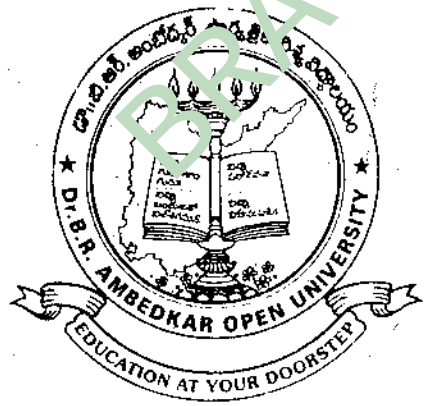
MATHEMATICS

Numerical Analysis and Principles of Computer Programming



DR. B.R. AMBEDKAR OPEN UNIVERSITY
Hyderabad
1996

Course Team

Editor
Prof. B. Kesava Rao

Associate Editors
Prof. K. Kuppuswamy Rao
Dr. N. Venkata Narayana

Writers
Dr. P.B. Bhaskara Rao
Dr. D. Rama Murthy
Dr. V.V. Ramana Rao
Dr. N. Venkata Narayana

Cover Design & Graphics
M. Ramesh

Dr. B.R. Ambedkar Open University
Hyderabad.

First Publsihed 1986

Further information on Open University Courses may be obtained from the Director, Dr.B.R.Ambedkar Open University, Road No. 46, Jubilee Hills, Hyderabad - 500 033. (A.P.)

SAI SRI PRINTERS, Offset Printers, 8-3-945, Shop No.21, Pancom Business Centre, Ameerpet, Hyderabad-500 016, Ph. No.290222
for Dr. B.R. Ambedkar Open University, 1994

PREFACE

This book deals with the topics in Numerical Analysis and Elements of Computer Programming (Fortran IV) included in the syllabus for the third year of the B.Sc. course offered by Dr. B.R. Ambedkar Open University. These topics are to be studied in the third year of the Three Year Degree Course in Science (B.Sc.). The Syllabus, for the sake of convenience, is divided into Blocks, each of which comprises a number of Units. Each Block generally covers a specific area of the subject. The units are prepared by specialists in accordance with a format so designed as to enable the student read and understand them without much difficulty. Each unit begins with a statement of its objectives followed by as introduction and has at its end assignments intended to test the student's comprehension of its subject matter. In cases which are likely to be beyond the level of the student, theorems are stated without proofs.

The interest of the mathematician and of the scientist in numerical methods has grown considerably during the last few decades. Electric desk calculators and electronic super-computers of the digital and the analogue type make possible today's camputations which could not have been tackled a few years ago. Many mathematical problems important for application in engineering and science but inaccessible by analytical methods can now be treated numerically.

The book is divided into two parts. Part one deals with Numerical Analysis and part two with Fortran-IV programming. Numerical methods are introduced in the first part, the program writing through Fortran–IV is explained in part two.

The Univesity hopes that this course material will help the student to get acquainted with the concepts of Numerical Methods and principles of computer programming.

Contents

PART – I

NUMERICAL ANALYSIS

BLOCK - 1 : INTERPOLATION

Interpolation is the process of evaluating a function whose graph goes through a given set of points. It is a techinique where the non - tabulated values of a tabular function are estimated on the assumption that the function behaves sufficiently smoothly between tabular points. This is possible only when the interpolation function could be approximated by polynomials of low degrees. With the introduction of pogrammed caluculators and computers the "art of reading bewteen the lines in a table" has become relatively easy. But the present day theory of interpolation uses advanced methods (like spline functions) to get better approximation and compute estimates of errors in situations not amenable to polynomial interpolations.

Unit - 1 : Ordinary finite differences

Unit - 2 : Divided differences

Unit - 3 : Central differences

Unit - 4 : Errors in Interpolation formula & Least squares approximation

Unit - 5 : Inverse Interpolation.

UNIT–1 : ORDINARY FINITE DIFFERENCES

Contents

1.1 AIMS AND OBJECTIVES

By the time you complete this unit you will able to : (i) Define the operators Δ, ∇, and E and establish various relations between them, (ii) State the problem of interpolation and derive Newton's forward and backward difference interpolation formula and should use them conveniently for estimating the interpolation polynomials, (iii) State the problem of subtabulation and tabulate the data within the required tabular form.

1.2 INTRODUCTION

Numerical analysis deals with tables of numbers. These tables may have arisen as sets of observations from a physical problem or may be obtained from evaluating a function at different points. We often come across examples of logarithmic tables and trigonometric tables where we are required to find values of these functions at points not given in the tables. To get a reasonably good approximation of the "interpolated value" it is very important that the interval between successive values is small enough to display the variation of the tabulated function. Since the time of Newton, Finite Differences have been used extensively to achieve this objective. The first differences are obtained by subtracting each value from the succeeding value in the given table. The second differences are obtained by repeating this operation on the first differences. This process is continued to obtain higher order differences. There are mainly three important sets of finite differences (forward, backward and central) that we can use depending on the nature of the problem though these represent the same sets of numbers. These in turn result in the more useful operators similar to differentiation operators.

1.3 FORMATION OF DIFFERENCE TABLE

1.3.1 Forward Difference Operator Δ

Let a table of functional values $f(a), f(a+h), f(a+2h)$ ... of a function f (not known explicitely and is to be determined by interpolation) be formed corresponding to the independent variable $x = a, a+h, a+2h$ .....

The difference between consecutive values of x (here a constant, h) is called the *interval of differencing.*

We define the *forward difference operator* Δ by the equation

$$\Delta f(x) = f(x+h) - f(x)$$

The quantity $\Delta f(x)$ is called the first difference of $f(x)$.

The second differences of $f(x)$ is given by

$$\begin{aligned}\Delta^2 f(x) &= \Delta\{f(x+h) - f(x)\} \\ &= \Delta f(x+h) - \Delta f(x)\end{aligned}$$

In general the n^{th} difference of $f(x)$ will be defined by the recursion relation.

$$\Delta^n f(x) = \Delta\left\{\Delta^{n-1} f(x)\right\} = \Delta^{n-1} f(x+h) - \Delta^{n-1} f(x)$$

From the definition of $\Delta f(x)$, corresponding to $x = a, a+h, a+2h, \ldots$ we have

$$\begin{aligned}\Delta f(a) &= f(a+h) - f(a) \\ \Delta f(a+h) &= f(a+2h) - f(a+h) \\ \Delta f(a+2h) &= f(a+3h) - f(a+2h) \text{ etc.}\end{aligned}$$

The respective second differences are

$$\begin{aligned}\Delta^2 f(a) &= \Delta f(a+h) - \Delta f(a) \\ &= f(a+2h) - 2f(a+h) + f(a), \\ \Delta^2 f(a+h) &= \Delta f(a+2h) - \Delta f(a+h) \\ &= f(a+3h) - 2f(a+2h) + f(a+h), \text{ etc.}\end{aligned}$$

Similarly the third difference $\Delta^3 f(x)$ at $x = a$ can be obtained as

$$\begin{aligned}\Delta^3 f(a) &= \Delta^2 f(a+h) - \Delta^2 f(a) \\ &= f(a+3h) - 3f(a+2h) + 3f(a+h) - f(a).\end{aligned}$$

SAQ1 : If $f(x) = x^2 + 2x + 1$ and $h = 2$, compute $\Delta f(x)$.

SAQ 2 : What is the forward difference of a constant function?

SAQ 3 : Show that $\Delta[f(x) + g(x)] = \Delta f(x) + \Delta g(x)$ and $\Delta[c f(x)] = c[\Delta f(x)]$, c is a constant.

Denoting $a, a+h, a+2h$ ..... by x_0, x_1, x_2 ..... and the functional values $f(a), f(a+h), f(a+2h)$ ... by y_0, y_1, y_2 ..... then $y_1 - y_0, y_2 - y_1, y_3 - y_2$ ... are called the first differences of the function y. Denoting these differences by $\Delta y_0, \Delta y_1, \Delta y_2$ ..., we have $\Delta y_0 = y_1 - y_0$; $\Delta y_1 = y_2 - y_1$, ... the 2nd differences are defined by

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = y_2 - 2y_1 + y_0.$$

Like this we can define the higher order differences. This type of notation is useful in the formation of difference table.

SAQ 4 : Compute $\Delta^3 y_1$.

1.3.2 The Difference Table

The above differences are shown in the following table known as the difference table.

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0				
		$\Delta y_0 = y_1 - y_0$			
x_1	y_1		$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$		
		$\Delta y_1 = y_2 - y_1$		$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$	
x_2	y_2		$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$		$\Delta^4 y_0 = \Delta^3 y_1 - \Delta^3 y_0$
		$\Delta y_2 = y_3 - y_2$		$\Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1$	
x_3	y_3		$\Delta^2 y_2 = \Delta y_3 - \Delta y_2$		
		$\Delta y_3 = y_4 - y_3$			
x_4	y_4				

Observe that Δy_0 is written in the coloumn of Δy and in between y_0 and y_1 and $\Delta^2 y_0$ under $\Delta^2 y$ column and in between Δy_0 and Δy_1 etc .... Such a table is called diagonal difference table.

Ex : Form the difference table for the data

x	0	2	4	6	8
y	1	9	25	49	81

Identify Δy_0, Δy_2, $\Delta^2 y_2$, $\Delta^3 y_2$.

Sol : Observe that the tabular values are the functional values for the function given in SAQ 1; i.e. $f(x) = x^2 + 2x + 1$.

Difference Table

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
0	1			
		8		
2	9		8	
		16		0
4	25		8	
		24		0
6	49		8	
		32		
8	81			

Here, $\Delta y_0 = 8$; $\Delta y_2 = 16$; $\Delta^2 y_2 = 8$, $\Delta^3 y_2 = 0$.

1.3.3 Differences of a polynomial

Theorem 1 :

> The n^{th} difference of a polynomial of n^{th} degree is a constant when the values of the independent variable are at equal intervals.

Proof : Let a polynomial of n^{th} degree in x be $f(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \ldots + a_{n-1} x + a_n$ where n is a positive integer, $a_0, a_1, a_2 \ldots a_n$ are constants and $a_0 \neq 0$. Now

$$\begin{aligned} f(x+h) &= a_0 (x+h)^n + a_1 (x+h)^{n-1} + \ldots + a_{n-1} (x+h) + a_n \\ \Delta f(x) &= f(x+h) - f(x) \\ &= a_0 \left[(x+h)^n - x^n\right] + a_1 \left[(x+h)^{n-1} - x^{n-1}\right] + \ldots + a_{n-1} \left[(x+h) - x\right] \\ &= a_0 . n . h\ x^{n-1} + b_1 x^{n-2} + \ldots + b_{n-1}, \end{aligned}$$

where $b_1, b_2 \ldots b_{n-1}$ are new constants.

This is a polynomial of degree $n - 1$.

Hence the first difference of a polynomial of degree n is a polynomial of degree $n - 1$.

To find the second difference of $f(x)$, we have

$$\begin{aligned} \Delta^2 f(x) &= \Delta [\Delta f(x)] = \Delta \{f(x+h) - f(x)\} \\ &= \Delta f(x+h) - \Delta f(x) \\ &= a_0 nh \left[(x+h)^{n-1} - x^{n-1}\right] + b_1 \left[(x+h)^{n-2} - x^{n-2}\right] + \ldots + b_{n-2} h. \\ &= a_0 n (n-1) h^2 . x^{n-2} + c_1 x^{n-3} + \ldots + c_{n-2} \end{aligned}$$

where $c_1, c_2 \ldots c_{n-2}$ are new constants.

This is a polynomial of degree $n - 2$. By continuing in this manner, we arrive at the result,

$$\Delta^n f(x) = a_0 \cdot n (n-1) \ldots 1 . h^n = a_0 n! h^n$$

The n^{th} difference of a polynomial of n^{th} degree is a constant and all the higher order differences are zero (why?)

The converse of the above theorem is also true; i.e;

If the n^{th} differences of a tabulated function are constant when the values of the independent variable are taken at equal intervals, the function is a polynomial of degree n. (Proof of this proposition is left to the student). This proposition enables us to replace any function by a polynomial of say r^{th} degree, if its r^{th} differences become constant.

1.3.4 Factorial Notation

The product of r factors of which the first factor is x and the successive factors decrease by a constant difference is called a factoral polynomial and is denoted by $x^{(r)}$ where r is a positive integer. Hence

$$x^{(r)} = x (x-h) (x-2h) \ldots \left(x - \overline{r-1}\ h\right);$$

we take $x^{(0)} = 1$.

$$\begin{aligned} \text{Then } \Delta x^{(r)} &= (x+h)^{(r)} - x^{(r)} \\ &= \left[(x+h)\, x\, (x-h) \ldots \left(x - \overline{r-2}\ h\right)\right] - \\ &\qquad \left[x (x-h) \ldots \left(x - \overline{r-1}\ h\right)\right] \\ &= rh\, x^{(r-1)} \end{aligned}$$

Similarly $\Delta^2 x^{(r)} = \Delta\left(\Delta x^{(r)}\right) = \Delta\left(rh\, x^{(r-1)}\right) = rh\left(x^{(r-1)}\right)$

$= rh\,(r-1)\,h\,x^{(r-2)} = r(r-1)\,h^2\,x^{(r-2)}$

Continuing this process we get

$\Delta^m x^{(r)} = r(r-1)\ldots\left(r - \overline{m-1}\right) h^m\, x^{(r-m)}$ if $m \le r$.

In particular if $h = 1$

$x^{(r)} = x(x-1)(x-2)\ldots\left(x - \overline{r-1}\right)$.

This notation could be extended to negative integers also

$$x^{(-r)} = \frac{1}{(x+h)(x+2h)\ldots(x+rh)} = \frac{1}{(x+rh)^{(r)}}, \text{ where } r \text{ is a positive integer}$$

$$\Delta x^{(-r)} = (x+h)^{(-r)} - x^{(-r)} = \frac{1}{(x+2h)\ldots\left(x + \overline{r+1}\ h\right)} - \frac{1}{(x+h)\ldots(x+rh)}$$

$$= \frac{(x+h) - \left(x + \overline{r+1}\ h\right)}{(x+h)(x+2h)\ldots(x+rh)\left(x + \overline{r+1}\ h\right)} = -rh\ x^{(-r-1)}$$

Similarly, $\Delta^2 x^{(-r)} = \Delta\left(\Delta x^{(-r)}\right) = -rh\,\Delta x^{(-r-1)}$

$= -rh\,(-r-1)\,h\,x^{(-r-2)}$

$= r(r+1)\,h^2\,x^{(-r-2)}$

and $\Delta^m\left(x^{-r}\right) = (-1)^m\, r(r+1)\ldots(r+m-1)\,h^m\,x^{(-r-m)}$, if $m \le r$

Note : Whenever $h = 1$,

$$x^{(-r)} = \frac{1}{(x+1)(x+2)\ldots(x+r)}$$

Clearly we have $x^{(r)}(x-r)^{(n)} = x^{(r+n)}$

where r and n are positive or negative integers. This is referred to as the exponential law for psuedo exponents, when $h = 1$.

Ex. : $x^{(2)} = x(x-1)$, $(x-2)^{(3)} = (x-2)(x-3)(x-4)$

$\therefore x^{(2)}(x-2)^{(3)} = x(x-1)(x-2)(x-3)(x-4)$

$= x^{(5)}$

The function $(x)_n$ defined as follows is called the factorial function

$$(x)_n = \prod_{r=1}^{n}(x+r-1)$$

$= x(x+1)(x+2)\ldots(x+n-1), \qquad n \ge 1$

$(x)_0 = 1$.

Ex. 1 : Represent the function $f(x) = x^4 - 12x^3 + 24x^2 - 30x + 9$ and its successive differences in factorial notation.

Let $f(x) = x^4 - 12x^3 + 24x^2 - 30x + 9$

$\equiv x(x-1)(x-2)(x-3) + bx(x-1)(x-2) + cx(x-1) + dx + 9$

where b, c and d are constants to be determined.

Putting $x = 1$, we get $d = -17$. Similarly, putting $x = 2$ and 3 in turn, we find $c = -5$, $b = -6$.

$$\therefore f(x) = x^{(4)} - 6x^{(3)} - 5x^{(2)} - 17x^{(1)} + 9.$$

$$\text{Hence } \Delta f(x) = 4x^{(3)} - 18x^{(2)} - 10x^{(1)} - 17,$$

$$\Delta^2 f(x) = 12x^{(2)} - 36x^{(1)} - 10,$$

$$\Delta^3 f(x) = 24x^{(1)} - 36.$$

$$\Delta^4 f(x) = 24; \Delta^5 f(x) = 0.$$

Note : The result of differencing $x^{(r)}$ is analogous to that of differentiating x^r.

Ex. 2 : Find the function whose first difference is $9x^2 + 11x + 5$.

Let $f(x)$ be the required function so that we have

$$\Delta f(x) = 9x^2 + 11x + 5$$

$$= 9x(x-1) + bx + c.$$

Putting $x = 0$ and 1 successively we find $c = 5$, $b = 20$.

$$\therefore \Delta f(x) = 9x^{(2)} + 20x^{(1)} + 5.$$

$$\text{Hence } f(x) = 3x^{(3)} + 10x^{(2)} + 5x^{(1)} + k$$

Where k is a constant.

$$\text{Thus } f(x) = 3x(x-1)(x-2) + 10x(x-1) + 5x + k$$

$$= 3x^3 + x^2 + x + k$$

1.3.5 The Backward Difference ∇ (Nabla)

Definition

Let f be a function defined at points $a, a+h, a+2h, \ldots$ in the domain of definition and f assumes the values $f(a), f(a+h), f(a+2h), \ldots$ at these points. Then the backward difference operator ∇ is defined as

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

Thus, for backward difference operator ∇, the points to the left of x are taken into consideration where as for forward difference operator Δ the points to the right of x are taken into consideration. The higher powers of ∇ are defined recursively as :

$$\nabla^2 f(x) = \nabla(\nabla f(x)) = \nabla(f(x) - f(x-h))$$

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

$$= [f(x) - f(x-h)] - [f(x-h) - f(x-2h)]$$

$$= f(x) - 2f(x-h) + f(x-2h)$$

$$\text{and } \nabla^k f(x) = \nabla^{k-1}(\nabla f(x)) = \nabla^{k-1}(f(x) - f(x-h))$$

$$= \nabla^{k-1} f(x) - \nabla^{k-1} f(x-h), \text{ where } k \text{ is an integer.}$$

Remark 1 : It is easy to observe that ∇ and Δ are related as $\nabla f(x) = \Delta f(x-h)$ and $\nabla^k f(x) = \Delta^k f(x-kh)$.

Recall $\Delta f(x) = f(x+h) - f(x)$ so that $\Delta f(x-h) = f(x-h+h) - f(x-h)$

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

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
x_0	y_0				
		$\nabla y_1 = y_1 - y_0$			
x_1	y_1		$\nabla^2 y_2 = \nabla y_2 - \nabla y_1$		
		$\nabla y_2 = y_2 - y_1$		$\nabla^3 y_3 = \nabla^2 y_3 - \nabla^2 y_2$	
x_2	y_2		$\nabla^2 y_3 = \nabla y_3 - \nabla y_2$		$\nabla^4 y_4 = \nabla^3 y_4 - \nabla^3 y_3$
		$\nabla y_3 = y_3 - y_2$		$\nabla^3 y_4 = \nabla^2 y_4 - \nabla^2 y_3$	
x_3	y_3		$\nabla^2 y_4 = \nabla y_4 - \nabla y_3$		
		$\nabla y_4 = y_4 - y_3$			
x_4	y_4				

Example 1 : The common logarithms of 10, 20, 30, 40 and 50 are known from the tables as 1.0000, 1.3010, 1.4771, 1.6021 and 1.6990 respectively. Construct the Backward difference table for this data and determine $\nabla^4 f(x+4h)$ where $h = 10$.

Backward difference table

x	y	$\nabla f(x)$	$\nabla^2 f(x)$	$\nabla^3 f(x)$	$\nabla^4 f(x)$
10	1.0000				
		0.3010			
20	1.3010		– 0.1249		
		0.1761		0.0738	
30	1.4771		– 0.0511		– 0.0508
		0.1250		0.0230	
40	1.6021		– 0.0281		
		0.0969			
50	1.6990				

Here $\nabla f(50) = f(50) - f(40) = 0.0969$ where as $\Delta f(40) = 0.0969$ etc. and $\nabla^4 f(50) = -0.0508$.

1.4 RELATIONS BETWEEN THE OPERATORS Δ, ∇, D and E

Let $y = f(x)$ be the function of x and let $a, a+h, a+2h, \ldots$ be the consecutive values of x. Then we know

$$\Delta f(a) = f(a+h) - f(a)$$

We define an operator E, called the *displacement operator*, by the equation

$$\mathrm{E}f(a) = f(a+h).$$

The operator E is also referred to as the *shift operator*, since it results in the next value of the function. Connecting E and Δ, we have

$$\begin{aligned}\Delta f(a) &= f(a+h) - f(a)\\ &= \mathrm{E}f(a) - f(a).\\ \text{Hence } \mathrm{E}f(a) &= f(a) + \Delta f(a)\\ &= (1+\Delta) f(a).\end{aligned}$$

Since $f(a)$ is arbitrary, this result suggests the following relation between the two operators Δ and E.

$$\boxed{\mathrm{E} \equiv 1 + \Delta}$$

the operand $f(a)$ on each side being understood. Unity is an operator with the property that it operates on $f(x)$ to leave $f(x)$ unaltered.

Let E^n stand for the operation E being carried n times.

Then

$$\begin{aligned}\mathrm{E}^2 f(a) &= \mathrm{E} \cdot \mathrm{E} f(a)\\ &= \mathrm{E} f(a+h)\\ &= f(a+2h),\\ &\cdots\\ \mathrm{E}^n f(a) &= \mathrm{E}^{n-1}\, \mathrm{E} f(a)\\ &= \mathrm{E}^{n-1} f(a+h) = f(a+nh).\end{aligned}$$

The inverse operator is defined by

$$\mathrm{E}^{-1} f(x) = f(x-h)$$

so that,

$$\begin{aligned}\mathrm{E}^{-1} f(a) &= f(a-h),\\ \mathrm{E}^{-1} f(a+h) &= f(a),\\ &\cdots,\\ \mathrm{E}^{-1} f(a+nh) &= f[a+(n-1)h].\end{aligned}$$

The backward difference operator ∇ is already defined as

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

so that we have

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

This relationship suggests

$$\boxed{\nabla = 1 - \mathrm{E}^{-1}.}$$

Thus we have established

$$\Delta f[a + (n-1)h] = \nabla f(a + nh),\ \Delta = E - 1,\ \nabla = 1 - E^{-1}.$$

In fact $\Delta^r f(a + kh) = \nabla^r f[a + (k + r)h]$.

Notes :

1. The expressions $E^2 f^2(x)$ and $[E f(x)]^2$ are not identical as evident from the following example when $f(x) = x$.

$$E^2(x^2) = E \cdot E(x^2) = E(x + h)^2 = (x + 2h)^2$$

and $(E x)^2 = (x + h)^2$. $\therefore E^2 x^2 \neq (E x)^2$.

2. $$E^2 f(a) = f(a + 2h) = f(a) + 2\Delta f(a) + \Delta^2 f(a)$$
$$= (1 + \Delta)^2 f(a),$$
$$\therefore E^2 = (1 + \Delta)^2.$$

In general, for any positive integer n,

$$\boxed{E^n = (1 + \Delta)^n}$$

3. $$E^0 = 1.$$

S A Q 5 : Prove that $f_{x+nh} = \sum_{i=0}^{\infty} \binom{n}{i} \Delta^i f_x$ and hence prove that $f_x = E^x f_0 = \sum_{i=0}^{\infty} \binom{x}{i} \Delta^i f_0$.

Operator D : If $D^n f(x)$ represents the n^{th} derivative of $f(x)$, Taylor's series expansion for $f(x + h)$ is given by,

$$f(x + h) = f(x) + h\,Df(x) + \frac{h^2}{2!} D^2 f(x) + \ldots + \frac{h^n}{n!} D^n f(x) + \ldots$$

$$\text{or} \quad Ef(x) = \left(1 + h\,D + \frac{h^2 D^2}{2!} + \ldots + \frac{h^n D^n}{n!} + \ldots\right) f(x)$$

$$= e^{hD} f(x),$$

so that we have the symbolic equivalence,

$$\boxed{E \equiv e^{hD} = 1 + \Delta.}$$

Taking logarithms both sides, we have

$$hD = \log_e(1 + \Delta) = \Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \ldots$$

$$\text{Also } E^{-1} = e^{-hD} = 1 - \nabla,$$

so that

$$hD = -\log(1 - \nabla) = \nabla + \frac{1}{2}\nabla^2 + \frac{1}{3}\nabla^3 + \ldots$$

1.4.1 Properties of the operators Δ, ∇ and E.

The symbols Δ, ∇ and E obey the following three laws of Algebra namely,

i) Distributive ii) Commutative and iii) Index laws.

(i) $\Delta [f(x) + g(x)]$

$$= [f(x+h) + g(x+h)] - [f(x) + g(x)]$$
$$= [f(x+h) - f(x)] + [g(x+h) - g(x)]$$
$$= \Delta f(x) + \Delta g(x).$$

Similarly,

$$\nabla [f(x) + g(x)] = \nabla f(x) + \nabla g(x)$$
$$E [f(x) + g(x)] = Ef(x) + Eg(x).$$

(ii) $$\Delta [cf(x)] = cf(x+h) - cf(x)$$
$$= c [f(x+h) - f(x)]$$
$$= c \Delta f(x).$$

Similarly

$$\nabla [cf(x)] = c \nabla f(x),$$
$$E [cf(x)] = c Ef(x),$$

where c is a constant.

(iii) $$\Delta^p \Delta^q f(x) = \Delta^p [\Delta^q f(x)]$$
$$= (\Delta\Delta\Delta \ldots p \text{ times}) \cdot (\Delta\Delta\Delta \ldots q \text{ times}) f(x)$$
$$= [\Delta\Delta\Delta \ldots (p+q) \text{ times}] f(x)$$
$$= \Delta^{p+q} f(x), p \text{ and } q \text{ being positive integers.}$$

Similarly,

$$\nabla^p \nabla^q f(x) = \nabla^{p+q} f(x),$$
$$E^p E^q f(x) = E^{p+q} f(x).$$

Notes

1. If m be a positive integer, then we can define symbol Δ^{-m} as

$$\Delta^m [\Delta^{-m} f(x)] = f(x).$$

Similarly

$$\nabla^m [\nabla^{-m} f(x)] = f(x),$$
$$E^m [E^{-m} f(x)] = f(x).$$

2. $E\Delta = \Delta E$.

$$E \Delta f(x) = E [f(x+h) - f(x)] = f(x+2h) - f(x+h).$$

$$\Delta E f(x) = \Delta f(x+h) = f(x+2h) - f(x+h).$$

Hence $$E\Delta = \Delta E$$

3. $\nabla E = \Delta = E \nabla$

$$\nabla E f(x) = \nabla f(x+h)$$
$$= f(x+h) - f(x)$$
$$= \Delta f(x)$$
$$\text{so that } \nabla E = \Delta.$$
$$\text{Also } E \nabla f(x) = E[f(x) - f(x-h)]$$
$$= f(x+h) - f(x)$$
$$= \Delta f(x)$$
$$\text{so that } E \nabla = \Delta.$$
$$\text{Hence } \nabla E = \Delta = E \nabla.$$

Ex. 1 : Prove the following operator relations :

(a) $(1+\Delta)(1-\nabla) = 1$

(b) $E = (1-\nabla)^{-1}$

(c) $\Delta \nabla = \Delta - \nabla$

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

(a) $(1+\Delta)(1-\nabla) f(x)$

$$= EE^{-1} f(x) = Ef(x-h)$$
$$= 1 \cdot f(x)$$
$$\therefore (1+\Delta)(1-\nabla) = 1.$$

(b) $Ef(x) = f(x+h)$

$$(1-\nabla)^{-1} f(x) = \left(E^{-1}\right)^{-1} f(x) \qquad \left(\because \left(E^m\right)^n = E^{mn}\right)$$
$$= Ef(x) = f(x+h).$$
$$\therefore E = (1-\nabla)^{-1}.$$

(c) $$\Delta \nabla f(x) = (E-1)\left(1 - E^{-1}\right) f(x)$$
$$= (E-1)[f(x) - f(x-h)]$$
$$= f(x+h) - f(x) - f(x) + f(x-h).$$
$$(\Delta - \nabla) f(x) = \left(E - 1 - 1 + E^{-1}\right) f(x)$$
$$= f(x+h) - 2f(x) + f(x-h)$$
$$\therefore \Delta \nabla = \Delta - \nabla.$$

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

$= f(x) - f(x-h),$

$E^{-1} \Delta f(x) = E^{-1} (E-1) f(x)$

$= E^{-1} [f(x+h) - f(x)]$

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

$\therefore \nabla = E^{-1} \Delta.$

Ex.2 : Prove the following :

(a) $\Delta \left(f_i g_i\right) = f_i \Delta g_i + g_{i+1} \Delta f_i$

(b) $e^x = \left(\frac{\Delta^2}{E}\right) e^x \cdot \frac{E\, e^x}{\Delta^2 e^x}$

the interval of differencing being h.

(a) $\Delta \left(f_i g_i\right) = (E-1) f_i g_i$

$= E f_i g_i - f_i g_i$

$= f_{i+1} g_{i+1} - f_i g_i$

$f_i \Delta g_i + \Delta g_{i+1} \Delta f_i = f_i (E-1) g_i + g_{i+1} (E-1) f_i$

$= f_i g_{i+1} - f_i g_i + g_{i+1} f_{i+1} - f_i g_{i+1}$

$= f_{i+1} g_{i+1} - f_i g_i$

$\therefore \Delta \left(f_i g_i\right) = f_i \Delta g_i + g_{i+1} \Delta f_i$

(b) $E e^x = e^{x+h}, \Delta e^x = e^{x+h} - e^x$

$= e^x \left(e^h - 1\right),$

$\Delta^2 e^x = e^x \left(e^h - 1\right)^2$

$\left(\frac{\Delta^2}{E}\right) e^x = \left(\Delta^2 E^{-1}\right) e^x = \Delta^2 e^{x-h}$

$= e^{-h} \Delta^2 e^x.$

$\left(\frac{\Delta^2}{E}\right) e^x \cdot \frac{E\, e^x}{\Delta^2 e^x} = \frac{e^{-h}\, e^{x+h}\, \Delta^2 e^x}{\Delta^2 e^x}$

$= e^x$

$\therefore e^x = \left(\frac{\Delta^2}{E}\right) e^x \cdot \frac{E\, e^x}{\Delta^2 e^x}.$

Ex.3 : Prove the following identity :

$$u_x = u_{x-1} + \Delta u_{x-2} + \Delta^2 u_{x-3} + \ldots + \Delta^{n-1} u_{x-n} + \Delta^n u_{x-n}.$$

We know $u_x - \Delta^n u_{x-n} = u_x - \Delta^n E^{-n} u_x$

$= \left\{1 - \frac{\Delta^n}{E^n}\right\} u_x = \left\{\frac{E^n - \Delta^n}{E^n}\right\} u_x$

$$= \frac{1}{E^n}\left(\frac{E^n - \Delta^n}{E - \Delta}\right) u_x \qquad (\because E - \Delta \equiv 1)$$

$$= E^{-n}\left(E^{n-1} + \Delta E^{n-2} + \Delta^2 E^{n-3} + \ldots + \Delta^{n-1}\right) u_x$$

$$= \left(E^{-1} + \Delta E^{-2} + \Delta^2 E^{-3} + \ldots + \Delta^{n-1} E^{-n}\right) u_x$$

$$= u_{x-1} + \Delta u_{x-2} + \Delta^2 u_{x-3} + \ldots + \Delta^{n-1} u_{x-n}$$

$$\therefore u_x = u_{x-1} + \Delta u_{x-2} + \Delta^2 u_{x-3} + \ldots + \Delta^{n-1} u_{x-n} + \Delta^n u_{x-n}.$$

1.5 THE PROBLEM OF INTERPOLATION

Interpolation means insertion or filling up intermediate terms of a series. It is the technique of estimating the value of a function for any intermediate value of the independent variable when the values of the function corresponding to a number of the values of the variable are given. The process of computing the value of a function outside the range of given values of the variable is called *extrapolation*. Assuming that the function can be represented to any desired degree of approximation by a polynomial, the calculus of finite differences is helpful in estimating the missing figures.

1.5.1 Newton's formula for forward interpolation

Let $(x_0, y_0), (x_1, y_1), \ldots (x_n, y_n)$ be the set of $(n+1)$ values of (x, y) where $x_0, x_1, \ldots x_n$ are the equidistant values such that $x_i = x_0 + ih$, $i = 0, 1, 2, \ldots n$. We are required to find a polynomial of n^{th} degree with these tabulated values. Any polynomial $\phi(x)$ of n^{th} degree can be written as

$$\phi(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$
$$+ a_3(x - x_0)(x - x_1)(x - x_2)$$
$$+ \ldots + a_n(x - x_0)(x - x_1)(x - x_2) \ldots (x - x_{n-1}).$$

We choose $a_0, a_1, \ldots, a_n$ so that

$$\phi(x_0) = y_0, \phi(x_1) = y_1, \phi(x_2) = y_2, \ldots \phi(x_n) = y_n$$

Substituting $x = x_0, x_1, x_2, \ldots, x_n$ in $\phi(x)$ and using the above relations, we get

$$y_0 = a_0 \text{ or } a_0 = y_0$$

$$y_1 = a_0 + a_1(x_1 - x_0)$$

$$= a_0 + a_1 h = y_0 + a_1 h$$

$$\therefore a_1 = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}$$

$$y_2 = a_0 + a_1 2h + a_2 \cdot 2h \cdot h$$

$$= y_0 + \frac{y_1 - y_0}{h} \cdot 2h + a_2 2h^2.$$

$$\therefore y_2 - 2(y_1 - y_0) - y_0 = a_2 \cdot 2h^2$$

$$\text{or } a_2 = \frac{y_2 - 2y_1 + y_0}{2h^2}$$

$$= \frac{\Delta^2 y_0}{2! \cdot h^2}$$

Similarly, $$a_3 = \frac{\Delta^3 y_0}{3!\, h^3}, \ldots, a_n = \frac{\Delta^n y_0}{n!\, h^n}$$

Substituting the values of $a_0, a_1, \ldots, a_n$ in $\phi(x)$, we get

$$\phi(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!\, h^2}(x - x_0)\cdot(x - x_1)$$

$$+ \frac{\Delta^3 y_0}{3! \cdot h^3}(x - x_0)(x - x_1)(x - x_2) + \ldots$$

$$+ \frac{\Delta^n y_0}{n! \cdot h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1})$$

This is Newton's formula for forward interpolation written in terms of x.

Substituting $\frac{x - x_0}{h} = u$ or $x = x_0 + hu$, the formula takes the form

$$\phi(x) = \phi(x_0 + hu) = y_0 + u\,\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0$$

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

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

$$\text{or } \phi(x) = y_0 + u^{(1)}\Delta y_0 + \frac{1}{2!}u^{(2)}\Delta^2 y_0$$

$$+ \frac{1}{3!}u^{(3)}\Delta^3 y_0 + \ldots + \frac{u^{(n)}}{n!}\Delta^n y_0.$$

Note : This formula is used mainly for interpolating the values of $f(x)$ near the beginning of a set of tabulated values and for extrapolating values of y at short distance backward (to the left) from y_0.

1.5.2 Newton's formula for backward interpolation

The Newtons formula for forward interpolation is not generally used for interpolating a value of $f(x)$ near the end of the tabular values. To derive a formula for this case, we write

$$\phi(x) = a_0 + a_1(x - x_n) + a_2(x - x_n)(x - x_{n-1})$$

$$+ a_3(x - x_n)(x - x_{n-1})(x - x_{n-2}) + \ldots$$

$$+ a_n(x - x_n)(x - x_{n-1})(x - x_{n-2})\ldots(x - x_1).$$

We determine $a_0, a_1, a_2 \ldots a_n$ so as to make

$$\phi(x_n) = y_n, \phi(x_{n-1}) = y_{n-1}, \ldots \phi(x_0) = y_0.$$

Thus $y_n = a_0$ or $a_0 = y_n$.

$$y_{n-1} = a_0 + a_1 (x_{n-1} - x_n) = y_n + a_1 (-h)$$

$$\text{or } a_1 = \frac{y_n - y_{n-1}}{h} = \frac{\nabla y_n}{h}$$

Where ∇ is the backward difference operator.

$$\text{Again, } y_{n-2} = a_0 + a_1 (x_{n-2} - x_n) + a_2 (x_{n-2} - x_n)(x_{n-2} - x_{n-1})$$

$$= y_n + \frac{y_n - y_{n-1}}{h} \cdot (-2h) + a_2 (-2h)(-h)$$

$$= y_n - 2y_n + 2y_{n-1} + a_2 2h^2.$$

$$\therefore a_2 = \frac{y_n - 2y_{n-1} + y_{n-2}}{2h^2} = \frac{\nabla^2 y_n}{2! \cdot h^2}$$

Similarly

$$a_3 = \frac{\nabla^2 y_n}{3! \cdot h^3}, \ldots a_n = \frac{\nabla^n y_n}{n! \cdot h^n}$$

[since $\nabla y_n = y_n - y_{n-1}, \nabla^2 y_n = \nabla y_n - \nabla y_{n-1}$

$$= y_n - y_{n-1} - (y_{n-1} - y_{n-2})$$

$$= y_n - 2y_{n-1} + y_{n-2}].$$

Substituting these values of $a_0, a_1, a_2 \ldots a_n$ in $\phi(x)$, we get

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

$$+ \frac{\nabla^3 y_n}{3! \cdot h^3}(x - x_n)(x - x_{n-1})(x - x_{n-2}) + \ldots$$

$$+ \frac{\nabla^n y_n}{n! \cdot h^n}(x - x_n)(x - x_{n-1})(x - x_{n-2}) \ldots (x - x_1)$$

This is Newton's formula for backward interpolation written in terms of x.

Substituting $\frac{x - x_n}{h} = u$ or $x = x_n + hu$ in the above, we get

$$\phi(x) = \phi(x_n + hu) = y_n + u \nabla y_n + \frac{u(u+1)}{2!} \nabla^2 y_n$$

$$+ \frac{u(u+1)(u+2)}{3!} \nabla^3 y_n + \ldots$$

$$+ \frac{u(u+1)(u+2) \ldots (u+n-1)}{n!} \nabla^n y_n.$$

Note : This formula is used mainly for interpolating values of y near the end of a set of tabuler values and also for extrapolating values of y a short distance ahead (to the right) of y_n.

Examples

Ex.1 : Construct a difference table from the fallowing values of x and $f(x)$.

x :	0	2	4	6	8
$f(x)$:	7	13	43	145	367

Extend the table to give $f(10)$. When is this possible?

Difference Table

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
0	7			
		6		
2	13		24	
		30		48
4	43		72	
		102		48
6	145		120	
		222		48
8	367		168	
		390		
10	757			

It is clear that $\Delta^4 f(x) = \Delta^5 f(x) = \ldots = 0$. Th[illegible] the property that $\Delta^3 f(x)$ is equal to a constant for all x may be used to extend the talbe of functional values as far as we please. This is done in the above table below the line drawn to give $f(10) = 757$.

This process of computing the value of the function (i.e. $f(10)$) outside the range of the given values is called extrapolation. This is possible if the function is known to run smoothly near the ends of the range of the given values and if h is taken as small as it should be.

Ex.2 : Find the cubic polynomial $f(x)$ which takes on the values $f(0) = -5, f(1) = 1, f(2) = 9, f(3) = 25, f(4) = 55, f(5) = 105$. Use Newton's formula for forward interpolation to compute $f(3.2)$.

Forming the difference table for the function, we have

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
0	–5			
		6		
1	1		2	
		8		6
2	9		8	
		16		6
3	25		14	
		30		6
4	55		20	
		50		
5	105			

Then using Newton's formula for forward interpolation we get

$$f(x) = f(0) + x\,\Delta f(0) + \frac{x(x-1)}{2}\Delta^2 f(0) + \frac{x(x-1)(x-2)}{6}\Delta^3 f(0)$$

Substituting $f(0) = -5$, $\Delta f(0) = 6$, $\Delta^2 f(0) = 2$, $\Delta^3 f(0) = 6$ we obtain the result

$$f(x) = x^3 - 2x^2 + 7x - 5.$$

Interpolating to obtain $f(3.2)$, we find

$$f(3.2) = f(3) + (3.2-3)\,\Delta f(3) + \frac{(3.2-3)(3.2-4)}{2}\Delta^2 f(3) + \frac{(3.2-3)(3.2-4)(3.2-5)}{6}\Delta^3 f(3)$$

$$= 25 + (0.2)(30) + \frac{(0.2)(-0.8)}{2}\times 20 + \frac{(0.2)(-0.8)(-1.8)}{6}\times 6$$

$$= 25 + 6.0 - (0.8)(20) + (.048)(6)$$

$$= 29.688$$

Ex.3 : The population of a town in the decennial census were as under. Estimate the population for the year 1925.

Year x :	1891	1901	1911	1921	1931
Population (in thousands) y :	46	66	81	93	101

The difference table is :

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1891	46				
		20			
1901	66		–5		
		15		2	
1911	81		–3		–3
		12		–1	
1921	93		–4		
		8			
1931	101				

Since five values are given, we assume that the fourth differences are constant. We require the entry for $x = 1925$.

We shall apply Newton's formula for backward interpolation.

$$\text{Hence } u = \frac{x - 1931}{h} = \frac{1925 - 1931}{10} = -0.6$$

$$\therefore y = 101 + (-0.6)(8) + \frac{(-0.6)(0.4)}{2}(-4)$$

$$+ \frac{(-0.6)(0.4)(1.4)}{6}(-1) + \frac{(-0.6)(0.4)(1.4)(2.4)}{24}(-3)$$

$$= 101 - 4.8 + 0.48 + 0.056 - 0.1008 = 96.6352 \text{ (approximately)}$$

1.6 SUBTABULATION

At times one is faced with the necessity of finding a large number of interpolated values for equally - spaced arguments. Thus the table might give u_x for $x = 0\ (5)\ 30$, and it might be desired to compute u_x for $x = 0\ (1)\ 30$. In such a case, interpolation by the formula $u_x = (1 + \Delta)^x u_0$ becomes cumbersome. In practice, one usually needs to subtabulate only for the cases when the finer subdivision is $\frac{1}{2}$, $\frac{1}{5}$, or $\frac{1}{10}$ the length of the original subdivision. We illustrate by means of the following example to obtain 30 (1) 35.

Ex. 1 :

x_0	$u_x = \cos x$	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$	$\Delta^4 u_x$
30	.86603				
		−.04688			
35	.81915		−.00623		
		−.05311		.00041	
40	.76604		−.00582		.00002
		−.05893		.00043	
45	.70711		−.00539		
		−.06432			
50	.64279				

We introduce a new operator θ which has the property

$$\theta u_x = u_{x+h/5} - u_x.$$

Thus θ moves the functional value on by 1° only, whereas Δ moves the function by jumps of 5°.

Consequently

$$(1 + \theta) u_x = u_{x+h/5}$$

$$\therefore (1 + \theta)^5 u_x = u_{x+h}$$

Thus we see that $(1 + \theta)^5 = 1 + \Delta$. Solving

$$\theta = (1+\Delta)^{1/5} - 1 = \frac{\Delta}{5} - \frac{2}{25}\Delta^2 + \frac{6}{125}\Delta^3 - \frac{21}{625}\Delta^4$$

$$\theta^2 = \frac{\Delta^2}{25} - \frac{4}{125}\Delta^3 + \frac{16}{625}\Delta^4,$$

$$\theta^3 = \frac{\Delta^3}{125} - \frac{6}{625}\Delta^4,$$

$$\theta^4 = \frac{\Delta^4}{625}.$$

These powers of θ terminate with Δ^4 since we assume $\Delta^5 = 0$. Substituting, we obtain

$$\theta u_{30} = -.009376 + .0004984 + .000019224 - .000000672$$

$$= -.008859048$$

$$\theta^2 u_{30} = -.000261504,$$

$$\theta^3 u_{30} = .000003012,$$

$$\theta^4 u_{30} = .000000032.$$

We can then build up the complete table for u_x, using θ as our difference operator. Thus

x	u_x	θu_x	$\theta^2 u_x$	$\theta^3 u_x$	$\theta^4 u_x$
30	.866030000				
		- .008859048			
31	.857170952		- .000261504		
		- .009120552		.000003012	
32	.848050400		- .0000258492		.000000032
		- .009379044		.000003044	
33	.838671356		- .0000255448		.000000032
		- .009634492		.000003076	
34	.829036864		- .0000252372		
		- .009886864			
35	.819150000				

Thus we have estimated u_x for $x = 30\ (1)\ 35$.

