,m,$ we get

$$I(f) = \sum_{j=1}^{m} \int_{x_{6j-6}}^{x_{6j}} f(x)\,dx$$

$$= \frac{3h}{10}\sum_{j=1}^{m}\left[f\left(x_{6j-6}\right) + 5f\left(x_{6j-5}\right) + f\left(x_{6j-4}\right) + 6f\left(x_{6j-3}\right) + f\left(x_{6j-2}\right) + 5f\left(x_{6j-1}\right) + \right.$$

$$\left. f\left(x_{6j}\right)\right] - \frac{h^7}{140}\sum_{j=1}^{m} f^{viii}\left(\xi_j'\right) - \frac{9h^9}{1400}\sum_{j=1}^{m} f^{vi}\left(\xi_j\right)$$

$$\left(x_{6j-6} < \xi_j, \xi_j' < x_{6j},\ j = 1,2,...,m\right)$$

$$\int_a^b f(x)\,dx = I_w^c + E_w^c$$

where $I_w^c = \frac{3h}{10}\Big[f(x_0) + f(x_n) + 5\{ f(x_1) + f(x_5) + f(x_7) + \ldots f(x_{n-5}) + f(x_{n-1})\}$

$+2\{ f(x_2) + f(x_4) + \ldots + f(x_{n-2})\}$

$+6\{ f(x_3) + f(x_9) + \ldots + f(x_{n-3})\}$ (6.5.5)

$+2\{ f(x_6) + f(x_{12}) + \ldots + f(x_{n-6})\}\Big] \quad (n \geq 12)$

$$E_W^{\in} = -\frac{nh^7}{840}\sum_{j=1}^{m} f^{viii}(\xi_j') - \frac{9nh^9}{8400}\sum_{j=1}^{m} f^{vi}(\xi_j) \quad (a < \xi, \xi' < b) \quad (6.5.6)$$

Example 6.5.1 : Evaluate $I = \int_0^6 \frac{1}{1+x}dx$ using (i) Trapezoidal rule, (ii) Simpson's 1/3rd rule, (iii) Weddle's rule. Also check by direct integration.

Solution: Here, we have $y = f(x) = \frac{1}{1+x}, 0 \leq x \leq 6.$

Divide the interval into six parts. So $h = \frac{6-0}{6} = 1$

Therefore, the values of $y = \frac{1}{1+x}$ are:

x	0	1	2	3	4	5	6
y = f (x)	1	0.5	1/3	1/4	1/5	1/6	1/7

(i) By Trapezoidal rule:

$$\int_0^6 \frac{1}{1+x}dx = \frac{h}{2}\Big[(y_0 + y_6) + 2(y_1 + y_2 + y_3 + y_4 + y_5)\Big]$$

$$= \frac{1}{2}\left[\left(1 + \frac{1}{7}\right) + 2\left(0.5 + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6}\right)\right]$$

= 2.021429

(ii) By Simpson's 1/3rd rule:

$$\int_0^6 \frac{1}{1+x}dx = \frac{h}{3}\Big[(y_0 + y_6) + 4(y_1 + y_3 + y_5) + 2(y_2 + y_4)\Big]$$

$$= \frac{1}{3}\left[\left(1+\frac{1}{7}\right)+4\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{6}\right)+2\left(\frac{1}{3}+\frac{1}{5}\right)\right]$$

= 1.9538730

(iii) By Weddle's rule

$$\int_0^6 \frac{1}{1+x}dx = \frac{3h}{10}\left[(y_0+y_6)+3(y_1+y_2+y_4+y_5)+2y_3\right]$$

$$= \frac{3}{8}\left[\left(1+\frac{1}{7}\right)+3\left(\frac{1}{2}+\frac{1}{3}+\frac{1}{5}+\frac{1}{6}\right)+2\left(\frac{1}{4}\right)\right]$$

= 1.952857

By actual integration,

$$\int_0^6 \frac{1}{1+x}dx = \left[\log(1+x)\right]_0^6$$

$$= \log 7 - \log 1$$

$$= 1.945910$$

Example 6.5.2 : The velocity v of a particle at distance s from a point on its path is given in the table below:

s in meter	0	10	20	30	40	50	60
v in m / sec	47	58	64	65	61	52	38

Estimate the time to travel 60 meters by using Simpson's 1/3rd rule.

Solution: Here, we have $h = 10.$

We know the $v = \frac{ds}{dt}$. Hence, $dt = \frac{ds}{v}$

To find the time taken to travel 60 metres we have to evaluate

$$\int_0^{60} dt = \int_0^{60} \frac{ds}{v}$$

Let $y = \frac{1}{v}$, then the table values of y for different values of s are given below

s	0	10	20	30	40	50	60
$y = \frac{1}{v}$	0.0213	0.0172	0.0156	0.0156	0.0164	0.0192	0.0263

By Simpson's 1/3d rule,

$$\int_0^{60} y\,ds = \frac{h}{3}\left[(y_0+y_6)+4(y_1+y_3+y_5)+2(y_2+y_4)\right]$$

$$=\frac{10}{3}\left[(0.0213+0.0263)+4(0.0172)+0.0154+0.0192\right]+2(0.0156+0.0164)$$

$$=1.0627$$

$\therefore$ Time taken to travel 60 meters is 1.0627 seconds.

6.6 Summary

In this unit the numerical integration by using Newton-Cotes formula(closed type), Trapezoidal rule, Simpson's1/3rd rule and Weddle's rule have been discussed and also the corresponding error terms are also studied.

6.7 Exercises

1. Define the degree of precision of mechanical quadrature formula. Show that the d.p. of trapezoidal is 1.

2. Deduce the trapezoidal, Simpson's 1/3rd and Weddle's rules (without error) by integrating Newton's forward interpolation formula.

3. Evaluate $\int_0^5 \frac{1}{4x+5}dx$ by Trapezoidal rule using 11 coordinate.

Ans: 0.4055

4. find the value of $\int_0^{\pi/2} \sqrt{\cos x}\,dx$ by (i) Trapezoidal rule and (ii) Simpson's one-third rule taking n = 6. Ans: (i) 1.170 (ii) 1.187)

5. When a train is moving at 30m/sec steam is shut off and brakes are applied. The speed of the train per second after *t* seconds is given by

$time(t)$	0	5	10	15	20	25	30	35	40
$speed(v)$	30	24	19.5	16	13.6	11.7	10.0	8.5	7.0

Using Simpson's rule, determine the distance moved by the train in 40 sec.

(Ans: 606.66 m.)

Unit 7 ❐ Computer Language

Structure

7.0 Objectives

After going through this unit one will be able to learn

- the concept of programming languages, interpreter, compiler, source and object program.

7.1 Introduction

We have seen that the hardware or physical parts that form a computer serve no purpose by themselves. To make a computer work, we must learn how to give instruction to it in a language that the computer will **understand**.

7.2 Concept of Programming Language

In a natural language we speaks in, we use words to convey ideas and even

emotions, feeling and sensations. A *computer language* is used to communicate with a machine which can react to only simple and very clear instructions conveyed through precise notations or words. *The notations and words which can be used to give instructions to a computer and the rules which the instructions must obey form a computer language.*

The first set of computer language that developed were based upon the internal structure of the computer. These languages were referred to as *codes* or low level languages. *Machine code* and *assembly code* which used binary or mnemonic symbols were first set of languages that were developed for computers.

7.3 Machine Language

A computer works on electricity and this enables it to receive and store information only in the form of electric pulses. If a pulse is present it codes it as 1 and if it is not present it codes it as 0. The computer's own language is, therefore, made up of the binary numbers 0 and 1 and is written in the form of a numeric code. This language is called ***machine language*** or code and is a part of a computer's electronic circuitry. When computers were first made, machine language was the only language.

The utility of a machine language is that since it is written in the machine code itself, the computer processes it quickly. On the other hand, the number of people who can without difficulty a series of instruction using zeroes and ones must indeed be very few. It requires long term expertise to do this. Coding and decoding are tedious processes and prone to errors. Further, machine languages vary with the make of each computer and one may need to learn a new machine language each time one works on a different make of machines.

7.4 Assembly Language

In the beginning, machine language was the only language. Then *assembly language* was developed. In an assembly language, 'mnemonics' (or alphanumeric codes) were used to substitute the binary machine coded to machine language. These 'mnemonics' were memory aids which helped the mind to relate things more easily. For example, mnemonics 'DIV' could be used to describe the operation 'divide'.

Assembly language made it easier for the user to write his instructions. But the

'mnemonics' had to be translated to the computer into its binary pattern before the machine could do the job. The translation was done by a special pre-stored set of instructions called an *assembler*. The assembler was supplied by the computer manufacturer and usually embedded in ROM chips.