1.7 SUMMARY

Construction of finite difference tables is important to estimate the missing information from the given tabulated values. Various relations between Δ, ∇ and E are established. The terms interpolation and extrapolation are defined. The Newtons forward difference formula is useful for interpolating the values at the beginning of the table and Newtons backward difference formula is useful for interpolating the values at the end of the table. The method of subtabulation is followed in case of finding a large number of interpolated values for equally spaced arguments.

1.8 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail.*

(i) a) Define ordinary difference of a function and obtain the Newton's formula for forward interpolation.

b) The population of a town was as given below. Estimate the population for the year 1895.

Year x :	1891	1901	1911	1921	1931
Population (in thousands) y :	46	66	81	93	101

[Ans : 54.85 Thousands]

(ii) a) Explain interpolation and obtain the Newton's formula for backward differences

b) Find the cubic polynomial which takes the following values

$y(0) = 1, y(1) = 0, y(2) = 1$ and $y(3) = 10$ and hence obtain $y(4)$

[Ans : $y = x^3 - 2x^2 + 1$, $y(4) = 33$]

(iii) a) What do you mean by Subtabulation? Explain.

b) The function $k(\alpha) = \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - \sin^2 \alpha \cdot \sin^2 \phi}}$ is tabulated below

α^0 :	0	[illegible]	10	15	20
$k(\alpha)$:	1.5708	1.5738	1.5828	1.5981	1.6200

Tabulate $k(\alpha)$ for $\alpha = 0(1)5$.

(iv) Let u_x be a function for which Δ^3 may be considered constant. Develop formula for subtabulation by means of the difference operator θ in the cases :

(a) $\theta u_x = u_{x+h/2} - u_x$

(b) $\theta u_x = u_{x+h/10} - u_x$

(v) Obtain various relations between Δ, ∇, E and state the properties of these operators.

(vi) Prove the following relations.

(a) $\Delta + \nabla = \frac{\Delta}{\nabla} - \frac{\nabla}{\Delta}$,

(b) $\Delta f_i^2 = (f_i + f_{i+1}) \Delta f_i$

(c) $\Delta \left(\frac{f_i}{g_i}\right) = \left(g_i \Delta f_i - f_i \Delta g_i\right) / g_i g_{i+1}$

(d) $\Delta \left(1/f_i\right) = -\Delta f_i / f_i f_{i+1}$.

II. *Briefly answer the following.*

(i) By constructing a difference table, find the 7^{th} term as well as the general term of the sequence.

0, 0, 2, 6, 12, 20

[Ans : 7^{th} term = 30; general term = $r(r-1)$]

(ii) Express the function $2x^3 - 3x^2 + 3x - 10$ and its differences in factorial notation.

[Ans : $2x^{(3)} + 3x^{(2)} + 2x^{(1)} - 10$;

$\Delta y = 6x^{(2)} + 6x^{(1)} + 2$; $\Delta^2 y = 12x^{(1)} + 6$; $\Delta^3 y = 12$]

(iii) Obtain a function whose first difference is $x^3 + 3x^2 + 5x + 12$.

$$\left[\text{Ans : } \frac{x^{(4)}}{4} + 2x^{(3)} + \frac{9}{2}x^{(2)} + 12x^{(1)} + c\right]$$

(iv) The table below gives the values of tan x for $0.01 \le x \le 0.30$.

x :	0.10	0.15	0.20	0.25	0.30
$y = \tan x$:	0.1003	0.1511	0.2027	0.2553	0.3093

Find (i) tan 0.26, (ii) tan 0.12

[Ans : tan (0.26) = 0.2662
tan (0.12) = 0.1205]

(v) In the table below the values of y are consecutive terms of a series of which the number 21.6 is the 6^{th} term. Find the first and tenth terms of the series.

x :	3	4	5	6	7	8	9
y :	2.7	6.4	12.5	21.6	34.3	51.2	72.9

[Ans $y(1) = 0.01$, $y(10) = 100$]

(vi) a) Prove that $y_4 = y_3 + \Delta y_2 + \Delta^2 y_1 + \Delta^3 y_1$

b) Show that $y_4 = y_0 + 4\,\Delta y_0 + 6\,\Delta^2 y_{-1} + 10\,\Delta^3 y_{-1}$ as far as third differences.

(vii) Show that $\sum_{k=0}^{n-1} \Delta^2 f_k = \Delta f_n - \Delta f_0$

(viii) Show that $u_0 + \binom{x}{1}\Delta u_1 + \binom{x}{2}\Delta^2 u_2 + \ldots$

$$= u_x + \binom{x}{1}\Delta^2 u_{x-1} + \binom{x}{2}\Delta^4 u_{x-2} + \ldots$$

where $\binom{x}{r} = \dfrac{x(x-1)\ldots(x-\overline{r-1})}{r!}$

(ix) Evaluate a) Δa^x b) $\Delta^4 (a\,e^x)$ c) $\Delta^2 x^3$ by taking $h = 1$.

[Ans : a) $(a-1)\,a^x$; b) $a\,(e-1)^4\,e^x$; c) $6x + 6$]

(x) Evaluate a) $\dfrac{\Delta^2}{E}x^3$ b) $\dfrac{\Delta^2 x^3}{Ex^3}$ by taking $h = 1$.

$$\left[\text{Ans : a) } 6x\text{; b) } \frac{6}{(1+x)^2}\right]$$

1.9 ANSWERS TO SAQ's

SAQ 1 :
$$\begin{aligned} f(x) &= x^2 + 2x + 1 \\ \Delta f(x) &= f(x+2) - f(x) \qquad (\because h = 2) \\ &= (x+2)^2 + 2(x+2) + 1 - \left\{x^2 + 2x + 1\right\} \\ &= 4(x+1) \end{aligned}$$

SAQ 2 : $f(x) = a$; $\therefore f(x+h) = a$, $\therefore \Delta f(x) = f(x+h) - f(x) = a - a = 0$.

SAQ 3 :
$$\begin{aligned} \Delta[f(x) + g(x)] &= [f(x+h) + g(x+h)] - [f(x) + g(x)] \\ &= [f(x+h) - f(x)] + [g(x+h) - g(x)] \\ &= \Delta f(x) + \Delta g(x). \end{aligned}$$
$$\Delta[cf(x)] = cf(x+h) - cf(x) = c[f(x+h) - f(x)] = c[\Delta f(x)].$$

SAQ 4 :
$$\begin{aligned} \Delta^3 y_1 = \Delta^2 y_2 - \Delta^2 y_1 &= \Delta y_3 - \Delta y_2 - \left\{\Delta y_2 - \Delta y_1\right\}. \\ &= y_4 - y_3 - 2\left\{y_3 - y_2\right\} + y_2 - y_1. \\ &= y_4 - 3y_3 + 3y_2 - y_1. \end{aligned}$$

SAQ 5 : We prove the result by induction. When $n = 1$, $f_{x+h} = Ef_x = f_x + \Delta f_x$, the result is ture. Assume that the result is true for the value of $n - 1$; then

$$E^n f_x = E\left[E^{n-1} f(x)\right] = E\sum_{i=0}^{\infty}\binom{n-1}{i}\Delta^i f_x. \text{ (By induction hypothesis)}$$

But we know that $E = 1 + \Delta$; then

$$\begin{aligned} E^n f_x &= (1+\Delta)E^{n-1} f_x = E^{n-1} f_x + \Delta E^{n-1} f_x \\ &= \sum_{i=0}^{\infty}\binom{n-1}{i}\Delta^i f_x + \sum_{i=0}^{\infty}\binom{n-1}{i}\Delta^{i+1} f_x \\ &= \sum_{i=0}^{\infty}\binom{n-1}{i}\Delta^i f_x + \sum_{j=1}^{\infty}\binom{n-1}{j-1}\Delta^j f_x \end{aligned}$$

The coefficient of $\Delta^k f_x$ $(k = 0, 1, 2 \ldots n)$ is given by

$$\binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n}{k}$$

$$\therefore E^n f_x = \sum_{k=0}^{\infty}\binom{n}{k}\Delta^k f_x.$$

When $n = x$ and $x = 0$,

$$f_x = E^x f_0 = \sum_{i=0}^{\infty}\binom{x}{i}\Delta^k f_0.$$

UNIT–2 : DIVIDED DIFFERENCES

Contents

2.1 AIMS AND OBJECTIVES

By the time you complete this unit you will be able to : (i) Derive Newton's divided difference formula and apply this to solve an interpolation problem, (ii) Derive Lagranges interpolation formula and apply it to solve interpolation problem and apply the formula for the problem of partial fractions.

2.2 INTRODUCTION

The interpolation formulae of unit 1 are useful only when the values of the function are given at equal intervals. But in many real life situations such a situation is very hard to be true. Hence there is a need to study interpolation formulae which could be applied to situations when the tabular values are not equally spaced. The Newton's divided difference formula is useful in situations where the tabular values are not equally spaced. The divided differences are differences obtained in the usual manner and then divided by certain differences of the argument. The Lagrange formula however does not use the differences. The Lagranges formula is a relation between two variables either of which could be treated as an independent variable.

2.3 DIVIDED DIFFERENCE TABLE

2.3.1 *Definition and notation for divided differences*

Suppose that the function u_x is given for distinct values $x = a, b, c, d, \ldots$, where the intervals $b - a, c - b, d - c, \ldots$ are not necessarily equal (note that we do not demand that $a, b, c, d, \ldots$ be arranged in ascending order of magnitude). Then we define the first divided difference of u_x at a and b by the equation.

$$\underset{b}{\Delta} u_a = \frac{u_b - u_a}{b - a}$$

The difference table may then be formed as :

x	u_x	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$
a	u_a			
		Δu_a b		
b	u_b		$\Delta^2 u_a$ bc	
		Δu_b c		$\Delta^2 u_a$ bcd
c	u_c		$\Delta^2 u_a$ cd	
		Δu_c d		
d	u_d			

The second and third divided differences in terms of the functional values can be obtained as :

$$\Delta^2_{bc} u_a = \frac{\Delta_c u_b - \Delta_b u_a}{c - a} = \frac{\dfrac{u_c - u_b}{c - b} - \dfrac{u_b - u_a}{b - a}}{c - a}$$

$$= \frac{u_a}{(a - b)(a - c)} + \frac{u_b}{(b - c)(b - a)} + \frac{u_c}{(c - a)(c - b)}$$

Similarly the third divided difference can be obtained as :

$$\Delta^3_{bcd} u_a = \frac{u_a}{(a - b)(a - c)(a - d)} + \frac{u_b}{(b - c)(b - d)(b - a)}$$

$$+ \frac{u_c}{(c - d)(c - a)(c - b)} + \frac{u_d}{(d - a)(d - b)(d - c)}$$

The n^{th} divided difference of a function u_x is defined by

$$\Delta^n_{bc...lm} u_a = \left(\Delta^{n-1}_{c, ...lm} u_b - \Delta^{n-1}_{bc...l} u_a \right) \Big/ (m - a),$$

where $a, b \ldots m$ are distinct values of x.

The above results are special cases of the following theorem :

Theorem. 1 :

If $a, b, c \ldots j, k$ are $r + 1$ values of the argument x, then

$$\Delta^r_{bc...jk} u_a = \frac{u_a}{(a - b)(a - c) \ldots (a - k)} + \frac{u_b}{(b - a)(b - c) \ldots (b - k)}$$

$$+ \ldots + \frac{u_k}{(k - a)(k - b) \ldots (k - j)}$$

This can be proved by mathematical induction and therefore omitted.

SAQ 1 : Find the values of $\underset{y}{\Delta} x^2$; $\underset{yz}{\Delta^2} x^2$.

Ex. 1 : Build up a divided difference table, given that

$u_{-2} = 5, u_0 = 3, u_3 = 15, u_4 = 47, u_9 = 687.$

x	u_x	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$
–2	5	$\underset{0}{\Delta} u_{-2} = \frac{3-5}{0-(-2)} = -1$		
0	3		$\underset{0,3}{\Delta^2} u_{-2} = \frac{4-(-1)}{3-(-2)} = 1$	
		$\underset{3}{\Delta} u_0 = \frac{15-3}{3-0} = 4$		$\underset{0,3,4}{\Delta^3} u_{-2} = \frac{7-1}{4-(-2)} = 1$
3	15		$\underset{3,4}{\Delta^2} u_0 = \frac{32-4}{4-0} = 7$	
		$\underset{4}{\Delta} u_3 = \frac{47-15}{4-3} = 32$		$\underset{3,4,9}{\Delta^3} u_0 = \frac{16-7}{9-0} = 1$
4	47		$\underset{4,9}{\Delta^2} u_3 = \frac{128-32}{9-3} = 16$	
		$\underset{9}{\Delta} u_4 = \frac{687-47}{9-4} = 128$		
9	687			

Ex. 2 : Build up a divided difference table, given that

$u_3 = 15, u_{-2} = 5, u_9 = 687, u_0 = 3, u_4 = 47$

(Obviously, we are dealing with the same function as in Ex. 1)

x	u_x	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$
3	15			
		2		
–2	5		1	
		–1		1
0	3		7	
		76		1
9	687		13	
		128		
4	47			

It might be noted that the triangular arrangement of a divided difference table is extremely convenient, we have

$\underset{-2,0}{\Delta^2} u_3 = \frac{-1-2}{0-3} = 1$ and the triangle drawn immediately points to the proper divisor 0 – 3.

We note in both the examples above, the third divided differences of the function (actually $x^3 - 5x + 3$) are constant. This is a special case of the following theorem.

SAQ 2 : Obtain the relation between divided differences and ordinary differences.

Note : In Ex. 1 we found $\underset{0,3}{\Delta^2} u_{-2} = 1$; in Ex. 2 $\underset{-2,0}{\Delta^2} u_3 = 1$.

In fact this is an immediate consequence of the following theorem.

Theorem. 3 : (The symmetry property).

> If a divided difference $\underset{bc\ldots jk}{\Delta^r} u_a$ is given, then it is unaltered by any permutation of the letters $a, b, c, \ldots, j, k$.

The proof of this theorem follows, when we note that the expression in theorem 1 is a symmetric function of all $(r + 1)$ letters $a, b, \ldots, j, k$.

SAQ 3 : What can you say about the n^{th} divided differences of a polynormal of n^{th} degree?

2.4 NEWTON'S DIVIDED DIFFERENCE FORMULA

Consider the function u_x for arguments $x, a, b, c, d, \ldots, j, k$; then

$$\underset{a}{\Delta} u_x = \underset{x}{\Delta} u_a = \frac{u_x - u_a}{x - a},$$

and solving for u_x,we obtain

$$u_x = u_a + (x - a) \underset{x}{\Delta} u_a \qquad \ldots (1)$$

From the symmetry property, we obtain

$$\underset{bx}{\Delta^2} u_a = \underset{ab}{\Delta^2} u_x = \frac{\underset{a}{\Delta} u_x - \underset{b}{\Delta} u_a}{x - b};$$

thence, using the symmetry property,

$$\underset{x}{\Delta} u_a = \underset{b}{\Delta} u_a + (x - b) \underset{bx}{\Delta^2} u_a,$$

and, substituting in (1), we obtain

$$u_x = u_a + (x - a) \underset{b}{\Delta} u_a + (x - a)(x - b) \underset{bx}{\Delta^2} u_a \qquad \ldots (2)$$

Finally, $$\underset{bcx}{\Delta^3} u_a = \underset{abc}{\Delta^3} u_x = \frac{\underset{ab}{\Delta^2} u_x - \underset{bc}{\Delta^2} u_a}{x - c}$$

and $$\underset{bx}{\Delta^2} u_a = \underset{bc}{\Delta^2} u_a + (x - c) \underset{bcx}{\Delta^3} u_a$$

Substitution in (2) gives the expression

$$u_x = u_a + (x - a) \underset{b}{\Delta} u_a + (x - a)(x - b) \underset{bc}{\Delta^2} u_a + (x - a)(x - b)(x - c) \underset{bcx}{\Delta^3} u_a .$$

Continue this process until n^{th} differences are reached under the assumtpion that u_x is represented by an n^{th} degree polynomial, all higher differences vanish and we have Newton's divided difference formula

$$u_x = u_a + A \underset{b}{\Delta u_a} + AB \underset{bc}{\Delta^2 u_a} + ABC \underset{bcd}{\Delta^3 u_a} + \ldots$$

$$+ ABC \ldots J \underset{bc\ldots k}{\Delta^n u_a} \qquad \ldots (3)$$

where there are $(n + 1)$ arguments $a, b, c, \ldots k$, and where we use the abbreviations

$x - a = A, x - b = B, \ldots, x - k = K.$

If the arguments $a, b, c, d, \ldots$ are taken as 0, 1, 2, ..., (i.e., equally spaced) then

$$\Delta^r u_a = \frac{\Delta^r u_0}{r!}$$

and (3) specializes to the result

$$u_x = u_0 + \binom{x}{1} \Delta u_0 + \binom{x}{2} \Delta^2 u_0 + \ldots + \binom{x}{n} \Delta^n u_0,$$

which is Newton's formula for forward interpolation of Unit 1.

Examples

Ex. 1 : Find the polynomial such that $f(0) = 1, f(1) = 3, f(3) = 55$, using Newton's divided difference formula.

We build up the divided difference table for the given data :

x	f_x	Δf_x	$\Delta^2 f_x$
0	1		
		2	
1	3		8
		26	
3	55		

We note $\underset{1}{\Delta f_0} = 2, \quad \underset{1,3}{\Delta^2 f_0} = 8.$

Applying Newton's divided difference formula, we get

$$f_x = f_0 + (x - 0) \underset{1}{\Delta f_0} + (x - 0)(x - 1) \underset{1,3}{\Delta^2 f_0}$$

$$= 1 + x.\, 2 + x(x - 1).\, 8 = 8x^2 - 6x + 1$$

the required polynomial.

Ex. 2 : Use Newton's formula to obtain a polynomial approximation to the data

$u_{10} = 355, u_0 = -5, u_8 = -21, u_1 = -14, u_4 = -125$

The divided difference table for the data is given below.

x	u_x	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$	$\Delta^4 u_x$
10	355				
		36			
0	–5		19		
		–2		2	
8	–21		1		0
		–1		2	
1	–14		9		
		–37			
4	–125				

(The line in the table indicates the path of the formula)

Applying Newton's divided difference formula, we get

$$u_x = 355 + (x - 10)\,36 + (x - 10)(x)(19) + (x - 10)(x)(x - 8)(2)$$

$$= 2x^3 - 17x^2 + 6x - 5, \text{ the required polynomial approximation.}$$

Ex. 3 : Using Newton's divided difference formula estimate the polynomial approximation to the data given by

$u_0 = 5$; $u_2 = 26$; $u_3 = 58$; $u_4 = 112$; $u_7 = 466$, $u_9 = 922$.

Sol : The Newton's divided difference formula is

$$u_x = u_0 + x\,\Delta u_0 + x(x-2)\,\Delta^2_{2,3} u_0 + x(x-2)(x-3)\,\Delta^3_{2,3,4} u_0$$

$$+ x(x-2)(x-3)(x-4)\,\Delta^4_{2,3,4,7} u_0$$

$$+ x(x-2)(x-3)(x-4)(x-7)\,\Delta^5_{2,3,4,7,9} u_0$$

We now construct the Newton's divided difference formula and substitute the values from the table to compute the polynomial u_x.

x	u_x	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$	$\Delta^4 u_x$
0	4				
		11			
2	26		7		
		32		1	
3	58		11		0
		54		1	
4	112		16		0
		118		1	
7	466		22		
		228			
9	922				

The Polynomial is obtained by substituting these values in u_x.

$$u_x = 4 + x(11) + x(x-2)(7) + x(x-2)(x-3)\,1$$

$$= 4 + 11x + 7x^2 - 14x + x^3 - 5x^2 + 6x$$

$$= x^3 + 2x^2 + 3x + 4.$$

2.5 THE LAGRANGE FORMULA FOR INTERPOLATION

In 2.4 we considered Newton's divided difference formula that makes use of divided differences. Now, we derive a formula useful for functions which are not tabulated at equal intervals called Lagrange formula, which does not use differences at all. However, Lagrange formula is quite combersome to workout and has a disadvantage in that the degree of the approximating polynomial must be chosen at the outset, so it is mainly of theoretical interest only.

Suppose that the function u_x is tabulated at $n + 1$ distinct values (not necessarily equidistant), $a, b, c \ldots, j, k$ and assume that u_x is represented by an nth degree polynomial.

Then the $(n + 1)^{th}$ order differences vanish. i.e.,

$$\Delta^{n+1}_{a,b,c,\ldots k} u_x = 0.$$

From theorem 1, we can write this equation as

$$\frac{u_x}{(x-a)(x-b)\ldots(x-k)} + \frac{u_a}{(a-b)(a-c)\ldots(a-k)(a-x)} + \ldots +$$

$$\frac{u_k}{(k-x)(k-a)\ldots(k-j)} = 0 \qquad \ldots (1)$$

Multiplying the equation (1) throughout by $(x-a)(x-b)\ldots(x-k)$, we get

$$u_x = \frac{(x-b)(x-c)\ldots(x-k)}{(a-b)(a-c)\ldots(a-k)} u_a + \frac{(x-a)(x-c)\ldots(x-k)}{(b-a)(b-c)\ldots(b-k)} u_b$$

$$+ \ldots + \frac{(x-a)(x-b)\ldots(x-j)}{(k-a)(k-b)\ldots(k-j)} u_k \qquad \ldots (2)$$

This formula (2) is called Lagranges interpolation formula. It expresses u_x in terms of the $n + 1$ given functional values $u_a, u_b, \ldots, u_k$.

Now replace $a, b, c, \ldots, j, k$ by $x_0, x_1, \ldots x_{n-1}, x_n$ and their corresponding functional values $u_a, u_b, \ldots, u_k$ by $y_0, y_1, \ldots y_n$ in (2).

Then formula (2) can be written as

$$y_x = \frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)} y_0 + \ldots$$

$$+ \frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})} \cdot y_n \qquad \ldots (3)$$

SAQ 4 : Obtain Lagranges interpolation formula for equal intervals.

Since Lagrange formula is merely a relation between two variables either of which may be taken as the independent variable, it is evident that by considering y as the independent variable we can write a formula giving x as a function of y by interchanging x and y in (3). Hence

$$x_y = \frac{(y-y_1)(y-y_2)\cdots(y-y_n)}{(y_0-y_1)(y_0-y_2)\cdots(y_0-y_n)}\,x_0 +$$

$$\frac{(y-y_0)(y-y_2)\cdots(y-y_n)}{(y_1-y_0)(y_1-y_2)\cdots(y_1-y_n)}\,.\,x_1$$

$$+\ \ldots\ + \frac{(y-y_0)(y-y_1)\cdots(y-y_{n-1})}{(y_n-y_0)(y_n-y_1)\cdots(y_n-y_{n-1})}\,.\,x_n \qquad \ldots (4)$$

This formula (4) is useful for inverse interpolation, i.e., to find the value of the independent variable corresponding to a given value of the function, which we shall see in Unit 5.

Lagrange formula (3) can be written in the form

$$y = \sum_{i=0}^{n} \frac{\Pi_n(x)}{(x-x_i)\,\Pi_n'(x_i)}\ y_i \qquad \ldots (5)$$

where $\Pi_n(x) = (x-x_0)(x-x_1)\ldots(x-x_n)$

and $\Pi_n'(x) = \frac{d}{dx}\left(\Pi_n(x)\right)$

SAQ 5 : Verify the formula (5).

2.5.2 Application to Partial Fractions

Equation (2) is intimately connected with the partial fraction break-up of a rational function. For instance, let $F(x) = u(x)/v(x)$ $(v(x) \neq 0)$ be a given rational function, where $u(x)$ has degree not greater than n, where $v(x) = (x-a)(x-b)\ldots(x-k)$, and the zeros of $v(x)$ are distinct. From (1),

$$F(x) = \frac{u(x)}{(x-a)(x-b)\ldots(x-k)} = \frac{u(a)}{(a-b)(a-c)\ldots(a-k)}\,\frac{1}{x-a}$$

$$+\ \ldots\ + \frac{u(k)}{(k-a)(k-b)\ldots(k-j)}\ \frac{1}{x-k}$$

The result can be written in the form

$$F(x) = \frac{u(a)}{v'(a)}\,.\,\frac{1}{x-a} + \ldots + \frac{u(k)}{v'(k)}\,\frac{1}{x-k},$$

which is the usual form of the partial fraction decomposition of $F(x)$.

Examples

Ex. 1 : Certain corresponding values of x and $\log_{10} x$ are (300, 2.4771), (304, 2.4829) (305, 2.4843), (307, 2.4871). Find $\log_{10} 301$.

From formula (4), we obtain

$$\log_{10} 301 = \frac{(-3)(-4)(-6)}{(-4)(-5)(-7)}(2.4771) + \frac{(1)(-4)(-6)}{(4)(-1)(-3)}(2.4829)$$

$$+ \frac{(1)(-3)(-6)}{(5)(1)(-2)}(2.4843) + \frac{(1)(-3)(-4)}{(7)(3)(2)}(2.4871)$$

$$= 1.2739 + 4.9658 - 4.4717 + 0.7106 = 2.4786.$$

Ex. 2 : If $y_1 = 4$, $y_3 = 12$, $y_4 = 19$ and $y_x = 7$, find x.

Using formula (5), we have

$$x = \frac{(-5)(-12)}{(-8)(-15)}(1) + \frac{(3)(-12)}{(8)(-7)}(3) + \frac{(3)(-5)}{(15)(7)}(4)$$

$$= \frac{1}{2} + \frac{27}{14} - \frac{4}{7} = 1.86.$$

Ex. 3 : The mode of a certain frequency curve $y = f(x)$ is very near $x = 9$ and the value of the frequency density $f(x)$ for $x = 8.9$, 9.0, and 9.3 are respectively equal to 0.30, 0.35 and 0.25. Calculate the approximate value of the mode.

We have

x :	8.9	9.0	9.3
$f(x)$:	0.30	0.35	0.25

We shall first find the form of the frequency density $f(x)$. By Lagrange's formula, we get

$$f(x) = \frac{(x-9)(x-9.3)}{(8.9-9)(8.9-9.3)} \times 0.30 + \frac{(x-8.9)(x-9.3)}{(9-8.9)(9-9.3)} \times 0.35$$

$$+ \frac{(x-8.9)(x-9)}{(9.3-8.9)(9.3-9)} \times 0.25$$

or $$f(x) = -\frac{25}{12}x^2 + \frac{453.5}{12}x - 2052.3$$

This is the form of the frequency density. It will be maximum for the value of x lying in the range (8.9, 9.3) for which $f'(x) = 0$ and $f''(x)$ is negative.

Now $f'(x) = 0$ gives

$$-\frac{5}{12}(10x - 90.7) = 0 \text{ or } x = 9.07$$

and $f''(x) = -\frac{25}{6}$ which is a negative quantity.

Hence the mode of the frequency curve $y = f(x)$ is 9.07.

Ex. 4 : Apply Lagrange's formula for interpolation and obtain the values of $f(5)$ and $f(6)$ for the data given below.

x	1	2	3	7
$f(x)$	2	4	8	128

Sol : The Lagranges formula states that

$$\frac{f(x)}{(x-x_0)(x-x_1)\ldots(x-x_n)} = \frac{f(x_0)(x-x_0)^{-1}}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)} +$$

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

Substituting the values of x_1, x_2, x_3 and x_7 in the formula we get

$$\frac{f(x)}{(x-1)(x-2)(x-3)(x-7)}=\frac{f(1)(x-1)^{-1}}{(1-2)(1-3)(1-7)}+$$

$$\frac{f(2)(x-2)^{-1}}{(2-1)(2-3)(2-7)}+\frac{f(3)(x-3)^{-1}}{(3-1)(3-2)(3-7)}$$

$$+\frac{f(7)(x-7)^{-1}}{(7-1)(7-2)(7-3)}$$

That is

$$\frac{f(x)}{(x-1)(x-2)(x-3)(x-7)}=-\frac{1}{12(x-1)}+\frac{4}{5(x-2)}-\frac{1}{(x-3)}+\frac{16}{15(x-7)}$$

$$\therefore f(x) = -\frac{1}{12}(x-2)(x-3)(x-7)+\frac{4}{5}(x-1)(x-3)(x-7)$$

$$-(x-1)(x-2)(x-7)+\frac{16}{15}(x-1)(x-2)(x-3)$$

$$f(5) = -\frac{1}{12}(5-2)(5-3)(5-7)+\frac{4}{5}(5-1)(5-3)(5-7)$$

$$-(5-1)(5-2)(5-7)+\frac{16}{15}(5-1)(5-2)(5-3)$$

$$= 1-\frac{64}{5}+24+\frac{128}{5}=\frac{189}{5}=37.8$$

$$f(6) = -\frac{1}{12}(6-2)(6-3)(6-7)+\frac{4}{5}(6-1)(6-3)(6-7)$$

$$-(6-1)(6-2)(6-7)+\frac{16}{15}(6-1)(6-2)(6-3)$$

$$= 1-12+20+64=73.$$

Remark : Even without interpolation it can be observed by simple inspection that the values of $f(x)$ for a given x are such that the relation between x and $f(x)$ is given by $f(x)=2^x$. As such $f(5)=32$ and $f(6)=64$. But the Lagranges interpolation gives $f(5)=37.8$ and $f(6)=73$. This error is due to the approximation of 2^x as a polynomial of x. Hence care should be taken in using Lagrange's formula.

Ex. 5 : Use Lagrange's formula to express

$$\frac{x^2+6x+1}{(x^2-1)(x-4)(x-6)}$$

as sum of partial fractions

Here $u(x) = x^2+6x+1$,

$v(x) = (x+1)(x-1)(x-4)(x-6)$.

$u(-1) = -4, u(1) = 8, u(4) = 41, u(6) = 73.$

$v'(-1) = (-2)(-5)(-7) = -70.$

$v'(1) = 2(-3)(-5) = 30,$

$v'(4) = (5)(3)(-2) = -30,$

$v'(6) = (7)(5)(2) = 70.$

Substitution in the form of the partial fraction decomposition of $u(x)/v(x)$, we have

$$\frac{x^2 + 6x + 1}{(x^2 - 1)(x - 4)(x - 6)} = \frac{2}{35} \cdot \frac{1}{(x + 1)} + \frac{4}{15} \cdot \frac{1}{(x - 1)}$$

$$-\frac{41}{30} \cdot \frac{1}{x - 4} + \frac{73}{70} \cdot \frac{1}{(x - 6)}$$

2.6 SUMMARY

When the arguments of a function tabulated are not equally spaced, we have introduced the difference operator called divided difference operator. The Newtons divided difference interpolation formula and the Lagrange's interpolation formula have been derived and have discussed their merits and demerits. Lagranges formula is independent of differences but cumbersome in calculations and also should be used with care. Lagrange formula is useful for the partial fraction break-up of a rational function.

2.7 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail*

i. a) Define the divided difference operator and obtain the Newton's divided difference formula.

b) If P (1) = 1, P (3) = 27, P (4) = 64, find the polynomial P (x) satisfying these values and obtain P (1.5).

$$\left[P(x) = 8x^2 - 19x + 12,\ P(1.5) = 1.5\right]$$

ii. a) Show that $\Delta^n f(x)$ is a constant if $f(x)$ is a polynomial of degree n.

b) Show that the nth difference x^n is equal to 1.

iii. a) Obtain the Lagrange interpolation formula for unequal intervals.

b) Using the above formula show that $y_1 = y_3 - 0.3\left(y_5 - y_{-3}\right) + 0.2\left(y_{-3} - y_{-5}\right)$

iv. a) Explain how the Lagrange's formula is useful for expressing $u(x) / v(x)$ as a partial fraction.

b) Express $\frac{3x^2 + x + 1}{(x - 1)(x - 2)(x - 3)}$ in partial fractions using lagrange's principle.

$$\left[\frac{5}{2(x - 1)} - \frac{15}{x - 2} - \frac{31}{2(x - 3)}\right]$$

II. *Briefly answer the following.*

i. Construct a divided difference table for the data

x :	–1	1	4	6
u_x :	1	–3	21	127

write the value $\Delta^2_{4,6} u_1$. Form another table using the order u_4, u_1, u_6, u_{-1} and read off the value of $\Delta^2_{1,6} u_4$.

ii. Show by applying Newton's formula a polynomial approximation to the data of the above example is $x^3 - 5x + 3$.

iii. Show that the nth divided difference of $f(x) = 1/x$ is $(-1)^n / (x_0, x_1, x_2, \ldots, x_n)$.

iv. Use the Newton's divided difference formula to calculate $f(2)$, $f(8)$ and $f(15)$ from the following table :

x :	4	5	7	10	11	13
$f(x)$:	48	100	294	900	1210	2028

[$f(2) = 4, f(8) = 448, f(15) = 3150$]

v. Use Newton's divided difference formula to calculate $f(3)$ from the following table :

x :	0	1	2	4	5	6
$f(x)$:	1	14	15	5	6	10

[$f(3) = 10$]

vi. Find a polynomial satisfied by :

(–4, 1245), (–1, 33), (0, 5), (2, 9) and (5, 1335).

$\left[3x^4 - 5x^3 + 6x^2 - 14x + 5\right]$

vii. Given $\log_{10} 654 = 2.8156$, $\log_{10} 658 = 2.8182$, $\log_{10} 659 = 2.8188$, $\log_{10} 661 = 2.8202$. Find $\log_{10} 656$.

[2.8169]

viii. The observed values of a function are respectively 168, 120, 72 and 63 at the four positions 3, 7, 9 and 10 of independent variable. What is the best estimate you can give for the value of the function at the position 6 of the independent variable?

[147]

ix. Given $y_0 = 2$, $y_1 = 3$, $y_5 = 147$ and $y_x = 12$, find x.

[$x = 2$]

2.8 ANSWERS TO SAQ'S

SAQ 1 By definition $\underset{y}{\Delta} f(x) = \dfrac{f(y) - f(x)}{y - x}$

$$\therefore \underset{y}{\Delta} x^2 = \frac{y^2 - x^2}{y - x} = y + x.$$

$$\Delta^2_{yz} x^2 = \frac{1}{z-x}\left[\Delta_z y^2 - \Delta_y x^2\right]$$

$$= \frac{1}{z-x}\left[y + z - (x + y)\right] = \frac{z-x}{z-x} = 1.$$

SAQ 2 : The relation between divided differences and ordinary differences can be found by starting with a set of functional values corresponding to equidistant values of the argument. Let us construct a table of ordinary differences and a table of divided differences of these functional values, and then comparing differences of the same order in the two tables.

Ordinary difference table :

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0				
		$y_1 - y_0$			
$x_0 + h$	y_1		$y_2 - 2y_1 + y_0$		
		$y_2 - y_1$		$y_3 - 3y_2 + 3y_1 - y_0$	
$x_0 + 2h$	y_2		$y_3 - 2y_2 + y_1$		$y_4 - 4y_3 + 6y_2 - 4y_1 + y_0$
		$y_3 - y_2$		$y_4 - 3y_3 + 3y_2 - y_2$	
$x_0 + 3h$	y_3		$y_4 - 2y_3 + y_2$		
		$y_4 - y_3$			
$x_0 + 4h$	y_4				

Divided difference table :

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0				
		$\frac{y_1 - y_0}{h}$			
$x_0 + h$	y_1		$\frac{y_2 - 2y_1 + y_0}{2h \,.\, h}$		
		$\frac{y_2 - y_1}{h}$		$\frac{y_3 - 3y_2 + 3y_1 - y_0}{3h \,.\, 2h \,.\, h}$	
$x_0 + 2h$	y_2		$\frac{y_3 - 2y_2 + y_1}{2h \,.\, h}$		$\frac{y_4 - 4y_3 + 6y_2 - 4y_1 + y_0}{4h \,.\, 3h \,.\, 2h \,.\, h}$
		$\frac{y_3 - y_2}{h}$		$\frac{y_4 - 3y_3 + 3y_2 - 3y_1}{3h \,.\, 2h \,.\, h}$	
$x_0 + 3h$	y_3		$\frac{y_4 - 2y_3 + y_2}{2h \,.\, h}$		
		$\frac{y_4 - y_3}{h}$			
$x_0 + 4h$	y_4				

By comparision from the tables, we see that

$$\Delta^3_{x_0+h,\, x_0+2h,\, x_0+3h} y_{x_0} = \frac{y_3 - 3y_2 + 3y_1 - y_0}{3!\, h^3} = \frac{\Delta^3 y_0}{3!\, h^3}$$

In general we have

$$\Delta^n_{x_k+h,\, x_k+2h,\ldots,x_k+nh} y_{x_k} = \frac{\Delta^n y_k}{n!\, h^n} \text{ where } k = 0, 1, 2 \ldots$$

SAQ 3 : In unit 1, we have seen that the nth order ordinary differences of a polynomial of nth degree are constant. From SAQ 2, the nth order divided difference is equal to the nth order ordinary difference divided by the constant product $n!\ h^n$, it follows that the nth divided differences of a polynomial of the nth degree are constant. Hence the $n+1$th and higher order divided differences of a polynomial of nth degree are zero.

SAQ 4 : If the arguments are equally spaced, $x_s = x_0 + sh$ and $x = x_0 + hu$ for $s = 1, 2, \ldots n$. Then from equation (3)

$$y_x = \frac{(-1)^n\ (u-1)\ (u-2)\ldots\ (u-n)}{n!} y_0 + \frac{(-1)^{n-1}\ (u-0)\ (u-2)\ldots(u-n)}{n!}\ y_1$$

$$+ \ldots + \frac{u\ (u-1)\ \ldots\ \overline{(u-n-1)}}{n!}\ y_n.$$

SAQ 5 : $\Pi_n(x) = (x - x_0)\ (x - x_1) \ldots (x - x_{i-1})(x - x_i)(x - x_{i+1}) \ldots (x - x_n)$

then $\Pi'_n(x_i) = \frac{d}{dx}\left[\Pi_n(x)\right]_{x=x_i}$

$$= (x_i - x_0)(x_i - x_1) \ldots (x_i - x_{i-1})\ (x - x_{i+1}) \ldots (x_i - x_n)$$

The denominator of (3) can be writte. as $(x - x_i)\ \Pi'_n(x_i)$ and so (3) can be written as $\sum_{i=0}^{n} \frac{\Pi_n(x)}{(x - x_i)\ \Pi'_n(x_i)} \cdot y_i.$

UNIT–3 : CENTRAL DIFFERENCES

Contents

3.1 AIMS AND OBJECTIVES

By the time you complete this unit, you will be able to (i) Establish various relations between central difference operator with other operators, (ii) Derive Gauss's, Stirling's Bessel's and Everett's interpolation formulae, (iii) Discuss the situations in which these formulae are appropriate to apply.

3.2 INTRODUCTION

Newton's forward and backward difference formulae, though fundamental become well suited when applied for interpolation near the beginning and end respectively of a table of differences. For interpolation near the middle of the difference table, these formulae do not converge so rapidly as central difference ones. In other words the central difference formulae are particularly suited for interpolation near the middle of the difference table.

In this unit we study the central difference formulae, particularly due to Gauss, Stirling, Bessel and Everett. Sheppard rule has been explained to derive these formulae.

3.3 CENTRAL DIFFERENCE OPERATOR

We write u_x to mean u as a function of x. Suppose for $x = a - 2h, a - h, a, a + h, a + 2h$, u is tabulated. Make the transformation $X = (x - a)/h$. Then the tabulated values of u are same as $u_{-2}, u_{-1}, u_0, u_1, u_2$ corresponding to $X = -2, -1, 0, 1, 2$. Writing u_X for u_x etc., we form the following difference table.

Table 3.1

x	X	u_X	Δu_X	$\Delta^2 u_X$	$\Delta^3 u_X$	$\Delta^4 u_X$
$a-2h$	-2	u_{-2}				
			Δu_{-2}			
$a-h$	-1	u_{-1}		$\Delta^2 u_{-2}$		
			Δu_{-1}		$\Delta^3 u_{-2}$	
a	0	u_0		$\Delta^2 u_{-1}$		$\Delta^4 u_{-2}$
			Δu_0		$\Delta^3 u_{-1}$	
$a+h$	1	u_1		$\Delta^2 u_0$		
			Δu_1			
$a+2h$	2	u_2				

We introduce the central difference operator δ as

$$\delta = E^{1/2} - E^{-1/2}$$

Then $\delta E^{1/2} = E - 1 = \Delta$

This permits us to write the first differences of table 3.1. in 'δ' notation.

Thus $\Delta u_{-2} = \delta E^{1/2} u_{-2} = \delta u_{-3/2}$,

$\Delta u_{-1} = \delta E^{1/2} u_{-1} = \delta u_{-1/2}$,

$\Delta u_0 = \delta E^{1/2} u_0 = \delta u_{1/2}$,

$\Delta u_1 = \delta E^{1/2} u_1 = \delta u_{3/2}$.

(Here we have used $E^n u_x = u_{x+nh}$ and $h = 1$).

Since $\delta^2 E = \Delta^2$

the second differences of table 3.1 follow as

$\Delta^2 u_{-2} = \delta^2 E u_{-2} = \delta^2 u_{-1}$,

$\Delta^2 u_{-1} = \delta^2 E u_{-1} = \delta^2 u_0$,

$\Delta^2 u_0 = \delta^2 E u_0 = \delta^2 u_1$

Similarly from $\delta^3 E^{3/2} = \Delta^3$, and $\delta^4 E^2 = \Delta^4$ we get

$\Delta^3 u_{-2} = \delta^3 u_{-1/2}$, $\Delta^3 u_{-1} = \delta^3 u_{1/2}$, $\Delta^4 u_{-2} = \delta^4 u_0$.

Let us rewrite table 3.1 in 'δ' notation as

Table 3.2

x	u_x	δu_x	$\delta^2 u_x$	$\delta^3 u_x$	$\delta^4 u_x$
-2	u_{-2}				
		$\delta u_{-3/2}$			
-1	u_{-1}		$\delta^2 u_{-1}$		
		$\delta u_{-1/2}$		$\delta^3 u_{-1/2}$	
0	u_0		$\delta^2 u_0$		$\delta^4 u_0$
		$\delta u_{1/2}$		$\delta^3 u_{1/2}$	
1	u_1		$\delta^2 u_1$		
		$\delta u_{3/2}$			
2	u_2				

One might have observed from table 3.2 that $\delta^r u_a$ appears on the line exactly opposite u_a. This is the chief advantage of 'δ' notation.

We define the average operator μ as

$$\mu = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)$$

Since $\mu u_0 = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right) u_0 = \frac{1}{2}\left(u_{1/2} + u_{-1/2}\right)$

We regard 'μ' as an averaging operator.