The advantages of an assembly language are that it helps in reducing errors and the time involved in writing instructions. The drawbacks are that it requires the user to have a fair knowledge of hardware and being machine dependent, the instructions for one machine cannot be executed on another.

7.5 High Level Language

In the initial phase of development, the use of computers was largely confined to a small group of scientists and computer specialists. With improvements in technology and fall in prices, there arose a need for languages that would permit even a non-expert to communicate with a computer. This led to the development of *high level* languages which enable a large number of people to use computer without having to know in detail its internal structure. These languages are *user-centred* and not *machine-centred* like the machine and assembly codes. A program written in high-level language can be run on different computers without any or much modifications.

Instructions in high level languages are given using certain words from a natural language, such as English, an a few notations. Each word or notation in these languages have one precise meaning and we must adhere to the *syntax* or the set of grammar, punctuation and spelling rules for the language. Today, virtually all work is undertaken by writing instructions in one of the high level languages.

The first high-level programming were designed in 1950s. Ada, Algo, LOGO, PILOT,BASIC, COBOL, C/C++,FORTRAN, Java, R, python etc. are popular examples of high-level languages.

The computer does not directly understand a high level language. A translation is undertaken by specially prepared software called *language processors or translators*.

7.6 Interpreter

An interpreters translates one instruction at a time and gets it immediately executed. Each instruction is checked for errors and corrections are made when necessary. Interpreters do not involves much storage space but they require more time to execute. Basic, R, Python are Interpreter based language

7.7 Compiler, Source program and object program

Compilers

Compilers take all the instructions together and then compile them into the corresponding machine code. The user written program (referred to as the **source**

Basis for comparison	**Compiler**	**interpreter**
input	It takes an entire program at a time.	It takes a single line of code or instruction at a time.
Output	It generates intermediate object code.	It does not produce any intermediate object code.
orking mechanism	The compilation is done before execution.	Compliation and execution take place simultaneously.
Speed Memory	Comparatively faster Memory requirement is more due to the creation of object code.	Slower It requires less memory as it does not create intermediate object code.
Errors	Display all errors after compilation, all at the same time.	Displays error of each line one by one.
Error detection	Diffucult	Easier comparatively
Pertaining Programming languages	C, C++, C#, Scala, typescript uses compiler.	PHP, Perl, Python, Ruby uses an interpreter.

program) is fed into the computer. The compiler translates the ***source program*** and produces a complete program in machine language known as the ***object program*** which is loaded into main memory for execution.

Some basic comparison between Compiler and Interpreter is given in the form of the table given belos :

7.8 Conclusion

Compiler and interpreter both are intended to do the same work but differ in operating procedure, Compiler takes source code in an aggregated way whereas Interpreter takes constituent parts of source code, i.e., statement by statement.

Although both compiler and interpreter have certain advantages and disadvantages like Interpreted languages are considered as cross-platform, i.e., the code is portable. It also doesn't need to compile instruction previously unlike compiler which is time-saving. Compiled languages are faster regarding compilation process.

7.9 Summary

In this unit the concept of programming language like machine language, assembly language, High level language is discussed. Also the difference between interpreter and compiler as well as the source and object program also discussed

7.10 Exercise

1) What do you understand by Machine language?
2) How the machine language differ from the assembly language?
3) Define the object and source program.
4) Write the difference between Interpreter and compiler.

Unit 8 ❐ Number System

Structure

8.0 Objectives

After going through this unit one will be able to learn

- different types of number systems and their conversion from one system to another system.

8.1 Introduction

We have heard of number systems like the whole numbers, the real numbers etc. But in the context of computer awareness, we define other types of number systems like the binary number system, the decimal system, the hexadecimal system and others. We will discuss the binary number system and others and how we can convert from one number system to the other.

The value of any digit in a number can be determined by

-The digit

-Its position in the number

-The base of the number system

Let *r* be the base of a number system. Then to represent any given integer number, say D, symbolically in this system, we use *r* number of different characters, namely

$0<1<2<...<(r-2)<(r-1)$ and represent D uniquely as

$$D=\pm(d_n d_{n-1} d_{n-2} d_{n-3}....d_2 d_1 d_0) \qquad (8.1)$$

According as the number is positive or negative, where *n* is a positive integer and each d_i ranges from $0 to (r-1)$, such that $d_n \neq 0,\ 0 \le d_i \le (r-1), i = 0,1,2,...(n-1)$

The magnitude of the number will be given by

$$|D| = d_n.(r)^n + d_{n-1}.(r)^{n-1} + ... + d_2.(r)^2 + d_1.(r)^1 + d_0.(r)^0$$

8.2 Decimal Number System

The most commonly used number system is Decimal Number System with base 10. In this system, the ten basic characters that are used to represent number are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. Thus in decimal number system the (n+1) digit number *D* represented by (8.1) has the magnitude

$$d_n.(10)^n + d_{n-1}.(10)^{n-1} + ... + d_2.(10)^2 + d_1.(10)^1 + d_0.(10)^0$$

For example, the decimal number represented by the symbol 4356 has the magnitude

$$4356 = 4.(10)^3 + 3.(10)^2 + 5.(10)^1 + 6.(10)^0$$

For a fractional number whose magnitude is less than 1, the symbolic representation starts with dot (.), called the decimal point, and the powers of the base will be negative from -1. For **example**,

$$\frac{83}{100} = .83 = 8\times10^{-1} + 3\times10^{-2}$$

Thus $607.03 = 6\times10^2 + 0\times10^1 + 7\times10^0 + 0\times10^{-1} + 3\times10^{-2}$

Exercise 8.2.1 : Write i) $\frac{22}{7}, \sqrt{5}$ in decimal number system.

8.3 Binary Number System

In binary number system, the base is 2 and the symbols used for representing a number are 0 and 1. Thus the number 110101 in binary system is equivalent to

$1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$

= 32+16+0+4+0+1 = 53 in decimal system.

Using the respective radix as subscript, we write this result as:

$(110101)_2=(53)_{10}$.

Just like decimal point, we also have binary point as:

$(1101.011)_2=1\times2^3+1\times2^2+0\times2^1+1\times2^0+0\times2^{-1}+1\times2^{-2}+1\times2^{-3}$

= 8+4+0+1+0+.25+.125 $=(13.375)_{10}$

Binary numbers play a vital role in the design of digital computers.

Exercise 8.3.1 : Write $(.1011)_2$ to decimal number system.

8.4 Octal Number System

Here the base is 8 and eight different symbols are 0, 1, 2, 3, 4, 5, 6 and 7. Thus a number $(7032)_8$ in octal system is equivalent to

$7\times8^3+0\times8^2+3\times8^1+2\times8^0$

= 3584 + 24 + 2 $=(3610)_{10}$

Again

$(71.34)_8=7\times8^1+1\times8^0+3\times8^{-1}+4\times8^{-2}$

= 56 + 1+ 0.375 = 0.0625 $=(57.4375)_{10}$

8.5 Hexadecimal Number System

The base is 16 and the required symbols to represent a number in this system are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F. The symbols A, B, C, D, E and F represent the decimal number 10, 11, 12, 13, 14 and 15 respectively. The number

$(BC6A)_{16}=11\times16^3+12\times16^2+6\times16^1+10\times16^0$

= 45056 + 3072 + 96 + 10 $=(48234)_{10}$

The symbol 0 and 1 are generally called BIT – the bit at the extreme left having the highest positional value is the Most Significant Bit (MSB) while the bit occupying the extreme right position having least positional value is called the Least Significant Bit (LSB)

8.6 Conversion

Conversion of binary to decimal: The decimal equivalent of a binary number is obtained by expanding it according to the place-value of each bit.

Exercise : Obtain the decimal equivalent of the following numbers:

i) 11011

ii) 10010

iii) 0.01101 Ans: i) $(27)_{10}$, ii) $(28)_{10}$, iii) $(0.40625)_{10}$.

Conversion from decimal to binary: There are several methods of converting a decimal number to its binary equivalent. The most commonly used methods are (i) *Expansion Method* and (ii) *Division and Multiplication Method*.

Expansion Method: The given decimal number is first expressed as summation terms each of which is a power (positive integral and negative integral) of 2.