Two important relations which exist among the operators Δ, E, δ and μ are

$$\mu^2 = \frac{1}{4}\left(E + E^{-1} + 2\right)$$

$$= \frac{1}{4}\left[\left(E^{1/2} - E^{-1/2}\right)^2 + 4\right] = 1 + \frac{\delta^2}{4}$$

$$\text{and } \mu\delta = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)\left(E^{1/2} - E^{-1/2}\right)$$

$$= \frac{1}{2}\left(E - E^{-1}\right)$$

$$= \frac{1}{2}\left(1 + \Delta - \frac{1}{1+\Delta}\right)$$

$$= \frac{1}{2}\Delta\left(\frac{\Delta + 2}{1+\Delta}\right) = \frac{\Delta}{2}\left(\frac{1}{1+\Delta} + 1\right)$$

$$= \frac{1}{2}\left(E^{-1} + 1\right) = \frac{1}{2}\Delta E^{-1} + \frac{1}{2}\Delta.$$

Assuming the operators μ and δ commute with each other i.e., $\mu\delta = \delta\mu$, we have

$$\mu\,\delta u_0 = \delta\mu u_0 = \delta\frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)u_0$$

$$= \frac{1}{2}\left(\delta u_{1/2} + \delta u_{-1/2}\right).$$

$$\text{Also } \mu\delta u_0 = \frac{1}{2}\left(\Delta E^{-1} + \Delta\right) u_0$$

$$= \frac{1}{2}\left(\Delta u_{-1} + \Delta u_0\right)$$

$$\text{Thus } \mu\delta u_0 = \frac{1}{2}\left(\delta u_{1/2} + \delta u_{-1/2}\right)$$

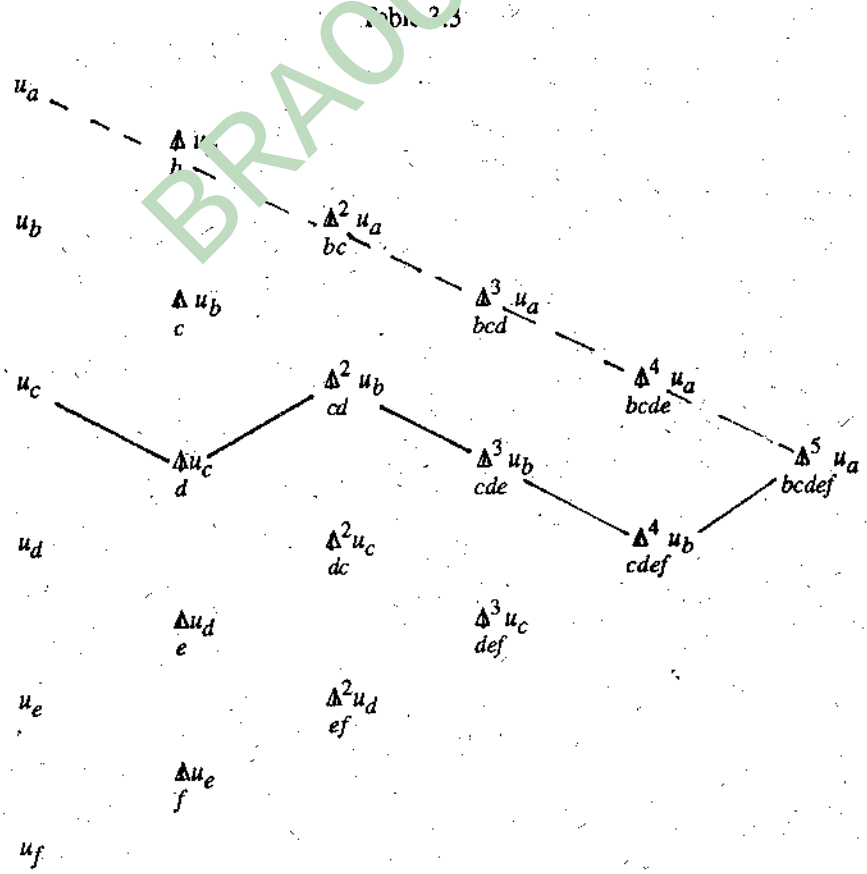
$$= \frac{1}{2}\left(\Delta u_{-1} + \Delta u_0\right)$$

SAQ 1 : Prove that $\delta\{x\} = h$ (the interval of differencing)

3.3.1 Sheppard's rule

Suppose u_x is tabulated for different values of x but not necessarily equally spaced. Let them be $u_a, u_b, u_c, u_d, u_e, u_f$ for $x = a, b, c, d, e, f$ respectively. Then the table of their divided differences is given in 3.3 on the lines given below.

If we wish to write u_x about any one of these x, say $x = c$, we connect u_c to the highest difference $\Delta^5_{bcdef} u_a$ in table 3.3. The line joing these two (solid line in the table) should necessarily pass through the first, second, third and fourth divided differences and in the order.



Also the line should move from u_c to its first adjacent difference or from a difference to its adjacent next higher order difference only. In so moving we have only two paths from one point to the other. We can choose one of them. Along the solid line c, d, b, e, f, a appear in order.

To obtain u_x we now state Sheppard's rule with reference to the solid line. Multiply each of

u_c, $\underset{d}{\Delta} u_c$, $\underset{cd}{\Delta^2} u_b$, $\underset{cde}{\Delta^3} u_b$, $\underset{cdef}{\Delta^4} u_b$, $\underset{bcdef}{\Delta^5} u_a$ in succession by each of 1, C, CD, CDB, CDBE, CDBEF respectively and add. Thus,

$$u_x = u_c + C \underset{d}{\Delta} u_c + CD \underset{cd}{\Delta^2} u_b + CDB \underset{cde}{\Delta^3} u_b$$

$$+ CDBE \underset{cdef}{\Delta^4} u_b + CDBEF \underset{bcdef}{\Delta^5} u_a$$

Note that capital letters appear in the same order as the small ones and

$A = x - a$, $B = x - b$, $C = x - c$, $D = x - d$, $E = x - e$, $F = x - f$.

Also the Newton's divided difference formula corresponds to the dotted line in table 3.3, that is, to the successive divided differences of u_a and the coefficient order is A, B, C, D, E. In other words, we have for Newton's divided difference formula

$$u_x = u_a + A \underset{b}{\Delta} u_a + AB \underset{bc}{\Delta^2} u_a + ABC \underset{bcd}{\Delta^3} u_a$$

$$+ ABCD \underset{bcde}{\Delta^4} u_a + ABCDE \underset{bcdef}{\Delta^5} u_a$$

In fact as many great formulae as possible can be de[illegible]ved from table 3.3 and all of them can be rigorously established analogous to Newton's divided difference formula.

Examples

Ex. 1 : Form a difference table, given $u_{-2} = 15$, $u_{-1} = 12$, $u_0 = 5$, $u_1 = 0$, $u_2 = 3$. Identify $\delta u_{1/2}$, $\delta^2 u_0$, $\delta^3 u_{-1/2}$ in the table and give the other names for these differences in terms of Δ and ∇.

x	u_x	δu_x	$\delta^2 u_x$	$\delta^3 u_x$	$\delta^4 u_x$
–2	15				
		–3			
–1	12		–4		
		–7		6	
0	5		2		0
		–5		6	
1	0		8		
		3			
2	3				

Comparing the above difference table with table 3.2, we identify

$\delta u_{1/2} = -5$, $\delta^2 u_0 = 2$, and $\delta^3 u_{-1/2} = 6$.

Let us form the backward difference table as below

x	u_x	∇u_x	$\nabla^2 u_x$	$\nabla^3 u_x$	$\nabla^4 u_x$
-2	u_{-2}				
		∇u_{-1}			
-1	u_{-1}		$\nabla^2 u_0$		
		∇u_0		$\nabla^3 u_1$	
0	u_0		$\nabla^2 u_1$		$\nabla^4 u_2$
		∇u_1		$\nabla^3 u_2$	
1	u_1		$\nabla^2 u_2$		
		∇u_2			
2	u_2				

Comparing tables 3.1, 3.2 and the above backward difference table, we have

$$\Delta u_0 = \nabla u_1 = \delta u_{1/2},$$

$$\Delta^2 u_{-1} = \nabla^2 u_1 = \delta^2 u_0,$$

$$\Delta^3 u_{-2} = \nabla^3 u_1 = \delta^3 u_{-1/2}$$

SAQ 2 : Prove the above relations using the definitions of the operators.

Ex. 2 : Two operators θ and ϕ commute if $\theta\phi = \phi\theta$. Check that μ, δ, E, Δ and ∇ commute with one another.

The two operators μ, δ commute with each other if we can verify $\mu\,\delta\,u_x = \delta\,\mu\,u_x$.

$$\mu\,\delta\,u_x = \frac{1}{2}\left(E^{1/2} - E^{-1/2}\right)u_x = \mu\left(u_{x+h/2} - u_{x-h/2}\right)$$

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

$$= \frac{1}{2}\left(u_{x+h} + u_x - u_x - u_{x-h}\right)$$

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

$$\delta\mu\,u_x = \delta\,\frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)u_x$$

$$= \frac{1}{2}\,\delta\left(u_{x+h/2} + u_{x-h/2}\right)$$

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

$$= \frac{1}{2}\left(u_{x+h} + u_x - u_x - u_{x-h}\right)$$

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

$\therefore \mu\delta u_x = \delta\mu u_x$ i.e., $\mu\delta = \delta\mu$.

(Note that in verifying the above we have assumed the obvious result that an operator and a constant commute with each other).

$$\mu E u_x = \mu u_{x+h} = \frac{1}{2}\left(u_{x+3h/2} + u_{x+h/2}\right)$$

$$E\mu\, u_x = \frac{1}{2} E\left(u_{x+h/2} + u_{x-h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+3h/2} + u_{x+h/2}\right)$$

$\therefore \mu E u_x = E\mu u_x$ i.e., $\mu E = E\mu$.

$$\mu\,\Delta\, u_x = \mu\left(u_{x+h} - u_x\right)$$

$$= \frac{1}{2}\left(u_{x+3h/2} - u_{x+h/2} + u_{x+h/2} - u_{x-h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+3h/2} - u_{x-h/2}\right)$$

$$\Delta\,\mu\, u_x = \frac{1}{2}\Delta\left(u_{x+h/2} + u_{x-h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+3h/2} - u_{x-h/2}\right) + \frac{1}{2}\left(u_{x+h/2} - u_{x-h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+3h/2} - u_{x-h/2}\right)$$

$\therefore \mu\,\Delta u_x = \Delta\mu u_x$ i.e., $\mu\Delta = \Delta\mu$.

$$\mu\nabla\, u_x = \mu\left(u_x - u_{x-h}\right)$$

$$= \frac{1}{2}\left(u_{x+h/2} + u_{x-h/2} - u_{x-h/2} - u_{x-3h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+h/2} - u_{x-3h/2}\right)$$

$$\nabla\mu\, u_x = \frac{1}{2}\nabla\left(u_{x+h/2} + u_{x-h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+h/2} - u_{x-h/2} + u_{x-h/2} - u_{x-3h/2}\right)$$

$$= \frac{1}{2}\left(u_{x+h/2} - u_{x-3h/2}\right)$$

$\therefore \mu\,\nabla u_x = \nabla\mu u_x$ i.e., $\mu\nabla = \nabla\mu$.

$$\delta E u_x = \delta u_{x+h} = u_{x+3h/2} - u_{x+h/2}$$

$$E\delta u_x = E\left(u_{x+h/2} - u_{x-h/2}\right) = u_{x+3h/2} - u_{x+h/2}$$

$\therefore \delta E u_x = E\delta u_x$ i.e., $\delta E = E\delta$.

$$\delta\,\Delta\, u_x = \delta\left(u_{x+h} - u_x\right) = u_{x+3h/2} - 2u_{x+h/2} + u_{x-h/2}$$

$$\Delta\,\delta\, u_x = \Delta\left(u_{x+h/2} + u_{x-h/2}\right)$$

$$= u_{x+3h/2} - 2u_{x+h/2} + u_{x-h/2}$$

$\therefore \delta\,\Delta\, u_x = \Delta\,\delta\, u_x$ i.e., $\delta\Delta = \Delta\delta$.

$$\delta \nabla u_x = \delta\left(u_x - u_{x-h}\right) = u_{x+h/2} - 2u_{x-h/2} + u_{x-3h/2}$$

$$\nabla \delta u_x = \nabla\left(u_{x+h/2} - u_{x-h/2}\right)$$

$$= u_{x+h/2} - 2u_{x-h/2} + u_{x-3h/2}$$

$\therefore \delta \nabla u_x = \nabla \delta u_x$ i.e., $\delta\nabla = \nabla\delta$.

$$E \Delta u_x = E\left(u_{x+h} - u_x\right) = u_{x+3h/2} - 2u_{x+h/2} + u_{x-h/2}$$

$$\Delta E u_x = \Delta\left(u_{x+h/2} - u_{x-h/2}\right)$$

$$= u_{x+3h/2} - 2u_{x+h/2} + u_{x-h/2}$$

$\therefore E\Delta u_x = \Delta E u_x$ i.e., $E\Delta = \Delta E$.

$$E \nabla u_x = E\left(u_x - u_{x-h}\right) = u_{x+h/2} - 2u_{x-h/2} + u_{x-3h/2}$$

$$\nabla E u_x = \nabla\left(u_{x+h/2} - u_{x-h/2}\right)$$

$$= u_{x+h/2} - 2u_{x-h/2} + u_{x-3h/2}$$

$\therefore E \nabla u_x = \nabla E u_x$ i.e., $E\nabla = \nabla E$.

$$\Delta \nabla u_x = \Delta\left(u_x - u_{x-h}\right) = u_{x+h} - 2u_x + u_{x-h}$$

$$\nabla \Delta u_x = \nabla\left(u_{x+h} - u_x\right) = u_{x+h} - 2u_x + u_{x-h}$$

$\therefore \Delta \nabla u_x = \nabla \Delta u_x$ i.e., $\Delta\nabla = \nabla\Delta$.

Hence all the given operators commute with one another.

Ex. 3 : Prove the following identities

(a) $\sqrt{1 + \delta^2 \mu^2} = 1 + \delta^2/2$

(b) $E^{1/2} = \mu + \delta/2$

(c) $\nabla = \delta E^{-1/2}$

(d) $E^{-1/2} = \mu - \delta/2$

(a) Since $\delta\mu = \frac{1}{2}\left(E - E^{-1}\right)$

we have,

$$1 + \delta^2 \mu^2 = 1 + \frac{1}{4}\left(E^2 + E^{-2} - 2\right) = \frac{E^2 + E^{-2} + 2}{4}$$

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

$$\therefore \sqrt{1 + \delta^2 \mu^2} = \frac{1}{2}\left(E + E^{-1}\right).$$

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

$$\therefore \sqrt{1 + \delta^2 \mu^2} = 1 + \frac{\delta^2}{2}$$

(b) $\mu + \delta/2 = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right) + \frac{1}{2}\left(E^{1/2} - E^{-1/2}\right) = E^{1/2}$

$\therefore E^{1/2} = \mu + \delta/2.$

(c) $\delta E^{-1/2} = \left(E^{1/2} - E^{-1/2}\right) E^{-1/2} = 1 - E^{-1}$

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

we have $1 - E^{-1} = \nabla$

$\therefore \delta E^{-1/2} = \nabla$ or $\nabla = \delta E^{-1/2}$.

(d) $\mu - \delta/2 = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right) - \frac{1}{2}\left(E^{1/2} - E^{-1/2}\right) = E^{-1/2}$

$\therefore E^{-1/2} = \mu - \delta/2.$

Ex. 5 : The derivative operator D is defined by $Du_x = \frac{du_x}{dx}$. Prove that D commutes with δ, E, Δ, μ, ∇.

We write $Du_x = u'_x$ to denote $\frac{du_x}{dx}$.

$D\delta u_x = D\left(u_{x+h/2} - u_{x-h/2}\right) = u'_{x+h/2} - u'_{x-h/2}$

$\delta D u_x = \delta u'_x = u'_{x+h/2} - u'_{x-h/2}$

$\therefore D\delta u_x = \delta D u_x$ i.e., $D\delta = \delta D$.

$DE u_x = D u_{x+h} = u'_{x+h}$

$ED u_x = E u'_x = u'_{x+h}$

$\therefore DE u_x = ED u_x$ i.e., $DE = ED$.

$D\Delta u_x = D\left(u_{x+h} - u_x\right) = u'_{x+h} - u'_x$

$\Delta D u_x = D u'_x = u'_{x+h} - u'_x$

$\therefore D\Delta u_x = \Delta D u_x$ i.e., $D\Delta = \Delta D$.

$D\mu u_x = D \cdot \frac{1}{2}\left(u_{x+h/2} + u_{x-h/2}\right)$

$= \frac{1}{2}\left(u'_{x+h/2} + u'_{x-h/2}\right)$

$\mu D u_x = \mu u'_x = \frac{1}{2}\left(u'_{x+h/2} + u'_{x-h/2}\right)$

$\therefore D\mu u_x = \mu D u_x$ i.e., $D\mu = \mu D$.

$D\nabla u_x = D\left(u_x - u_{x-h}\right) = u'_x - u'_{x-h}$

$\nabla D u_x = \nabla u'_x = u'_x - u'_{x-h}$

$\therefore D\nabla u_x = \nabla D u_x$ i.e., $D\nabla = \nabla D$.

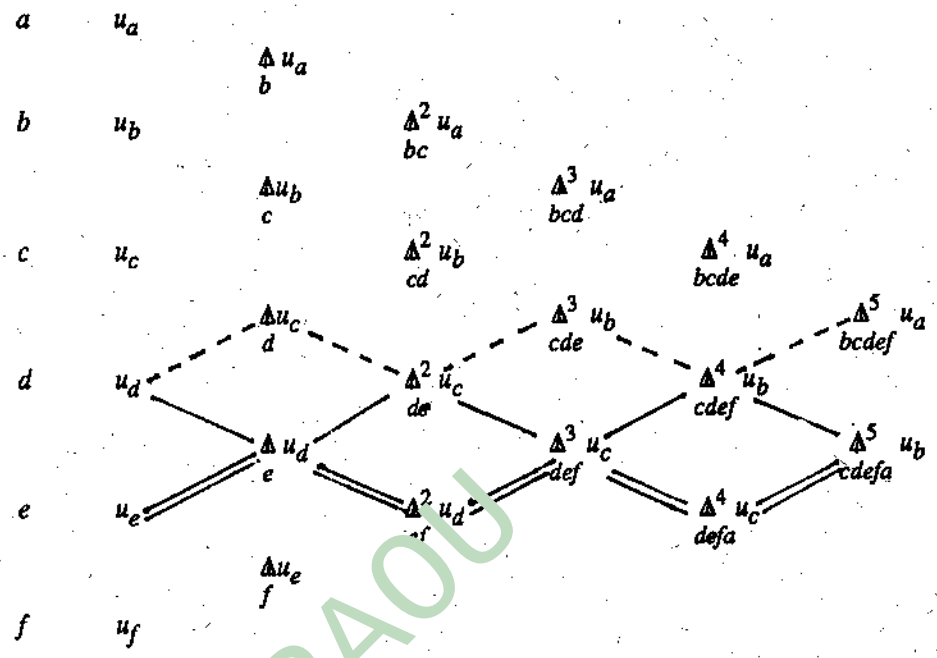
Hence D commutes with δ, E, Δ, μ, ∇.

3.4 GAUSS FORMULA

3.4.1 Gauss forward formula

Suppose u_x is tabulated for six different (not necessarily equally spaced arguments) say, $x = a, b, c, d, e, f$. We then have table 3.4 as their divided difference table.

Table 3.4



If, however, a, b, c, d, e, f are equidistant with 'h' as the inverval of differencing, then along the dotted and solid paths above, we have

$$\underset{d}{\Delta} u_c = \Delta u_c = \frac{1}{h\ 1!}\ \Delta u_c,$$

$$\underset{de}{\Delta^2} u_c = \Delta^2 u_c = \frac{1}{h^2\ 2!}\ \Delta^2 u_c,$$

$$\underset{cde}{\Delta^3} u_b = \Delta^3 u_b = \frac{1}{h^3\ 3!}\ \Delta^3 u_b,$$

$$\underset{cdef}{\Delta^4} u_b = \Delta^4 u_b = \frac{1}{h^4\ 4!}\ \Delta^4 u_b,$$

$$\underset{bcdef}{\Delta^5} u_a = \Delta^5 u_a = \frac{1}{h^5\ 5!}\ \Delta^5 u_a,$$

$$\underset{e}{\Delta} u_d = \Delta u_d = \frac{1}{h\ 1!}\ \Delta u_d,$$

$$\underset{def}{\Delta^3} u_c = \Delta^3 u_c = \frac{1}{h^3\ 3!}\ \Delta^3 u_c,$$

$$\mathbb{\Delta}^5_{cdefa} u_b = \mathbb{\Delta}^5 u_b = \frac{1}{h^5\, 5!} \Delta^5 u_b,$$

and in general $\mathbb{\Delta}^r u_x = \frac{1}{h^r\, r!} \Delta^r u_x$.

Choose $a = -3, b = -2, c = -1, d = 0, e = 1, f = 2$ so that

A = $x + 3$, B = $x + 2$, C = $x + 1$, D = x, E = $x - 1$, F = $x - 2$.

Applying Sheppard's rule along the solid line, we get since $h = 1$,

$$u_x = u_0 + \frac{x}{1!} \Delta u_0 + \frac{x\,(x-1)}{2!} \Delta^2 u_{-1} + \frac{x\,(x-1)\,(x+1)}{3!} \Delta^3 u_{-1}$$

$$+ \frac{x\,(x-1)\,(x+1)\,(x-2)}{4!} \Delta^4 u_{-2}$$

$$+ \frac{x\,(x-1)\,(x+1)\,(x-2)\,(x+2)}{5!} \Delta^5 u_{-2} + \ldots$$

or $$u_x = u_0 + \binom{x}{1} \Delta u_0 + \binom{x}{2} \Delta^2 u_{-1} + \binom{x+1}{3} \Delta^3 u_{-1} + \binom{x+1}{4} \Delta^4 u_{-2}$$

$$+ \binom{x+2}{5} \Delta^5 u_{-2} + \ldots$$

known as Gauss forward formula.

Aliter : From table 3.5 given below

Table 3.5

x	u_x	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$	$\Delta^4 u_x$	$\Delta^5 u_x$
–2	u_{-2}					
		Δu_{-2}				
–1	u_{-1}		$\Delta^2 u_{-2}$			
		Δu_{-1}		$\Delta^3 u_{-2}$		
0	u_0		$\Delta^2 u_{-1}$		$\Delta^4 u_{-2}$	
		Δu_0		$\Delta^3 u_{-1}$		$\Delta^5 u_{-2}$
1	u_1		$\Delta^2 u_0$		$\Delta^4 u_{-1}$	
		Δu_1		$\Delta^3 u_0$		$\Delta^5 u_{-1}$
2	u_2		$\Delta^2 u_1$		$\Delta^4 u_0$	
		Δu_2		$\Delta^3 u_1$		
3	u_3		$\Delta^2 u_2$			
		Δu_3				
4	u_4					

we infer

$$\Delta^3 u_{-1} = \Delta^2 u_0 - \Delta^2 u_{-1},$$

$$\Delta^4 u_{-1} = \Delta^3 u_0 - \Delta^3 u_{-1},$$

$$\Delta^5 u_{-1} = \Delta^4 u_0 - \Delta^4 u_{-1},$$

$$\Delta^5 u_{-2} = \Delta^4 u_{-1} - \Delta^4 u_{-2}.$$

These give us

$$\Delta^2 u_0 = \Delta^2 u_{-1} + \Delta^3 u_{-1},$$

$$\Delta^3 u_0 = \Delta^3 u_{-1} + \Delta^4 u_{-1}$$

$$\Delta^4 u_0 = \Delta^4 u_{-1} + \Delta^5 u_{-1}$$

$$\Delta^4 u_{-1} = \Delta^4 u_{-2} + \Delta^5 u_{-2}.$$

Substituting the above in Newton's forward formula

$$u_x = u_0 + \binom{x}{1}\Delta u_0 + \binom{x}{2}\Delta^2 u_0 + \binom{x}{3}\Delta^3 u_0 + \binom{x}{4}\Delta^4 u_0 + \dots$$

we get

$$u_x = u_0 + \binom{x}{1}\Delta u_0 + \binom{x}{2}\left(\Delta^2 u_{-1} + \Delta^3 u_{-1}\right)$$

$$+ \binom{x}{3}\left(\Delta^3 u_{-1} + \Delta^4 u_{-2} + \Delta^5 u_{-2}\right)$$

$$+ \binom{x}{4}\left(\Delta^4 u_{-2} + \Delta^5 u_{-2} + \Delta^5 u_{-1}\right) + \dots$$

Using,

$$\binom{x}{r} + \binom{x}{r-1} = \binom{x+1}{r}$$

we get the Gauss forward formula

$$u_x = u_0 + \binom{x}{1}\Delta u_0 + \binom{x}{2}\Delta^2 u_{-1} + \binom{x+1}{3}\Delta^3 u_{-1} + \binom{x+1}{4}\Delta^4 u_{-2} + \dots$$

3.4.2 Gauss Backward formula

To derive Gauss backward formula, we apply Sheppard's rule along the dotted line. Thus

$$u_x = u_0 + \binom{x}{1}\Delta u_{-1} + \binom{x+1}{2}\Delta^2 u_{-1} + \binom{x+1}{3}\Delta^3 u_{-2}$$

$$+ \binom{x+2}{4}\Delta^4 u_{-2} + \binom{x+2}{5}\Delta^5 u_{-3} + \dots$$

Aliter : From table 3.5 we have

$$\Delta u_0 - \Delta u_{-1} = \Delta^2 u_{-1}$$

$$\Delta^3 u_{-1} - \Delta^3 u_{-2} = \Delta^4 u_{-2}$$

$$\Delta^2 u_0 - \Delta^2 u_{-1} = \Delta^3 u_{-1},$$

$$\Delta^3 u_0 - \Delta^3 u_{-1} = \Delta^4 u_{-1} \text{ etc.}$$

These give us

$$\Delta u_0 = \Delta u_{-1} + \Delta^2 u_{-1}$$

$$\Delta^3 u_{-1} = \Delta^3 u_{-2} + \Delta^4 u_{-2}$$

$$\Delta^2 u_0 = \Delta^2 u_{-1} + \Delta^3 u_{-1},$$

$$\Delta^3 u_0 = \Delta^3 u_{-1} + \Delta^4 u_{-1} \text{ and so on.}$$

Substituting the above in Newton's forward formula, we get

$$u_x = u_0 + \binom{x}{1}\left(\Delta u_{-1} + \Delta^2 u_{-1}\right) + \binom{x}{2}\left(\Delta^2 u_{-1} + \Delta^3 u_{-2} + \Delta^4 u_{-2}\right)$$

$$+ \binom{x}{3}\left(\Delta^3 u_{-2} + \Delta^4 u_{-2} + \Delta^4 u_{-1}\right) + \ldots$$

Using the relation

$$\binom{x}{n} + \binom{x}{n-1} = \binom{x+1}{n}$$

we get Gauss backward formula

$$u_x = u_0 + \binom{x}{1}\Delta u_{-1} + \binom{x+1}{2}\Delta^2 u_{-1} + \binom{x+1}{3}\Delta^3 u_{-2} + \ldots$$

3.4.3 Gauss Third Formula

Applying Sheppard's rule along the double solid line, we get

$$u_x = u_1 + (x-1)\,\Delta u_0 + x(x-1)\frac{\Delta^2 u_0}{2} + x(x-1)(x-2)\frac{\Delta^3 u_{-1}}{3!}$$

$$+ x\left(x^2-1\right)(x-2)\frac{\Delta^4 u_{-1}}{4!} + x\left(x^2-1\right)(x-2)(x-3)\frac{\Delta^5 u_{-2}}{5!} + \ldots$$

$$= u_1 + \binom{x-1}{1}\Delta u_0 + \binom{x}{2}\Delta^2 u_0 + \binom{x}{3}\Delta^3 u_{-1}$$

$$+ \binom{x+1}{4}\Delta^4 u_{-1} + \binom{x+1}{5}\Delta^5 u_{-2} + \ldots$$

This formula known as Gauss third formula is also obtained from Gauss backward formula by advancing all its subscripts by one unit and replacing x by $(x-1)$.

3.5 STIRLING'S FORMULA

Stirling's formula is now obtained by taking the average of Gauss forward and backward formulas.

Thus $$u_x = u_0 + \frac{x\left(\Delta u_{-1} + \Delta u_0\right)}{2} + \frac{x^2}{2}\Delta^2 u_{-1}$$

$$+ \frac{x\left(x^2-1\right)}{3!}\,\frac{\left(\Delta^3 u_{-2} + \Delta^3 u_{-1}\right)}{2} + \frac{x^2\left(x^2-1\right)}{4!}\,\Delta^4 u_{-2}$$

$$+ \frac{x\left(x^2-1\right)\left(x^2-2^2\right)}{5!}\,\frac{\left(\Delta^5 u_{-3} + \Delta^5 u_{-2}\right)}{2} + \ldots$$

Since $$\mu\delta u_0 = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)\Delta E^{-1/2} u_0$$

$$= \frac{1}{2}\Delta\left(E^{-1} + 1\right) u_0 = \frac{1}{2}\left(\Delta u_{-1} + \Delta u_0\right)$$

$$\delta^2 u_0 = \Delta^2 E^{-1} u_0 = \Delta^2 u_{-1},$$

$$\mu\delta^3 u_0 = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)\Delta^3 E^{-3/2} u_0 = \frac{1}{2}\Delta^3\left(E^{-1} + E^{-2}\right) u_0$$

$$= \frac{1}{2}\left(\Delta^3 u_{-2} + \Delta^3 u_{-1}\right),$$

$$\delta^4 u_0 = \Delta^4 E^{-2} u_0 = \Delta^4 u_{-2}$$

$$\mu\delta^5 u_0 = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)\Delta^5 E^{-5/2} u_0 = \frac{1}{2}\Delta^5\left(E^{-3} + E^{-2}\right) u_0$$

$$= \frac{1}{2}\left(\Delta^5 u_{-3} + \Delta^5 u_{-2}\right).$$

We rewrite the Stirling's formula in central difference notation 'δ' as

$$u_x = u_0 + x\mu\,\delta u_0 + \frac{x^2}{2!}\,\delta^2 u_0 + \frac{x\left(x^2-1\right)}{3!}\,\mu\delta^3 u_0$$

$$+ \frac{x^2\left(x^2-1\right)}{4!}\,\delta^4 u_0 + \frac{x\left(x^2-1\right)\left(x^2-2^2\right)}{5!}\,\mu\delta^5 u_0 + \ldots$$

Aliter : Since $$\Delta u_0 = \delta E^{1/2} u_0 = \delta u_{1/2},\ \Delta^2 u_0 = \delta^2 u_1,$$

$$\Delta^3 u_0 = \delta^2 u_{3/2},\ \Delta^4 u_0 = \delta^4 u_2, \ldots$$

the Newton's forward interpolation may be written as

$$u_x = u_0 + x\,\delta u_{1/2} + \frac{x(x-1)}{2!}\,\delta^2 u_1 + \frac{x(x-1)(x-2)}{3!}\,\delta^3 u_{3/2}$$

$$+ \frac{x(x-1)(x-2)(x-3)}{4!}\,\delta^4 u_2 + \ldots$$

Now $\delta u_{1/2} = \delta E^{1/2} u_0 = \delta\left(\mu + \frac{\delta}{2}\right)u_0 = \mu\,\delta u_0 + \frac{1}{2}\delta^2 u_0$

$$\delta^2 u_1 = \delta^2 E u_0 = \delta^2\left(\mu + \frac{\delta}{2}\right)u_0 = \delta^2\left(\mu^2 + \frac{\delta^2}{4} + \mu\delta\right)u_0$$

$$= \delta^2\left(1 + \frac{\delta^2}{4} + \frac{\delta^2}{4}\mu\delta\right)u_0$$

$$= \delta^2\left(1 + \mu\delta + \frac{\delta^2}{2}\right)u_0$$

$$= \delta^2 u_0 + \mu\delta^3 u_0 + \frac{1}{2}\delta^4 u_0 .$$

$$\delta^3 u_{3/2} = \mu\delta^3 u_0 + \frac{3}{2}\delta^4 u_0 + \mu\delta^5 u_0 + \frac{1}{2}\mu\delta^6 u_0,$$

$$\delta^4 u_2 = \delta^4 u_0 + 2\,\mu\delta^5 u_0 + 2\,\delta^6 u_0 + \mu\,\delta^7 u_0 + \frac{1}{2}\delta^8 u_0 \text{ and so on.}$$

Substituting these in the above Newton's interpolation formula and simplyfying we get

$$u_x = u_0 + x\,\mu\delta u_0 + \frac{x^2}{2!}\delta^2 u_0 + \frac{x\left(x^2-1\right)}{3!}\mu\delta^3 u_0 + \frac{x^2\left(x^2-1\right)}{4!}\delta^4 u_0 + \ldots$$

3.6 BESSEL'S FORMULA

In order to derive Bessel's formula let us recall the Gauss third formula i.e.,

$$u_x = u_1 + \binom{x-1}{1}\Delta u_0 + \binom{x}{2}\Delta^2 u_0 + \binom{x}{3}\Delta^3 u_{-1}$$

$$+ \binom{x+1}{4}\Delta^4 u_{-1} + \binom{x+1}{5}\Delta^5 u_{-2} + \ldots$$

We know the Gauss forward formula as given by

$$u_x = u_0 + \binom{x}{1}\Delta u_0 + \binom{x}{2}\Delta^2 u_{-1} + \binom{x+1}{3}\Delta^3 u_{-1}$$

$$+ \binom{x+1}{4}\Delta^4 u_{-2} + \binom{x+2}{5}\Delta^5 u_{-2} + \ldots$$

Taking the average of the above two formulas, we get

$$u_x = \frac{u_0 + u_1}{2} + \left(x - \frac{1}{2}\right)\Delta u_0 + \frac{x(x-1)}{2}\frac{\left(\Delta^2 u_{-1} + \Delta^2 u_0\right)}{2}$$

$$+ \frac{x\left(x-\frac{1}{2}\right)(x-1)}{3!}\Delta^3 u_{-1} + \frac{x\left(x^2-1\right)(x-2)}{4!}$$

$$\frac{\left(\Delta^4 u_{-2} + \Delta^4 u_{-1}\right)}{2} + \frac{x\left(x - \frac{1}{2}\right)\left(x^2 - 1\right)(x - 2)}{5!}\Delta^5 u_{-2} + \ldots$$

which is one form of Bessel's formula.

Since

$$\frac{u_0 + u_1}{2} - \frac{1}{2}\Delta u_0 = \frac{u_0 + u_1}{2} - \frac{1}{2}\left(u_1 - u_0\right) = u_0,$$

the above form can also be written as

$$u_x = u_0 + x\,\Delta u_0 + \frac{x(x-1)}{2}\cdot\left(\frac{\Delta^2 u_{-1} + \Delta^2 u_0}{2}\right)$$

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

$$+ \frac{x\left(x^2 - 1\right)(x-2)}{4!}\cdot\frac{\left(\Delta^4 u_{-2} + \Delta^3 u_{-1}\right)}{2}$$

$$+ \frac{x\left(x - \frac{1}{2}\right)\left(x^2 - 1\right)(x-2)}{5!}\Delta^5 u_{-2} + \ldots$$

If we put $x = \frac{1}{2}$, we get the simple formula

$$u_{1/2} = \frac{u_0 + u_1}{2} - \frac{1}{8}\cdot\frac{\left(\Delta^2 u_{-1} + \Delta^2 u_0\right)}{2} + \frac{3}{128}\cdot\frac{\left(\Delta^4 u_{-2} + \Delta^4 u_{-1}\right)}{2} \ldots$$

This important special case of Bessel's formula is called the formula for *interpolating to halves*. It is used for computing values of the function midway between any two given argument values.

Bessel's formula can be put in a more elegent form by setting $x = z + \frac{1}{2}$, we then obtain

$$u_{z+1/2} = \frac{u_0 + u_1}{2} + z\,\Delta u_0 + \frac{z^2 - 1/4}{2!}\cdot\frac{\left(\Delta^2 u_{-1} + \Delta^2 u_0\right)}{2}$$

$$+ \frac{z\left(z^2 - 1/4\right)}{3!}\Delta^3 u_{-1}$$

$$+ \frac{\left(z^2 - 1/4\right)\left(z^2 - 9/4\right)}{4!}\cdot\frac{\left(\Delta^4 u_{-2} + \Delta^4 u_{-1}\right)}{2}$$

$$+ \frac{z\left(z^2 - 1/4\right)\left(z^2 - 9/4\right)}{5!}\Delta^5 u_{-2} + \ldots$$

Noting $\mu u_{1/2} = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right)u_{1/2} = \frac{1}{2}\left(u_0 + u_1\right)$

$$\Delta u_0 = \delta E^{1/2} u_0 = \delta u_{1/2};$$

$$\frac{\left(\Delta^2 u_{-1} + \Delta^2 u_0\right)}{2} = \frac{\left(\delta^2 u_0 + \delta^2 u_1\right)}{2} = \delta^2 \mu u_{1/2} = \mu\delta^2 u_{1/2};$$

$$\Delta^3 u_{-1} = \delta^3 E^{3/2} u_{-1} = \delta^3 u_{1/2};$$

$$\frac{\left(\Delta^4 u_{-2} + \Delta^4 u_{-1}\right)}{2} = \frac{\left(\delta^4 u_0 + \delta^4 u_1\right)}{2} = \delta^4 \mu u_{1/2} = \mu\delta^4 u_{1/2};$$

$$\Delta^5 u_{-2} = \delta^5 E^{5/2} u_{-2} = \delta^5 u_{1/2};$$

We get from the above,

$$u_{z+1/2} = \mu u_{1/2} + z\,\delta u_{1/2} + \frac{\left(z^2 - 1/4\right)}{2!}\mu\delta^2 u_{1/2}$$

$$+ \frac{z\left(z^2 - 1/4\right)}{3!}\delta^3 u_{1/2} + \frac{\left(z^2 - 1/4\right)\left(z^2 - 9/4\right)}{4!}\mu\delta^4 u_{1/2}$$

$$+ \frac{z\left(z^2 - 1/4\right)\left(z^2 - 9/4\right)}{5!}\delta^5 u_{1/2} + \ldots$$

Putting $z = 0$ gives the important formula in central difference notation, useful for halving the interval of tabulation

$$u_{1/2} = \mu u_{1/2} - \frac{1}{8}\mu\delta^2 u_{1/2} + \frac{3}{128}\mu\delta^4 u_{1/2} - \ldots.$$

We should note that by proper choice of origin, x in any central difference formula may be taken in the range $0 \le x \le 1$ or in the range $-\frac{1}{2} \le x \le \frac{1}{2}$.

Aliter : We give an alternative proof for Bessel's formula. We know

$$\delta^2 u_1 = \delta^2 E^{1/2} u_{1/2} = \delta^2\left(\mu + \frac{\delta}{2}\right)u_{1/2}$$

$$= \mu\delta^2 u_{1/2} + \frac{1}{2}\delta^3 u_{1/2},$$

$$\delta^3 u_{3/2} = \delta^3 E\, u_{1/2} = \delta^3\left(\mu + \frac{\delta}{2}\right)^2 u_{1/2}$$

$$= \delta^3\left(\mu^2 + \mu\delta + \frac{\delta^2}{4}\right)u_{1/2}$$

$$= \delta^3\left(1 + \frac{\delta^2}{4} + \mu\delta + \frac{\delta^2}{4}\right)u_{1/2}$$

$$= \delta^3\left(1 + \mu\delta + \frac{\delta^2}{2}\right)u_{1/2}$$

$$= \delta^3 u_{1/2} + \mu\,\delta^4 u_{1/2} + \frac{1}{2}\delta^5 u_{1/2},$$

$$\delta^4 u_2 = \delta^4 E^{3/2} u_{1/2} = \delta^4 \left(\mu + \frac{\delta}{2}\right)^3 u_{1/2}$$

$$= \delta^4 \left(\mu^3 + 3\mu^2 \frac{\delta}{2} + 3\mu \frac{\delta^2}{4} + \frac{\delta^3}{8}\right) u_{1/2}$$

$$= \delta^4 \left(\mu + \frac{\mu\delta^2}{4} + \frac{3\delta}{2} + \frac{3\delta^3}{8} + \frac{3}{4}\mu\delta^2 + \frac{\delta^3}{3}\right) u_{1/2}$$

$$= \delta^4 \left(\mu + \mu\delta^2 + \frac{3\delta}{2} + \frac{\delta^3}{2}\right) u_{1/2}$$

$$= \mu\delta^4 u_{1/2} + \mu\delta^6 u_{1/2} + \frac{3}{2}\delta^5 u_{1/2} + \frac{1}{2}\delta^7 u_{1/2}.$$

We substitute the above in the Newton's forward interpolation formula expressed in 'δ' notation as

$$u_x = u_0 + x\delta u_{1/2} + \frac{x(x-1)}{2!}\delta^2 u_1 + \frac{x(x-1)(x-2)}{3!}\delta^3 u_{3/2}$$

$$+ \frac{x(x-1)(x-2)(x-3)}{4!}\delta^4 u_2 + \ldots$$

We then obtain

$$u_x = u_0 + x\delta u_{1/2} + \frac{x(x-1)}{2!}\mu\delta^2 u_{1/2} + \frac{x(x-1)(x-1/2)}{3!}\delta^3 u_{1/2}$$

$$+ \frac{x(x-1)(x-2)(x+1)}{4!}\mu\delta^4 u_{1/2} + \ldots$$

Noting $\mu u_{1/2} = \frac{1}{2}\left(E^{1/2} + E^{-1/2}\right) u_{1/2} = \frac{1}{2}\left(u_0 + u_1\right)$

$$= u_0 + \frac{1}{2}\left(u_1 - u_0\right) = u_0 + \frac{1}{2}\delta u_{1/2}$$

and putting $x = z + \frac{1}{2}$ in the above, we get

$$u_{z+1/2} = \mu u_{1/2} + z\,\delta u_{1/2} + \frac{\left(z^2 - 1/4\right)}{2!}\mu\delta^2 u_{1/2}$$

$$+ \frac{z\left(z^2 - 1/4\right)}{3!}\delta^3 u_{1/2} + \frac{\left(z^2 - 1/4\right)\left(z^2 - 9/4\right)}{4!}\mu\delta^4 u_{1/2} + \ldots$$

as Bessel's formula.

3.7 EVERETT FORMULA

Before we prove Everett formula, we prove the following lemma

$$\binom{z}{j}\delta^j u_a + \binom{z+1}{j+1}\delta^{(j+1)} u_{(a+1/2)} = \binom{z+1}{j+1}\delta^j u_{a+1} - \binom{z}{j+1}\delta^j u_a$$

Proof : We know

$$\binom{z}{j} + \binom{z}{j-1} = \binom{z+1}{j}$$

or $$\binom{z}{j-1} = \binom{z+1}{j} - \binom{z}{j}$$

Replacing j by $(j + 1)$, we get

$$\binom{z}{j} = \binom{z+1}{j+1} - \binom{z}{j+1}.$$

Substituting this on the L.H.S. of lemma, we get

$$\binom{z}{j}\delta^j u_a + \binom{z+1}{j+1}\delta^{(j+1)} u_{(a+1/2)}$$

$$= \left(\binom{z+1}{j+1} - \binom{z}{j+1}\right)\delta^j u_a + \binom{z+1}{j+1}\delta^{(j+1)} u_{(a+1/2)}$$

$$= \binom{z+1}{j+1}\delta^j\left(1 + \delta E^{1/2}\right)u_a - \binom{z}{j+1}\delta^j u_a$$

$$= \binom{z+1}{j+1}\delta^j u_{(a+1)} - \binom{z}{j+1}\delta^j u_a$$

$$\left(\because 1 + \delta E^{1/2} = 1 + \Delta = E\right)$$

Let us write the Gauss forward formula in [illegible]ion i.e.,

$$u_x = u_0 + \binom{x}{1}\delta u_{1/2} + \binom{x}{2}\delta^2 u_0 + \binom{x+1}{3}\delta^3 u_{1/2}$$

$$+ \binom{x+1}{4}\delta^4 u_0 + \binom{x+2}{5}\delta^5 u_{1/2} + \ldots$$

By the above lemma,

$$\binom{x-1}{0}u_0 + \binom{x}{1}\delta u_{1/2} = \binom{x}{1}u_1 - \binom{x-1}{1}u_0$$

Since $\binom{x-1}{0} = 1$, we have

$$u_0 + \binom{x}{1}\delta u_{1/2} = \binom{x}{1}u_1 - \binom{x-1}{1}u_0.$$

Similarly

$$\binom{x}{2}\delta^2 u_0 + \binom{x+1}{3}\delta^3 u_{1/2} = \binom{x+1}{3}\delta^2 u_1 - \binom{x}{3}\delta^2 u_0.$$

$$\binom{x+1}{4}\delta^4 u_0 + \binom{x+2}{5}\delta^5 u_{1/2} = \binom{x+2}{5}\delta^4 u_1 - \binom{x+1}{5}\delta^4 u_0.$$

Substituting these in the above Gauss forward furmula, we get

$$u_x = \binom{x}{1} u_1 + \binom{x+1}{3} \delta^2 u_1 + \binom{x+2}{5} \delta^4 u_1 + \ldots$$
$$- \binom{x-1}{1} u_0 - \binom{x}{3} \delta^2 u_0 - \binom{x+1}{5} \delta^4 u_0 - \ldots$$

known as Everrett's first formula.

Putting $x + y = 1$ in the above yields a more symmetric and convenient form

$$u_x = x\left[u_1 + \frac{(x^2-1)}{3!}\delta^2 u_1 + \frac{(x^2-1)(x^2-4)}{5!}\delta^4 u_1 + \ldots\right]$$
$$+ y\left[u_0 + \frac{(y^2-1)}{3!}\delta^2 u_0 + \frac{(y^2-1)(y^2-4)}{5!}\delta^4 u_0 + \ldots\right]$$

known as Laplace-Everett formula.

The Gauss Backward formula in 'δ'notation is given by

$$u_x = u_0 + \binom{x}{1}\delta u_{-1/2} + \binom{x+1}{2}\delta^2 u_0 + \binom{x+1}{3}\delta^3 u_{-1/2}$$
$$+ \binom{x+2}{4}\delta^4 u_0 + \ldots$$

By lemma, we have

$$\binom{x}{1}\delta u_{-1/2} + \binom{x+1}{2}\delta^2 u_0 = \binom{x+1}{2}\delta u_{1/2} - \binom{x}{2}\delta u_{-1/2}$$

$$\binom{x+1}{3}\delta^3 u_{-1/2} + \binom{x+2}{4}\delta^4 u_0 = \binom{x+2}{4}\delta^3 u_{1/2} - \binom{x+1}{4}\delta^3 u_{-1/2}$$

Substituting these in the Gauss Backward formula, we get

$$u_x = u_0 + \binom{x+1}{2}\delta u_{1/2} + \binom{x+2}{4}\delta^3 u_{1/2} + \ldots$$
$$- \binom{x}{2}\delta u_{-1/2} - \binom{x+1}{4}\delta^3 u_{-1/2} - \ldots$$

known as Everett's second formula. Putting

$p = \frac{1}{2} + x$, $q = \frac{1}{2} - x$ gives

$$u_x = u_0 + \left[\frac{(p^2-1/4)}{2!}\delta u_{1/2} + \frac{(p^2-1/4)(p^2-9/4)}{4!}\delta^3 u_{1/2} + \ldots\right]$$
$$- \left[\frac{(q^2-1/4)}{2!}\delta u_{-1/2} + \frac{(q^2-1/4)(q^2-9/4)}{4!}\delta^3 u_{-1/2} + \ldots\right]$$

Examples

Ex. 1 : Use Gauss forward formula to find the values of u when $x = 3.75$ given the following table :

x :	2.5	3.0	3.5	4.0	4.5	5.0
u :	24.145	22.043	20.225	18.644	17.262	16.047

Take 3.5 as the origin and 0.5 as the unit. Define

$$t = \frac{x - 3.5}{.5}$$

Then we require the value of u for which

$$t = \frac{3.75 - 3.5}{.5} = 0.5$$

The difference table is :

t	u	Δu_t	$\Delta^2 u_t$	$\Delta^3 u_t$	$\Delta^4 u_t$	$\Delta^5 u_t$
–2	24.145					
		–2.102				
–1	22.043		.284			
		–1.818		.047		
0	20.225		.237		.009	
		–1.581		–.038		–.003
1	18.644		.199		.006	
		–1.382		–.032		
2	17.262		.167			
		–1.215				
3	16.047					

Gauss forward formula is :

$$u_t = u_0 + t\Delta u_0 + \frac{t(t-1)}{2}\Delta^2 u_{-1} + \frac{(t+1)\,t\,(t-1)}{6}\Delta^3 u_{-1}$$

$$+ \frac{(t+1)\,t\,(t-1)\,(t-2)}{24}\Delta^4 u_{-2}$$

$$+ \frac{(t+2)\,(t+1)\,t\,(t-1)}{120}\Delta^5 u_{-2} + \ldots$$

$$\therefore u_{0.5} = 20.225 + .5\,(-1.581) + \frac{(.5)\,(.5-1)}{2}\,(.237)$$

$$+ \frac{(.5+1)(.5)(.5-1)}{6}(-.038)$$

$$+ \frac{(.5+1)(.5)(.5-1)(.5-2)}{24}(.009)$$

$$+ \frac{(.5+2)(.5+1)(.5)(.5-1)(.5-2)}{120}(-.003)$$

$$= 20.225 - .7905 - .029625 + .002375 + .0002109 - .0000351$$

$$= 19.407426$$

Hence the estimated value of u, when $x = 3.75$ is 19.41 approximately.

Ex. 2: Interpolate by means of Gauss backward formula the population for the year 1936, given the following table :

Year :	1901	1911	1921	1931	1941	1951
Population :	12	15	20	27	39	52

Take 1931 as the origin and 10 years as the unit. Then the population u in thousands is to be estimated for

$$x = \frac{1936 - 1931}{10} = 0.5$$

x	u	Δu_x	$\Delta^2 u_x$	$\Delta^3 u_x$	$\Delta^4 u_x$	$\Delta^5 u_x$
–3	12					
		3				
–2	15		2			
		5		0		
–1	20		2		3	
		7		3		–10
0	27		5		–7	
		12		–4		
1	39		1			
		13				
2	52					

Gauss backward formula is :

$$u_x = u_0 + \binom{x}{1}\Delta u_{-1} + \binom{x+1}{2}\Delta^2 u_{-1} + \binom{x+1}{3}\Delta^3 u_{-2}$$

$$+\binom{x+2}{4}\Delta^4 u_{-2} + \binom{x+2}{5}\Delta^5 u_{-3} + \cdots$$

$$\therefore u_{0.5} = 27 + .5\,(7) + \frac{1.5 \times .5}{2}(5) + \frac{1.5 \times .5 \times (-.5)}{6}(3)$$

$$+ \frac{(2.5)\,(1.5)\,(.5) \times (-.5)}{24}(-7)$$

$$+ \frac{(2.5)\,(1.5)\,(.5)\,(-.5)\,(-.15)}{120}(-10)$$

$$= 27 + 3.5 + 1.875 - .1875 + .2734375 - .1171875$$

$$= 32.34375 \text{ thousands.}$$

Hence the estimated population for 1936 is 32.34375 thousands.

Ex. 3 : The following table gives the values of the probability integral

$$u(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2}\,dx$$

for cert ain equidistant values of x. Using Stirling's formula, find the value of this integral when $x = 0.5437$.

x	.51	.52	.53	.54	.55	.56	.57
u	.5292437	.5378987	.5464641	.5549392	.5633233	.5716157	.5798158

We take .54 as origin and 0.01 as unit. Then it is required to find the value of u when

$$t = \frac{.5437 - .54}{.01} = 0.37.$$

The difference table is :

t	u	Δu_t	$\Delta^2 u_t$	$\Delta^3 u_t$	$\Delta^4 u_t$
–3	.5292437				
		86550×10^{-7}			
–2	.5378987		-896×10^{-7}		
		85654×10^{-7}		-7×10^{-7}	
–1	.5464641		903×10^{-7}		0
		85751×10^{-7}		-7×10^{-7}	
0	.5549392		-910×10^{-7}		0
		83841×10^{-7}		-7×10^{-7}	
1	.5633233		916×10^{-7}		1×10^{-7}
		82924×10^{-7}		6×10^{-7}	
2	.5716157		-923×10^{-7}		
		82001×10^{-7}			
3	.5798158				

Stirling's interpolation formula is :

$$u_t = u_0 + t\mu\delta u_0 + \frac{t^2}{2!}\delta^2 u_0 + \frac{t^2\left(t^2-1\right)}{3!}\mu\delta^3 u_0 + \frac{t^2\left(t^2-1\right)}{4!}\delta^4 u_0 + \ldots$$

$$\therefore\ u_{0.37} = .5549392 + \frac{.37\ (.0084751 + .0083841)}{2} + \frac{(.37)^2\ (-.0000910)}{2}$$

$$+ \frac{(.37)\left(.37^2-1\right)}{6}\ \frac{(-.0000007 - .0000007)}{2}$$

$$= .5549392 + .00311895 - .00000623 + .00000004$$

$$= .5580520.$$

Thus the value of the probability integral when $x = .5437$ is .5580520

Ex. 4 : Employ Stirling's formula to compute $u_{12.2}$ from the following table $\left(u_x = 1 + \log_{10}\sin x\right)$

$x°$:	10	11	12	13	14
$10^5\ u_x$:	23967	28060	31788	35209	38368

Take 12 as origin and 1 as unit. Then we require v_t i.e., u_x for which

$$t = \frac{12.2 - 12}{1} = .$$

The difference table is :

t	v_t	δv_t	$\delta^2 v_t$	$\delta^3 v_t$	$\delta^4 v_t$
–2	.23967				
		.04093			
–1	.28060		–.00365		
		.03728		.00058	
0	.31788		–.00307		–.00013
		.03421		.00045	
1	.35209		–.00262		
		.03159			
2	.38368				

Stirling's formula is :

$$v_t = v_0 + t\mu\delta v_0 + \frac{t^2}{2!}\delta^2 v_0 + \frac{t\left(t^2-1\right)}{3!}\mu\delta^3 v_0 + \frac{t^2\left(t^2-1\right)}{4!}\delta^4 v_0 + \ldots$$

Here $v_0 = .31788,$

$$\mu\delta v_0 = \frac{1}{2}\ (.03728 + .03421) = .035745,$$

$$\delta^2 v_0 = -.00307,$$

$$\mu\delta^3 y_0 = \frac{1}{2}(.00058 + .00045) = .000515,$$

$$\delta^4 y_0 = -.00013$$

$$\therefore y_{.2} = .31788 + (.2)(.035745) + \frac{(.2)^2}{2}(-.00307)$$

$$+ \frac{(.2)(-.96)}{6}(.000515) + \frac{(.04)(-.96)}{24}(-.00013)$$

$$= .31788 + .007149 - .0000614 - .0000164 + .0000002$$

$$= .3249514$$

Thus $u_{12.2} = .32495$ approximately

Ex. 5 : The value of e^{-x} for certain equidistant values of x are given in the following table. Using Bessel's formula find the value of e^{-x} when $x = 1.7489$.

x	1.72	1.73	1.74	1.75
e^{-x}	.1790661479	.1772844100	.1755204006	.1737739435

1.76	1.77	1.78
.1720448638	.1703329888	.1686381473

Choose 1.74 as the origin and .01 as unit. Then we have to find e^{-x} for which

$$u = \frac{1.7489 - 1.74}{.01} = .89$$

Then $v = u - \frac{1}{2} = .89 - .50 = .39$.

The difference table is :

u	$e^{-x} = y$	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
–2	.1790661479				
		–.0017817379			
–1	.1772844100		.0000177285		
		–.0017640094		–.0000001762	
0	.1755204006		.0000175523		.0000000013
		–.0017464571		–.0000001749	
1	.1737739435		.0000173774		.0000000022
		–.0017290797		–.0000001727	
2	.1720448638		.0000172047		.0000000015
		–.0017118750		–.0000001712	
3	.1703329888		.0000170335		
		–.0016948415			
4	.1686381473				

Bessel's formula is :

$$y_u = \frac{y_0 + y_1}{2} + \nu\Delta y_0 + \frac{(\nu^2 - 1/4)}{2}\,\frac{(\Delta^2 y_{-1} + \Delta^2 y_0)}{2}$$

$$+ \frac{\nu(\nu^2 - 1/4)}{3!}\Delta^2 y_{-1} + \frac{(\nu^2 - 1/4)(\nu^2 - 9/4)}{4!}\,\frac{(\Delta^4 y_{-2} + \Delta^4 y_{-1})}{2} + \ldots$$

$$y_{0.89} = \frac{.1755204006 + .1737739435}{2} + .39\,(-.00174664571)$$

$$+ \left(\frac{.39^2 - .25}{2}\right)\frac{(.0000175523 + .0000173774)}{2}$$

$$+ .39\left(\frac{.39^2 - .25}{2}\right)(-.0000001749) + \frac{(.39^2 - .25)(.39^2 - 2.25)}{24}$$

$$\frac{(.0000000013 + .0000000022)}{2}$$

$$= .17464717205 - .00068111827 - .00000085490$$

$$+ .00000000111 + .00000000001$$

$$= .1739652000$$

Thus the value of e^{-x} when $x = 1.7400$ is .1739652000

Ex. 6 : The values of $x^{1/3}$ for certain equidistant values of x are given in the following table. Compute $344.5^{1/3}$.

x :	342	343	344	345	346	347
u_x :	6.993191	7.000000	7.006796	7.013579	7.020349	7.027106

Take origin at 344 and 1 as unit. The difference table is :

t	u_t	δu_t	$\delta^2 u_t$
–2	6.993191		
		.006809	
–1	7.00000		– .000013
		.006796	
0	7.006796		– .000013
		.006783	
1	7.013579		– .000013
		.006770	
2	7.020349		– .000013
		.006557	
3	7.027106		

It is required to compute at $x = 344.5$ i.e., at the midpoint of the interval (344, 345). We therefore compute at $t = .5$, the midpoint of the interval (0, 1). Then we use the interpolation formula for halving the interval of tabulation, namely

$$u_{1/2} = \mu u_{1/2} - \frac{1}{8}\mu\delta^2 u_{1/2} + \frac{3}{128}\mu\delta^4 u_{1/2} - \ldots$$

$$= \frac{1}{2}(7.006796 + 7.013579) - \frac{1}{8}\frac{(-.000013 - .000013)}{2}$$

$$= 7.0101875 + .0000016$$

$$= 7.010189$$

Thus the value of $x^{1/3}$ when $x = 344.5$ is 7.010189.

Ex. 7 : Given $y_0, y_1, y_2, y_3, y_4, y_5$, (fifth differences constant), prove that

$$y_{2.5} = \frac{1}{2}c + \frac{25(c-b) + 3(a-c)}{256}$$

where $a = y_0 + y_5$, $b = y_1 + y_4$, $c = y_2 + y_3$.

If the fifth differnces are constant, then Bessel's interpolation formula for halving the inverval of tabulation gives

$$y_{1/2} = \frac{1}{2}(y_0 + y_1) - \frac{1}{8}\frac{(\Delta^2 y_{-1} + \Delta^2 y_0)}{2} + \frac{3}{128}\frac{(\Delta^4 y_{-2} + \Delta^4 y_{-1})}{2}$$

Shifting the origin to 2, this formula gives

$$y_{2.5} = \frac{1}{2}(y_2 + y_3) - \frac{1}{8}\frac{(\Delta^2 y_1 + \Delta^2 y_2)}{2} + \frac{3}{128}\frac{(\Delta^4 y_0 + \Delta^4 y_1)}{2}$$

We know

$$\Delta y_1 = y_3 - 2y_2 + y_1.$$

$$\Delta y_0 = y_4 - 4y_3 + 6y_2 - 4y_1 + y_0, \text{ etc.}$$

Substituting these in the above formula, we get

$$y_{2.5} = \frac{1}{2}(y_2 + y_3) + \frac{25(y_2 + y_3) - 25(y_1 + y_4) + 3(y_0 + y_5) - 3(y_2 + y_3)}{256}$$

Replacing $(y_2 + y_3)$ by c, $(y_1 + y_4)$ by b and $(y_0 + y_5)$ by a,

we get $y_{2.5} = \frac{1}{2}c + \frac{25(c-b) + 3(a-c)}{256}$

as the required result.

Ex. 8 : Given the table

x :	310	320	330	340	350	360
$u_x = \log_x$:	2.4913617	2.5051500	2.5185139	2.5314789	2.5440680	2.5563025

Find the value of $u_{337.5}$ by Laplace – Everett formula.

Taking 330 as the origin and 10 as the unit, we have to find u when

$$v = \frac{337.5 - 330}{10} = .75$$

Thus $s = 1 - v = 1 - 0.75 = 0.25$

The difference table is

v	u	δu	$\delta^2 u$	$\delta^3 u$	$\delta^4 u$	$\delta^5 u$
–2	2.4913617					
		.0137883				
–1	2.5051500		– .0004244			
		.0133639		.0000255		
0	2.5185139		– .0003989		– .0000025	
		.0129650		.0000230		.0000008
1	2.5314789		– .0003759		– .0000017	
		.0125891		.0000213		
2	2.5440680		– .0003546			
		0.122345				
3	2.5563025					

Laplace- Everett formula is :

$$u = v\left[u_1 + \frac{(v^2-1)}{3!}\delta^2 u_1 + \frac{(v^2-1)(v^2-4)}{5!}\delta^4 u_1 + \ldots\right]$$

$$+ s\left[u_0 + \frac{(s^2-1)}{3!}\delta^2 u_0 + \frac{(s^2-1)(s^2-4)}{5!}\delta^4 u_0 + \ldots\right]$$

$$\therefore u_{0.75} = 0.75\left[2.5314789 + \frac{(.75^2-1)}{6}(-.0003759)\right.$$

$$\left. + \frac{(.75^2-1)(.75^2-4)}{120}(-.0000017)\right]$$

$$+ .25\left[2.5185139 + \frac{(.25^2 - 1)}{6}(-.0003989)\right.$$

$$\left. + \frac{(.25^2 - 1)(.25^2 - 4)}{120}(-.0000025)\right]$$

$$= .75\,(2.5314789 + .0000274) + .25\,(2.5185139 + .0000623 - .0000001)$$

$$= 1.8986297 + .629644 = 2.5282737$$

Thus the value of $u_{337.5}$ by Laplace – Everett formula is 2.5282737.

3.8 SUMMARY

In this unit we have derived certain interpolation formulae which are useful when we are required to estimate the values of a function at places near the middle point of a given table of values. Though the formulae look difficult we see that there is a pattern in the relationship between the coefficients and exponents of the variable. These have been indicated through arrows leading the path of the formula. We have introduced the central difference operators δ and μ, and their relationships to Δ, we relied upon the formulae involving Δ alone in solving the problems.

3.9 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail*

1. Prove the following identities.

(a) $\mu = \dfrac{2 + \Delta}{2\sqrt{1 + \Delta}} = \dfrac{2 - \nabla}{2\sqrt{1 - \nabla}} = \sqrt{1 + \dfrac{\delta^2}{4}}$

(b) $\Delta = \dfrac{1}{2}\delta^2 + \delta\sqrt{1 + \dfrac{\delta^2}{4}}$

(c) $\Delta\nabla = \nabla\Delta = \Delta - \nabla = \delta^2$

(d) $\mu\delta = \dfrac{1}{2}(\Delta + \nabla) = \sinh(hD)$

2. Show that

(a) $\delta[f(x)\,g(x)] = \mu f(x)\,\delta g(x) + \mu g(x)\,\delta f(x)$

(b) $\delta\left[\dfrac{f(x)}{g(x)}\right] = \dfrac{\mu g(x)\,\delta f(x) - \mu f(x)\,\delta g(x)}{g(x - 1/2)\,g(x + 1/2)}$

(c) $\mu[f(x)\,g(x)] = \mu f(x)\,\mu g(x) + \dfrac{1}{4}\delta f(x)\,\delta g(x)$

(d) $\mu\left[\dfrac{f(x)}{g(x)}\right] = \dfrac{\mu f(x)\,\mu g(x) - 1/4\,\delta f(x)\,\delta g(x)}{g(x - 1/2)\,g(x + 1/2)}$

3. If D, E, δ and μ be the operators with usual meaning and if $hD = V$ where h is the interval of differencing prove the following relations.

(a) $\delta = 2 \sinh\left(\frac{V}{2}\right)$

(b) $\mu = 2 \cosh\left(\frac{V}{2}\right)$

(c) $e^{-V} = 1 - \nabla$

(d) $(E + 1)\,\delta = 2\,(E - 1)\,\mu$

(e) $\left(\frac{\delta}{V}\right)^2 = 2\,(\cosh V - 1)/\,V^2$

(f) $\frac{\delta}{\mu V} = (\tanh V/2)/\frac{V}{2}$

(g) $\frac{\mu\delta}{V} = \frac{\sinh V}{V}$

4. Prove that

(i) $\delta^2 y_0 = y_1 - 2y_0 + y_{-1}$

(ii) $\delta^3 y_{1/2} = y_2 - 3y_1 + 3y_0 - y_{-1}$

5. From a difference table, given $u_{-3} = 720$, $u_{-2} = 277$, $u_{-1} = 76$, $u_0 = 9$, $u_1 = -8$, $u_2 = -35$ and $u_3 = -108$. Identify $\delta u_{3/2}$, $\delta^2 u_{-1}$, $\delta^3 u_{1/2}$, $\delta^4 u_1$ in the table and give other names for these differences in terms of Δ and ∇.

6. (a) Use Gauss forward formula to find y_{30}, given that

$y_{21} = 18.4708$, $y_{25} = 17.8144$, $y_{29} = 17.1070$,

$y_{33} = 16.3432$, $y_{37} = 15.5154$ [Ans. 16.9216]

(b) Given $y_2 = 10$, $y_1 = 9$, $y_0 = 5$, $y_{-1} = 10$, find $y_{1/2}$ by Gauss forward formula. [6.66]

7. Given that

$\sqrt{12500} = 111.803399$; $\sqrt{12510} = 111.848111$

$\sqrt{12520} = 111.892806$; $\sqrt{12530} = 111.937487$

show by Gauss backward formula that

$\sqrt{12516} = 111.874930$.

8. Find cos 0.806595 by Stirling's formulae, given

cos 0.8050 = 0.693111235

cos 0.8055 = 0.692750733

cos 0.8060 = 0.692390058

cos 0.8065 = 0.692029210

cos 0.8070 = 0.691668188

cos 0.8075 = 0.691306994

cos 0.8080 = 0.690945627 [Ans. 0.691960629]

9. Use Stirling's formula to find y_{35}, given

$y_{20} = 512,\ y_{30} = 439, y_{40} = 346,\ y_{50} = 243$

where y_x represents the number of persons at age x years in a life table. [Ans. 395]

10. Apply Bessel's formula to obtain y_{25}, given

$y_{20} = 2854,\ y_{24} = 3162, y_{28} = 3544, y_{32} = 3992.$ [Ans. 3250.875]

11. The function $u_x = 1.015^{-x}$ is tabulated below

x	u_x
48	.48936170
50	.47500468
52	.46106887
54	.44754092
56	.43441182

Compute 1.015^{-53} by Bessel's formula. [Ans. 0.454255.05]

12. If third differences are constant prove that

$$y_{x+1/2} = \frac{1}{2}\left(y_x + y_{x+1}\right) - \frac{1}{16}\left(\Delta^2 y_{x-1} + \Delta^3 y_x\right)$$

13. Use Everett's formula to obtain u_{1031}, given

x	u_x
1000	0
1010	43214
1020	86002
1030	128372
1040	170333
1050	211893
1060	253059
1070	293838

[Ans: 132586]

3.10 ANSWERS TO SAQ'S

SAQ 1 By definition of δ, we have

$$\delta u_x = \left(E^{1/2} - E^{-1/2}\right) u_x = u_{x+h/2} - u_{x-h/2}$$

Since $u_x = x$, we have on replacing x by $x + h/2$,

$$u_{x+h/2} = x + h/2$$

Similarly, $u_{x-h/2} = x - h/2.$

Hence $\delta x = (x + h/2) - (x - h/2) = h$

SAQ2 From definition of δ, we have

$$\delta u_x = \left(E^{1/2} - E^{-1/2}\right) u_x = u_{x+h/2} - u_{x-h/2}$$

Also $\Delta E^{-1/2} u_x = \Delta u_{x-h/2} = u_{x+h/2} - u_{x-h/2}$

and $\nabla E^{1/2} u_x = \nabla u_{x+h/2} = u_{x+h/2} - u_{x-h/2}$

$$\therefore \delta u_x = \Delta E^{-1/2} u_x = \nabla E^{1/2} u_x$$

i.e., $\delta = \Delta E^{-1/2} = \nabla E^{1/2}$

Squaring and cubing 'δ', we get

$$\delta^2 = \Delta^2 E^{-1} = \nabla^2 E$$

$$\delta^3 = \Delta^3 E^{-3/2} = \nabla^3 E^{3/2}$$

Since in this problem $h = 1$, we have

$$\delta u_{1/2} = \Delta E^{-1/2} u_{1/2} = \Delta u_0$$

$$\delta u_{1/2} = \nabla E^{1/2} u_{1/2} = \nabla u_1$$

$$\therefore \Delta u_0 = \nabla u_1 = \delta u_{1/2}$$

Also $\delta^2 u_0 = \Delta^2 E^{-1} u_0 = \Delta^2 u_{-1}$

$$\delta^2 u_0 = \nabla^2 E u_0 = \nabla^2 u_1$$

$$\therefore \Delta^2 u_{-1} = \nabla^2 u_1 = \delta^2 u_0.$$

and $\delta^3 u_{-1/2} = \Delta^2 E^{-3/2} u_{-1/2} = \Delta^3 u_{-2}$

$$\delta^3 u_{-1/2} = \nabla^3 E^{3/2} u_{-1/2} = \nabla^3 u_1$$

$$\therefore \Delta^3 u_{-2} = \nabla^3 u_1 = \delta^3 u_{-1/2}$$

UNIT–4 : ERRORS IN INTERPOLATION FORMULAE AND LEAST SQUARES APPROXIMATION

Contents

4.1 AIMS AND OBJECTIVES

After going through this unit you will be able to : (i) derive the error in Lagranges, Newton's forward and backward, Stirling's, Bessel's interpolation formulae (ii) derive a formula for the maximum error inherent in linear interpolation estimator for a straight line, parabola and a curve of the form αx^m.

4.2 INTRODUCTION

What all that had discussed so far is approximating a function by a polynomial. Consequently, the values we are estimating using the interpolation formulae are the values of the interpolation polynomials but not the exact values of the function. This gives rise to the error between the exact and approximate values of the function. In this unit we obtain expressions for the errors involved in various interpolation formulae. We also study here, the technique of least squares approximation.

Suppose we are given $f(x_0), f(x_1), \ldots, f(x_n)$ of a function $f(x)$. Then with the help of their difference table, we know from the preceeding units how to construct an interpolating polynomial $\phi(x)$ such that

$$\phi(x_0) = f(x_0),\ \phi(x_1) = f(x_1),\ \ldots\ \phi(x_n) = f(x_n).$$

As n increases indefinitely, it was our feeling all through that $\phi(x)$ conicides with $f(x)$ for any x in the interval $x_0 \le x \le x_n$. Supposing it does i.e., $\phi(x)$ converges to $f(x)$, (proof of convergence not attempted, being beyond our scope of present reading), we define $\{f(x) - \phi(x)\}$ as the error in $\phi(x)$ for that x.

In approximating functions by polynomials through interpolation we have actually found a polynomial curve exactly passing through all the given points (x_i, y_i), $i = 1, 2, \ldots n$. This is actually good when the data is an accurate one. Now, when the data corresponds to that obtained from an experiment, usually there will be errors of observation in the values of x_i or y_i or both. In such a situation no purpose will be served by asking for a polynomial curve going through points which are themselves not accurate. Then what we do is to find a curve which goes nearer to the given points but need not pass through all the given points (The curve of best fit). This is done in method of least squares approximation. The method of least squres envisages a procedure by which the sum of the squares of the difference between the expected and theoretical values is least. Based on this principle the least squares estimators for a straight line, parabola and a curve of the form $y = \alpha x^m$ (m fixed) have been derived. A formula for the minimal sum of squared differences between observed and theoretical values of a straight line has also been obtained.

4.3 ERROR IN LAGRANGE'S AND NEWTON'S FORWARD INTERPOLATION FORMULAS

We shall assume $f(x)$ is continuous and possesses continuous derivatives of all orders within the interval from x_0 to x_n. We write down the arbitrary function

$$F(z) = f(z) - \phi(z) - [f(x) - \phi(x)] \frac{(z - x_0)(z - x_1) \ldots (z - x_n)}{(x - x_0)(x - x_1) \ldots (x - x_n)}$$

where $f(x)$ denotes the given function, $\phi(z)$ a polynomial interpolation formula and z is a real variable.

Now $F(z)$ vanishes for $(n + 2)$ values $z = x, x_0, x_1, \ldots, x_n$ and since $f(x)$ is continuous and has continuous derivatives of all orders, the same is true of $f(z)$ and hence of $F(z)$. $F(z)$ therefore satisfies the conditions of Rolle's theorem. Hence the first derivative of $F(z)$ vanishes at least once between every two consecutive zero values of $F(z)$. Therefore in the interval from x_0 to x_n, $F'(z)$ must vanish $(n + 1)$ times; $F''(z)$, n times; $F'''(z)$, $(n - 1)$ times; etc. Hence $(n + 1)^{th}$ derivative of $F(z)$ will vanish atleast once at some point say ξ, where ξ lies in an interval containing (x_0, x_n) or x as end points.

Since $\phi(z)$ is a polynomial of n^{th} degree, its $(n + 1)^{th}$ derivative is zero. Furthermore, since the expression $(z - x_0)(z - x_1)(z - x_2) \ldots (z - x_n)$ is a polynomial of degree $(n + 1)$, it follows that its $(n + 1)^{th}$ derivative is the same as the $(n + 1)^{th}$ derivative of z^{n+1}, which is $(n + 1)!$. On differentiating $F(z)$ defined above, $(n + 1)$ times with respect to z, we therefore have

$$F^{(n+1)}(z) = f^{(n+1)}(z) - [f(x) - \phi(x)] \frac{(n + 1)!}{(x - x_0)(x - x_1) \ldots (x - x_n)}$$

Since $F^{(n+1)}(z) = 0$ at some point $z = \xi$, we have

$$0 = f^{(n+1)}(\xi) - [f(x) - \phi(x)] \frac{(n + 1)!}{(x - x_0)(x - x_1) \ldots (x - x_n)}$$

Hence we have

$$\text{Error} = R_n = \frac{f^{(n+1)}(\xi)}{(n + 1)!}(x - x_0)(x - x_1) \ldots (x - x_n),$$

where ξ is some value of x between x_0 and x_n. This is the remainder term in both Lagrange's and Newton's forward interpolation formulas.

Putting $x - x_0 = hu, x - x_1 = h(u-1),$

$x - x_2 = h(u-2), \ldots x - x_n = h(u-n)$ in R_n, we get

$$R_n = \frac{h^{(n+1)} f^{(n+1)}(\xi)}{(n+1)!} u(u-1)(u-2)\ldots(u-n),$$

as also the remainder in Newton's forward interpolation formula. When the analytic form of $f(x)$ is not known, we replace $f^{(n+1)}(\xi)$ by its value in terms of differences as below.

We have the fundamental relation between differences and derivatives; namely;

$$\Delta^n f(x) = (\Delta x)^n f^{(n)}(x + \theta n \Delta x), 0 < \theta < 1.$$

putting $x = x_0$, and $\Delta x = h$, we get

$$f^{(n)}\left(x_0 + \theta n h\right) = \frac{\Delta^n f\left(x_0\right)}{h^n}$$

Now since $\left(x_0 + \theta n h\right)$ and ξ are values of x at points within the interval of interpolation x_0 to x_n we put $\xi = x_0 + \theta nh$, Therefore

$$f^{(n)}(\xi) = \frac{\Delta^n f\left(x_0\right)}{h^n}.$$

Hence we have

$$f^{(n+1)}(\xi) = \frac{\Delta^{n+1} f\left(x_0\right)}{h^{n+1}}.$$

Substituting this value of $f^{(n+1)}(\xi)$ in R_n above, we get

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

Since $f\left(x_0\right) = y_0$.

The smaller the interval h is taken, the more nearly does R_n give the actual error.

4.4 ERROR IN NEWTON'S BACKWARD INTERPOLATION FORMULA

To find the error in Newton's formula for backward interpolation we write down the function.

$$F(z) = f(z) - \phi(z) - [f(x) - \phi(x)] \frac{\left(z - x_n\right)\left(z - x_{n-1}\right)\ldots\left(z - x_0\right)}{\left(x - x_n\right)\left(x - x_{n-1}\right)\ldots\left(x - x_0\right)}$$

Differentiating this with respect to z, $(n+1)$ times and setting $F^{(n+1)}(z) = 0$ for $z = \xi$, we find

$$f(x) - \phi(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_n)(x - x_{n-1}) \ldots (x - x_0)$$

or $$\text{Error} = R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_n)(x - x_{n-1}) \ldots (x - x_0)$$

which is the remainder in Newton's backward interpolation formula.

To write it in terms of u, we recall

$$x - x_n = hu, \; x - x_{n-1} = h(u+1), \ldots (x - x_0) = h(u+n)$$

Substituting them in R_n above, we get

$$R_n = \frac{h^{n+1} f^{(n+1)}(\xi)}{(n+1)!} u(u+1)(u+2) \ldots (u+n).$$

To obtain R_n when the analytical form of the given function is unkown, we replace

$f^{(n+1)}(\xi)$ by $\frac{\nabla^{n+1} y_n}{h^{n+1}}$ where $f(x_n) = y_n$.

The result is $$R_n = \frac{\nabla^{n+1} y_n}{(n+1)!} u(u+1)(u+2) \ldots (u+n).$$

4.5 ERROR IN STIRLING'S FORMULA

To find the remainder term in Stirling's formula, we write down the arbitrary function

$$F(z) = f(z) - \phi(z) - [f(x) - \phi(x)] \frac{(z - x_0)(z - x_1)(z - x_{-1}) \ldots (z - x_n)(z - x_{-n})}{(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})}$$

This function vanishes for the $(2n+2)$ values $z = x, x_0, x_1, \ldots, x_n, x_{-1}, \ldots, x_{-n}$. We assume $f(x)$ is continuous and has continuous derivatives of all orders upto $(2n+1)$. Hence $F(z)$ satisfies the conditions of Rolle's theorem. Since $\phi(z)$ is a polynomial of degree '$2n$' its $(2n+1)^{th}$ derivative is zero. Therefore on differentiating $F(z)$, $(2n+1)$ times and putting $F^{(2n+1)}(z) = 0$ for some value $z = \xi$, we get

$$0 = f^{(2n+1)}(\xi) - [f(x) - \phi(x)] \frac{(2n+1)!}{(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})}$$

for which

$$f(x) - \phi(x) = \frac{f^{(2n+1)}(\xi)}{(2n+1)!}(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})$$

or $$\text{Error} = R_n = \frac{f^{(2n+1)}(\xi)}{(2n+1)!}(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})$$

We now write

$$x - x_0 = hu, \; x - x_1 = h(u-1), \ldots, x - x_n = h(u-n),$$

and $(x - x_{-1}) = h(u + 1), \ldots, x - x_{-n} = h(u + n)$.

Thus $$R_n = \frac{h^{2n+1} f^{(2n+1)}(\xi)}{(2n+1)!} u(u^2 - 1)(u^2 - 2^2) \ldots (u^2 - n^2),$$

where ξ is some value of x between x_{-n} and x_n.

If the analytic form of $f(x)$ is unknown, we replace

$f^{(2n+1)}(\xi)$ by $\frac{m_{2n+1}}{h^{2n+1}}$

where $$m_{2n+1} = \frac{\Delta^{2n+1} y_{-n-1} + \Delta^{2n+1} y_{-n}}{2}$$

Therefore $$R_n = \frac{m_{2n+1}}{(2n+1)} u(u^2 - 1)(u^2 - 2^2) \ldots (u^2 - n^2)$$

4.6 ERROR IN BESSEL'S FORMULA

The remainder term in Bessel's formula is derived by first writing down the arbitrary function

$$F(z) = f(z) - \phi(z) - [f(x) - \phi(x)] \frac{(z - x_0)(z - x_1)(z - x_{-1}) \ldots (z - x_n)(z - x_{-n})(z - x_{n+1})}{(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})(x - x_{n+1})}$$

$F(z)$ vanishes at the $(2n + 3)$ points $z = x, x_0, x_1, x_{-1}, \ldots, x_n, x_{-n}, x_{n+1}$. Since $\phi(z)$ is a polynomial of degree $(2n + 1)$, its $(2n + 2)^{th}$ derivative is zero. Hence on differentiating $F(z)$, $(2n + 2)$ times with respect to z and putting $F^{(2n+2)}(z) = 0$ for some value $z = \xi$, we get

$$0 = f^{(2n+2)}(\xi) - [f(x) - \phi(x)] \frac{(2n+2)!}{(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})(x - x_{n+1})}$$

from which

$$f(x) - \phi(x) = R_n = \text{error} = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}(x - x_0)(x - x_1)(x - x_{-1}) \ldots (x - x_n)(x - x_{-n})(x - x_{n+1})$$

Putting $x - x_0 = hu$, $x - x_1 = h(u - 1)$, $(x - x_{-1}) = h(u + 1)$, etc.,

we get $$R_n = \frac{h^{2n+2} f^{(n+2)}(\xi)}{(2n+2)!} u(u - 1)(u + 1)(u - 2)(u + 2) \ldots (u - n)(u + n)(u - n - 1)$$

which is the remainder in Bessel's formula.

In case $f(x)$ is not known, we replace $f^{(2n+2)}(\xi)$ by

$$\frac{m_{2n+2}}{h^{2n+2}}, \text{ where}$$

$$m_{2n+2} = \frac{\Delta^{2n+2} y_{-n-1} + \Delta^{2n+2} y_{-n}}{2}$$

Then $$R_n = \frac{m_{2n+2}}{(2n+2)!} u(u-1)(u+1)(u-2)(u+2)$$

$$\ldots (u-n)(u+n)(u-n-1)$$

Putting $u = v + \frac{1}{2}$ we get,

$$R_n = \frac{h^{2n+2} f^{(2n+2)}(\xi)}{(2n+2)!} \left(v^2 - \frac{1}{4}\right)\left(v^2 - \frac{9}{4}\right) \ldots \left(v^2 - \frac{(2n+1)^2}{4}\right) \text{or}$$

$$R_n = \frac{m_{2n+2}}{(2n+2)!} \left(v^2 - \frac{1}{4}\right)\left(v^2 - \frac{9}{4}\right) \ldots \left(v^2 - \frac{(2n+1)^2}{4}\right)$$

Putting $v = 0$ we get the remainder terms in the formulas for interpolating to halves. Thus

$$R_n = \frac{h^{2n+2} f^{(2n+2)}(\xi)}{(2n+2)!} (-1)^{n+1} \frac{[1.3.5 \ldots (2n+1)]^2}{2^{2n+2}} \text{ or}$$

$$R_n = \frac{h^{2n+2} f^{(2n+2)}(\xi)}{(2n+2)!} (-1)^{n+1} \frac{[1.3.5 \ldots (2n+1)]^2}{2^{2n+2}}$$

4.7 THE ACCURACY OF LINEAR INTERPOLATION FROM TABLES

We shall now derive a simple formula for the maximum error inherent in linear interpolation from tables.

Substituting $n = 1$ in the error of Newton's forward interpolation formula, we get

$$R_1 = \frac{h^2 f^{(2)}(\xi)}{2} u(u-1) = \frac{h^2 M}{2}\left(u^2 - u\right)$$

Where M denotes the maximum absolute value of $f''(x)$ in any interval of width h. To find the maximum numerical value of R_1, we differentiate it with respect to u, put the derivative equal to zero, solve for u, and then substitute this value of u in R_1. Hence we have

$$\frac{dR_1}{du} = \frac{h^2 M}{2}(2u-1) = 0$$

$$\therefore u = \frac{1}{2}, \text{ Also } \frac{d^2 R_1}{du^2}\bigg|_{u=\frac{1}{2}} = h^2 M, \text{ a positive quantity.}$$

Hence $u = \frac{1}{2}$ corresponds to a maximum point and the maximum value of R_1 is $\frac{h^2 M}{8}$ in magnitude.

$$\therefore |R_{max}| = \frac{h^2 M}{8}.$$

The formula for the maximum error is therefore

$$E \leq \frac{h^2 M}{8}$$

Examples

Ex. 1 : Determine the maximum error in the interpolation of $\log_{10} 1.024$ by Newton's formula for forward interpolation when the following table is given.

x :	1.02	1.03	1.04	1.05	1.06
$f(x)$:	0.0086002	0.0128372	0.0170333	0.0211893	0.0253059

Here $h = .01 = 10^{-2}$, $u = \frac{1.024 - 1.02}{.01} = .4$

$\therefore u - 1 = -.6$, $u - 2 = -1.6$, $u - 3 = -2.6$

Newton's forward interpolation formula is

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

The difference table is :

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1.02	.0086002				
		.004237			
1.03	.0128372		–0.0000409		
		.0041961		.0000008	
1.04	.0170333		–.0000401		– .0000001
		.004156		.0000007	
1.05	.0211893		– .0000394		
		.0041166			
1.06	.0253059				

We neglect fourth differences and consider only upto third difference i.e., $n = 3$

$$y_{.4} = .0086002 + 4 \times 004237 + \frac{(.4)(-0.6)}{2} (- .0000409)$$

$$+ \frac{(.4)(-0.6)(-1.6)}{6} (.0000008)$$

$$= .0086002 + .0016948 + .0000049 - .0000001$$

$$= .0102998$$

$\therefore \log_{10} 1.024 = .0102998$

Error in Newton's formula for forward interpolation is given by

$$R_n = \frac{h^{n+1} f^{(n+1)}(\xi)}{(n+1)!} u(u-1) \ldots (u-n)$$

In this problem $f(x) = \log_{10} x$. Also $n = 3$ as already stated

$$R_3 = \frac{h^4}{4!} u(u-1)(u-2)(u-3) \frac{d^4}{dx^4} \left(\log_{10} x\right)\Big|_{x=\xi}$$

where $1.02 < \xi < 1.05$ since we have not effectively used $x = 1.06$.

$$\therefore R_3 = \frac{10^{-8}}{24} (.4)(-0.6)(-1.6)(-2.6) \frac{d^4}{dx^4} \left(\log_e x\right)\Big|_{x=\xi} \times \log_{10} e$$

$$= -10^{-8} \times .0416 \times \frac{-6}{x^4}\Big|_{x=\xi} \times 0.4342944$$

$$= 10^{-8} \times 0.2496 \times \frac{1}{x^4}\Big|_{x=\xi} \times 0.4342944$$

At $\xi = 1.02$ $R_3 = 10^{-8} \times .1083998 \times .923845$

$= 10^{-8} \times .1001446$

At $\xi = 1.05$, $R_3 = 10^{-8} \times 0.1083998 \times .82270$

$= 10^{-8} \times 0.08[illegible]1807$

Thus the error in the interpolation must lie in the range

$10^{-9} \times 0.891807 < E < 10^{-8} \times 1.001446$

and the maximum error is $10^{-8} \times 1.001446$.

Ex. 2 : Determine the accuracy of the interpolation by Lagranges formula for $\log_{10} 47$ given

x :	40	42	45	48	49	50
$y = \log_{10} x$:	1.6020600	1.6232493	1.6532126	1.6812413	1.6901960	1.989700

Letting $x_0 = 40$, $x_1 = 42$, $x_2 = 45$, $x_3 = 48$, $x_4 = 49$, $x_5 = 50$ and applying Lagrange's interpolation formula we get

$$y = \frac{(5)(2)(-1)(-2)(-3)}{(-2)(-5)(-8)(-9)(-10)} (1.6020600) + \frac{(7)(2)(-1)(-2)(-3)}{(2)(-3)(-6)(-7)(-8)} (1.6232493)$$

$$+ \frac{(7)(5)(-1)(-2)(-3)}{(5)(3)(-3)(-4)(-5)} (1.6532126) + \frac{(7)(5)(2)(-2)(-3)}{(8)(6)(3)(-1)(-2)} (1.6812413)$$

$$+ \frac{(7)(5)(2)(-1)(-3)}{(9)(7)(4)(1)(-1)} (1.6901960) + \frac{(7)(5)(2)(-1)(-2)}{(10)(8)(5)(2)(1)} (1.989700)$$

$$= 0.0133505 - 0.06763543 + 0.3857496 + 2.4518102 - 1.4084967$$

$$+ 0.3481975$$

$$= 1.7229758$$

Now $R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \ldots (x - x_n)$

Here $n = 5$.

$\therefore R_5 = \frac{f^{(6)}(\xi)}{720}(7)(5)(-1)(-2)(-3)$

$= -\frac{7}{12} f^{(6)}(\xi)$

where $40 < \xi < 50$.