Example 8.6.1 : Convert the decimal numbers (i) 47 (ii) 195 (iii) 88.5625 to their binary equivalents:

Solution: (i) $(47)_{10} = 32 + 15 = 32 + 8 + 7 = 32 + 8 + 4 + 3$

$= 32 + 8 + 4 + 2 + 1$

$= 2^5 + 2^3 + 2^2 + 2^1 + 2^0$

$= 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0$

$= (101111)_2$

(ii) $(195)_{10} = 128 + 64 + 2 + 1$

$= 2^7 + 2^6 + 2^1 + 2^0$

$= 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^1 + 1\times 2^0$

$= (11000011)_2$

(iii) $(88.5625)_{10} = 64 + 16 + 8 + 0.5 + 0.0625$

$= 2^6 + 2^4 + 2^3 + 2^{-1} + 2^{-4}$

$= 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 0\times 2^1$

$$+0\times 2^{0}+1\times 2^{-1}+0\times 2^{-2}+0\times 2^{-3}+1\times 2^{-4}$$

$$= =(1011000.1001)_2$$

Division and Multiplication Method: The above method is laborious and not suitable for large numbers. We may however use the division and multiplication method which is described as follows:

The decimal number has both an integral and fractional part, then we first convert the integral part to its binary equivalent by the division method. The fractional part must next be converted by multiplication process and the two results should be linked up after that.

For decimal integral: The given decimal integer is repeatedly divided by the base 2 of the binary number system. The remainder (which is either 0 or 1) is noted in each division. The process continues till the quotient is zero. The first remainder is the least significant bit and the last one is the most significant bit. Thus the binary equivalent is obtained by writing down the remainder in the reversed order, i.e. from bottom to upward.

Example 8.6.2 : Convert $(47)_{10}$ to binary equivalent.

Solution:

2	47			
2	23	1	←	LSB
2	11	1		
2	05	1		
2	02	1		
2	01	0	↑	
	00	1	←	MSB

Thus $(47)_{10}=(101111)_2$

For decimal fraction: The given decimal fraction is multiplied by 2, the fractional part is again multiplied by 2 and the process is repeated till the fraction part of the product is zero. The integral part obtained each time, which can be either 0 or 1, is taken in top to bottom order and arranged from left to right to provide the binary equivalent to the decimal number.

Example 8.6.3 : Convert the following decimal fractions to its binary equivalent $(.37)_{10}$

Solution : The result of repeated multiplication is shown below

Multiplication	Integral Part	Fractional Part	Binary Position
0.375 × 2 = 0.75	0 ↓	0.75	0×2^{-1}
0.75 × 2 = 1.50	1	0.50	1×2^{-2}
0.5 × 2 = 1.00	1	0.00	1×2^{-3}

Thus the equivalent binary fraction is $(.375)_{10} = (.011)_2$

Exercise 8.6.4 : Convert the decimal fractions to its binary equivalent $(.435)_{10}$

Example 8.6.5 : Convert $(47.375)_{10}$ to binary equivalent.

Solution: As we have already done the binary equivalent of the integral part

$(47)_{10} = (101111)_2$

and the decimal fraction to binary is $(.375)_{10} = (.011)_2$

Linking the two results, we have $(47)_{10} + (.375)_{10} = (101111)_2 + (.011)_2$

Or, $(47.375)_{10} = (101111.011)_2$

Conversion of decimal number to octal: The conversion method follows similar rules as in the case of binary number system. Here we divide the number by the base 8 instead of 2. It will clear in the following example

Example 8.6.6 : i) Convert $(347)_{10}$ to octal equivalent.

Solution:

8	347			
8	43	3	←	LSB
8	05	3	↑	
8	00	5	←	MSB

Therefore $(347)_{10} = (533)_8$

ii) Convert $(0.30)_{10}$ to octal equivalent.

Solution:

Multiplication	Integral Part	Fractional Part	Binary Position
$0.30 \times 8 = 2.40$	2 ↓	.40	2×8^{-1}
$0.40 \times 8 = 3.20$	3	.20	3×8^{-2}
$0.20 \times 8 = 1.60$	1	.60	1×8^{-3}
$0.60 \times 8 = 4.80$	4	.80	4×8^{-4}
$0.80 \times 8 = 6.40$	6	.40	6×8^{-5}
$0.40 \times 8 = 3.20$	3	.20	3×8^{-6}

(Recurring Starts)

Hence $(0.30)_{10} = (.2\overline{3146})_8$

Conversion of binary number to octal: The base of the octal system is 8 or (2x2x2). Thus the octal base 8 is a power of the base 2 in the binary system.

A binary number is converted to its octal equivalent by grouping of three successive bits starting from the least significant bit or the right-most digit.

Example 8.6.7 : Convert $(10101111011)_2$ to octal.

Solution: Three successive bits of the binary string are grouped from the right.