Now $f^{(6)}(x) = \frac{d^6}{dx^6}(\log_{10} x) = \frac{d^6}{dx^6}(\log_e x) \cdot \log_{10} e$

$= -\frac{120}{x^6} \log_{10} e$

$\therefore f^{(6)}(\xi) = -\frac{120}{\xi^6} \times .4342944$

$= -\frac{52.115333}{\xi^6}$

$R_5 = \frac{30.400613}{\xi^6}$

At $\xi = 40$, $R_5 = 10^{-6} \times 0.007422 = 10^{-8} \times .7422$

At $\xi = 50$, $R_5 = 10^{-6} \times .0019456 = 10^{-8} \times 0.19456$

Hence $10^{-8} \times 0.19456 < \text{Error} < 10^{-8} \times 0.7422$

and the interpolation error is less than one unit in the eighth place.

Ex. 3 : The following table contains values of the function $y = x^4 + 10x^5$ for certain values of x. Find y when $x = 2.27$ by using Bessel's formula. Obtain the error.

x :	2.0	2.1	2.2	2.3	2.4	2.5
y :	336.0000	427.8582	538.7888	671.6184	829.4400	1015.6250

The difference table is :

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
2.0	336.0000				
		91.8582			
2.1	427.8582		19.0724		
		110.9306		2.8266	
2.2	538.7888		21.8990		0.2664
		132.8296		3.0930	
2.3	671.6184		24.9920		0.2784
		157.8216		3.3714	
2.4	829.4400		28.3634		
		186.1850			
2.5	1015.6250				

We take $x_0 = 2.2, x = 2.27, h = .1,$

Then $u = \frac{2.27 - 2.20}{.1} = 0.7$

$\therefore v = u - \frac{1}{2} = .2$

We also stop with third differences.

$$y_{2.27} = \frac{1}{2}(538.7888 + 671.6184) + (0.2)(132.8296) + \frac{(.04 - .25)}{2}\frac{(21.8990 + 24.992)}{2} + (.2)\frac{(0.04 - 0.25)}{6}(3.0930)$$

$$= 605.2036 + 26.56592 - 2.46178 - .02165$$

$$= 629.28609.$$

$\therefore y_{2.27} = 629.28609.$

To find R_n, we have $2n + 2 = 4$

$\therefore n = 1$

Also $f^{(4)}(x) = 24 + 1200x$

$\therefore f^{(4)}(\xi) = 24 + 1200\xi$

Since ξ lies between 2.0 and 2.5, we express

$\xi = 2.2 + 0.1\eta$

where η lies between -2.5 and $+2.5$. Substituting ξ in $f^{(4)}(\xi)$, we get $f^{(4)}(\xi) = 2724 + 120\eta$.

$$R_1 = \frac{h^4 f^{(4)}(\xi)}{4!}\left(v^2 - \frac{1}{4}\right)\left(v^2 - \frac{9}{4}\right)$$

$$= \frac{(.1)^4}{24}(2724 + 120\eta)(.04 - .25)(.04 - 2.25)$$

$$= .00527 + .000232\eta$$

$$= .00527 \pm .00058$$

$\therefore$ .00469 < Error < .00585

$y_{2.27}$ when corrected lies between 629.2919 and 629.2908, obtained by **additing .00585 and** .00469 to 629.28609 respectively. The mean of these is 629.2914.

If we substitute differences instead of the derivative in R_n, we have

$$m_4 = \frac{(.2664 + .2784)}{2} = .2724$$

$$\therefore R_1 = \frac{m_4}{4!}\left(v^2 - \frac{1}{4}\right)\left(v^2 - \frac{9}{4}\right)$$

$$= \frac{.2724}{24}(.04 - .25)(.04 - 2.25)$$

$$= .00527$$

We then have $y_{2.27} = 629.28609 + .00527 = 629.29136$ or **629.2914 correct to four decimal** places.

Ex. 4 : The function $\frac{1}{N}$ is tabulated in Barlow's tables at unit intervals from 1 to 12,500. Find the possible error in the linear interpolation of this function when N = 650.

Since $f(N) = \frac{1}{N}$, $f'(N) = -\frac{1}{N^2}$, $f''(N) = \frac{2}{N^3}$

Taking $h = 1$, N = 650 and substituting in

$$E \leq \frac{h^2 M}{8}, \text{ where } M = f''(N)$$

we get

$$E \leq \frac{1}{4 \times (650)^3} = \frac{1}{1098500000}$$

$$\text{or } E \leq .000000001$$

Note : Linear interpolation is permissible only when the first differences are constant. One should therefore check the constancy of few first differences before using linear interpolation.

4.8 LEAST SQUARES APPROXIMATION

The method of least-squares says that the best representative curve is that for which the sum of the squares of the residuals is a minimum. The residuals of a series of plotted points are the vertical distances of these points from the best representative curve. Some of the residuals will be positive and others negative. Since the squares of the residuals are positive quantitites, the requirement that their sum shall be as small as possible ensures that the numerical value of the residuals will be small; and this means that in the case of a series of plotted points, the best representative curve will pass as closely as possible to all the points. We shall apply this method to different curves and illustrate them by way of examples.

4.8.1 Least-squares estimator for a straight line

Given the data (x_i, y_i) for $i = 1, 2, \ldots, n$, we shall find the least-squares estimator for the true curve

$$F(x) = \alpha + \beta x$$

The theoretical values for $y_1, \ldots, y_n$ are $\alpha + \beta x_1, \alpha + \beta x_2, \ldots, \alpha + \beta x_n$ respectively. Hence we may form the sum of the squared differences between observed and theoretical values as

$$S(\alpha, \beta) = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2.$$

$S(\alpha, \beta)$ is a function of α and β; consequently, if it is to be minimized, we require that

$$\frac{\partial S}{\partial \alpha} = \frac{\partial S}{\partial \beta} = 0.$$

The required estimates a, b corresponding to α and β respectively will thus be the solutions of the two linear equations found by equating

$$\frac{\partial S}{\partial \alpha} = -2\sum_{i=1}^{n}\left(y_i - \alpha - \beta x_i\right)$$

and $$\frac{\partial S}{\partial \beta} = -2\sum_{i=1}^{n} x_i\left(y_i - \alpha - \beta x_i\right)$$

to zero. Note that x_i and y_i are not variables, but are observed numbers. Simplifying the resulting equations, we end up with the result

$$\sum_{i=1}^{n} y_i = na + b\sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} x_i\, y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2$$

This gives a line $f(x) = a + bx$ which estimates the true line $F(x) = \alpha + \beta x$. The above equations are known as normal equations for a straight line.

Ex. 1 : Fit up a straight line to the numerical data

x :	2	4	6	8	10	12
y :	7.32	8.24	9.20	10.19	11.01	12.05

We form the table

	x_i	x_i^2	y_i	$x_i\, y_i$
	2	4	7.32	14.64
	4	16	8.24	32.96
	6	36	9.20	55.20
	8	64	1.096	81.52
	10	100	11.01	110.10
	12	144	12.05	144.60
Sum	42	364	58.01	439.02

The normal equations for a straight line $f(x) = a + bx$, then become

$$6a + 42b = 58.01$$

$$42a + 364b = 439.02$$

Solving we get $a = 6.3733333$, $b = .4707143$. Consequently we have fitted a straight line to the given data as

$$f(x) = 6.3733333 + 0.4707143x$$

and this line may reasonably be used as an estimator of the true value $F(x)$ for any x in the range $2 \le x \le 12$.

4.8.2 Least-squares estimator for a parabola

Given the data (x_i, y_i) for $i = 1, 2, \ldots, n$ we shall find the least-squares estimator for the true curve

$$f(x) = \alpha + \beta x + \gamma x^2$$

For the i^{th} observation, the y – deviation between observed and theoretical values is

$$y_i - \alpha - \beta x_i - \gamma x_i^2$$

and the sum of the squared deviations is

$$S = \sum_{i=1}^{n} \left(y_i - \alpha - \beta x_i - \gamma x_i^2\right)^2.$$

Then
$$\frac{\partial S}{\partial \alpha} = -2 \sum_{i=1}^{n} \left(y_i - \alpha - \beta x_i - \gamma x_i^2\right),$$

$$\frac{\partial S}{\partial \beta} = -2 \sum_{i=1}^{n} x_i \left(y_i - \alpha - \beta x_i - \gamma x_i^2\right),$$

$$\frac{\partial S}{\partial \gamma} = -2 \sum_{i=1}^{n} x_i^2 \left(y_i - \alpha - \beta x_i - \gamma x_i^2\right).$$

Equating these three partial derivatives to zero, we obtain a, b and c (the estimates of α, β, γ) as solutions of the equations

$$\Sigma y = na + b\Sigma x + c\Sigma x^2,$$
$$\Sigma xy = a\Sigma x + b\Sigma x^2 + c\Sigma x^3,$$
$$\Sigma x^2 y = a\Sigma x^2 + b\Sigma x^3 + c\Sigma x^4$$

called the normal equations for a parabola and where in, for simplicity we omit the range of summation.

Ex. 2 : Fit up a parabola to the following data

x :	2	4	6	8	10
y :	3.07	12.85	31.47	57.38	91.29

In this case, since there is an odd number of items, we greatly simplify matters by introducing $X = (x - 6)/2$. Then we can form

	X	y	Xy	X^2y
	–2	3.07	–6.14	12.38
	–1	12.85	–12.85	12.85
	0	31.47	0.00	0.00
	1	57.38	57.38	57.38
	2	91.29	182.58	365.16
Sum	0	196.06	220.97	447.67

The normal equations are now given as :

$$\Sigma y = na + b\Sigma X + c\Sigma X^2,$$
$$\Sigma Xy = a\Sigma X + b\Sigma X^2 + c\Sigma X^3,$$
$$\Sigma X^2 y = a\Sigma X^2 + b\Sigma X^3 + c\Sigma X^4.$$

Nothing $n = 5$, $\Sigma X^2 = 10$, $\Sigma X^3 = 0$, $\Sigma X^4 = 34$ these equations become

$$196.06 = 5a + 10c,$$
$$220.97 = 10b,$$
$$447.67 = 10a + 34c$$

Solving, we get $a = 31.276286$, $b = 22.097$, $c = 3.967857$ and our estimator is

$$f(X) = 31.276286 + 22.097X + 3.967857X^2.$$

In terms of the original variables, this becomes

$$f(x) = 0.695999 - 0.855071x + 0.991964x^2.$$

4.8.3 Least-squares estimator for the curve $F(x) = \alpha x^m$, where m is a fixed number

Given data (x_i, y_i) for $i = 1, 2, \ldots, n$ we shall find the least-squares estimator for the true curve

$$F(x) = ax^m$$

where m is a fixed number.

In this problem, we have only one parameter α; we readily find

$$S(\alpha) = \sum_{i=1}^{n} \left(y_i - \alpha x_i^m\right)^2,$$

$$\frac{\partial S}{\partial \alpha} = -2 \sum_{i=1}^{n} x_i^m \left(y_i - \alpha x_i^m\right).$$

Equating $\frac{\partial S}{\partial \alpha}$ to zero, we obtain the least-squares estimate a for α as

$$a = \frac{\sum_{i=1}^{n} x_i^m y_i}{\sum_{i=1}^{n} x_i^{2m}}$$

Other reasonable estimates of a exist; however they are not least-squares estimates. for example one might compute y_i/x_i^m for each of the n items of the data and then compute

$$a_1 = \frac{1}{n} \sum_{i=1}^{n} \frac{y_i}{x_i^m}.$$

A different estimator a_2 can be found by taking logarithms of the given data. Instead of a theoretical curve

$$F(x) = \alpha x^m$$

consider

$$\log F(x) = \log \alpha + m \log x$$

This can be written as

$$F^*(x) = \alpha^* + mx^*$$

where $F^*(x) = \log F(x)$, $a^* = \log \alpha$ and $x^* = \log x$.

We take the data in the form (x_i^*, y_i^*). The least-squares estimate of α^* is immediately found to be

$$\alpha^* = \frac{1}{n}\left[\sum_{i=1}^{n} y_i^* - m \sum_{i=1}^{n} x_i^*\right]$$

and our estimator a_2 of α is found from the relation

$$\log a_2 = a^*.$$

Ex. 3 : Apply the three methods of estimation of α, given the following data and the prescribed curve $F(x) = \alpha x^2$.

x_i :	2	4	6	8	10
y_i :	0.973	3.839	8.641	15.987	23.794

It is easy to verify

$$\sum_{i=1}^{5} x_i^4 = 15664, \quad \sum_{i=1}^{5} x_i^2 y_i = 3778.96$$

$$a = \frac{\sum_{i=1}^{5} x_i^2 y_i}{\sum_{i=1}^{5} x_i^4} = \frac{3778.96}{15664} = 0.24125$$

$$a_1 = \frac{1}{5}\sum_{i=1}^{n} \frac{y_i}{x_i^2} = \frac{1.210952}{5} = 0.24219$$

The estimate a_2 requires a table

x_i^*	y_i^*
0.30103	1.98811
0.60206	0.58422
0.77815	0.93656
0.90309	1.20377
1.00000	1.37647

Here we use logarithms to base 10, although any base would serve. Then

$$\sum_{i=1}^{5} y_i^* = 4.08913, \sum_{i=1}^{5} x_i^* = 3.58433,$$

$$a^* = -0.61591$$

Taking its antilogarithm, we have

$$a_2 = 0.24214.$$

4.8.4 Computation of the minimal sum of squared deviations

For the curve $f(x) = a + bx$, we note that

$$m = \sum_{i=1}^{n} \left(y_i - a - bx_i\right)^2$$

$$m = \sum_{i=1}^{n} \left(y_i - a - bx_i\right) y_i - a \sum_{i=1}^{n} \left(y_i - a - bx_i\right)$$

$$- b \sum_{i=1}^{n} x_i \left(y_i - a - bx_i\right).$$

The second and the third terms vanish, in view of the normal equations of the straight line; then

$$m = \sum_{i=1}^{n} y_i - a \sum_{i=1}^{n} y_i - b \sum_{i=1}^{n} x_i y_i .$$

Ex. 4 : Determine the minimal sum of the squared deviations for the data given in Ex. 1 to find a straight line.

In Ex. 1, a and b have been found to be 6.3733333, .4707143 respectively to fit a straight line $f(x) = a + bx$.

Also we have

$$\sum_{i=1}^{6} y_i = 58.01, \sum_{i=1}^{6} x_i y_i = 439.02.$$

A bit of calculation gives further

$$\sum_{i=1}^{6} y_i^2 = 576.3787$$

The minimal sum of squared deviations for a straight line $f(x) = a + bx$ is

$$m = \sum_{i=1}^{6} y_i^2 - a \sum_{i=1}^{6} y_i - b \sum_{i=1}^{6} x_i y_i$$

$$= 576.3787 - 6.3733333\ (58.01) - 439.02\ (.4707143)$$

$$= .00864328.$$

4.9 SUMMARY

In this unit, you have obtained formula for the remainder terms in each of the interpolation formulae Newton forward, backward, Lagrange, Stirling's and Bessel, and determined the accuracy of these formulae for the given data.

The least squares estimators for linear, parabola and a curve of the form αx^m are obtained.

4.10 SAMPLE EXAMINATION QUESTIONS

I. Answer the following questions in detail.

1. Find $\log_{10} \sin 37' 22''$ given

$\log \sin 37' = 8.0319195 - 10$

$\log \sin 38' = 8.0435009 - 10$

$\log \sin 39' = 8.0547814 - 10$

$\log \sin 40' = 8.0657763 - 10$

$\log \sin 41' = 8.0764997 - 10$

$\log \sin 42' = 8.0869646 - 10$

$\log \sin 43' = 8.0971832 - 10$

Estimate the error using Newton's forward interpolation formula.

[Ans. 8.0363956 – 10; $0.3 \times .0000001$]

2. Find cos 0.806595 by Stirling's formula given

cos .8050 = .693111235

cos .8055 = .692750733

cos .8060 = .692390058

cos .8065 = .692029210

cos .8070 = .691668188

cos .8075 = .691306994

cos .8080 = .690945627

Estimate the Error. [0.691960629; $.015 \times 0.000000001$]

3. Using sin (0.1) = .09983 and sin (0.2) = .19867 find an approximate vlue of sin (0.15) by Lagrange's interpolation. Obtain the maximum absolute error.

[0.14925; 0.00025]

4. Determine the stepsize h to be used in the tabulation of $f(x) = \sin x$ in the interval [1, 3] so that linear interpolation will be correct to four decimal places.

[$h \leq 0.02$]

5. Find by least-squares approximation the equation of the straight line which comes nearest to passing through the following points :

x :	0.5	1.0	1.5	2.0	2.5	3.0
y :	0.31	0.82	1.29	1.85	2.51	3.02

[$y = -0.285 + 1.096x$]

6. Find the least-squares approximation of second degree for the discrete data

x :	–2	–1	0	1	2
$f(x)$:	15	1	1	3	19

$\left[y = \frac{1}{35}(-37 + 35x + 155x^2)\right]$

7. Find by the method of least-squares a formula of the form $y = a + bx^2$ which will fit the following data.

x :	20	24	29	36	43
y :	2100	2980	4310	6600	9360

[$94.87 + 5.0132x^2$)

8. The indicated horse power, I, required to drive a ship of displacement D tons at ten-knot speed is given by the following data. Find a formula of the from $I = a\,D^n$ which will fit the data.

D :	1720	2300	3200	4100
I :	655	789	1000	1164

[$I = 4.473\ D^{0.6691}$]

9. Compute the minimal sum of squared deviations in Q5. [0.01688].

UNIT–5 : INVERSE INTERPOLATION

Contents

5.1 AIMS AND OBJECTIVES

After going through this unit you will be able to : (i) state the problem of inverse interpolation, (ii) solve the problem of inverse interpolation by using Lagrange interpolation formula, sucessive approximation and by the method of reversion of series, (iii) solve an algebriac equation by the method of invese interpolation.

5.2 INTRODUCTION

Suppose that we are given a table of functional values; then the problem of direct interpolation, which we have so far been considering, can be phrased briefly as "Given x, find u_x". The problem of inverse interpolation takes the form "Given u_x, find x". In otherwords inverse interpolation is the process of finding the value of the argument corresponding to a given value of the function when the latter is intermediate between two tabulated values.

For certain functions, the problem of inverse interpolation does not offer any difficulty. For example when $y = \sin x$, $x = \sin^{-1} y$. Similarly when $y = \log_e x$, $x = e^y$. In such examples the required values of x can be computed easily on substituting the values of y. In other cases, especially when the corresponding numerical values of x and u_x are given, the problem of inverse interpolation offers difficulty and can be solved by several methods, most important of them being

(i) Use of Lagrange's formula

(ii) Method of successive approximations or iteration

(iii) Method of reversion of series

5.3 BY LAGRANGE'S INTERPOLATION FORMULA

When the values of x are at unequal invervals, the most obvious way of performing the process of inverse interpolation is by interchanging x and y or u_x in Lagrange's interpolation formula given by eq. (2) of 2.5. When this is done, we obtain eq. (4) of 2.5 and quote it below for our ready reference.

$$x = \frac{(y-y_1)(y-y_2)\cdots(y-y_n)}{(y_0-y_1)(y_0-y_2)\cdots(y_0-y_n)} x_0 + \cdots$$

$$+ \frac{(y-y_0)(y-y_1)\cdots(y-y_{n-1})}{(y_n-y_0)(y_n-y_1)\cdots(y_n-y_{n-1})} x_n \qquad \ldots (1)$$

We used the above formula in Ex. 2 of the same article. However, we shall yet illustrate another example, below to apply Lagrange's formula inversely.

Ex. 1 : Apply Lagrange's formula inversely to find, to one decimal place, the age for which the annuity value is 13.6, given the following table.

Age :	x	30	35	40	45	50
Annuity value (of $4\frac{1}{2}\%$) :	u_x	15.9	14.9	14.1	13.3	12.5

Here $x_0 = 30, x_1 = 35, x_2 = 40, x_3 = 45, x_4 = 50.$

$y_0 = 15.9, y_1 = 14.9, y_2 = 14.1, y_3 = 13.3, y_4 = 12.5.$

Applying Lagrange's formula of eq. (1) above giving x as a function of y, we obtain x when $y = 13.6$.

$$x = \frac{(13.6-14.9)(13.6-14.1)(13.6-13.3)(13.6-12.5)}{(15.9-14.9)(15.9-14.1)(15.9-13.3)(15.9-12.5)} \times 30$$

$$+ \frac{(13.6-15.9)(13.6-14.1)(13.6-13.3)(13.6-12.5)}{(14.9-15.9)(14.9-14.1)(14.9-13.3)(14.9-12.5)} \times 35$$

$$+ \frac{(13.6-15.9)(13.6-14.9)(13.6-13.3)(13.6-12.5)}{(14.1-15.9)(14.1-14.9)(14.1-13.3)(14.1-12.5)} \times 40$$

$$+ \frac{(13.6-15.9)(13.6-14.9)(13.6-14.1)(13.6-12.5)}{(13.3-15.9)(13.3-14.9)(13.3-14.1)(13.3-12.5)} \times 45$$

$$+ \frac{(13.6-15.9)(13.6-14.9)(13.6-14.1)(13.6-12.5)}{(12.5-15.9)(12.5-14.9)(12.5-14.1)(12.5-13.3)} \times 50$$

$$= 40.1$$

5.4 BY SUCCESSIVE APPROXIMATIONS OR ITERATION

We start with Newton's forward difference formula written as

$$y_u = y_0 + u\,\Delta y_0 + \frac{u(u-1)}{2}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{6}\Delta^3 y_0 + \ldots$$

From this we obtain

$$u = \frac{1}{\Delta y_0}\left[y_u - y_0 - \frac{u(u-1)}{2}\Delta^2 y_0 - \frac{u(u-1)(u-2)}{6}\Delta^3 y_0 + \ldots\right]$$

Neglecting the second and higher differences, we obtain the first approximation to u and write this as

$$u_1 = \frac{1}{\Delta y_0}\left(y_u - y_0\right)$$

Next, we obtain the second approximation to u by including the term containing the second differences. Thus

$$u_2 = \frac{1}{\Delta y_0}\left[y_u - y_0 - \frac{u_1\,(u_1 - 1)}{2}\Delta^2 y_0\right],$$

where we have used the value of u_1 for u in the coefficient of $\Delta^2 y_0$. Similarly we obtain

$$u_3 = \frac{1}{\Delta y_0}\left[y_u - y_0 - \frac{u_2\,(u_2 - 1)}{2}\Delta^2 y_0 - \frac{u_2\,(u_2 - 1)\,(u_2 - 2)}{6}\Delta^3 y_0\right]$$

and so on. This process should be continued till two successive approximations to u agree with each other to the required accuracy. The method is illustrated by means of the following example.

Ex. 2 : Tabulated $y = x^3$ for $x = 2, 3, 4$ and 5. Calculate the cube root of 10 correct to three decimal places.

Here $y_u = 10$, $y_0 = 8$, $\Delta y_0 = 19$, $\Delta^2 y_0 = 18$ and $\Delta^3 y_0 = 6$. The successive approximations to u are therefore

$$u_1 = \frac{1}{19}(2) = 0.1$$

$$u_2 = \frac{1}{19}\left[2 - \frac{0.1\,(0.1 - 1)}{2} \times 18\right] = 0.1$$

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
2	8			
		19		
3	27		18	
		37		6
4	64		24	
		61		
5	125			

$$u_3 = \frac{1}{19}\left[2 - \frac{0.15\,(0.15 - 1)}{2}(18) - \frac{0.15\,(0.15 - 1)\,(0.15 - 2)}{6}(6)\right]$$

$$= 0.1532.$$

$$u_3 = \frac{1}{19}\left[2 - \frac{0.1532\,(0.1532 - 1)}{2}(18) - \frac{0.1532\,(0.1532 - 1)\,(0.1532 - 2)}{6}(6)\right]$$

$$= 0.1541$$

$$u_5 = \frac{1}{19}\left[2 - \frac{0.1541\,(0.1541 - 1)}{2}(18) - \frac{0.1541\,(0.1541 - 1)\,(0.1541 - 2)}{6}(6)\right]$$

$$= 0.1542.$$

We therefore take $u = 0.154$ correct to three decimal places. Hence the value of x (which corresponds to $y = 10$), i.e., the cube root of 10 is given by $x_0 + uh = 2 + (.154)\ 1 = 2.154$

Ex. 3 : Given a table of values of the probability integral $\left(\frac{2}{\sqrt{\pi}}\right)\int_0^x e^{-x^2}\,dx$, for what value of x is this integral equal to $\frac{1}{2}$?

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0.45	0.4754818				
		.0091737			
0.46	0.48446555		– .0000840		
		.0090897		– .0000011	
0.47	0.4937452		– .0000851		.0000001
		.0090046		– .0000010	
0.48	.5027498		–.0000861		.0000002
		.0089185		– .0000008	
0.49	0.5116683		– .0000869		
		.0088316			
0.50	0.5204999				

For this problem we use the Bessel's formula. Inspection shows that the desired value of x lies between 0.47 and 0.48.

We take $x_0 = 0.47$, $h = .01$, $v = \frac{1}{2} = 0.5$

Substituting in Bessel's formula

$$y = \frac{y_0 + y_1}{2} + v\,\Delta y_0 + \frac{\left(v^2 - \frac{1}{4}\right)}{2}\,\frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \frac{v\left(v^2 - \frac{1}{4}\right)}{3!}\,\Delta^3 y_{-1},$$

We get $$0.5 = 0.4982475 + 0.0090046\,v + \frac{\left(v^2 - 0.25\right)}{2}(-0.0000856) + \frac{v\left(v^2 - 0.25\right)}{2}(-.0.0000010).$$

Transposing and dividing throughout by 0.0090046, we get

$$v = 0.194623 - \left(v^2 - 0.25\right)(-0.004753) - v\left(v^2 - 0.25\right)(-0.000185) \quad \ldots (2)$$

A first approximation for v is obtained by neglecting all terms beyond the first in the right-hand memeber of eq. (2). Hence

$$v_1 = 0.194623.$$

Substituting this for v in the right-hand member of eq. (2), we find the second approximation to be

$$v_2 = 0.194623 - \left[(0.194623)^2 - 0.25\right](-0.004753)$$
$$- 0.194623\left[(0.194623)^2 - 0.25\right](-0.0000185)$$
$$= 0.194623 - 0.001008 - 0.000001 = 0.193614.$$

Now substituting this value for v in the right-hand member of eq. (2), we find

$$v_3 = 0.194623 - 0.0010101 - 0.000001 = 0.193612.$$

This value differs only slightly from the proceeding and we therefore make no further approximation.

Since $u = v + \frac{1}{2}$ and, $x = x_0 + hu$ we have $u = 0.693612$,

$$x = 0.47 + 0.01\,(0.693612) = 0.47693612$$

This value is correct to six decimal places.

5.5 BY THE METHOD OF REVERSION OF SERIES

The most obvious method of solving the problem of invese interpolation is by reversion of series; for all the interpolation formulas thus so far developed are in the form of a power series and any covergent power series can be reverted. Thus the power series

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n + \ldots \quad \ldots (3)$$

when reverted becomes

$$x = \left(\frac{y - a_0}{a_1}\right) + c_1\left(\frac{y - a_0}{a_1}\right)^2 + c_2\left(\frac{y - a_0}{a_1}\right)^3 + c_3\left(\frac{y - a_0}{a_1}\right)^4 + \ldots$$
$$+ c_{n-1}\left(\frac{y - a_0}{a_1}\right)^n + \ldots \quad \ldots (4)$$

where

$$\left.\begin{aligned} c_1 &= -\frac{a_2}{a_1}, \\ c_2 &= -\frac{a_3}{a_1} + 2\left(\frac{a_2}{a_1}\right)^2, \\ c_3 &= -\frac{a_4}{a_1} + 5\left(\frac{a_2 a_3}{a_1^2}\right) - 5\left(\frac{a_2}{a_1}\right)^3, \\ c_4 &= -\frac{a_5}{a_1} + 6\frac{a_2 a_4}{a_1^2} + 3\left(\frac{a_3}{a_1}\right)^2, \\ &-21\frac{a_2^2 a_3}{a_1^3} + 14\left(\frac{a_2}{a_1}\right)^4, \text{ etc.} \end{aligned}\right\} \quad (5)$$

When reverting a series with numerical coefficients, it is better to compute c's from eqs. (5) and substitute thier values in eq. (4).

We shall now write Newton's, Stirling's and Bessel's formulas in the form of power series and then write down the values of $a_0, a_1, \ldots, a_4$ in each case. We stop with fourth differences, but the reader will have no difficulty in extending them to higher differences if necessary.

(a) Newton's forward formula

$$y = y_0 + u\,\Delta y_0 + \frac{u(u-1)}{2}\Delta^2 y_0 + \frac{u(u-1)(u-2)}{6}\Delta^3 y_0 + \frac{u(u-1)(u-2)(u-3)}{24}\Delta^4 y_0$$

$$= y_0 + \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4}\right)u + \left(\frac{\Delta^2 y_0}{2} - \frac{\Delta^3 y_0}{3} + 11\,\frac{\Delta^4 y_0}{24}\right)u^2 + \left(\frac{\Delta^3 y_0}{6} - \frac{\Delta^4 y_0}{4}\right)u^3 + \frac{\Delta^4 y_0}{24}u^4.$$

Here $a_0 = y_0$,

$a_1 = \Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4}$,

$a_2 = \frac{\Delta^2 y_0}{2} - \frac{\Delta^3 y_0}{3} + 11\,\frac{\Delta^4 y_0}{24}$,

$a_3 = \frac{\Delta^3 y_0}{6} - \frac{\Delta^4 y_0}{4}$

$a_4 = \frac{\Delta^4 y_0}{24}$

(b) Stirling's formula

$$y = y_0 + u\,\frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{u^2}{2}\Delta^2 y_{-1} + \frac{u(u^2-1)}{6}\,\frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \frac{u^2(u^2-1)}{24}\Delta^4 y_{-2}$$

$$= y_0 + \left(\frac{\Delta y_{-1} + \Delta y_0}{2} - \frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{12}\right)u + \left(\frac{\Delta^2 y_{-1}}{2} - \frac{\Delta^4 y_{-2}}{24}\right)u^2 + \left(\frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{12}\right)u^3 + \frac{\Delta^4 y_{-2}}{24}u^4.$$

Here $a_0 = y_0$,

$$a_1 = \frac{1}{2}\left(\Delta y_{-1} + \Delta y_0\right) - \frac{1}{12}\left(\Delta^3 y_{-2} - \Delta^3 y_{-1}\right),$$

$$a_2 = \frac{1}{2}\Delta^2 y_{-1} - \frac{1}{24}\Delta^4 y_{-2},$$

$$a_3 = \frac{1}{12}\left(\Delta^3 y_{-2} - \Delta^3 y_{-1}\right),$$

$$a_4 = \frac{1}{24}\Delta^4 y_{-2}.$$

(c) Bessel's formula

$$y = \frac{y_0 + y_1}{2} + v\Delta y_0 + \frac{v^2 - \frac{1}{4}}{2}\,\frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2}$$

$$+ \frac{v\left(v^2 - \frac{1}{4}\right)}{6}\Delta^3 y_{-1}$$

$$+ \frac{\left(v^2 - \frac{1}{4}\right)\left(v^2 - \frac{9}{4}\right)}{24}\,\frac{\Delta^4 y_{-2} + \Delta^4 y_{-1}}{2}$$

$$= \frac{1}{2}\left(y_0 + y_1\right) - \frac{1}{16}\left(\Delta^2 y_{-1} + \Delta^2 y_0\right)$$

$$+ \frac{3}{256}\left(\Delta^4 y_{-2} + \Delta^4 y_{-1}\right) + \left(\Delta y_0 - \frac{1}{24}\Delta^3 y_{-1}\right)v +$$

$$\left[\frac{1}{4}\left(\Delta^2 y_{-1} + \Delta^2 y_0\right) - \frac{5}{96}\left(\Delta^4 y_{-2} + \Delta^4 y_{-1}\right)\right]v^2$$

$$+ \frac{1}{16}\Delta^3 y_{-1}\,v^3 + \frac{1}{48}\left(\Delta^4 y_{-2} + \Delta^4 y_{-1}\right)v^4.$$

Here we have

$$a_0 = \frac{1}{2}\left(y_0 + y_1\right) - \frac{1}{16}\left(\Delta^2 y_{-1} + \Delta^2 y_0\right)$$

$$+ \frac{3}{256}\left(\Delta^4 y_{-2} + \Delta^4 y_{-1}\right)$$

$$a_1 = \Delta y_0 - \frac{1}{24}\Delta^3 y_{-1},$$

$$a_2 = \frac{1}{4}\left(\Delta^2 y_{-1} + \Delta^2 y_0\right) - \frac{5}{96}\left(\Delta^4 y_{-2} + \Delta^4 y_{-1}\right)$$

$$a_3 = \frac{1}{16}\Delta^3 y_{-1},$$

$$a_4 = \frac{1}{48}\left(\Delta^4 y_{-2} + \Delta^4 y_{-1}\right).$$

Ex. 4 : If $\sin hx = 62$, find x by reversion of series, using the following data.

x	$y = \sin hx$	Δy	$\Delta^2 y$	$\Delta^3 y$
4.80	60.7511			
		0.6106		
4.81	61.3617		0.0062	
		0.6168		0
4.82	61.9785		0.0062	
		0.6230		0
4.83	62.6015		0.0062	
		0.6292		
4.84	63.2307			

To solve the problem by reversion of series we use Stirling's formula. Here

$$a_0 = y_0 = 61.9785,$$

$$a_1 = 0.6199$$

$$a_2 = \frac{.0062}{2} = .0031,$$

$$a_3 = a_4 = 0.$$

Since $y = 62$, we have

$$y - a_0 = 62 - 61.9785 = 0.0215$$

$$\therefore \frac{y - a_0}{a_1} = \frac{0.0215}{06199} = 0.034683$$

$$\frac{a_2}{a_1} = \frac{0.0031}{0.6199} = 0.005001.$$

Hence, $c_1 = -.005001,$

$$c_2 = 2(.005001)^2 = 0.00005002$$

$c_3 = 0$ Practically.

Therefore, $u = 0.034683 - 0.005001\ (0.034683)^2 = 0.0347$

$$\therefore x = 4.82 + 0.01\ (0.0347) = 4.8203.$$

5.6 ROOTS OF AN ALGEBRAIC EQUATION BY INVERSE INTERPOLATION

Suppose that a real root of $f(x) = 0$ has been isolated, then $f(x)$ may be tabulated using a finite interval in the neighbourhood of the root and represented closely by a central difference formula.

This produces a polynomial equation in which the terms of higher degree have very small influence. Consequently an accurate approximation to the required root can be found and this approximation can be improved by an interative procedure. The method is illustrated by the following example.

Ex. 5 : Find by the method of inverse interpolation the real root of the equation $x^3 + x - 3 = 0$ which lies between 1.2 and 1.3.

x	X	$y = (x^2+x-3)$	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1	–0.2	–1				
			0.431			
1.1	–0.1	–0.569		0.066		
			0.497		0.006	
1.2	0	–0.072		0.072		0
			0.569		0.006	
1.3	0.1	0.497		0.078		
			0.647			
1.4	0.2	1.144				

The root of the equation is $1.2 + 0.1\,u$ where u is to be determined.

Take the origin at 1.2; interval of difference is 0.1 and using Stiriling's formula viz.,

$$y = y_0 + u\,\frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{u^2}{2}\,\Delta^2 y_{-1} + \frac{u\,(u^2-1)}{6}\,\frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2}$$

we get
$$0 = -0.072 + u\,\frac{.569 + .497}{2} + \frac{u^2}{2}\,(0.072) + \frac{u\,(u^2-1)}{6}\,(.006)$$

$$= -.072 + .532\,u + .036\,u^2 + .001\,u^3$$

This cubic equation can be solved by successive approximations and can be written as

$$u = \frac{.072}{.532} - \frac{.036}{.532}\,u^2 - \frac{.001}{.532}\,u^3. \qquad \ldots (6)$$

First approximation of u is

$$u_1 = \frac{.072}{.532} = 0.1353$$

For second approximation, substitute this value of u in eq. (6), we get

$$u_2 = .1353 - .0677\,(.1353)^2 - \frac{.001}{.532}\,(.1353)^3$$

$$= .134 \text{ approx.}$$

$$\therefore \text{Real root} = 1.2 + 0.1 \times 0.134 = 1.2134.$$

5.7 SUMMARY

Inverse interpolation is the process of finding the value of the argument corresponding to a given value of the function when the latter is intermediate between two tabulated values. We have solved the problem of inverse interpolation by using Lagrange's formula, by sucessive approximations and by the method of reversion series.

We have also illustrated a method of finding a real root of an algebriac equation using inverse interpolation process.

5.8 SAMPLE EXAMINATION QUESTIONS

1. Apply Lagrange's formula inversely to find, to one decimal place, the value of x when u_x is 0.163 given

x :	80	82	84	86	88
u_x :	0.134	0.154	0.176	0.200	0.227

[Ans : $x = 82.8$]

2. Given cosh $x = 1.285$ find x by inverse interpolation (use the method of successive approximations for Bessle's formula), using the following data

x :	0.736	0.737	0.738	0.739	0.740	0.741
$y = \cosh x$:	1.2832971	1.2841023	1.2849085	1.2857179	1.2865247	1.2873348

[Ans : $x = 0.738110$]

3. Solve the equation $x = \log_{10} x$, given the following data

x :	1.35	1.36	1.37	1.38
$\log_{10} x$:	0.1303	0.1335	0.1367	0.1399

[Ans : $x = 1.37136$]

4. Find by the method of inverse interpolation (using Stirling's formula) the real root of the equation $x^3 + x^2 - 7x + 1 = 0$ which lies between 2.1 and 2.2.

[Ans : $x = 2.1027751$]

BLOCK - 2 : SOLUTIONS OF EQUATIONS

Introduction

You have learnt in the lower classes, how to solve literal equations upto the fourth degree. However, we haven't come across the solving of equations like $my + n \log y = l$, $c e^{-x} + d \tan x = 8$ *etc. These type of equations are called transcendental equations and no general method exists for finding their roots in terms of their coefficients. However, when the coefficients of such equations are pure numbers, it is always possible to compute the roots to any desired degree of accuracy.*

In unit 6, we will be discussing some of the useful methods for finding the roots of any equation having numerical coefficients. In unit-7, we discuss various methods for numerical solution of simultaneous linear equations. Unit-8 deals with the solution of difference equations.

Unit - 6 : Solutions of Algebraic and Transcendental Equations

Unit - 7 : Numerical solution of Simultaneous Linear Equations

Unit - 8 : Difference Equations

UNIT–6 : SOLUTIONS OF ALGEBRAIC AND TRANSCENDENTAL EQUATIONS

Contents

6.1 AIMS AND OBJECTIVES

After going through this unit you will be able to (i) find an approximate real root of an equation with single unknown, i.e., $f(x) = 0$ using (a) Bisection method, (b) Regula Falsi, (c) Newton – Rephson and (d) Iteration methods (ii) interpret each of these methods geometrically, (iii) extend Newton – Raphson and Iteration methods to obtain a pair of real roots of two simultaneous equations in two unkwons $\phi(x, y) = 0$; $\psi(x, y) = 0$.

6.2 INTRODUCTION

The first non-linear equation encountered in algebra courses is usually the quadratic equation

$$ax^2 + bx + c = 0,$$ and all of you are familiar with the formula for its roots :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

The formula for the roots of a general cubic is some what more complicated and that for a general quartic usually takes several pages to describe. It is already proved that there is no such formula for general polynomials of degree higher than four. Accordingly, except in special cases, we prefer in practice to use a numerical method to solve polynomial equation of degree higher than two.

Another class of non-linear equations consists of those which involve transcendental functions such as e^x, $\log x$, $\sin x$ and $\tan x$. Analytic solutions of such equations are rare and hence we are forced to use numerical methods.

6.3 LOCATION OF ROOTS

In finding the real roots of a numerical equation $f(x) = 0$, whether algebraic or transcendental, it is necessary first to find an approximate value of the root from a graph or otherwise. Hence $f(x)$ can be a polynomial of any degree or an expression involving transcendental functions like $1 + \cos x - 5x$; $x \tan x - \cosh x$, $e^{-x} - \sin x$ etc. If we take a set of rectangular co-ordinate axes and plot the graph of $y = f(x)$, it is evident that the abscissas of the points where the graph crosses the x-axis are the real roots of the given equation, for at these points y is 0 and so $f(x) = 0$. Approximate values for the real roots of any numerical equation can therefore be found from the graph of the given equation. It is not necessary, however, to draw the complete graph. Only the portions in the neighbourhood of the points where it crosses the x-axis are needed.

A very useful theorem which determines the location of a real root of given numerical equation and also forms a basis of the bisection method, as we shall see later in this unit, is stated below.

"If $f(x)$ is continuous from $x = a$ to $x = b$ and if $f(a)$ and $f(b)$ have opposite signs, then there is at least one real root between a and b".

The theorem is evident from an inspection of the figure 1 below, for if $f(a)$ and $f(b)$ have opposite signs, the graph must cross the x-axis at least once between $x = a$ and $x = b$.

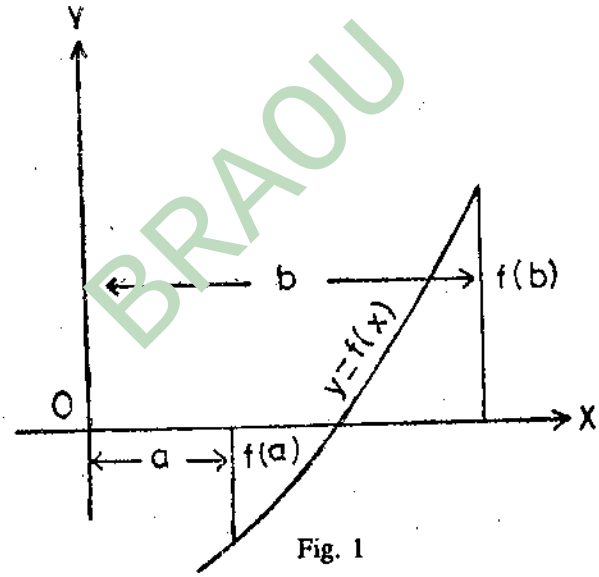


Fig. 1

In most cases the approximate values of the real roots of $f(x) = 0$ are most easily found by writing the equation in the form

$$f_1(x) = f_2(x)$$

and then plotting on the same axes the two equations

$$y_1 = f_1(x), y_2 = f_2(x).$$

The abscissas of the points of intersection of these two curves are the real roots of the given equation, for at these points $y_1 = y_2$ and therefore $f_1(x) = f_2(x)$. Consequently $f(x) = 0$ is satisfied.

Ex. 1 : Find the approximate value of the root of $3x - \cos x - 1 = 0$. We write this equation in the form

$$3x - 1 = \cos x$$

Plot the graphs of $y_1 = 3x - 1$ and $y_2 = \cos x$ on the same set of co-ordinate axes as in figure 2 below. The abscissa of the point of intersection of the graphs of these equations is seen to be about 0.6.

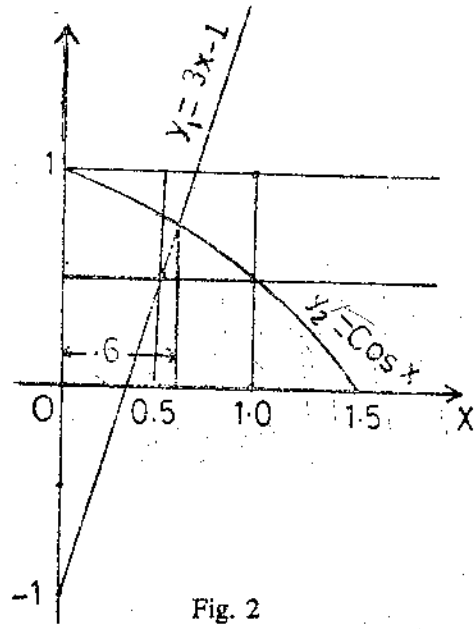


Fig. 2

By applying the above theorem repeatedly we get the real roots of a given y desired degree of accuracy. This is done by the following exam

Ex. 2 : Compute the real root of the equation $x \log_{10} x - 1.2 = 0$. We shall write the given equation in the form

$$f(x) = x \log_{10} x - 1.2 = 0$$

Since $f(2) = -.6$ and $f(3) = 0.23$ a[illegible] therefore have opposite signs, a root of the given equation lies between 2 and 3 according to the theorem. We tabulate $f(x)$ as below

x	$f(x)$
2	–0.60
3	0.23
2.7	–0.035
2.8	0.052
2.74	–0.00056
2.75	0.0082
2.740	–0.00056
2.741	0.00031
2.7406	–0.000040
2.7407	0.000047

The last pair or values shows that the root is about halfway between 2.7406 and 2.7407 or 2.740645.

This method of computing roots is not rapid, but it is simple and can be applied to any type of numerical equation.

6.4 THE BISECTION METHOD

In our theorem for definiteness, let $f(a)$ be negative and $f(b)$ be positive. Then the root lies between a and b and let its approximate value be given by $x_0 = \frac{a+b}{2}$. If $f(x_0) = 0$, we conclude that x_0 is a root of the equation $f(x) = 0$. Otherwise, the root lies either between x_0 and b or between x_0 and a depending on whether $f(x_0)$ is negative or positive. Then, as before, we bisect the interval and repeat the process until the root is known to the desired accuracy. The method is shown graphically in the figure below.

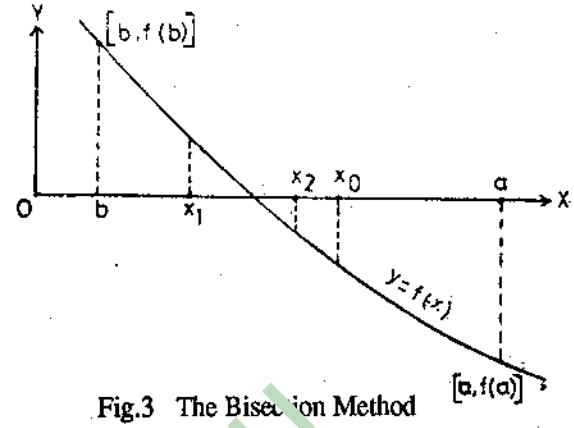


Fig.3 The Bisection Method

Ex. 3 : Find a real root of the equation $x^3 - x - 1 = 0$ by bisection method.

Write the given equation in the form

$$f(x) = x^3 - x - 1 = 0$$

Since $f(1)$ is negative and $f(2)$ is positive, a root lies between 1 and 2 and therefore we take $x_0 = \frac{3}{2}$. Then

$$f(x_0) = \frac{7}{8} \text{ which is positive.}$$

Hence the root lies between 1 and 1.5 and we obtain

$$x_1 = \frac{1 + 1.5}{2} = 1.25$$

We find $f(x_1) = \frac{-19}{64}$, which is negative. We therefore conclude that the root lies between 1.25 and 1.5. It follows that

$$x_2 = \frac{1.25 + 1.5}{2} = 1.375$$

The procedure is repeated and the successive approximations are

$x_3 = 1.3125$, $x_4 = 1.34375$, $x_5 = 1.328125$ etc.

6.5 THE REGULA FALSI METHOD

This is an oldest method for finding the real root of an equation and closely resembles the bisection method. In this method we choose two points x_0 and x_1 such that $f(x_0)$ and $f(x_1)$ are of opposite sings. Since the graph of $y = f(x)$ crosses the x-axis between these two points, a root must

lie in between these points. Now the equation of the chord joining the two points $\left[x_0, f\left(x_0\right)\right]$ and $\left[x_1, f\left(x_1\right)\right]$ is

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

Geometrically the method consists in replacing the part of the curve between the points $\left[x_0, f\left(x_0\right)\right]$ and $\left[x_1, f\left(x_1\right)\right]$ by means of the chord joining these points and taking the point of intersection of the chord with the x-axis as an approximation to the root. The point of intersection in the present case is given by putting $y = 0$ in eq. (1). Thus we obtain

$$x = x_0 - \frac{f\left(x_0\right)}{f\left(x_1\right) - f\left(x_0\right)}\left(x_1 - x_0\right)$$

Hence the second approximation to the root of $f(x) = 0$ is given by

$$x_2 = x_0 - \frac{f\left(x_0\right)}{f\left(x_1\right) - f\left(x_0\right)}\left(x_1 - x_0\right) \qquad \ldots (2)$$

If now $f\left(x_2\right)$ and $f\left(x_0\right)$ are opposite signs, then the root lies between x_0 and x_2 and we replace x_1 by x_2 in eq. (2) and obtain the next approximation. We replace, otherwise, x_0 by x_2 and generate the next approximation. The procedure is repeated till the root is obtained to the desired accuracy. Figure 4 below, gives a graphical representation of the method.

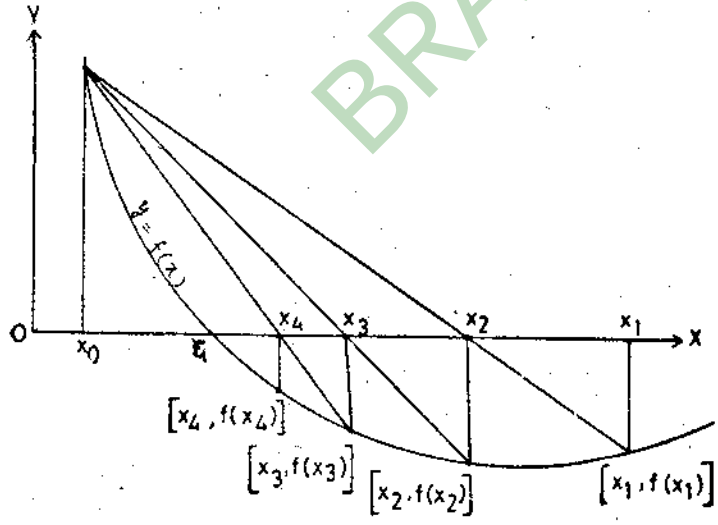


Fig. 4 The Regula-Falsi Method

Ex. 4 : Find a real root of Walli's equation $x^3 - 2x - 5 = 0$

We write the given equation in the form $f(x) = x^3 - 2x - 5 = 0$

We observe $f(2) = -1$ and $f(3) = 16$ and hence a root lies between 2 and 3. Equation (2) then gives

$$x_2 = 2 + \frac{1}{17} = 2.059$$

Now $f(x_2) = -0.386$ and hence the root lies between 2.059 and 3 using eq. (2) once again, we obtain

$$x_3 = 2.059 + \frac{0.386}{16.386}(3 - 2.059) = 2.0812$$

Repeating the process, we obtain successively

$$x_4 = 2.0904,$$

$$x_5 = 2.0934 \text{ etc.}$$

The correct value is 2.0945 ..., so that x_5 is correct to two decimal places only. It is clear that the process is very slow and is therefore unsuitable for hand computation. This example was used by Wallis in 1685 to illustate the Newton-Raphson method (see below) and the real root of this equation is now known atleast to 100 decimal places.

6.6 THE NEWTON–RAPHSON METHOD

This method is generally used to improve the result obtained by one of the previous methods. let x_0 be an approximate root of $f(x) = 0$ and let $x_1 = x_0 + h$ be the correct root so that $f(x_1) = 0$ and let $x_1 = x_0 + h$ be the correct root so that $f(x_1) = 0$. Expanding $f(x_0 + h)$ by Taylor's series, we obtain

$$f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \ldots = 0$$

Neglecting the second and higher order derivatives, we have

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

which gives $$h = -\frac{f(x_0)}{f'(x_0)}.$$

A better approximation than x_0 is therefore given by x_1 where

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Successive approximations are given by $x_2, x_3, \ldots, x_{n+1}$ where

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

which is the Newton-Raphson formula.

Geometrically, the method consists in replacing the part of the curve between the point $[x_0, f(x_0)]$ and the x-axis by means of the tangent to the curve at that point and is described graphically in figure 5.

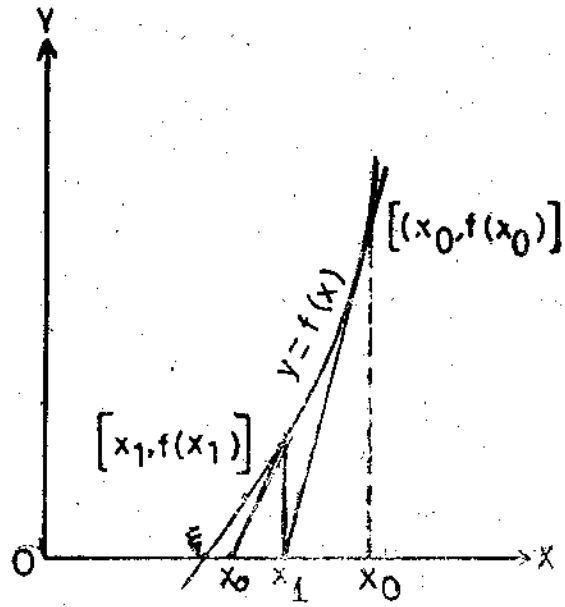


Fig. 5 The Newton–Raphson Method

Ex. 5 : Apply Newton–Raphson's method to find a root of the equation $x^3 - 3x - 5 = 0$.

Here $f(x) = x^3 - 3x - 5$ and $f'(x) = 3x^2 - 3$.

The Newton–Raphson formula gives

$$x_{n+1} = x_n - \frac{x_n^2 - 3x_n - 5}{3x_n^2 - 3}.$$

Since $f(2) = -3$ and $f(3) = 13$, a root lies between 2 and 3. We choose $x_0 = 3$ and obtain succesively

$$x_1 = 3 - \frac{13}{24} = 2.46,$$

$$x_2 = 2.295, \; x_3 = 2.279, \; x_4 = 2.279.$$

6.7 THE METHOD OF ITERATION

When a numerical equation $f(x) = 0$ can be expressed in a form $x = \phi(x)$, the real roots can be found by the process of Iteration. We first find from a graph or otherwise an approximate value of x_0 of the desired root. A better aproximation $x^{(1)}$ is given by the equation

$$x^{(1)} = \phi(x_0)$$

Then the succeeding approximations are

$$x^{(2)} = \phi\left(x^{(1)}\right),$$

$$x^{(3)} = \phi\left(x^{(2)}\right)$$

.....

$$x^{(n)} = \phi\left(x^{(n-1)}\right).$$

When $|\phi(x)| < 1$, the successive approximations converge. In fact the smaller the value of $\phi'(x)$, the more rapid is the convergence. Its proof is ommitted here, being beyond the scope of our

present reading. However care must be taken in writing $f(x) = 0$ in the form $x = \phi(x)$ for in some forms the process will not converge at all.

It is instructive to look at the geometric picture of the Iteration process. For simplicity we denote the successive approximations to the root by $x_0, x_1, x_2, \ldots, x_n$. Then the relations

$$x_1 = \phi(x_0),\ x_2 = \phi(x_1), x_3 = \phi(x_2) \text{ etc.,}$$

can be pictured as points by the following geometric construction.

Draw the graphs $y_1 = x$ and $y_2 = \phi(x)$ as shown in figure 6. Since $|\phi'(x)| < 1$ for convergence, the inclination of the curve $y_2 = \phi(x)$ must be less than 45° in the neighbourhood of x_0. This fact has been observed in constructing the graph.

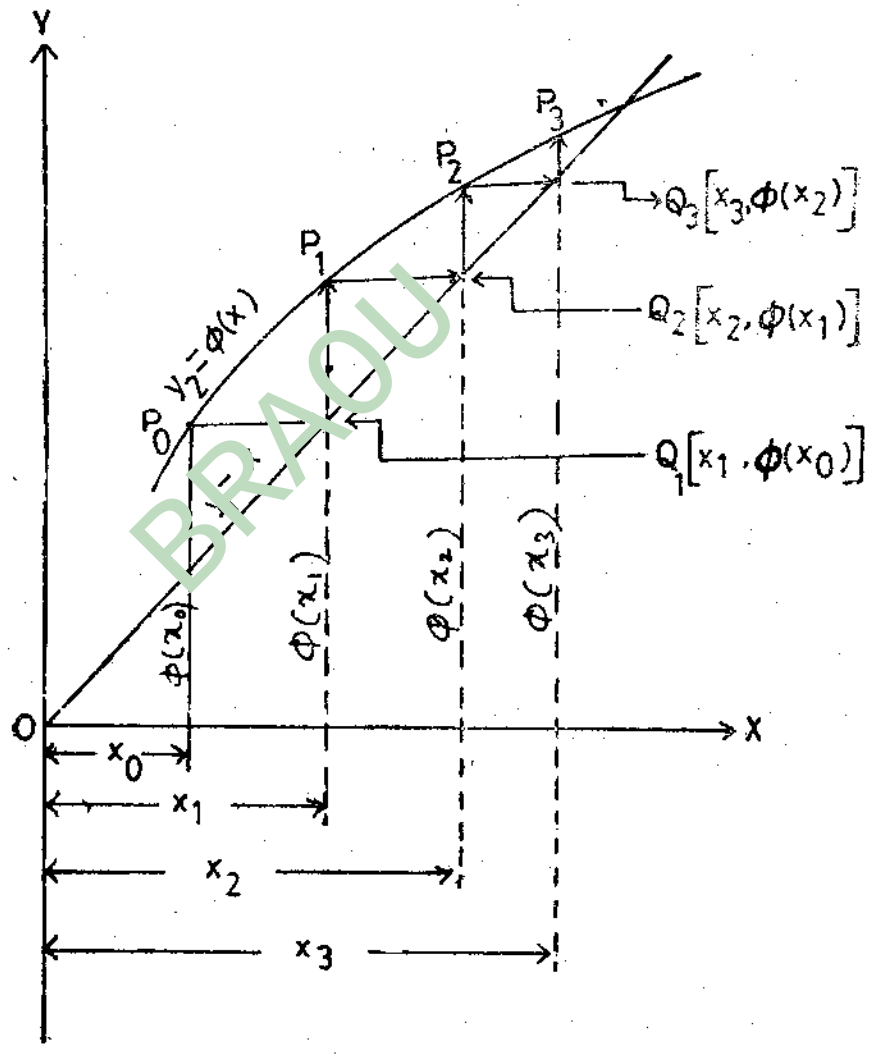


Fig. 6 The Iteration Method

Now to trace the convergence of the Iteration process, draw the ordinate $\phi(x_0)$. Then from the Point P_0 draw a line parallel to OX until it intersects the line $y_1 = x$ at the point $Q_1[x_1, \phi(x_0)]$. Note that this point Q_1 is the geometric representation of the first iteration equation $x_1 = \phi(x_0)$. Then draw $Q_1 P_1$, $P_1 Q_2$, $Q_2 P_2$, $P_2 Q_3$ etc; as indicated by the arrows in fig. 6. The points Q_1, Q_2, Q_3... thus approach the point of intersection of the curves $y_1 = x$ and $y_2 = \phi(x)$ as the iteration proceeds.

Note that the co-ordinates of these Q's satsify the corresponding Iteration equations.

The reader should draw a curve $y_1 = \phi(x)$ with inclination greater than 45° in the neighbourhood of x_0 and then proceed with the construction as outlined above. He will find that the points Q_1, Q_2 ... etc., recede farther and farther from the intersection point of graphs and that the successive approximations x_1, x_2, etc., get worse as the iteration proceeds.

Ex. 6 : Find by the method of iteration a real root of

$$2x - \log_{10} x = 7$$

The given equation can be written in the form

$$x = \frac{1}{2}\left(\log_{10} x + 7\right)$$

We find from the intersection of the graphs $y_1 = 2x - 7$ and $y_2 = \log_{10} x$ that an approximate value of the root is 3.8. Hence we have

$$x^{(1)} = \frac{1}{2}\left(\log_{10} 3.8 + 7\right) = 3.79,$$

$$x^{(2)} = \frac{1}{2}\left(\log_{10} 3.79 + 7\right) = 3.7893,$$

$$x^{(3)} = \frac{1}{2}\left(\log_{10} 3.7893 + 7\right) = 3.7893$$

Since $x^{(3)}$ is the same as $x^{(2)}$, we do not repeat the process but take 3.7893 as the correct result to four figures.

6.8 SIMULTANEOUS EQUATIONS IN SEVERAL UNKNOWNS

The real roots of simultaneous algebraic and transcendental equations in several unkowns can be found eighter by the Newton-Raphson method or by the method of iteration. We shall give an outline of each method for the case of two unkowns. The reader will have no difficulty in extending both methods to the case of any number of unknowns.

(i) Newton-Raphson Method : Let (x_0, y_0) be an initial apprxomation to the root of the system

$$\phi(x, y) = 0,$$

$$\psi(x, y) = 0.$$

If $(x_0 + h, y_0 + k)$ is the root of the system, then we must have,

$$\phi(x_0 + h, y_0 + k) = 0,$$

$$\psi(x_0 + h, y_0 + k) = 0,$$

By Taylor's series, we obtain

$$\phi_0 + h\left(\frac{\partial \phi}{\partial x_0}\right) + k\left(\frac{\partial \phi}{\partial y_0}\right) + \ldots = 0,$$

$$\psi_0 + h\left(\frac{\partial \psi}{\partial x_0}\right) + k\left(\frac{\partial \psi}{\partial y_0}\right) + \ldots = 0$$

$$\text{Here } \frac{\partial \phi}{\partial x_0} = \left[\frac{\partial \phi}{\partial x}\right]_{x = x_0,\, y = y_0},$$

$$\phi_0 \equiv \phi\left(x_0, y_0\right) \text{ etc.}$$

Neglecting the second and higher order terms we obtain the system of linear equations.

$$h\left(\frac{\partial \phi}{\partial x}\right)_0 + k\left(\frac{\partial \phi}{\partial y}\right)_0 = -\phi_0,$$

$$\text{and } h\left(\frac{\partial \psi}{\partial x}\right)_0 + k\left(\frac{\partial \psi}{\partial y}\right)_0 = -\psi_0,$$

which can be readily solved for h and k. We have thus obtained a new apporximation given by

$$x_1 = x_0 + h \text{ and } y_1 = y_0 + k.$$

The process is to be repeated till we obtain the value to the desired accuracy.

(ii) The Method of Iteration : We assume the system of equations

$$\phi(x, y) = 0,$$

$$\psi(x, y) = 0$$

may be written in the form

$$x = F(x, y)$$

$$\text{and } y = G(x, y)$$

where F and G satisfy the conditions of convergence

$$\left|\frac{\partial F}{\partial x}\right| + \left|\frac{\partial G}{\partial x}\right| < 1$$

$$\text{and } \left|\frac{\partial F}{\partial y}\right| + \left|\frac{\partial G}{\partial y}\right| < 1.$$

Let $\left(x_0, y_0\right)$ be the initial approximation. Then the sequence of successive approximations is

$$x_1 = F\left(x_0, y_0\right), \quad y_1 = G\left(x_1, y_0\right)$$

$$x_2 = F\left(x_1, y_1\right), \quad y_2 = G\left(x_2, y_1\right)$$

$$x_3 = F\left(x_2, y_2\right), \quad y_3 = G\left(x_3, y_2\right) \text{ etc.}$$

The sequence is continued until the values for x and y converge to the required accuracy.

Ex. 7 : Find by Newton-Raphson method a real root of equations $x^2 - y^2 = 4x^2 + y^2 = 16$, given that the first approximation is $x_0 = y_0 = 2.828$

Here we have $\phi = x^2 - y^2 - 4$ and $\psi = x^2 + y^2 - 16$;

so that $\frac{\partial \phi}{\partial x} = 2x, \frac{\partial \phi}{\partial y} = -2y,$

$$\frac{\partial \psi}{\partial x} = 2x, \quad \frac{\partial \psi}{\partial y} = 2y,$$

The system of linear equations of Newton-Raphson method become

$$h - k = 0.707$$

$$\text{and } h + k = 0$$

$$\text{so that } h = -k = 0.354.$$

Hence the second approximation to the root is given by

$$x_1 = x_0 + h = 3.182,$$

$$\text{and } y_1 = y_0 + k = 2.474.$$

Replacing (x_0, y_0) by (x_1, y_1) and repeating the above process, we obtain $h = -.0204$ and $k = -0.0242$ and therefore $x_2 = 3.162$ and $y_2 = 2.450$.

Ex. 8 : Compute by the method of iteration a real solution of the equations

$$x + 3 \log_{10} x - y^2 = 0,$$

$$2x^2 - xy - 5x + 1 = 0,$$

given $x_0 = 3.4$, $y_0 = 2.2$ as first approximation.

We rewrite the above equations in the form

$$\left.\begin{aligned} x &= y^2 - 3 \log_{10} x, \\ y &= \frac{1}{x} + 2x - 5 \end{aligned}\right\} \quad \ldots (A)$$

Then we have $x^{(1)} = (2.2)^2 - 3 \log_{10} 3.4 = 3.25,$

$$y^{(1)} = \frac{1}{3.25} + 2(3.25) - 5 = 1.81,$$

$$x^{(2)} = (1.81)^2 - 3 \log_{10} (3.25) = 1.74,$$

$$y^{(2)} = \frac{1}{1.74} + 2(1.74) - 5 = 0.95.$$

The value of x and y are evidently getting worse with each application of the iteration process. We must therefore write the given equations in some other form before attempting the iteration process.

We rewrite the given equations in the form

$$\left.\begin{aligned} x &= \sqrt{\frac{x(y+5)-1}{2}}, \\ y &= \sqrt{x + 3 \log_{10} x} \end{aligned}\right\} \quad \ldots (B)$$

Then the successive approximations are

$$x^{(1)} = \sqrt{\frac{3.4(2.2+5) - 1}{2}} = 3.426,$$

$$y^{(1)} = \sqrt{3.426 + 3 \log_{10} 3.426} = 2.243,$$

$$x^{(2)} = \sqrt{\frac{3.426\,(2.243+5)-1}{2}} = 3.451,$$

$$y^{(2)} = \sqrt{3.451 + 3\log_{10} 3.451} = 2.2505$$

$$x^{(3)} = 3.466, \quad y^{(3)} = 2.255.$$

$$x^{(4)} = 3.475, \quad y^{(4)} = 2.258,$$

$$x^{(5)} = 3.480, \quad y^{(5)} = 2.259,$$

$$x^{(6)} = 3.483, \quad y^{(6)} = 2.260.$$

This example points out that one should be careful in writing the given equations in the form (A) or (B). If they are written in the form (A), instead of improving the roots at each step, we make them decidedly worse. On the other hand if they are written in the form (B), we not only improve the roots at each step but also obtain them to the desired accuracy.

6.9 SUMMARY

Various methods for solving algebraic and transcendental equations have been discussed. The bisection method is not rapid but is simple and reliable and is applicable to any type of numerical equation. It is preferable to the regula falsi method. From the formula of Newton-Raphson method, it is evident that the larger the derivative $f'(x)$ the smaller is the correction which must be applied to get the correct value of the root. This means that when the graph is nearly vertical where it crosses the x-axis, the correct value of the root can be found with great rapidity and very little labour and the Newton-Raphson's method should not be used when the graph of $f(x)$ is nearly horizontal. In such cases the regula falsi method should be used. When the given equation $f(x) = 0$ can be expressed in the form $x = \phi(x)$, we use the method of iteration.

6.10 SAMPLE EXAMINATION QUESTIONS

1. Find graphically the approximate value of a real root of the equation $2x - \log_{10} x = 7$.
2. Obtain a root of the equation $x^3 + x^2 + x + 7 = 0$ by the bisection method correct to three decimal places.
3. Obtain a root of the equation $x^3 - 4x - 9 = 0$ by the method of False position correct to three decimal places.
4. Find to six decimal places by the Newton-Raphson method a real root of $2x - 3\sin x - 5 = 0$.
5. Solve $\sin x = 10\,(x - 1)$ by the iteration process.
6. Find a real solution of

$$4.2\,x^2 + 8.8y^2 = 1.42,$$

$$(x - 1.2)^2 + (y - 0.6)^2 = 1$$

by Newton-Raphson method.

7. Find to five decimal places a solution of

$$\sin x = y - 1.32,$$

$$\cos y = x - 0.85$$

by the iteration process.

8. Find the smallest root of

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

Answers

1. 3.7893
2. −2.1049
3. 2.7065
4. 2.883238
5. 1.088
6. $x = 0.22684$, $y = 0.36962$
7. $x = 0.567325$, $y = 1.857378$
8. 1.44575.

UNIT–7 : NUMERICAL SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

Contents

7.1 AIMS AND OBJECTIVES

After going through this unit you will be able to ( ) Solve a given system of linear equations by using various methods like Cramers method, Matrix method and Gauss - Jordan (direct methods) Jacobi and Gauss - Seidel methods (inderect methods).

7.2 INTRODUCTION

Various methods for the numerical solution of a system of simultaneous linear equations have been discussed. Some methods are more of general nature, while the others are some what restricted in their application. Here we assumed that the students are familiar with the elementary properties of determinats and matrices.

7.3 CRAMERS METHOD

A simple method of solving simultaneous linear equations by determinants was discussed by a Swiss mathematician Gabriel Cramer (1704-1752).

To derive the Cramer's rule, we consider a system of equations

$$a_1 x + b_1 y + c_1 z = d_1$$

$$a_2 x + b_2 y + c_2 z = d_2$$

$$a_3 x + b_3 y + c_3 z = d_3 \qquad \ldots (1)$$

If the determinant of coefficients be

$$\Delta = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

then

$$x\Delta = \begin{vmatrix} xa_1 & b_1 & c_1 \\ xa_2 & b_2 & c_2 \\ xa_3 & b_3 & c_3 \end{vmatrix}$$

Operating $C_1 + y\,C_2 + z\,C_3$, we get

$$x\Delta = \begin{vmatrix} a_1x + b_1y + c_1z & b_1 & c_1 \\ a_2x + b_2y + c_2z & b_2 & c_2 \\ a_3x + b_3y + c_3z & b_3 & c_3 \end{vmatrix}$$

$$= \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix} \text{ by (1)}$$

Thus

$$x = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix} \div \Delta = \frac{\Delta_1}{\Delta} \quad \text{provided } \Delta \neq 0 \qquad \ldots (2)$$

Similarly

$$y = \begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix} \div \Delta = \frac{\Delta_2}{\Delta}, \qquad \ldots (3)$$

and

$$z = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix} \div \Delta = \frac{\Delta_3}{\Delta}, \qquad \ldots (4)$$

Note :

(i) Cramer's rule gives a unique solution of the equations (1) only when $\Delta \neq 0$.

(ii) If $\Delta = 0$ and any of the numerator determinants $\Delta_1, \Delta_2, \Delta_3 \neq 0$, the given equations (1) do not have any solution.

(iii) If $\Delta = 0$ and each $\Delta_1, \Delta_2, \Delta_3$ is also zero, then the given equations have infinite number of solutions.

Ex .1 : Apply Cramer's rule to solve the equations

$$3x + y + 2z = 3$$
$$2x - 3y - z = -3$$
$$x + 2y + z = 4$$

Here

$$\Delta = \begin{vmatrix} 3 & 1 & 2 \\ 2 & -3 & -1 \\ 1 & 2 & 1 \end{vmatrix} = 3(-3+2) - 2(1-4) + (-1+6) = 8$$

$$\therefore x = \frac{\Delta_1}{\Delta_2} = \frac{1}{\Delta}\begin{vmatrix} 3 & 1 & 2 \\ -3 & -3 & -1 \\ 4 & 2 & 1 \end{vmatrix}$$

$$= \frac{1}{8}[3(-3+2) + 3(\quad -1) + 4(-1+6)] = 1$$

Similarly,

$$y = \frac{\Delta_2}{\Delta} = \frac{1}{\Delta}\begin{vmatrix} 3 & 3 & 2 \\ 2 & -3 & -1 \\ 1 & 4 & 1 \end{vmatrix} = 2$$

and

$$z = \frac{\Delta_3}{\Delta} = \frac{1}{\Delta}\begin{vmatrix} 3 & 1 & 3 \\ 2 & -3 & -3 \\ 1 & 2 & 4 \end{vmatrix} = -1$$

Hence $x = 1, y = 2, z = 1$.

Ex. 2 : Find the values of λ for which the equations

$$(\lambda - 1)x + (3\lambda + 1)y + 2\lambda z = 0$$
$$(\lambda - 1)x + (4\lambda - 2)y + (\lambda + 3)z = 0$$
$$2x + (3\lambda + 1)y + 3(\lambda - 1)z = 0$$

are consistant and find the ratios $x : y : z$ when λ has the smallest of these values. What happens when λ has the greater of these values?

The given equations are homogeneous and linear since $d_1 = d_2 = d_3 = 0$ and will be consistent if $\Delta = 0$ ie.,

$$\begin{vmatrix} \lambda - 1 & 3\lambda + 1 & 2\lambda \\ \lambda - 1 & 4\lambda - 2 & \lambda + 3 \\ 2 & 3\lambda + 1 & 3(\lambda - 1) \end{vmatrix} = 0 \left(\text{Operate } R_2 - R_1\right)$$

or if

$$\begin{vmatrix} \lambda - 1 & 3\lambda + 1 & 2\lambda \\ 0 & \lambda - 3 & 3 - \lambda \\ 2 & 3\lambda + 1 & 3(\lambda - 1) \end{vmatrix} = 0 \left(\text{Operate } C_3 + C_2\right)$$

or if

$$\begin{vmatrix} \lambda - 1 & 3\lambda + 1 & 5\lambda + 1 \\ 0 & \lambda - 3 & 0 \\ 2 & 3\lambda + 1 & 6\lambda - 2 \end{vmatrix} = 0 \left(\text{Expand by } R_2\right)$$

or if

$$(\lambda - 3) \begin{vmatrix} \lambda - 1 & 5\lambda + 1 \\ 2 & 2(3\lambda - 1) \end{vmatrix} = 0$$

or if $2(\lambda - 3)[(\lambda - 1)(3\lambda - 1) - (5\lambda + 1)] = 0$

or if $6\lambda(\lambda - 3)^2 = 0$

or $\lambda = 0$ or 3.

(i) when $\lambda = 0$, the given equations become

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

Solving the second and third of above equations, we get

$$\frac{x}{6 - 3} = \frac{y}{6 - 3} = \frac{z}{-1 + 4}$$

Hence $x = y = z$.

(ii) When $\lambda = 3$, the equations became identical.

Ex. 3 : Does the system

$$\begin{aligned} x - 2y + 3z &= 0 \\ 2x + y - 4z &= 0 \\ x - y + z &= 0 \end{aligned}$$

possess non-trivial solutions? If so, find them.

A solution of the above equations is obviously $x = y = z = 0$. Such a solution is called a trivial solution. Other solutions are called non-trival solutions. The condition for a system of homogeneous linear equations to possess non-trivial solutions is $\Delta = 0$. Now,

$$\Delta = \begin{vmatrix} 1 & -2 & 3 \\ 2 & 1 & -4 \\ 1 & -1 & 1 \end{vmatrix} \quad (\text{Operate } C_1 + C_2, C_3 + C_2)$$

$$= \begin{vmatrix} -1 & -2 & 1 \\ 3 & 1 & -3 \\ 0 & -1 & 0 \end{vmatrix} \quad (\text{Expand by } R_3)$$

$$= (-1)(-3) - 3 = 0.$$

Hence the given equations possess non-trivial solutions. To get these solutions, solving any two of these equations say,

$$2x + y - 4z = 0$$

and $$x - y + z = 0$$

we get $$\frac{x}{1-4} = \frac{y}{-4-2} = \frac{z}{-2-1}.$$

$\therefore x = z$ and $y = 2z$ which satisfy the first equation. By giving z various values, we obtain an infinite number of non-trivial solutions. We conclude this section with a few remarks. Although Cramer's rule is simple and easy to apply, its use requires a great deal of labour when the number of equations exceeds four or five, because of the labour in evaluating the determinants involved.

7.4 MATRIX METHOD

Consider the equations

$$\left.\begin{aligned} a_1 x + b_1 y + c_1 z &= d_1 \\ a_2 x + b_2 y + c_2 z &= d_2 \\ a_3 x + b_3 y + c_3 z &= d_3 \end{aligned}\right\}$$

If $$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \text{ and } D = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

then the equations (1) are equivalent to the matrix equation

$$AX = D, \qquad \ldots (2)$$

where A is the coefficient matrix.

Multiplying both sides of (2) by the reciprocal matrix A^{-1}, we get

$$A^{-1} AX = A^{-1} D$$

$$\text{or } IX = A^{-1} D \qquad [\because A^{-1} A = I]$$

$$\text{or } X = A^{-1} D$$

$$\text{ie., } \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \\ C_1 & C_2 & C_3 \end{bmatrix} \cdot \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \qquad \ldots (3)$$

where A_1, B_1, etc. are the co-factors of a_1, b_1, etc. in the determinant Δ or |A|, given by

$$\Delta = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

Hence equating the values of x, y, z to the corresponding elements in the product on the right hand side of (3), we get the desired solution.

Ex.4 : Solve the equations with the help of matrices

$$2x_1 - 2x_2 + 4x_3 = -12$$

$$2x_1 + 8x_2 + 2x_3 = 8$$

$$-x_1 + x_2 - x_3 = 7/2$$

Hence

$$\Delta = \begin{vmatrix} 2 & -2 & 4 \\ 2 & 8 & 2 \\ -1 & 1 & -1 \end{vmatrix} = 20$$

$$\therefore \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{20} \begin{bmatrix} -10 & 2 & -36 \\ 0 & 2 & 4 \\ 10 & 0 & 20 \end{bmatrix} \begin{bmatrix} -12 \\ 8 \\ 7/2 \end{bmatrix}$$

$$= \begin{bmatrix} -1/2 & 1/10 & -9/5 \\ 0 & 1/10 & 1/5 \\ 1/2 & 0 & 1 \end{bmatrix} \begin{bmatrix} -12 \\ 8 \\ 7/2 \end{bmatrix}$$

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

Hence $x_1 = 1/2$, $x_2 = 3/2$, $x_3 = 5/2$.

Ex. 5 : Solve the equations with the help of matrices

$$x + y + z = 6$$
$$x - y + 2z = 5$$
$$3x + y + z = 8$$

Here

$$\Delta = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -1 & 2 \\ 3 & 1 & 1 \end{vmatrix} = 6$$

$$\therefore \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -3 & 0 & 3 \\ 5 & -2 & -1 \\ 4 & 2 & -2 \end{bmatrix} \begin{bmatrix} 6 \\ 5 \\ 8 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

which gives

$x = 1$, $y = 2$, and $z = 3$.

7.5 GAUSS ELIMINATION AND ITS MODIFICATION (GAUSS JORDAN) METHODS

This is the elementary elimination method and it reduces the system of equations to an equivalent upper triangular system which can be solved by back substitution. Although quite general, we shall describe this method by considering a system of three equations for the sake of clarity and simplicity.

Let the system be

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}\right\} \quad \ldots (1)$$

We first form the augmented matrix of the above system.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \quad \ldots (2)$$

To eliminate x_1 from the second equation, we multiply the first equation by $-\frac{a_{21}}{a_{11}}$ and then add it to the second equation. Similarly, to eliminate x_1 from the third equation, we miltiply the first equation by $-\frac{a_{31}}{a_{11}}$ and add it to the third. The procedure can be shown thus :

$$\begin{matrix} \\ -a_{21}/a_{11} \\ -a_{31}/a_{11} \end{matrix}\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \qquad \ldots (3)$$

where $-\frac{a_{21}}{a_{11}}$ and $-\frac{a_{31}}{a_{11}}$ are called the miltipliers for the first stage of elimination. In this stage we have assumed $a_{11} \neq 0$. The first equation is called the pivotal equation and a_{11} is called the first pivot. At the end of the first stage, the augmented matrix (3) becomes

$$\begin{matrix} \\ \\ -\frac{a_{32}'}{a_{22}'} \end{matrix}\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}' & a_{23}' & b_2' \\ 0 & a_{32}' & a_{33}' & b_3' \end{bmatrix} \qquad \ldots (4)$$

where $a_{22}', a_{23}' \ldots$ are all changed elements. a_{22}' is the new pivot and the multiplier is $-\frac{a_{32}'}{a_{22}'}$.

At the end of second stage, we have the upper triangular system

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22}' & a_{23}' & b_2' \\ 0 & 0 & a_{33}'' & b_3'' \end{bmatrix} \qquad \ldots (5)$$

from which the values of x_1, x_2 and x_3 can be obtained by back substitution.

It is clear that the method will fail if one of the pivots a_{11}, a_{22}' or a_{33}'' vanishes. In such a case the method can be modified by rearranging the rows so that the pivot is non-zero. This procedure is called partial pivoting.

Instead of eliminating x_2 only in the third equation, we could have obtained it from the first equation also so that at the end of the second stage, the augmented matrix becomes

$$\begin{bmatrix} a_{11}' & 0 & a_{12}' & b_1' \\ 0 & a_{22}' & a_{23}' & b_2' \\ 0 & 0 & a_{33}'' & b_3'' \end{bmatrix} \qquad \ldots (6)$$

from which the values of x_1, x_2 and x_3 can be obtained directly without further computation. This modification of the Gaussian elimination is called *Gauss-Jordan method.*

Ex. 6 : Solve the following system

$$\begin{aligned} 2x + y + z &= 10 \\ 3x + 2y + 3z &= 18 \\ x + 4y + 9z &= 16 \end{aligned}$$

by (a) Gauss method and (b) Gauss-Jordan method.

a) Gauss method :

In the first stage the multipliers are $-\frac{3}{2}$ and $-\frac{1}{2}$. We multiply the first equation successively by $-\frac{3}{2}$ and $-\frac{1}{2}$ and add it to the second and third equations respectively, to obtain the equations.

$$\frac{1}{2}y + \frac{3}{2}z = 3$$

and

$$\frac{7}{2}y + \frac{17}{2}z = 11.$$

The augmented matrix therefore becomes

$$\begin{matrix} \\ -3/2 \\ -1/2 \end{matrix} \begin{bmatrix} 2 & 1 & 1 & 10 \\ 0 & 1/2 & 3/2 & 3 \\ 0 & 7/2 & 17/2 & 11 \end{bmatrix}$$

At the second stage, we eliminate y from the third equation by multiplying the second equation by -7 and adding it to the third. The resulting system will be upper triangular,

$$\begin{aligned} 2x + y + z &= 10 \\ \frac{y}{2} + \frac{3z}{2} &= 3 \\ -2z &= -10 \end{aligned}$$

Back substitution gives the solution

$x = 7, y = -9, z = 5.$

b) Gauss-Jordan method :

At the end of the first stage, we have as in (a) above

$$\begin{bmatrix} 2 & 1 & 1 & 10 \\ 0 & 1/2 & 3/2 & 3 \\ 0 & 7/2 & 17/2 & 11 \end{bmatrix}$$

Now, instead of eliminating y from the third equation we shall eliminate it from the first equation also. This reduces the system to

$$\begin{aligned} -7x + 0 \cdot y + 5z &= -24 \\ \frac{1}{2} \cdot y + \frac{3}{2}z &= 3 \\ -2z &= -10 \end{aligned}$$

The elimination is now trivial and gives the same result as before.

We shall now work two examples illustrating two types of iterative procedures called Jacobi method and Gauss Seidel method and finally consider how best to adapt an arbitrary system of equations of these methods.

7.6 JACOBI METHOD

Ex. 7 : Solve the system of equations

$$13x_1 + 5x_2 - 3x_3 + x_4 = 18,$$
$$2x_1 + 12x_2 + x_3 - 4x_4 = 13,$$
$$3x_1 - 4x_2 + 10x_3 + x_4 = 29,$$
$$2x_1 + x_2 - 3x_3 + 9x_4 = 31,$$

The first step is to use the successive equations to solve for each unknown in terms of the others.

Thus we obtain

$$x_1 = \frac{1}{13}\left(18 - 5x_2 + 3x_3 - x_4\right),$$
$$x_2 = \frac{1}{12}\left(13 - 2x_1 - x_3 + 4x_4\right),$$
$$x_3 = \frac{1}{10}\left(29 - 3x_1 + 4x_2 - x_4\right),$$
$$x_4 = \frac{1}{9}\left(31 - 2x_1 - x_2 + 3x_3\right)$$

Let us start with the trial solution $x_1 = x_2 = x_3 = x_4 = 0$ and substitute in the right-hand side of this system of equations. Then

$x_1 = 1.385, x_2 = 1.083, x_3 = 2.900, x_4 = 3.444.$

This set of values then becomes our second trial solution.

Substituting it on the right-hand side of our system of equations, we obtain,

$x_1 = 1.372, x_2 = 1.759, x_3 = 2.573, x_4 = 3.983.$

Continue this procedure at each stage, the answer obtained is used as the next trial solution for substitution. The remainder of the work can be arranged as

0.995	0.980	1.025	1.005	.996	1.001	1.001
1.968	1.952	1.981	2.001	1.999	1.998	2.000
2.794	3.009	2.993	2.984	3.000	3.002	2.999
3.802	3.936	4.013	3.994	3.993	4.001	4.001

Since it is easily checked that the exact answer is

$x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 4,$

we see that the iteration has finally converged to the solution.

7.7 GAUSS - SEIDEL METHOD

Ex. 8 : We solve the equations of Ex. 7 by this method. We arrange them in the form

$$x_1 = \frac{1}{13}\left(18 - 5x_2 + 3x_3 - x_4\right),$$

$$x_2 = \frac{1}{12}\left(13 - 2x_1 - x_3 + 4x_4\right),$$

$$x_3 = \frac{1}{10}\left(29 - 3x_1 + 4x_2 - x_4\right),$$

$$x_4 = \frac{1}{9}\left(31 - 2x_1 - x_2 + 3x_3\right)$$

We again start with the trial solution $x_1 = x_2 = x_3 = x_4 = 0$. However, this time we substitute only in the first equation to give $x_1 = 1.385$. We then substitute $x_1 = 1.385$, $x_2 = x_3 = x_4 = 0$ in the second equation to obtain $x_2 = .853$. Put $x_1 = 1.385$, $x_2 = .853$, $x_3 = x_4 = 0$ in the third equation to give $x_3 = 2.826$. Finally putting $x = 1.385$, $x_2 = .853$, $x_3 = 2.826$, $x_4 = 0$ in the fourth equation yields $x_4 = 3.984$.

We see that in this method of iteration, known as the Gauss – Seidel method, the result of any stage within a step is used in succeeding stages of the same step. The remaining steps of the iteration, proceeding from the second trial solution follow.

1.385	1.402	1.000	1.[illegible]12	1.000
0.853	1.942	1.9[illegible]	1.999	1.999
2.826	2.858	3.0[illegible]	2.996	3.000
3.984	3.870	4.0[illegible]4	3.996	4.000

c) *A method of arrangement of a system of equations for iteration by the above methods is illustrated below.*

Ex. 9 : Arrange the following system for iteration

$$3x_1 - 5x_2 + 47x_3 + 20x_4 = 18,$$

$$56x_1 + 23x_2 + 11x_3 - 19x_4 = 36,$$

$$12x_1 + 16x_2 + 17x_3 + 18x_4 = 25,$$

$$17x_1 + 65x_2 - 13x_3 + 7x_4 = 84.$$

We begin by writing down the matrix of coefficients in the form

3	–5	47	20
56	23	11	–19
12	16	17	18
17	65	–13	7

We must now select for large matrix elements with the property that only one is in each row and only one is in each column. If we proceed by always choosing the maximal entry, we start with 65 and deleting to find the corresponding minor, obtain

$$\begin{matrix} 3 & 47 & 20 \\ 56 & 11 & -19 \\ 12 & 17 & 18 \end{matrix}$$

we now choose 56, 47, and 18, in that order; so our pivoted coefficients become 65, 56, 47, 18. With the choice of these pivotal coefficients, the equations can be written as

$$\begin{aligned} \underline{65}x_2 + 17x_1 - 13x_3 + 7x_4 &= 84, \\ 23x_2 + \underline{56}x_1 + 11x_3 - 19x_4 &= 36, \\ -5x_2 + 3x_1 + \underline{47}x_3 + 20x_4 &= 18, \\ 16x_2 + 12x_1 + 17x_3 + \underline{18}x_4 &= 25. \end{aligned}$$

The iteration scheme then sets

$$\begin{aligned} x_2 &= \frac{1}{65}\left(84 - 17x_1 + 13x_3 - 7x_4\right), \\ x_1 &= \frac{1}{56}\left(36 - 23x_2 - 11x_3 + 19x_4\right), \\ x_3 &= \frac{1}{47}\left(18 + 5x_2 - 3x_1 - 20x_4\right), \\ x_4 &= \frac{1}{18}\left(25 - 16x_2 - 12x_1 - 17x_3\right). \end{aligned}$$

we are now prepared for.