Binary:	010	101	111	011
Octal equivalent:	2	5	7	3

Hence $(10101111011)_2 = (2573)_8$

Note: A non-significant '0' has been added in the left-most group to make it a string of 3 bits. This is only for convenience of grouping.

Conversion of octal number to binary: The octal equivalent of binary number may be found through the same process of referring to the conversion table and arranging the bits in order.

Example 8.6.8 : Convert $(412)_8$ to binary

Solution: We have:

	4	1	2	(in Octal)
=	100	001	010	(in Binary)

Arranging in order, we get

$(412)_8 = (100001010)_2$

Exercise: Convert (i) $(1110101110)_2$

(ii) $(10.11)_2$

(iii) $(1011.1011011)_2$ to their octal equivalent.

Ans: (i) $(1656)_8$, (ii) $(2.6)_8$, (iii) $(13.554)_8$

Conversion from decimal system to hexadecimal system:

The procedure for conversion from decimal to hexadecimal is same as that of octal. Here in this case repeated divisions is by 16.

Example 8.6.9 : Convert $(116)_{10}$ to hexadecimal.

Solution:

16	116	
16	7	4
16	0	7 ↑

Hence $(116)_{10} = (74)_{16}$

Conversion method from binary to system to hexadecimal system is similar to octal but here instead of grouping by 3-bits, we arrange the binary string in groups of 4-bits

Example 8.6.10 : Convert $(111001)_2$ to hexadecimal.

Solution: $:(111001)_2 = (00111001)_2 = (39)_{16}$

Example 8.6.11 : Convert i) $(A748)_{16}$ and (ii) $(BA_2.C4)_{16}$ to binary number system.

Solution: i) $(A748)_{16} = (1010011101001000)_2$

(ii) $(BA2.C4)_{16} = (101110100010.11000100)_2$

8.7 Summary

In this unit, the detailed study of Number system like decimal, binary, octal, hexadecimal and their conversion from one system to other have been studied with proper examples.

8.8 Exercises

1. What do you understand by binary number system? How it is differ from decimal number system?

2. Convert the following decimal numbers into its binary equivalents:

a) $(131)_{10}$ b) $(395)_{10}$ c) $(423.25)_{10}$

Ans : (a) $(10000011)_2$ (b) $(395)_{10}$ (c) $(423.25)_{10}$

3. Convert the following binary numbers to its decimal equivalent:

(a) $(11001)_2$, (b) $(11.01)_2$, (c) $(10.011)_2$

Ans : (a) $(25)_{10}$ (b) $(3.25)_{10}$, (c) $(2.375)_{10}$

4. Convert the following decimal numbers into its octal and hexadecimal equivalents:

(a) $(231)_{10}$ (b) $(153)_{10}$

Ans : (a) $(347)_8$ $(E7)_{16}$, (b) $(231)_8$, $(99)_{16}$.

5. Convert the following octal numbers into its binary equivalents:

(a) $(346)_8$ (b) $(135)_8$

Ans : (a) $(1100110)_2$ (b) $(1011101)_2$.

6. Convert the following hexadecimal numbers into its binary equivalents:

(a) $(4B5)_{16}$ (b) $(A3BF)_{16}$

Ans : (a) $(10010110110)_2$ (b) $(1010001110111111)_2$.

References

1. Gupta, A and S. Bose, S. C.: Introduction to numerical Analysis Academic Publisher, 2009.

2. Jain, M. K., Iyenger, S.R.K., Jain, R.K., Numerical methods for scientific and engineering computation, New Age International Publishers, New Delhi, 2019

3. Froberg, C.E., Introduction to Numerical Analysis, Addision Wiley, 1969.

4. Conte, S. D. and deBoor, C., Elementary Numerical Analysis , McGraw- Hill, New York, 1982.

5. Atkinson, K. E., Elementary Numerical Analysis, John Wiely & Sons., New York, 1985

6. Patel, V. A., Elementary Numerical Analysi, John Wiely & Sons., New York, 1985

7. Scarborough, J. B., Numerical Mathematicsl Analysis, Oxford and IBH publishing Co. Pvt. Ltd., New Delhi, 2003

8. Pal, M. , Numerical Analysis for Scientist and Engineering: Theory and C. Programs Narosa Publisher, 2008

9. Garg, R. abd Goel R.S., Numerical Techniques Computing with C and MATLAB, CBS Publisher, 2018