7.8 SUMMARY

We have solved the system of simultaneous linear equations using some of the direct and indirect methods. The direct methods are Cramer's method and Matrix Inversion method and the Gauss method. The Cramer's rule is simple and easy to apply compared to other methods, but involves a great deal of labour if the number of equations exceeds four or five. For matrix inversion we have followed the cofactor method. Gauss method is an elimination method and we reduce the representative matrix to an upper triangular matrix which can be solved by back substitution. Gauss-Jordan method is a slight modification to Gauss method. Jacobi and Gauss-Seidel methods are iterative procedures and in these methods we use successive equations to solve for each unkown and with all the variables equated to zero as the initial solution. We make use of these solutions for the second iteration and continue this process till a final solution is obtained.

7.9 SAMPLE EXAMINATION QUESTIONS

1. Solve by using determinants

$$\begin{aligned} 3x - 4y + 5z &= 8, \\ x + 2y - 6z &= 7, \\ 2x - y + 5z &= 3, \end{aligned}$$

2. Find the values of λ for which the equations

$$(2-\lambda)x + 2y + 3 = 0,$$

$$2x + (4-\lambda)y + 7 = 0,$$

$$2x + 5y + (6-\lambda) = 0$$

are consistent and find the values of x and y corresponding to each of these values of λ.

3. Does this system

$$x + 3y + z = 0$$

$$9x + 7y + 3z = 0$$

$$55x + 5y + 7z = 0$$

possess non trivial solutions? If so, find these solutions.

4. Solve the following equations with the help of matrices

$$4x - y - 2z = 0$$

$$-x + 3y + z = 1$$

$$2x - 2y - 6z = 3.$$

5. In a given electrical net work, the equations for the currents i_1, i_2, i_3 are

$$3i_1 + i_2 + i_3 = 8,$$

$$2i_1 - 3i_2 - 2i_3 = -5$$

$$7i_1 + 2i_2 - 5i_3 = 0.$$

Solve these equations by matrix method.

6. Solve the system in Ex. 1 by (a) Gauss Elimination method and (b) Gauss-Jordan method.

7. Find the solution, to three decimals, of the system

$$83x + 11y - 4z = 95$$

$$7x + 52y + 13z = 104$$

$$3x + 8y + 29z = 71$$

using Jacobi and Gauss-Seidel methods.

8. Solve the system in Ex. 9 by the Jacobi iteration technique.

Answers

1. $x = 3.255, y = .545, z = -.818$
2. $\lambda = -1, 1, 12;$

   $x = \frac{-1}{11}, y = \frac{-15}{11},$

   $x = -5, y = 1,$

   $x = 1/2, y = 1.$
3. Yes : $z = -10x, z = \frac{-10y}{3}$
4. $x = -1/4, y = 1/2, z = -3/4.$
5. $i_1 = 3/2, i_2 = 1, i_3 = 5/2.$
7. $x = 1.06, y = 1.37, z = 1.96$
8. $x_2 = 1.6182, \quad x_1 = -.3788$

   $x_3 = 0.8241, \quad x_4 = -.5733$

UNIT–8 : DIFFERENCE EQUATIONS

Contents

8.1 AIMS AND OBJECTIVES

After going through this unit, (i) given a function and the interval of differencing, you will be able to formulate the difference equation, (ii) given a difference equation, you will be abel to solve the equation by finding the complimentary function and particular integral.

8.2 INTRODUCTION

One of the important problems in mathematics is to find the solution of a boundary value problem. Many methods have been developed but are all long and laborious. Certain types of boundary value problems can be solved by replacing the differential equation by the corresponding difference equation and solving the latter by iteration. In this unit, you will be learning how to find an exact solution of a difference equation. The numerical methods of solving a difference equation will not be discussed here as it would be above the level of the student.

8.3 FORMATION OF DIFFERENCE EQUATIONS

In calculus, integration is treated as the reverse process of differentiation. The analogous problem in finite differences is that given Δy to find y.

For example, if $\Delta y = 2x$ and the interval of differencing is h, we at once verify that

$$y = \frac{x^2}{h} - x + c.$$

Here, the contant c is not necessarily a constant for all values of x; it need merely be constant for the values of x under consideration, that is those separated by intevals of h.

Let us consider the orgin of differntial and difference equations.

Suppose a one-parameter family of curves $y = cx^2$ is given; then $\frac{dy}{dx} = 2cx$.

Eliminating the parameter leaves with the differential equation

$$\frac{dy}{dx} - 2\frac{y}{x} = 0 \qquad \ldots (1)$$

In the similar manner, we obtain the difference equation by assuming the interval of differencing $h = 1$.

$$y = cx^2.$$

then $$\Delta y = c\,(2x + 1).$$

Eliminating the parameter c, we obtain

$$\Delta y = \frac{y}{x^2}(2x + 1)$$

$$\text{or } \Delta y = \frac{2y}{x} + \frac{y}{x^2} \qquad \ldots (2)$$

Just as (1) is the differential equation corresponding to the one parameter system $y = cx^2$, (2) is the difference equation for the same one-parameter system.

Definition :

A difference equation or a recurrence relation is a relation between the differences of an unkown function at one or more general values of the argument.

$$\Delta y_{(n+1)} + y_{(n)} = 2 \qquad \ldots (1)$$

$$\text{and } \Delta y_{(n+1)} + \Delta^2 y_{(n-1)} = 1 \qquad \ldots (2)$$

are examples of difference equations.

Since $\Delta y_{(n+1)} = y_{(n+1)} - y_{(n+1)}$

(1) may be written as

$$y_{(n+2)} - y_{(n+1)} + y_{(n)} = 2 \qquad \ldots (3)$$

Also since $\Delta^2 y_{(n-1)} = y_{(n+1)} - 2y_{(n)} + y_{(n-1)}$

(2) takes the form

$$y_{(n+1)} - 2y_{(n)} + y_{(n-1)} = 1 \qquad \ldots (4)$$

Definition :

Order of a difference equation is the difference between the largest and the smallest arguments occuring in the difference equation divided by the unit of increment.

Thus (3) above is of the second order, for

$$\frac{\text{Largest argument} - \text{Smallest argument}}{\text{Unit of increment}} = \frac{(n + 2) - n}{1} = 2$$

and (4) is of the third order, for

$$\frac{(n + 2) - (n - 1)}{1} = 3.$$

Note : While finding the order of a difference equation, it must always be expressed in a form free of Δ^i for highest power of Δ does not give order of the difference equation.

Defintions :

Solution of a difference equation is an expression for $y_{(n)}$ which satsifies the given difference equation.

The general solution of a difference equation is that in which the number of arbitrary constants is equal to the order of the difference equation.

A particular solution or (particular integral) is that solution which is obtained from the general solution by giving particular values to the constants.

Ex. 1 : From $y_n = A.2^n + B(-3)^n$, derive a difference equation not containing the constants A and B.

$$\text{We have } y_n = A.2^n + B(-3)^n$$

$$y_{n+1} = 2.A.2^n - 3B(-3)^n$$

$$\text{and } y_{n+2} = 4.A.2^n + 9B(-3)^n.$$

Eliminating A and B, we get

$$\begin{vmatrix} y_n & 1 & 1 \\ y_{n+1} & 2 & -3 \\ y_{n+2} & 4 & 9 \end{vmatrix} = 0 \text{ or } y_{n+2} + y_{n+1} - 6y_n = 0$$

as the desired equation.

8.4 LINEAR DIFFERENCE EQUATIONS

Defintions :

A linear difference equation is that in which y_n, y_{n+1}, y_{n+2} etc. occur to the first degree only and are not multiplied together.

A linear difference equation with constant coefficients is of the form

$$y_{n+r} + a_1 y_{n+r-1} + a_2 y_{n+r-2} + \ldots + a_r y_n = f(n) \quad \ldots (1)$$

where $a_1, a_2, \ldots a_n$ are constants.

Now we shall deal with linear difference equations with constant coefficients only.

Elementary properties : If $u_1(n), u_2(n), \ldots, u_r(n)$ be r independent solutions of the equation

$$y_{n+r} + a_1 y_{n+r-1} + \ldots + a_r y_n = 0, \quad \ldots (2)$$

then its complete solution is

$$u_n = c_1 u_1(n) + c_2 u_2(n) + \ldots + c_r u_r(n)$$

where $c_1, c_2, \ldots, c_r$ are arbitrary constants.

If v_n is a particular solution of (1), then the complete solution of (1) is

$$y_n = u_n + v_n.$$

The part u_n is called the complimentary function (CF) and the part v_n is called the particular integral (PI) of (1).

Thus the complete solution (CS) of (1) is

$$y_n = \text{CF} + \text{PI}$$

8.4.1 Rules for finding the complementary function i.e., rules to find u_n of (2)

(i) Consider the first order linear equation

$$y_{n+1} - \lambda y_n = 0 \text{ where } \lambda \text{ is a constant}$$

Rewriting it as

$$\frac{y_{n+1}}{\lambda^{n+1}} - \frac{y_n}{\lambda^n} = 0$$

we have $\Delta\left(\frac{y_n}{\lambda^n}\right) = 0$

which gives $y_n/\lambda^n = c$, a constant.

Thus the solution of $(E - \lambda)\, y_n = 0$ is

$$y_n = c\,\lambda^n$$

(ii) Consider the general second order linear equation

$$y_{n+2} + a\, y_{n+1} + b y_n = 0$$

which is in symbolic form

$$\left(E^2 + aE + b\right) y_n = 0 \qquad \ldots (1)$$

we call $E^2 + aE + b = 0$

is the auxiliary equation.

Let its roots be λ_1, λ_2.

Case. 1 : If the roots are real and distinct, then (1) is equivalent to

$$\left(E-\lambda_1\right)\left(E-\lambda_2\right) y_n = 0 \qquad \ldots (2)$$

$$\text{or } \left(E-\lambda_2\right)\left(E-\lambda_1\right) y_n = 0 \qquad \ldots (3)$$

If y_n satsifies $\left(E - \lambda_1\right) y_n = 0$, then it satisfies (3). Similarly if y_n satsifies $\left(E-\lambda_2\right) y_n = 0$, then it satsifies (2). ∴ In order that y_n satisfies both (2) and (3), we solve

$$\left(E-\lambda_1\right) y_n = 0 \text{ and } \left(E-\lambda_2\right) y_n = 0.$$

The solutions are respectively

$$y_n = c_1\left(\lambda_1\right)^n, \text{ and } y_n = c_2\left(\lambda_2\right)^n.$$

where c_1 and c_2 are arbitrary constants.

Thus the general solution of (1) is

$$y_n = c_1\left(\lambda_1\right)^n + c_2\left(\lambda_2\right)^n.$$

Case II : If the roots are real and equal $(\text{i.e., } \lambda_1 = \lambda_2)$, then (2) becomes

$$(E - \lambda_1)^2 y_n = 0 \qquad \ldots (4)$$

$$\text{Let } y_n = (\lambda_1)^n z_n$$

where z_n is a new independent variable. Then (4) takes the form

$$(\lambda_1)^{n+2} z_{n+2} - 2\lambda_1 (\lambda_1)^{n+1} z_{n+1} + \lambda_1^2 (\lambda_1)^n z_n = 0$$

or $z_{n+2} - 2z_{n+1} + z_n = 0$

i.e., $\Delta^2 z_n = 0$

$$\therefore z_n = c_1 + c_2 n$$

where c_1, c_2 are arbitrary constants.

Thus the solution of (1) becomes

$$y_n = (c_1 + c_2 n)(\lambda_1)^n.$$

Case III : If the roots are imaginary $(\text{i.e., } \lambda_1 = \alpha + i\beta, \lambda_2 = \alpha - i\beta)$, then the solution of (1) is

$$y_n = c_1 (\alpha + i\beta)^n + c_2 (\alpha - i\beta)^n$$

Put $\alpha = r \cos\theta, \beta = r \sin\theta.$

Then

$$y_n = c_1 r^n (\cos n\theta + i \sin n\theta) + c_2 r^n (\cos n\theta - i \sin n\theta)$$

$$= r^n [A_1 \cos n\theta + A_2 \sin n\theta]$$

where A_1, A_2 are arbitrary constants and

$$r = \sqrt{\alpha^2 + \beta^2}, \theta = \tan^{-1}(\beta/\alpha)$$

Ex. 2 : Solve the difference equation

$$u_{n+3} - 2u_{n+2} - 5u_{n+1} + 6u_n = 0$$

The auxilary equation is

$$E^3 - 2E^2 - 5E + 6 = 0$$

$$\text{or } (E - 1)(E + 2)(E - 3) = 0$$

$$\therefore E = 1, -2, 3.$$

Thus the complete solution is

$$u_n = c_1 (1)^n + c_2 (-2)^n + c_3 (3)^n$$

Ex. 3 : Solve $u_{n+1} - 2u_{n+1} + u_n = 0$

The auxiliary equation is

$$E^2 - 2E + 1 = 0$$

$$\text{or } (E-1)^2 = 0$$

Thus the required solution is

$$u_n = (c_1 + c_2 n)(1)^n \text{ i.e., } u_n = c_1 + c_2 n.$$

Ex. 4 : Solve $y_{n+1} - 2y_n \cos\alpha + y_{n-1} = 0$

Its symbolic form is

$$(E^2 - 2\cos\alpha \cdot E + 1)\, y_{n-1} = 0$$

The auxiliary equation is

$$E^2 - 2E\cos\alpha + 1 = 0.$$

$$\therefore E = \frac{2\cos\alpha \pm \sqrt{4\cos^2\alpha - 4}}{2} = \cos\alpha \pm i\sin\alpha.$$

Thus the complete solution is

(Since $r = 1, \theta = \alpha$)

$$y_{n-1} = 1)^{n-} [c_1 \cos(n-1)\alpha + c_2 \sin(n-1)\alpha]$$

$$\text{or } y_n = {}_1 \cos n\alpha + c_2 \sin n\alpha.$$

8.5 METHODS FOR PARTICULAR INTEGRALS

Consider the equation $y_{n+r} + a_1 y_{n+r-1} + \ldots + a_r y_n = f(n)$

which is symbolic form is

where

$$\phi(E)\, y_n = f(n) \qquad \ldots (1)$$

$$\phi(E) = E^r + a_1 E^{r-1} + \ldots + a_r.$$

Then the particular integral is given by

$$\text{P.I.} = \frac{1}{\phi(E)} f(n).$$

Case 1. $f(n) = a^n$

$$\text{P.I.} = \frac{1}{\phi(E)} \cdot a^n = \frac{1}{\phi(a)} a^n \text{ provided } \phi(a) \neq 0.$$

If $\phi(a) = 0$, then for the equation

(i) $(E - a)\, y_n = a^n$

$$\text{P.I.} = \frac{1}{E-a} \cdot a^n = n a^{n-1}$$

(ii) $(E-a)^2 y_n = a^n$

$$\text{P.I.} = \frac{1}{(E-a)^2} a^n = \frac{n(n-1)}{2!} \cdot a^{n-2}$$

(iii) $(E-a)^3 y_n = a^n$

$$\text{P.I.} = \frac{1}{(E-a)^3} \cdot a^n = \frac{n(n-1)(n-2)}{3!} a^{n-3}$$

and so on

Case.2 (1) $f(n) = \sin kn$

$$\text{P.I.} = \frac{1}{\phi(E)} \sin kn = \frac{1}{\phi(E)} \left\{ \frac{e^{ikn} - e^{-ikn}}{2i} \right\}$$

$$= \frac{1}{2i} \left[\frac{1}{\phi(E)} \cdot a^n - \frac{1}{\phi(E)} \cdot b^n \right]$$

where $a = e^{ik}$ and $b = e^{-ik}$

Now proceed as in case 1.

(2) $f(n) = \cos kn$

$$\text{P.I.} = \frac{1}{\phi(E)} \cdot \cos kn = \frac{1}{\phi(E)} \left\{ \frac{e^{ikn} - e^{-ikn}}{2} \right\}$$

$$= \frac{1}{2} \left[\frac{1}{\phi(E)} \cdot a^n + \frac{1}{\phi(E)} b^n \right]$$ as before.

Now proceed as in case 1.

Case. 3 $f(n) = n^p$

$$\text{P.I.} = \frac{1}{\phi(E)} n^p = \frac{1}{\phi(1+\Delta)} \cdot n^p$$

(1) Expand $[\phi(1+\Delta)]^{-1}$ in ascending powers of Δ by the Binominal theorem as far as the term in Δ^p.

(2) Express n^p in the factorial form and operate on it with each term of the expansion.

Case.4 $f(n) = a^n F(n)$, $F(n)$ being a polynomial of finite degree in n.

$$\text{P.I.} = \frac{1}{\phi(E)} a^n F(n)$$

$$= a^n \cdot \frac{1}{\phi(aE)} F(n)$$

Now $F(n)$ being a polynomial in n, proceed as in case 3.

Ex.5 : Solve $y_{n+2} - 4y_{n+1} + 3y_n = 5^n$.

Given equation in symbolic form is

$$\left(E^2 - 4E + 3\right) y_n = 5^n$$

$\therefore$ the auxiliary equation is

$$E^2 - 4E + 3 = 0$$

$$\text{or } (E-1)(E-3) = 0$$

$$\therefore E = 1, 3$$

$$\therefore \text{C.F.} = c_1 (1)^n + c_2 (3)^n = c_1 + c_2 3^n$$

$$\text{and P.I.} = \frac{1}{E^2 - 4E + 3} 5^n = \frac{1}{5^2 - 4 \times 5 + 3} 5^n = \frac{1}{8} 5^n$$

Thus the complete solution is

$$y_n = c_1 + c_2 (3)^n + \frac{5^n}{8}.$$

Ex.6 : Solve $u_{n+2} - 4u_{n+1} + 4u_n = 2^n$.

Given equation in symbolic form is

$$\left(E^2 - 4E + 4\right) u_n = 2^n$$

The auxiliary equation is $E^2 - 4E + 4 = 0$; $\therefore E = 2, 2$

$$\therefore \text{C.F.} = \left(c_1 + c_2 n\right) 2^n$$

$$\text{P.I.} = \frac{1}{(E-2)^2} \cdot 2^n = \frac{n(n-1)}{2!} \cdot 2^{n-2} = n(n-1) 2^{n-3}$$

Hence the complete solution is

$$y_n = \left(c_1 + c_2 n\right) (2)^n + n(n-1) 2^{n-3}$$

Ex.7 : Solve $y_{n+2} - 2 \cos \alpha \cdot y_{n+1} + y_n = \cos \alpha n$.

Given equation in symbolic form is

$$\left(E^2 - 2E \cos \alpha + 1\right) y_n = \cos \alpha n$$

The auxiliary equation is $E^2 - 2 \cos \alpha \cdot E + 1 = 0$

$$\therefore E = \frac{2 \cos \alpha \pm \sqrt{4 \cos^2 \alpha - 4}}{2} = \cos \alpha \pm i \sin \alpha$$

$$\therefore \text{C.F.} = (1)^n \left[c_1 \cos \alpha n + c_2 \sin \alpha n\right]$$

$$= c_1 \cos \alpha n + c_2 \sin \alpha n$$

$$\text{P.I.} = \frac{1}{E^2 - 2 \cos \alpha E + 1} \cos \alpha n$$

$$= \frac{1}{E^2 - E\left(e^{i\alpha} + e^{-i\alpha}\right) + 1} \left(\frac{e^{i\alpha n} + e^{-i\alpha n}}{2}\right)$$

$$= \frac{1}{2}\left[\frac{1}{(E - e^{i\alpha})(E - e^{-i\alpha})} e^{i\alpha n} + \frac{1}{(E - e^{i\alpha})(E - e^{-i\alpha})} e^{-i\alpha n}\right]$$

$$= \frac{1}{2}\left[\frac{1}{(E - e^{i\alpha})(e^{i\alpha} - e^{-i\alpha})} e^{i\alpha n} + \frac{1}{(e^{-i\alpha} - e^{i\alpha})(E - e^{-i\alpha})} e^{-i\alpha n}\right]$$

$$= \frac{1}{2}\left[\frac{1}{2i \sin \alpha} \frac{1}{(E - e^{i\alpha})} e^{i\alpha n} - \frac{1}{2i \sin \alpha} \frac{1}{(E - e^{-i\alpha})} e^{-i\alpha n}\right]$$

$$= \frac{1}{4i \sin \alpha}\left[\frac{1}{E - e^{i\alpha}} e^{i\alpha n} - \frac{1}{E - e^{-i\alpha}} e^{-i\alpha n}\right]$$

$$= \frac{1}{4i \sin \alpha}\left[n e^{i\alpha(n-1)} - n e^{-i\alpha(n-1)}\right]$$

$$= \frac{n}{4i \sin \alpha}\left[2i \sin \alpha\ (n-1)\right] = \frac{n \sin\left[\alpha\ (n-1)\right]}{2 \sin \alpha}$$

Hence the complete solution is

$$y_n = c_1 \cos \alpha n + c_2 \sin \alpha n + \frac{n \sin\left[(n-1)\right] \alpha}{2 \sin \alpha}$$

Ex. 8 : Solve $y_{n+2} - 4y_n = n^2 + n - 1$.

Given equation is $\left(E^2 - 4\right) y_n = n^2 + n - 1$.

The auxiliary equation is $E^2 - 4 = 0$. $\therefore E = \pm 2$.

$\therefore$ C.F. $= c_1 (2)^n + c_2 (-2)^n$.

$$\text{P.I.} = \frac{1}{E^2 - 4}\left(n^2 + n - 1\right) = \frac{1}{(1 + \Delta)^2 - 4}[n\ (n-1) + 2n - 1]$$

$$= \frac{1}{\Delta^2 + 2\Delta - 3}\left\{[n]^2 + 2[n] - 1\right\}$$

$\because [n]^2 = n\ (n-1)$,

$[n] = n$

$$= -\frac{1}{3\left(1 - \frac{2\Delta}{3} - \frac{\Delta^2}{3}\right)}\left\{[n]^2 + 2[n] - 1\right\}$$

$$= \frac{-1}{3}\left[1 - \left(\frac{2\Delta}{3} + \frac{\Delta^2}{3}\right)\right]^{-1}\left\{[n]^2 + 2[n] - 1\right\}$$

$$= \frac{-1}{3}\left[1 + \frac{2\Delta}{3} + \frac{\Delta^2}{3} + \left(\frac{2\Delta}{3} + \frac{\Delta^2}{3}\right)^2 + \ldots\right] \times \left\{[n]^2 + 2[n] - 1\right\}$$

$$= \frac{-1}{3}\left[1 + \frac{2\Delta}{3} + \frac{7}{9}\Delta^2 + \ldots\right]\left\{[n]^2 + 2[n] - 1\right\}$$

$$= \frac{-1}{3}\left\{[n]^2 + 2[n] - 1 + \frac{2}{3}(2[n]+2) + \frac{7}{9}\times 2\right\}$$

$\because \Delta[n]^2 = 2[n]$,
$\Delta[n] = 1$,
$\Delta^2[n]^2 = 2$.

$$= \frac{-1}{3}\left\{[n]^2 + \frac{10}{3}[n] + \frac{17}{9}\right\}$$

$$= \frac{-1}{3}\left\{n(n-1) + \frac{10n}{3} + \frac{17}{9}\right\}$$

$$= \frac{-1}{3}\left\{n^2 + \frac{7n}{3} + \frac{17}{9}\right\} = \frac{-n^2}{3} - \frac{7n}{9} - \frac{17}{27}$$

Hence the complete solution is

$$y_n = c_1 2^n + c_2 (-2)^n - \frac{n^2}{3} - \frac{7n}{9} - \frac{17}{27}.$$

Ex. 9 : Solve $y_{n+2} - 2y_{n+1} + y_n = n^2 \cdot 2^n$

Given the equation is $\left(E^2 - 2E + 1\right) y_n = n^2 \cdot 2^n$

Its C.F. $= c_1 + c_2 n$

and P.I. $= \frac{1}{(E-1)^2} \cdot n^2 2^n = 2^n \cdot \frac{1}{(2E-1)^2} \cdot n^2$

$$= 2^n \cdot \frac{1}{(1+2\Delta)^2} \cdot n^2$$

$$= 2^n (1+2\Delta)^{-2} n^2 = 2^n \left(1 - 4\Delta + 12\Delta^2 - \ldots\right)\{n(n-1) + n\}$$

$$= 2^n \left(1 - 4\Delta + 12\Delta^2 - \ldots\right)\left\{[n]^2 + [n]\right\}$$

$$= 2^n \left\{[n]^2 + [n] - 4(2[n]+1) + 12 \times 2\right\}$$

$$= 2^n \left\{[n]^2 - 7[n] + 20\right\} = 2^n \left(n^2 - 8n + 20\right).$$

Hence the complete solution is

$$y_n = c_1 + c_2 n + 2^n \left(n^2 - 8n + 20\right)$$

8.6 SUMMARY

The difference equations are analogous to the differential equations of calculus. We have defined the terms order, solution and general solution and particular solutions of a difference equation. We have delt with the linear difference equations only. Various method for finding particular integral are discussed.

8.7 SAMPLE EXAMINATION QUESTIONS

1. Assuming $\frac{\log(1-z)}{1+z} = y_0 + y_1 z + y_2 z^2 + \ldots + y_n z^n + \ldots$ find the difference equation satisfied by y_n.

2. Derive the difference equations in each of the following cases.

   (i) $y_n = A \cdot 2^n + B \cdot 3^n$

   (ii) $y_n = (A + Bn)\, 2^n$

3. Solve the difference equation

$$y_{n+2} - y_{n+1} - 2y_n = 0,$$

4. Solve $u_{n+2} - 2u_{n+1} + 4u_n = 0$.
5. Solve $u_{p+2} - 6u_{p+1} + 9u_p = 0$.
6. Solve $\left(\Delta^2 - 3\Delta + 2\right) f(n) = 0$.
7. Solve $y_{n+2} - 5y_{n+1} - 6y_n = 2^n$.
8. Solve $y_{n+2} - 4y_n = 2^n$.
9. Solve $u_{n+2} + u_n = \cos n/2$.
10. Solve $u_{n+2} - 2u_{n+1} + u_n = 3n + 5$.
11. Solve $u_{n+2} + u_{n+1} + u_n = n^2 + n + 1$.
12. Solve $y_{n+2} - 6y_{n+1} + 8y_n = 2^n + 6n$.
13. Solve $u_{n+2} - 7u_{n+1} - 8u_n = 2^n [n]^2$.
14. Solve $u_{(x+3)} + 8u_{(x)} = (2x + 3)\, 2^x$.

Answers

1. $\Delta y_n = \dfrac{(-1)^{n+1}}{n+1}$
2. (i) $y_{n+2} - 5y_{n+1} + 6y_n = 0$ (ii) $y_{n+2} - 4y_{n+1} + 4y_n = 0$
3. $y_n = c_1\, 2^n + c_2\, (-2)^n$
4. $u_n = 2^n \left[c_1 \cos \dfrac{n\pi}{3} + c_2 \sin \dfrac{n\pi}{3}\right]$
5. $u_p = \left(c_1 + c_2\, p\right) 3^p$.
6. $f(n) = c_1\, 2^n + c_2 \,.\, 3^n$
7. $y_n = c_1\, (-1)^n + c_2\, (6)^n - \dfrac{2^n}{12}$
8. $y_n = c_1\, 2^n + c_2\, (-2)^n + n \,.\, 2^{n-3} - 2^{n-4}$.
9. $u_n = c_1 \cos \dfrac{n\pi}{2} + c_2 \sin \dfrac{n\pi}{2} + \dfrac{1}{2} \cos \dfrac{n-1}{2} \sec \dfrac{1}{2}$
10. $u_n = c_1 + c_2\, n + \dfrac{1}{2} n\, (n-1)\, (n+3)$
11. $u_n = c_1 \cos \dfrac{2n\pi}{2} + c_2 \sin \dfrac{2n\pi}{3} + \dfrac{1}{3}\left(n^2 - n + \dfrac{1}{3}\right)$
12. $y_n = c_1\, 4^n + \left(c_2 - \dfrac{1}{4} n\right) 2^n + 2n - \dfrac{8}{3}$
13. $u_n = c_1\, (-1)^n + c_2\, 8^n - \dfrac{2^n}{54}\left(3n^2 - 5n + 2\right)$
14. $u_x = 2^x \left(c_1 \cos \dfrac{n\pi}{3} + c_2 \sin \dfrac{n\pi}{3}\right) + c_3\, (-2)^x + x \,.\, 2^{x-3}$

BLOCK- 3 : NUMERICAL DIFFERENTIATION AND INTEGRATION

Introduction

In solving engineering and scientific problems, sometimes, it may be necessary to evaluate the derivative or the integral of a given function f (x), which may not explicitly known, at some values of the independent variable x. Normally the information available is a set of values of the unknown function f (x) at some $x = x_i$ ($i = 1, 2, \ldots, n$).

The process by which we can find the derivative of f (x) at a desired point u is called numerical differentiation. This is achieved by first approximating the function by an interpolation formuala and then differentiating it.

The concept of a definite integral as the limit of a sum is taken as the basis for the evaluation of a definte integral numerically. The given set of values of the function are used for numerical integration by choosing a suitable interpolation formula and then integrating it between the desired limits.

The Euler – Maclaurin series method to evaluate the given series is also considered in this block.

UNIT–9 : NUMERICAL DIFFERENTIATION

Contents

9.1 AIMS AND OBJECTIVES

After going through this unit, you will be able to : (i) obtain the derivative of a given function at a required point, when the information about the function is given in the form of a table, (ii) obtain the differential coefficient using different interpolation formulae for different cases.

9.2 INTRODUCTION

Numerical differentiation is the process of calculating the derivatives of a function by means of a set of given values of that function. The problem is solved by representing the function by an interpolation formula and then differentiating this formula as many times as required.

If the function is given by a table of values for equidistant values of the independent variable, it should be represented by an interpolation formula, employing differences, such as Newton's, Stirling's or Bessel's. But if the given values of the function are not for equidistant values of the independent variable, we must represent the function by Lagrange's or Divided difference formula. The considerations governing the choice of a formula employing differences are the same as in the case of interpolation.

9.3 DIFFERENTIATION OF NEWTON'S FORWARD DIFFERENCE FORMULA

Suppose the values of x and the corresponding $f(x)$ are

$$x: \quad x_0, x_1, x_2, \ldots x_n$$

$$y = f(x): \quad y_0, y_1, y_2, \ldots y_n$$

where x_i are equally spaced at the interval h. Newton forward difference interpolation formula is employed if the point at which the derivative is required is near the beginning of the interval (x_0, x_n).

The Newton's difference interpolation formula is

$$f(x_0 + xh) = f_0 + \binom{x}{1}\Delta f_0 + \binom{x}{2}\cdot\Delta^2 f_0 + \ldots + \binom{x}{n}\cdot\Delta^n f_0$$

where $f_0 = f(x_0)$, $\Delta f_0 = \Delta f(x_0)$, ... and so on.

$$\text{(i.e) } f(x_0 + xh) = f_0 + \frac{x}{1!}\Delta f_0 + \frac{x(x-1)}{2!}\Delta^2 f_0 + \ldots + \frac{x(x-1)\ldots(x-n+1)}{n!}\Delta^n f_0$$

Differentiating with respect to x we get

$$1)\quad hf'(x_0 + xh) = \Delta f_0 + \frac{2x-1}{2!}\Delta^2 f_0 + \frac{3x^2 - 6x + 2}{6}\Delta^3 f_0 + \frac{4x^3 - 21x^2 + 28x - 84}{24}\Delta^4 f_0 + \frac{5x^2 - 48x^3 + 147x^2 - 156x + 40}{120}\Delta^5 f_0 + \ldots$$

Taking $x = 0$ in (1) we get

$$2)\quad hf'(x_0) = \Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \ldots + \frac{(-1)^{n-1}}{n}\Delta^n f_0.$$

To get the derivative $f'(x_1)$ put $x = 1$, for $f'(x_2)$ put $x = 2$ and so on.

To get second order derivative, we differentiate (1) again w.r.t. 'x'.

$$h^2 f''(x_0 + xh) = \Delta^2 f_0 + (x-1)\Delta^3 f_0 + \ldots \text{ upto } (n-1) \text{ terms so that}$$

$$3)\quad h^2 f''(x_0) = \Delta^2 f_0 - \Delta^3 f_0 + \frac{7}{6}\Delta^4 f_0 - \frac{13}{10}\Delta^5 f_0 + \ldots$$

Let us consider the following examples.

Ex.1 : Find the first and second derivatives of the function tabulated below, at the point $x = 0$.

x :	0	1	2	3	4	5
$f(x)$:	4.21	5.11	6.20	12.80	23.70	36.80

Solution

In this problem the interval h is 1.

Using the formula (2) we have

$$1.\ f'(0) = f'(x_0) = \Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \frac{1}{4}\Delta^4 f_0 + \frac{1}{5}\Delta^5 f_0 + \ldots$$

The difference table is

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
0	4.21					
		0.90				
1	5.11		0.19			
		1.09		5.32		
2	6.20		5.51		–8.11	
		6.60		–2.79		7.00
3	12.80		4.30		–1.11	
		10.90		–3.90		
4	23.70		2.20			
		13.10				
5	36.80					

$$\text{Now } f'(0) = 0.90 - \frac{1}{2}(0.19) + \frac{1}{3}(5.32) - \frac{1}{4}(-8.11) + \frac{1}{5}(7.00) = 6.006$$

Using the formula (3) we have

$$h^2 f''(0) = \Delta^2 f_0 - \Delta^3 f_0 + \frac{7}{6}\ \Delta^4 f_0 - \frac{15}{10}\Delta^5 f_0$$

$$\therefore 1 \cdot f''(0) = 0.19 - (5.32) + \frac{7}{6}(-8.11) - \frac{15}{10}(7.00)$$

$$= -23.69$$

Ex. 2 : Find the first and second derivatives of $f(x)$ at $x = 16$ from the following table.

x :	15	17	19	21	23	25
$f(x)$:	3.873	4.123	4.359	4.583	4.790	5.000

Solution

The point at which the derivatives are required is near the beginning of the table.

The difference table is

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
15	3.873					
		0.250				
17	4.123		–0.014			
		0.236		0.002		
19	4.359		–0.012		–0.002	
		0.224		0.000		.007
21	4.583		–0.012		0.005	
		0.212		0.005		
23	4.795		–0.007			
		0.205				
25	5.000					

Here $h = 2$. Taking upto third order differences only and $x = \frac{1}{2}$

$$2 \cdot f'(16) = 0.250 + 0 - \frac{1}{24}(0.002) = 0.250 - .00008$$

$$= .24992$$

$$\therefore f'(16) = 0.12496$$

$$\text{and } 4 \cdot f''(16) = -0.014 + \left(-\frac{1}{2}\right)(0.002) = -0.015$$

$$\therefore f''(16) = -0.00375.$$

9.4 DIFFERENTIATION OF NEWTON'S BACKWARD DIFFERENCE FORMULA

If the derivative is required at the point near the end of the table, one can use the Newton's backward difference formula.

4) $$f(a + nh + xh) = f(a + nh) + \Delta f(a + nh) + \frac{x^2 + x}{2!}\Delta^2 f(a + nh) + \frac{x^3 + 3x^2 + 2x}{3!}\Delta^3 f(a + nh) + \ldots$$

Differentiating (4) w.r.t. 'x' we get

5) $$hf'(a + nh + xh) = \Delta f(a + nh) + \frac{2x + 1}{2!}\Delta^2 f(a + nh) + \frac{3x^2 + 6x + 2}{3!}\Delta^3 f(a + nh) + \ldots$$

Substituting $x = 0$ in (5), we get

6) $$hf'(a + nh) = \Delta f(a + nh) + \frac{1}{2!}\Delta^2 f(a + nh) + \frac{1}{3}\Delta^3 f(a + nh) + \ldots\ldots$$

The higher order derivatives can be obtained by repeated differentiation of (5) w.r.t. 'x'.

Ex. 3 : Find the value of $f'(x)$ at $x = .06$ from the following table.

x :	.01	.02	.03	.04	.05	.06
$f(x)$:	.102	.105	.107	.110	.112	.115

Solution

The difference table is

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$
.01	.102			
		.003		
.02	.105		−.001	
		.002		.002
.03	.107		+.001	
		.003		−.002
.04	.110		−.001	
		.002		.002
.05	.112		+.001	
		.003		
.06	.115			

Since the point at which the derivative required is at the end of the table we use the formula (6).

$$\text{here } h = .01, \text{ and } a + nh = .06$$

$$\therefore (0.1) f'(0.6) = .003 + \frac{1}{2}(.001) + \frac{1}{3}(.002) = .00416.$$

9.5 DIFFERENTIATION OF CENTRAL DIFFERENCE FORMULA

When the derivatives are required at points near the middle of the interval one can use a central difference formula like Bessel's formula or Stirling formula.

9.5.1 Bessel's formula

Let x_k be a point near the middle of the interval (x_1, x_n)

$$\text{writing } u = \frac{x - x_k}{h} \text{ or } x = x_k + uh$$

$$\frac{d}{dx} f(u) = \frac{d}{du} f(u) \cdot \frac{du}{dx} =$$

$$= \frac{d}{du} f(u) \cdot \frac{1}{h} = \frac{f'(u)}{h}$$

Bessel's formula is

$$f(u) = \frac{1}{2}(f(0) + f(1)) + \left(u - \frac{1}{2}\right) \Delta f(0) + \frac{u(u-1)}{2!}$$

$$\times \frac{1}{2}\left(\Delta^2 f(-1) + \Delta^2 f(0)\right)$$

$$+ \frac{\left(u - \frac{1}{2}\right) u (u-1)}{3!} \Delta^3 f(-1) + \frac{(u+1) u (u-1) (u-2)}{4!}$$

$$\frac{1}{2}\left(\Delta^4 f(-1) + \Delta^4 f(-2)\right) + \ldots$$

Differentiating w.r.t. 'u' and putting $u = 0$ we get

$$f'(0) = \Delta f(0) - \frac{1}{4}\left(\Delta^2 f(-1) + \Delta^2 f(0)\right) + \frac{1}{12}\Delta^3 f(-1)$$

$$+ \frac{1}{24}\left(\Delta^4 f(-1) + \Delta^4 f(-2)\right) + \ldots$$

7) $$\therefore f'(x) = \frac{1}{h} f'(0) = \frac{1}{h}\left[\Delta f(0) - \frac{1}{4}\left(\Delta^2 f(-1) + \Delta^2 f(0)\right) + \right.$$

$$\left. \frac{1}{12}\Delta^3 f(-1) + \frac{1}{24}\left(\Delta^4 f(-1) + \Delta^4 f(-2)\right) + \ldots\ldots\right]$$

9.5.2 Stirling's formula

Stirling's interpolation formula is

$$f(u) = f(0) + u \cdot \frac{\Delta f(0) + \Delta f(-1)}{2} + \frac{u^2}{2!}\Delta^2 f(-1) +$$

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

$$\frac{u^2\left(u^2-1\right)}{4!}\Delta^4 f(-2)+\ldots$$

Differentiating w.r.t. 'u' we get

8) $$f'(u) = \frac{1}{2}[\Delta f(0)+\Delta f(-1)]+u\,\Delta^2 f(-1)$$

$$+\frac{3u^2-1}{12}\left[\Delta^3 f(-1)+\Delta^3 f(-2)\right]+$$

$$+\frac{u\left(2u^2-1\right)}{12}\Delta^4 f(-2)+\ldots$$

Ex.4 : Find $f'(x)$ for $x = 1.4$ from the following table

x :	1.2	1.3	1.4	1.5	1.6
$f(x)$:	1.510	1.698	1.904	2.129	2.376

Solution

The difference table is

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$
1.2	1.510				
		.188			
1.3	1.698		.018		
		.206		.001	
1.4	1.904		.019		.002
		.225		.003	
1.5	2.129		.022		
		.247			
1.6	2.376				

Here $h = .1$, $u = \dfrac{x-1.4}{.1}$

Using formula (7) we have

$$f'(1.4) = \frac{1}{.1}\left[.225-\frac{1}{4}(.019+.022)+\frac{1}{12}(.003)+\frac{1}{24}(.002)+\ldots\right]$$

$$= 10\times .215 = 2.15$$

Using the formula (8) we have

$$f'(1.4) = \frac{1}{.1}\left[\frac{.225+.206}{2}-\frac{1}{12}(.003+.001)+\ldots\right]$$

Ex.5 : Find $f'(7)$ using the following table.

x :	2	4	6	8	10	12	14
$f(x)$:	104	17	0	–1	8	69	272

Solution

In this problem $h = 2$ and $u = \frac{x-6}{2} = \frac{7-6}{2} = 0.5$

Using the formula (7) we have

$$f'(u) = \frac{1}{2}(f(0) + f(1)) + \left(u - \frac{1}{2}\right)\Delta f(0) + \frac{u(u-1)}{4}$$

$$\left(\Delta^2 f(-1) + \Delta^2 f(0)\right) + \frac{\left(u - \frac{1}{2}\right) u(u-1)}{6}\Delta^3 f(-1)$$

$$+ \frac{(u+1)\,u\,(u-1)\,(u-2)}{4!} \cdot \frac{1}{2}\left(\Delta^4 f(-1) + \Delta^4 f(-2)\right) + \ldots$$

The difference table is

x	u	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$
2	–2	104				
			–87			
4	–1	17		70		
			17		–54	
6	0	0		16		48
			–1		–6	
8	1	–1		10		48
			9		42	
10	2	8		52		48
			61		90	
12	3	69		142		
			203			
14	4	272				

$$f'(u) = \frac{1}{2}(0 + (-1)) + \frac{1}{2}\frac{\left(-\frac{1}{2}\right)}{4}(16 + 70) +$$

$$\frac{\frac{3}{2} \cdot \frac{1}{2}\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)}{24} \cdot \frac{1}{2}(48 + 48) + \ldots$$

$$= -5.775$$

9.6 DIFFERENTIATION OF NEWTON's DIVIDED DIFFERENCE FORMULA

Newton's divided difference formula is used when the given points are unequally spaced.

9) $f(x) = f(x_0) + (x - x_0)\,\Delta f(x_0) + (x - x_0)(x - x_1)\,\Delta^2 f(x_0) + \cdots$

Differentiating w.r.t. 'x' we obtain $f'(x)$.

Ex. 6 : Find $f'(6)$ from the following table

x :	0	2	3	4	7	9
$f(x)$:	14	26	35	50	90	122

Solution

From (9) we obtain

10) $$f'(x) = \Delta f(0) + (2x - 2)\,\Delta^2 f(0) + (3x^2 - 10x + 6)\,\Delta^3 f(0) + (4x^3 - 18x^2 + 52x - 24)\,\Delta^4 f(0) + (5x^4 - 4\cdot\cdot - 58x^2 - 308x + 120)\,\Delta^5 f(0) + \cdots$$

The divided difference table is

x	$f(x)$	$\Delta f(x)$	$\Delta^2 f(x)$	$\Delta^3 f(x)$	$\Delta^4 f(x)$	$\Delta^5 f(x)$
0	14					
		6				
2	26		1			
		9		0.5		
3	35		3		–0.16	
		15		–0.65		0.03
4	50		–.25		0.10	
		14		0.07		
7	92		.20			
		15				
9	122					

$$f'(6) = 6 + 10(-1) + 54(0.5) + 504(-0.16) + 948(0.03)$$

$$= -29.2$$

Remark : Numerical differentiation is useful in problems of mechanics to find velocity, acceleration etc. But the errors that creep in will be sometimes so high that it is not applicable when one needs correct values.

9.7 SUMMARY

When the explicit form of a function is not given and when it is required to find the derivative of that function at some given points, we follow the process of numerical differentiation. We approximated the function by an interpolation polynomial. We followed the same principle as that of the problem of interpolation for choosing the interpolation formula.

9.8 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail.*

1. (a) Explain the concept of numerical differentiation. Obtain the formula for $f'(x)$ and $f''(x)$ when x is a point near the beginning of the interval (x_0, x_n).

   (b) Caluculate $f'(0.6)$ and $f''(0.6)$ from the following table.

x :	0.5	0.7	0.9	1.1	1.3	1.5	1.7
$f(x)$:	0.479	0.644	0.783	0.891	0.964	0.998	0.992

2. (a) Derive the formula for $f'(x)$ and $f''(x)$ when x is a point near the middle point of the interval (x_0, x_n).

   (b) Use the following table and find $f'(0.4)$ and $f''(0.4)$

x :	0.398	0.399	0.400	0.401	0.402
$f(x)$:	0.40859	0.40967	0.41075	0.41183	0.41292

3. (a) What interpolation formula would use to find the derivatives of $f(x)$ at a desired point of (x_0, x_n). when x_i are unequally spaced.

   (b) Find $f'(1.6)$ and $f''(3.3)$ from the table of values

x :	1.5	1.9	2.5	3.2	4.2	5.9
$f(x)$:	3.38	6.06	13.63	22.37	73.91	196.58

II. *Briefly answer the following*

1. Derive the formula for finding $f'(x)$ and $f''(x)$ numerically at a point near the end of the interval (x_0, x_n).

2. Evaluate $f''(0.55)$ from the following table

x :	.2	.3	.4	.5	.6
$f(x)$:	.5095	.6984	.9043	1.1293	1.3756

3. Find $f''(3)$ from the following table

x :	2.94	2.96	3.02	3.06
$f(x)$:	0.1826	0.1811	0.1769	0.1742

UNIT–10 : NUMERICAL INTEGRATION

Contents

10.1 AIMS AND OBJECTIVES

After going through this unit you will be able to : (i) derive quadrature formula for a function whose values are given at certain points, by choosing Newton's interpolation formula for the function, (ii) calculate the remainder terms in quadrature formulas like trapezoidal, Simpson's and Weddles rules.

10.2 INTRODUCTION

Numerical integration is the process of computing the value of a definite intergral from a set of numerical values of the integrand. The problem of numerical integration, like that of numerical differentiation, is solved by representing the integrand by an interpolation formula and then integrating this formula between the desired limits. Thus, to find the value of the definite integral $\int_a^b y\,dx$, we replace the function y by an interpolation formula, usually one involving differences and then integrate this formula between the limits a and b. In this way we can derive quadrature formulas for the approximate integration of any function for which numerical values are known. We shall now derive some of the simplest and most useful of the qudrature formulas.

Also we calculate the remainder terms in quadrature formulas. In numerical methods, generally, three types of errors arise. They are (i) inherent errors, (ii) round off errors and, (iii) truncated errors. In the case of computation of an infinite series expansion, we truncate the series after a finite number of terms. This leads to truncation errors. The remainder term gives the measure of truncation error.

10.3 GENERAL QUADRATURE FORMULA

Suppose the Values of x_i and the corresponding values of the Function are

x :	x_0	x_1	x_2	.....	x_n
$f(x)$:	y_0	y_1	y_2	.....	y_n

where $x_i = x_0 + i\,h$ and $y_i = f(x_i)$

We want to evaluate the definite integral $\int_a^b f(x)\,dx$ when the integrand is a function of a single variable. We shall first derive a general quadrature formula.

To evaluate $I = \int_a^b f(x)\,dx$, divide the interval $[a, b]$ into n equal parts each of width $h = \frac{b-a}{n}$

Let us denote

$a = x_0$, $b = x_n = x_0 + nh$, $x_k = x_0 + kh$,

and $y_k = f(x_k) = f(x_0 + k\,h)$.

Newton's forward interpolation formula is

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

where $u = \frac{x - x_0}{h}$

$$I = \int_a^b f(x)\,dx = \int_a^b y\,dx = \int_{x_0}^{x_0+nh} y\,dx$$

$$= h\int_0^n \left(y_0 + u\,.\,\Delta y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + \ldots\right) du$$

$$\left(\because \frac{du}{dx} = \frac{1}{h}\right)$$

$$\therefore I = h\left[n y_0 + \frac{n^2}{2}\Delta y_0 + \left(\frac{n^3}{3} - \frac{n^3}{2}\right)\frac{\Delta^2 y_0}{2!}\right.$$

$$\left. + \left(\frac{n^4}{4} - n^3 + n^2\right)\frac{\Delta^3 y_0}{3!} + \ldots\right]$$

By giving $n = 1, 2, 3$ and 6 we get different quadrature formulas.

10.4 TRAPEZOIDAL RULE

If we put $n = 1$ in (1) we find

$$\int_{x_0}^{x_0+h} y\,dx = h\left[y_0 + \frac{1}{2}\Delta y_0\right]$$

neglecting $\Delta^2 y_0$, $\Delta^2 y_0$ etc.,

$$= h\left[y_0 + \frac{1}{2}(y_1 - y_0)\right] = \frac{h}{2}\left[y_0 + y_1\right]$$

Similarly $\int_{x_0+h}^{x_0+2h} y\,dx = \frac{h}{2}\left[y_1 + y_2\right]$, and so on

Finally $\int_{x_0+(n-1)h}^{x_0+nh} y\,dx = \frac{h}{2}\left[y_{n-1} + y_n\right]$

Adding these n integrals we get

$$\int_{x_0}^{x_0+nh} y\,dx = \frac{h}{2}\left[(y_0 + y_n) + 2(y_1 + y_3 + \ldots + y_{n-1})\right]$$

10.5 SIMPSON'S $\frac{1}{3}$ RULE

Taking $n = 2$ in formula (1) we get

$$\int_{x_0}^{x_0+2h} y\,dx = h\left[2y_0 + 2\,.\,\Delta y_0 + \left(\frac{8}{3} - 2\right)\frac{\Delta^2 y_0}{2!}\right]$$

(neglecting the third and higher order differences)

$$= h\left[2y_0 + 2\left(y_1 - y_0\right) + \frac{1}{3}\left(y_2 - 2y_1 + y_0\right)\right]$$

$$= \frac{h}{3}\left[y_0 + 4y_1 + y_2\right]$$

Similarly, $\int_{x_0+2h}^{x_0+4h} y\,dx = \frac{h}{3}\left(y_2 + 4y_3 + y_4\right)$ and so on

$$\int_{x_0+(n-2)h}^{x_0+nh} y\,dx = \frac{h}{3}\left(y_{n-2} + 4y_{n-1} + y_n\right)$$

(the given interval in this case must be divided into an even number of intervals)

Thus

$$\int_{x_0}^{x_0+nh} y\,dx = \int_{x_0}^{x_0+2h} y\,dx + \int_{x_0+2h}^{x_0+4h} y\,dx + \ldots \int_{x_0+(n-2)h}^{x_0+nh} y\,dx$$

(3) $$\int_{x_0}^{x_0+nh} y\,dx = \frac{h}{3}\left[\left(y_0 + y_n\right) + 4\left(y_1 + y_3 + \ldots + y_{n-1}\right) + 2\left(y_2 + y_4 + \ldots + y_{n-2}\right)\right]$$

Formula (3) is called Simpson's $\frac{1}{3}$ Rule.

10.6 SIMPSON'S $\frac{3}{8}$ ths RULE

Taking $n = 3$ in (1), neglecting $\Delta^4 y_0$, $\Delta^5 y_0$ etc... we get

$$\int_{x_0}^{x_0+3h} y\,dx = h\left[3y_0 + \frac{9}{2}\Delta y_0 + \frac{9}{4}\Delta^2 y_0 + \frac{3}{8}\Delta^3 y_0\right]$$

$$= h\left[3y_0 + \frac{9}{2}\left(y_1 - y_0\right) + \frac{9}{4}\left(y_2 - 2y_1 + y_0\right) + \frac{3}{8}\left(y_3 - 3y_2 + 3y_1 - y_0\right)\right]$$

$$= \frac{3h}{8}\left[y_0 + 3y_1 + 3y_2 + y_3\right]$$

Similarly,

$$\int_{x_0+3h}^{x_0+6h} y\,dx = \frac{3h}{8}\left[y_3 + 3y_4 + 3y_5 + y_6\right] \text{ and so on}$$

Thus

$$\int_{x_0+(n-3)h}^{x_0+nh} y\,dx = \frac{3h}{8}\left[y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n\right]$$

(4) $$\int_{x_0}^{x_0+nh} y\,dx = \frac{3h}{8}\left[(y_0 + y_n) + 3\left(y_1 + y_2 + y_4 + \ldots + y_{n-1}\right)\right.$$

$$\left. + 2\left(y_3 + y_6 + \ldots + y_{n-3}\right)\right]$$

In this case the number of intervals must be a multiple of 3. This formula (4) is called Simpson's $\frac{3}{8}$ths rule.

10.7 WEDDLE'S RULE

If we put $n = 6$ and neglect the differences of order higher than sixth we get

$$\int_{x_0}^{x_0+6h} y\,dx = h\left[6y_0 + 18\,\Delta y_0 + 27\,\Delta^2 y_0 + 24\,\Delta^3 y_0 + \frac{123}{10}\Delta^4 y_0\right.$$

$$\left. + \frac{33}{10}\Delta^5 y_0 + \frac{41}{140}\Delta^6 y_0\right]$$

Taking $\frac{41}{140} \simeq \frac{3}{10}$, the above simplifies to

$$\int_{x_0}^{x_0+6h} y\,dx = \frac{3h}{10}\left[y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6\right]$$

Simialrly,

$$\int_{x_0+6h}^{x_0+12h} y\,dx = \frac{3h}{10}\left[y_6 + 5y_7 + y_8 + 6y_9 + y_{10} + 5y_{11} + y_{12}\right] \text{ and so on}$$

Finally

$$\int_{x_0+(n-6)h}^{x_0+nh} y\,dx = \frac{3h}{10}\left[y_{n-6} + 5y_{n-5} + y_{n-4} + 6y_{n-3} + y_{n-2} + 5y_{n-1} + y_n\right]$$

Adding all these integrals

(5) $$\int_{x_0}^{x_0+nh} y\,dx = \frac{3h}{10}\left[y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + 2y_6 + 5y_7 + \ldots\right]$$

This is Weddle's rule. It is more accurate than Simpson's rule. It requires at least seven consecutive values of the function and is applicable only when n is a multiple of six.

Let us now consider some examples

Ex. 1 : Evaluate the intergal $\int_0^2 1 + x^2\, dx$

Solution :

The exact value of the integral $= x + \frac{x^3}{3}\Big|_0^2 = \frac{14}{3} = 4.67$

Let us first consider the trapeziodal rule.

Taking $h = .2$ (i.e., into ten sub-invervals)

x_0	= 0	y_0	= 1
x_1	= 0.2	y_1	= 1.04
x_2	= 0.4	y_2	= 1.16
x_3	= 0.6	y_3	= 1.36
x_4	= 0.8	y_4	= 1.64
x_5	= 1.0	y_5	= 2.0
x_6	= 1.2	y_6	= 2.44
x_7	= 1.4	y_7	= 2.96
x_8	= 1.6	y_8	= 3.56
x_9	= 1.8	y_9	= 4.24
x_{10}	= 2.0	y_{10}	= 5.0

Using formula (2)

$$\int_0^2 1 + x^2\, dx = \frac{h}{2}\left[\left(y_0 + y_{10}\right) + \left(y_1 + y_3 + \ldots + y_9\right)\right]$$

$$= (.1)\,(6 + 25.4) = 3.14$$

Using formula (3), with $h = 0.2$ and 10 intervals

$$\int_0^2 1 + x^2\, dx = \left(\frac{.2}{3}\right)\left[\left(y_0 + y_{10}\right) + 4\left(y_1 + y_3 + y_5 + y_7 + y_9\right)\right.$$

$$\left. + 2\left(y_2 + y_4 + y_6 + y_8\right)\right]$$

$$= \left(\frac{.2}{3}\right)\,(6 + 46.4 + 17.6) = 4.67$$

Comparison of the values obtained using trapezoidal rule and Simpson's rule with the exact value of the integral we find that the Simpson's rule gives a better approximation.

Ex. 2 : Find an approximation of the value of π by applying Simpson's $\frac{1}{3}$ rule to $\int_0^1 \frac{dx}{1+x^2}$.

Solution :

$$\int_0^1 \frac{dx}{1+x^2} = \tan^{-1} x \Big|_0^1 = \frac{\pi}{4}$$

Taking $n = 10$ we have $h = 0.1$. The values of x_i and y_i are shown in the following table.

x :	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y :	1	.99	.962	.917	.862	.8	.735	.671	.610	.553	0.5

$$\int_0^1 \frac{dx}{1+x^2} = \frac{0.1}{3}\,[(1 + .5) + 4\,(.99 + .917) + .8 + .671 + .553)$$

$$+ 2\,(.962 + .862 + .735 + .610)]$$

$$= 0.7854$$

Hence $\frac{\pi}{4} = 0.7854$ approximately

or $\pi = 3.1416.$

Ex. 3 : Obtain the value of $\int_0^6 \frac{dx}{1+x}$ using Simpson's $\frac{3}{8}$ths rule and Weddle's rule.

Solution:

Taking $h = 1$ and $n = 6$

x	0	1	2	3	4	5	6
y	1	.5	.333	.25	.2	.167	.143

By Simpson's rule $\left(\frac{3}{8}\text{ths}\right)$

$$\int_0^6 \frac{dx}{1+x} = \frac{3}{8}\left[(y_0 + y_6) + 3\,(y_1 + y_2 + y_4 + y_5) + 2y_3\right]$$

$$= \frac{3}{8}\,[1.143 + 3.6 + 0.50] = 1.966$$

By Weddle's rule

$$\int_0^6 \frac{dx}{1+x} = \frac{3}{10}\,[1 + 5\,(.5) + .333 + 6\,(.25) + .2 + 5\,(.167) + .143]$$

$$= 1.9533$$

The actual value of the integral is 1.946.

Hence we find that Weddle's rule is giving a better approximation than Simpson's $\frac{3}{8}$ ths rule

Ex. 4 : A curve is drawn through the points given by the following table

x	1	2	3	4	5
y	10	50	70	80	100

Estimate the area bounded by the curve, the x-axis and the lines $x = 1$ and $x = 5$.

Solution :

The required area $= \int_1^5 y\,dx$

Using Simpson's $\frac{1}{3}$ rule

$$\int_1^5 y\,dx = \frac{h}{3}\left[\left(y_0 + y_4\right) + 4\left(y_1 + y_3\right) + 2y_2\right]$$

$$= \frac{1}{3}\,[(10 + 100) + 4\,(50 + 80) + 2 \times 70]$$

$$= 256.7 \text{ square units}$$

Ex. 5 : The velocity of a particle during the first 80 secs are given below

t (secs) :	0	10	20	30	40	50	60	70	80
v (m/sec) :	30.0	31.63	33.34	35.47	37.75	40.33	43.25	46.69	50.67

Find the distance travelled in 80 secs.

Solution : If S is the distance travelled in time t secs then the velocity is given by $\frac{ds}{dt}$.

Hence total distance travelled in 80 secs $\int_0^{80} \frac{ds}{dt}\,dt$.

Taking $h = 10$ secs, $n = 8$ and using Simpson's $\frac{3}{8}$ rule we have

$$\int_0^{80} \frac{ds}{dt}\,dt = \frac{3 \times 10}{8}\,[(30 + 50.67) + 3\,(31.63 + 33.44 + 37.75 + 40.33 + 46.69) + 2\,(35.47 + 43.25)]$$

$$= 3028.61\ m.$$

10.8 REMAINDER TERMS IN QUADRATURE FORMULAS

10.8.1 Remainder Term in Trapezoidal Rule

Let $f(x)$ be a continuous function in the inverval $(x_0, x_0 + h)$ and possess continuous derivatives upto second order.

Let $\int_a^x f(x)\,dx = g(x)$

Then exact value of the integral (denoted by I_e) $\int_{x_0}^{x_0+h} f(x)\,dx$

$$= \int_a^{x_0+h} f(x)\,dx - \int_a^{x_0} f(x)\,dx = g(x_0 + h) - g(x_0)$$

The value of the integral, using trapezoidal rule, (denoted as I_a)

$$= \frac{h}{2}\left[f(x_0) + f(x_0 + h)\right]$$

(1) $\text{Error} = I_e - I_a = \left[g(x_0 + h) - g(x_0)\right] - \frac{h}{2}\left[f(x_0) + f(x_0 + h)\right]$

From Taylor's series expansion

$$f(x_0 + h) = f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \ldots$$

(2) $$\therefore I_a = \frac{h}{2}\left[f(x_0) + f(x_0 + h)\right]$$

$$= hf(x_0) + \frac{h^2}{2}hf'(x_0) + \frac{h^3}{4}f''(x_0) + \ldots$$

Now $$g(x_0 + h) = g(x_0) + hg'(x_0) + \frac{h^2}{2!}g''(x_0) + \frac{h^3}{3!}g'''(x_0) + \frac{h^4}{4!}g^{iv}(x_0) + \ldots$$

$$g(x_0 + h) - g(x_0) = hg'(x_0) + \frac{h^2}{2!}g''(x_0) + \frac{h^3}{6!}g'''(x_0) + \ldots$$

From the definition of $g(x)$ we have

(3) $\therefore g(x_0 + h) - g(x_0) = hf(x_0) + \frac{h^2}{2} f'(x_0) + \frac{h^3}{6} f''(x_0) + \ldots$

Substituting (2) and (3) in (1) we get

$$\text{Error} = \left(\frac{h^3}{6} - \frac{h^3}{4}\right) f''(x_0) + \ldots = -\frac{h^3}{12} f''(x_0) + \ldots$$

Neglecting derivatives of order higher than two, we find that the error in the case trapezoidal rule is given by $-\frac{h^3}{12} f''(x_0)$.

Suppose we are considering the integral $\int_a^b f(x)\, dx$.

Taking $x_0 = a, x_i = x_0 + ih$ where $i = 1, 2, \ldots n$ and $b = x_0 + nh$

we find the error is given by

$$-\frac{h^3}{12}\left[f''(x_0) + f''(x_1) + \ldots + f''(x_{n-1})\right]$$

If $f''(x_m)$ is the largest of all $f''(x_i)$, $i = 1, 2, \ldots, n-1$, then

$$\text{Error} \leq -\frac{n}{12} h^3 f''(x_m) = -\frac{(b-a)}{12} h^2 f''(x_m)$$

since $nh = b - a$

It can be observed that if $f(x)$ is a first degree polynomial then trapezoidal rule gives the exact value of the integral.

10.8.2 Error in Simpson's $\frac{1}{3}$ Rule

Let $f(x)$ be a function continutous in the interval $(x_0 - h, x_0 + h)$ which possess continuous derivatives upto fourth order.

Proceeding as in 10.8.1

$$I_e = \int_{x_0 - h}^{x_0 + h} f(x)\, dx = g(x_0 + h) - g(x_0 - h)$$

By Simpson's 1/3 rule we have

$$I_a = \frac{h}{3}\left[f(x_0 - h) + 4f(x_0) + f(x_0 + h)\right]$$

(4) $\therefore$ $\text{Error} = I_e - I_a = \left[g(x_0 + h) - g(x_0 - h)\right]$

$$-\frac{h}{3}\left[f(x_0 - h) + 4f(x_0) + f(x_0 + h)\right]$$

Using Taylor series expansion we get

(5) $f(x_0 - h) + 4f(x_0) + f(x_0 + h) = 6f(x_0) + h^2 f''(x_0) + \frac{h^4}{12} f^{iv}(x_0) + \ldots$

(6) $g(x_0 + h) - g(x_0 - h) = 2h\, g'(x_0) + \frac{2h^3}{3!} g'''(x_0) + \frac{2h^5}{5!} g^{v}(x_0) + \ldots$

$= 2h f(x_0) + \frac{h^3}{3} f''(x_0) + \frac{h^5}{60} f^{iv}(x_0) + \ldots$

Substituting (5) and (6) in (4) we get

(7) $\text{Error} = h^2 \left[\frac{1}{60} - \frac{1}{36}\right] f^{iv}(x_0) + \ldots$

$= \frac{-h^5}{90} f^{iv}(x_0) + \ldots$

(neglecting derivatives of order higher than four)

When the interval of integration is (a, b), let $x_0 = a$,

$x_i = x_0 + ih$, and $b = x_0 + nh$

The estimate of the error for the intervals $(x_1 - h, x_1 + h)$ $(x_3 - h, x_3 + h)$, ... can similarly be obtained as in (7)

Thus for the integral $\int_a^b f(x)\, dx$, the error is given by

$\frac{-h^5}{90} \left[f^{iv}(x_1) + f^{iv}(x_3) + \ldots + f^{iv}(x_{n-1})\right]$

If $f^{iv}(x_m)$ is the largest of $f^{iv}(x_1), f^{iv}(x_3)$ ... then

$$\text{Error} \leq -\frac{n}{2} \cdot \frac{h^5}{90} f^{iv}(x_m) = \frac{b - a}{180} h^4 f^{iv}(x_m)$$

$(\because b - a = nh)$

When $f^{iv}(x) = 0$, it is clear that error vanishes which means that if $f(x)$ is a polynomial of degree less than four then simpson's $\frac{1}{3}$ rule gives the exact value of the integral.

10.8.3 Errors is Simpson's 3/8 Rule

In this case we take the interval from x_0 to $x_0 + 3h$.

$$I_e = \int_{x_0}^{x_0+3h} f(x)\, dx = g(x_0 + 3h) - g(x_0)$$

$$\text{and } I_a = \frac{3h}{8} \left[f(x_0) + 3f(x_0 + h) + 3f(x_0 + 2h) + f(x_0 + 3h)\right]$$

(8) $= \frac{3h}{8} \left[8 f(x_0) + 12h f'(x_0) + 12h^2 f''(x_0) + 9h^3 f'''(x_0)\right.$

$\left. + \frac{71}{8} h^4 f^{iv}(x_0) + \ldots\right]$

$$\text{and } I_e = 3h\,f(x_0) + \frac{9h^2}{2} f'(x_0) + \frac{9}{2} h^3 f''(x_0) + \frac{27}{8} h^4 f'''(x_0)$$

$$+ \frac{81}{40} h^5 f^{iv}(x_0) + \cdots$$

$$\therefore \text{Error} = I_e - I_a = -\frac{3}{80} h^5 f^{iv}(x_0) + \cdots$$

(neglecting higher order derivatives)

For the interval (a, b) we get the error as

$$\text{Error} = -\frac{3}{80} h^5 \left[f^{iv}(x_0) + f^{iv}(x_3) + \cdots + f^{iv}(x_{n-3}) \right]$$

If $f^{iv}(x_m)$ is the largest of $f^{iv}(x_0)$, $f^{iv}(x_3)$, ... we find that

$$\text{Error} \leq -\frac{3}{80} h^5 \cdot \frac{n}{3} f^{iv}(x_m) = -\frac{b-a}{80} h^4 f^{iv}(x_m)$$

$$(a < x_m < b).$$

10.8.4 Error in Weddle's Rule

Taking the inverval from x_0 to $x_0 + 6h$ we have

$$I_e = \int_{x_0}^{x_0+6h} f(x)\,dx - g(x_0 + 6h) - g(x_0)$$

$$\text{and } I_a = \frac{3h}{10} \left[f(x_0) + 5f(x_0 + h) + f(x_0 + 2h) + 6f(x_0 + 3h) \right.$$

$$\left. + f(x_0 + 4h) + 5f(x_0 + 5h) + f(x_0 + 6h) \right]$$

Expanding each of these functions using Taylor series and on simplification we get

(9) $$I_a = \frac{3h}{10} \left[20f(x_0) + 60h\,f'(x_0) + 120h^2 f''(x_0) + 180h^3 f'''(x_0) \right.$$

$$\left. + 216h^4 f^{iv}(x_0) + 216h^5 f^{v}(x_0) + \frac{1111}{6} f^{vi}(x_0) \right]$$

(10) $$I_e = 6h\,g'(x_0) + 18h^2 g''(x_0) + 36h^3 g'''(x_0) + 54h^4 g^{iv}(x_0)$$

$$+ \frac{324}{5} h^5 g^{v}(x_0) + \frac{324}{5} h^6 g^{vi}(x_0) + \frac{7776}{140} h^7 g^{vii}(x_0) + \cdots$$

$$= 6hf(x_0) + 18h^2 f'(x_0) + 36h^3 f''(x_0) + 54h^4 f'''(x_0)$$

$$+ \frac{324}{5} h^5 f^{v}(x_0) + \frac{324}{5} h^6 f^{vi}(x_0) + \frac{7776}{140} h^7 f^{vii}(x_0) + \cdots$$

Substituting from (9) and (10)

$$I_e - I_a = -\frac{1}{140} h^7 f^{vi}(x_0) + \cdots$$

(neglecting derivatives of order higher than sixth)

In the case of the integral $\int_a^b f(x)\,dx$

$$\text{Error} = -\frac{1}{140}h^7\left[f^{vi}(x_0) + f^{vi}(x_6) + \ldots + f^{vi}(x_{n-6})\right]$$

If $f^{vi}(x_m)$ is the largest of $f^{vi}(x_0), f^{vi}(x_6), \ldots$, we find that

$$\text{Error} \leq -\frac{1}{140}h^7 \cdot \frac{n}{6} f^{vi}(x_m) = -\frac{b-a}{840} h^6 f^{vi}(x_m)$$

$$(a < x_m < b)$$

Weddle's rule gives the exact value of the integral if $f(x)$ is a polynomial of degree less than six.

10.9 SUMMARY

We have derived the quadrature formulae by taking the Newton's forward formula for interpolation. Similarly, one can derive the quadrature formulae for central and other interpolation formulas. We have seen that the Weddle's rule give us a better approximation than Simpson's rules. But it requires atleast seven consecutive values of the function and is applicable only when the total number of observations is a multiple of six.

In order to measure the truncation error one need to estimate the remainder term. We have estimated the remainder terms in the case of Trapezoidal, Simpson's $\frac{1}{3}$ rule, Simpson's $\frac{3}{8}$ rule and Weddles rule.

10.10 SAMPLE EXAMINATION QUESTIONS

I. Answer the following questions in detail

1. a) Explain Numerical integration and derive the general quadrature formula.
   b) Find the area under the curve $y = \sin x$ from $x = 0$ to $x = \frac{\pi}{2}$.

2. a) Derive Simpson's and trapezodial rules.
   b) Find an approximate value of $\int_0^6 y\,dx$ from the following table

x :	0	1	2	3	4	5	6
y :	14.6	16.1	17.6	19.0	20.4	21.7	23.0

   using Weddle's Rule

3. a) Obtain Simpson's $\frac{3}{8}$ ths rule and Weddle's rule.
   b) Find approximate value of $\int_0^{12} \frac{dx}{1+x^2}$ by using these two rules.

II. Briefly answer the following.

1. Find an approximate value of π by applying Simpson's 3/8ths rule to $\int_0^1 \frac{dx}{1+x^2}$.

2. Evaluate $\int_3^5 \frac{1}{2+x^2}\,dx$ by dividing the range into 8 equal parts.

3. Find $\int_0^1 f(x)\,dx$ from the following table

x :	0	0.25	0.50	0.75	1
$f(x)$:	0.5	0.4794	0.4594	0.4398	0.4207

4. Explain different types of errors that arise in numerical methods.

   Estimate the error in $\int_a^b f(x)\,dx$ using Weddle's rule.

5. Estimate the errors in the value of a definite integral which is evaluated using Simpson's 1/3 rule or Simpson's 3/8 rule.

6. What is the order of the error that arises when trapezoidal rule is employed for evaluating $\int_a^b f(x)\,dx$.

UNIT–11 : EULER TRANSFORMATION AND ASYMPTOTIC EXPANSIONS

Contents

11.1 AIMS AND OBJECTIVES

After going through this unit, you will be able to (i) find the sum of a given series, which may converge slowly, using first term and its leading differences by (a) Euler transformation method and (b) Euler - Maclaurin series methods. (ii) Define an asymptotic series and expand a given function in asymptotic series and identify the term from which we truncate the series to get the most accurate results. (iii) If $f(x)$ is a function which can be expanded as a covergent series in x, then you will be able to use Lagrange series formula to express x as a power series in y from a given implicit relation of the form $x = y f(x)$.

11.2 INTRODUCTION

In mathematics, very often we come across different types of series. Mostly we will be interested in the convergent series. In the case when the given series converges slowly, it is very difficult to compute the sum. In such cases we follow the Euler - transformation method which uses the first term of the series and its leading differences. We have also derived the Euler - Maclaurin series formula to evaluate the given series. This method establishes a relation between an integral and sum of the series of a given function. Usually, the Euler-Maclaurin expansion provides an asymptotic series. An asymptotic series is an infinite series which converges for a certain number of terms and then begins to diverge. In computing with such a series, it is important to know what term to stop with inorder to get the most accurate result. Also we have obtained Lagrange series formula to express x as a poweer series in y from a given implicit relation of the form $x = y f(x)$.

11.3 EULER TRANSFORMATION

Consider the alternating series

$$S = u_1 - u_2 + u_3 - u_4 + \ldots$$

Where (i) $u_n > 0$, (ii) $u_n > u_{n+1}$, (iii) $u_n \to 0$ as $n \to \infty$

In the case when the series converges slowly it is not of much use for the computation of the sum. Hence we try to evaluate S in terms of the first term and its leading differences.

The difference table is

u_1				
	Δu_1			
u_2		$\Delta^2 u_1$		
	Δu_2		$\Delta^3 u_1$	.....
u_3		$\Delta^2 u_2$	⋮	⋮
	Δu_3	⋮	⋮	⋮
u_4	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮

Here

$$u_2 - u_1 = \Delta u_1. \text{ Therefore}$$

$$u_2 = (1 + \Delta)\, u_1 = Eu_1$$

Similarly, $u_2 = (1 + \Delta)\, u_2 = E u_2 = E^2 u_1$, etc.,

Hence we can write

$$S = u_1 - Eu_1 + E^2 u_1 - E^3 u_1 + \ldots$$

$$= \left(1 - E + E^2 - E^3 + \ldots\right) u_1$$

$$= \frac{1}{1 + E} u_1 = \frac{1}{2 + \Delta} u_1 = \frac{1}{2}\left(1 + \frac{\Delta}{2}\right)^{-1} u_1$$

$$= \frac{1}{2}\left(1 - \frac{\Delta}{2} + \frac{\Delta^2}{4} - \ldots\right) u_1$$

$$= \frac{1}{2}\left(u_1 - \frac{1}{2}\Delta u_1 + \frac{1}{4}\Delta^2 u_1 - \ldots\right) \quad (1)$$

Formula (1) is called the Euler transformation.

Ex. 1 : Given that $\frac{1}{1^2} - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \ldots = \frac{\pi^2}{12}$, compute the value of $\frac{\pi^2}{12}$ with the help of Euler transformation.

Solution

In order to have the differences smaller let us add the first ten terms separately.

$$\frac{1}{1^2} - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \ldots - \frac{1}{10^2} = 0.81796$$

$$\sum_{x=1}^{10} \frac{(-1)^{n-1}}{x^2} = .81796$$

For the terms after the tenth let us form the difference table.

x	$u = u_x = \frac{1}{x^2}$	Δu	$\Delta^2 u$	$\Delta^3 u$	$\Delta^4 u$	$\Delta^5 u$
11	.0082645					
		−.0013201				
12	.0069444		.0002929			
		−.0010272		−.0000809		
13	.0059172		.0002120		.0000265	
		−.0008152		−.0000544		−.0000102
14	.0051020		.0001576		−.0000163	
		−.0006576		−.0000381		
15	.0044444		.0001195			
		−.0005381				
16	.0039063					

we have

$$\sum_{x=11}^{\infty} \frac{(-1)^{n-1}}{x^2} = \frac{1}{2}\left[.0082645 + \frac{1}{2}(.00132\,01) + \frac{1}{4}(.0002929)\right.$$

$$\left. + \frac{.0000809}{8} + \frac{.0000265}{16} + \frac{.0000102}{32} + \ldots\ldots\right]$$

$$= .0045049.$$

Thus the sum of the series excluding first ten terms = .0045049

$$\text{Hence } \frac{\pi^2}{12} = \sum_{x=1}^{\infty} \frac{(-1)^{n-1}}{x^2} = .81796 + .0045049$$

$$= 0.8224649.$$

Note : In order to obtain this value from the given series in its original form it would have been necessary to add more than 10^2 terms.

11.4 EULER - MACLAURIN SERIES

Suppose $g(x)$ is a function such that

(2) $\qquad \Delta g(x) = f(x)$

(ie.,) $g(x+1) - g(x) = f(x)$ taking $h = 1$

put $x = 0$ we get $g(1) - g(0) = f(0)$

$x = 1$ we get $g(2) - g(1) = f(1)$

$x = 2$ we get $g(3) - g(2) = f(2)$

... ...

$x = n - 1$ $g(n) - g(n-1) = f(n-1)$

(3) Summing we get $g(n) - g(0) = \sum_{n=0}^{n-1} f(n) = \sum_{n=0}^{n-1} f(x)$

By Taylor Series

$$f(c+h) = \sum_{n=0}^{\infty} \frac{f^n(x)}{n!} h^n$$

Writing $c = x$ and $h = 1$

(4) $$f(x+1) = \sum_{n=0}^{\infty} \frac{f^n(x)}{n!} = \left(\sum_{n=0}^{\infty} \frac{D^n}{n!} \right) f(x) = e^D f(x)$$

(5) But $f(x+1) = Ef(x) = (1 + \Delta) f(x)$

From (4) and (5) we have

$$e^D = 1 + \Delta$$

From (2)

$$g(x) = \frac{1}{\Delta} f(x) = \frac{1}{e^D - 1} f(x) = D^{-1} \frac{D}{e^D - 1} f(x)$$

Now $$\frac{D}{e^D - 1} = \frac{1}{1 + \left(\frac{D}{2!} + \frac{D^2}{3!} + \frac{D^3}{4!} + \ldots \right)}$$

$$= 1 - \left(\frac{D}{2!} + \frac{D^2}{3!} + \ldots \right) + \left(\frac{D}{2!} + \frac{D^2}{3!} + \ldots \right)^2 - \ldots$$

$$= 1 - \frac{1}{2} D + \frac{1}{12} D^2 - \frac{1}{720} D^4 + \frac{1}{30240} D^6 - \ldots$$

Hence $$g(x) = \left[D^{-1} - \frac{1}{2} + \frac{1}{12} D - \frac{1}{720} D^3 + \frac{1}{30240} D^5 - \ldots \right] f(x)$$

$$= D^{-1} f(x) - \frac{1}{2} f(x) + \frac{1}{12} f'(x) - \frac{1}{720} f'''(x) + \ldots$$

Writing $x = 0$ we have

$$g(0) = D^{-1} f(0) - \frac{1}{2} f(0) + \frac{1}{12} f'(0) - \frac{1}{720} f'''(0) + \ldots$$

and $x = n$ gives

$$g(n) = D^{-1} f(n) - \frac{1}{2} f(n) + \frac{1}{12} f'(n) - \frac{1}{720} f'''(n) + \ldots$$

Substituting in (3) we get

$$\sum_{x=0}^{n-1} f(x) = g(n) - g(0) = \int_0^n f(x)\,dx - \frac{1}{2}[f(n) - f(0)]$$

$$+ \frac{1}{12}[f'(n) - f'(0)] - \frac{1}{720}[f'''(n) - f'''(0)] + \ldots$$

Additing $f(n)$ both sides we get

$$\sum_{x=0}^{n} f(x) = \int_0^n f(x)\,dx + \frac{1}{2}[f(n) + f(0)] \tag{6}$$

$$+ \frac{1}{12}[f'(n) - f'(0)] - \frac{1}{720}[f'''(n) - f'''(0)] + \ldots$$

Relation (6) is called the Euler-Maclaurin formula.

This relation is more useful in summing the series rather than for evaluating integrals.

Ex. 2 : Use the Euler–Maclaurin formula to find $\sum_{x=1}^{n} x^2$.

Solution :

Taking $f(x) = x^2$ and using (6) we get

$$\sum_{x=0}^{n} x^2 = \int_0^n f(x)\,dx + \frac{1}{2}\left(n^2 + 0\right) + \frac{1}{12}(2n - 0)$$

$$= \frac{n^3}{2} - 0 + \frac{1}{2} n^2 + \frac{n}{6} = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{x=1}^{n} x^2 = \sum_{x=0}^{n} x^2 = \frac{n(n+1)(2n+1)}{6}$$ (as the term vanishes for $x = 0$)

Ex. 3 : Prove that

$$f(a) + f(a+h) + \ldots + f(a+nh) = \frac{1}{h}\int_a^{a+nh} f(x)\,dx + \frac{1}{2}[f(a) + f(a+nh)]$$

$$+ \frac{h}{12}[f'(a+nh) - f'(a)] - \frac{h^3}{720}[f'''(a+nh) - f'''(a)] + \ldots$$

Solution :

Let $F(y) = f(a + yh)$

$$f(a) + f(a+h) + f(a+2h) + \ldots + f(a+nh)$$

$$= \sum_{y=0}^{n} f(a+yh) = \sum_{y=0}^{n} F(y)$$

$$= \int_0^n F(y)\,dy + \frac{1}{2}[F(n) + F(0)] + \frac{1}{12}[F'(n) - F'(0)]$$

$$- \frac{1}{720}[F'''(n) - F'''(0)] + \ldots$$

Since $F'(y) = f'(a+yh) \cdot h$

$F''(y) = f''(a+yh) \cdot h^2$

.....

$$= \int_0^n f(a+yh)\,dy + \frac{1}{12}[f(a+nh) - f(a)]$$

$$\frac{1}{12}[f'(a+nh) - f'(a)]$$

$$- \frac{1}{720}h^3[f'''(a+nh) - f'''(a)] + \ldots$$

Put $x = a + yh$ then

$$\int_0^n F(y)\,dy = \frac{1}{h}\int_a^{a+nh} f(x)\,dx$$

$$\text{Hence } \sum_{r=0}^{n} f(a+rh) = \frac{1}{h}\int_a^{a+nh} f(x)\,dx + \frac{1}{2}[f(a+nh) + f(a)]$$

$$+ \frac{h}{12}[f'(a+nh) - f'(a)] - \frac{h^3}{720}[f'''(a+nh) - f'''(a)] + \ldots$$

Ex. 4 : Compute $f(n) = \sum_{r=1}^{n} \frac{1}{r} - \log n$ for $n = 10^5$

Solution :

From Euler–Maclaurin formula, we have

$$\sum_{r=1}^{n} \frac{1}{r} = \int_{1}^{n} \frac{1}{x}dx + \frac{1}{2}\left[\frac{1}{n} + 1\right] + \frac{1}{12}\left[-\frac{1}{n^2} + 1\right] - \frac{1}{720}\left[-\frac{6}{n^4} + 6\right] + \ldots$$

$$= \log n - \log 1 + \frac{1}{2}(n+1) + \frac{1}{12}\left(-\frac{1}{n^2} + 1\right) - \frac{1}{720}\left[-\frac{6}{n^4} + 6\right] + \ldots$$

$$f(n) = \sum_{r=1}^{n=10^5} \frac{1}{r} - \log n = \frac{1}{2}(1 + .00001)$$

$$+ \frac{1}{12}(1 - .0000000001) - .0083333$$

$$= 0.575005.$$

11.5 ASYMPTOTIC EXPANSIONS

An asymptotic series is an infinite series which converges for a certain number of terms and then begins to diverge. In computing with such a series it is important to know what term to stop with in order to get the most accurate result. We should stop not with the smallest term but with the term just before the smallest; for the error committed is usually less than twice the first neglected term and is therefore least when the first term neglected is the smallest term in the series.

An asymptotic series for a function $f(x)$ is an expression $f(x) = S_n(x) + R_n(x)$ where $S_n(x)$ is a finite series of n terms and $R_n(x)$ is a term such that $\underset{n\to\infty}{\text{Lt}}\ R_n(x) = \infty$ (x fixed) but $\underset{n\to\infty}{\text{Lt}}\ R_n(x) = 0$ (n fixed).

Poincare's definition of an asymptotic series for a cuntion $f(x)$ is a divergent series $\sum_{r=0}^{\infty} a_r x^{-r}$ such that

$$\underset{n\to\infty}{\text{Lt}} \left(\sum_{r=0}^{n} a_r x^{-r} - f(x)\right) x^n = 0 \quad (n \text{ fixed})$$

$$\underset{n\to\infty}{\text{Lt}} \left(\sum_{r=0}^{n} a_r x^{-r} - f(x)\right) x^n = \infty \quad (x \text{ fixed})$$

This definition is quite differnt from that of an ordinary convergent series or divergent series. A rough definition of an asymptotic series is given above. Most of the asymptotic series are generated by integration by parts. This point shall be explaiend by an example.

Suppose we want to evaluate $\int_x^\infty e^{-t^2/2}\,dt$ given that x is large.

One can write

$$\int_x^\infty e^{-t^2/2}\,dt = \int_x^\infty -\frac{1}{t}\left(-t\,e^{-t^2/2}\right)dt$$

$$= \left[-\frac{1}{t}e^{-t^2/2} - \int e^{-t^2/2}\cdot\frac{1}{t^2}dt\right]_x^\infty$$

(treating $\left(-te^{-t^2/2}\right)$ as the second function and integrating by parts)

$$= \frac{1}{x}e^{-x^2/2} - \int_\infty^x e^{-t^2/2}\,dt$$

$$= \frac{1}{x}e^{-x^2/2} - \left\{-\int_x^\infty \frac{1}{t^3}\left(-te^{-t^2/2}\right)dt\right\}$$

$$= \frac{1}{x}e^{-x^2/2} - \left[-\frac{1}{t^3}e^{-t^2/2} - \int \frac{3}{t^4}e^{-t^2/2}\,dt\right]_x^\infty$$

$$= \frac{1}{x}e^{-x^2/2} - \frac{1}{x^3}e^{-x^2/2} + \int_x^\infty \frac{3}{t^4}e^{-t^2/2}\,dt$$

Proceeding in this matter we get

$$\int_x^\infty e^{-t^2/2}\,dt = e^{-x^2/2}\left[\frac{1}{x} - \frac{1}{x^3} + \frac{1.3}{x^5} - \frac{1.3.5}{x^7}\right.$$

$$\left. + \ldots + (-1)^{n-1}\frac{1.3.5\ldots(2n-3)}{x^{2n-1}}\right]$$

$$+ (-1)^n\ R_n$$

where $R_n = 1.3.5\ldots(2n-1)\int_x^\infty e^{-t^2/2}\cdot\frac{1}{t^{2n}}\,dt$

Continuing this process indefinitely we can write

(1) $$\int_x^\infty e^{-t^2/2}\,dt \sim e^{-x^2/2}\left[\frac{1}{x} - \frac{1}{x^3} + \frac{1.3}{x^5} - \frac{1.3.5}{x^7} + \ldots\right]$$

(is asymptotically equal to denoted as ~)

(Note that ~ does not mean convergence)

written in gradual steps

$$\int_x^\infty e^{-t^2/2}\,dt = e^{-x^2/2}\cdot\frac{1}{x} - R_1$$

$$= e^{-x^2/2}\left[\frac{1}{x} - \frac{1}{x^3}\right] + R_2$$

$$= e^{-x^2/2}\left[\frac{1}{x} - \frac{1}{x^3} + \frac{3}{x^5}\right] - R_3$$

... ...

In asymptotic expansions, the behaviour of R_n is most crucia.

(2) $$R_n = 1.3.5 \ldots (2n-1)\int_x^\infty e^{-t^2/2}\cdot\frac{1}{t^{2n}}\,dt$$

$$= \frac{1.3.5 \ldots (2n-1)}{x^{2n+1}}\,(-x^{-1}) - R_{n+1}$$

Here both R_n and R_{n+1} are positive being integrals of positive functions. Hence R_n is less than the first term of the right hand side of (2). This allows us to decide the numbeer of terms to be taken in the asymptotic series. It can be observed that R_n first decreases and then increases and ultimately becomes very large.

Writing $x = 2$, we have

$$\int_2^\infty e^{-t^2/2}\,dt \sim e^{-2}\left[\frac{1}{2} - \frac{1}{2^3} + \frac{1.3}{2^5} - \ldots\right]$$

Here $R_3 < e^{-2}\cdot\frac{1.3.5}{2^7}$ (this is the minimal bound on R_3)

∴ We select the particular finite series

$$\int_2^\infty e^{-t^2/2}\,dt = e^{-2}\left[\frac{1}{2} - \frac{1}{2^3} + \frac{1.3}{2^5}\right] + R_3$$

$$= .46875\,e^{-2} + R_3$$

where $R_3 < .11719\,e^{-2}$

Hence we may write

$$.46875\, e^{-2} < \int_2^{\infty} e^{-t^2/2}\, dt < (.46875 + .11719)\, e^{-2}$$

If x is large the terms of the asymptotic series will become very small before they start to increase. So one will be able to get an excellent value. In an asymptotic series there is a bound on the accuracy possible which is determined by the minimum value of R_n.

It may be noted that Euler-Maclaurin expansion usually provides an asymptotic series.

Ex.1 : Evaluate $\int_4^{\infty} e^{-t^2/2}\, dt$ as accurately as possible.

Solution

$$\int_4^{\infty} e^{-t^2/2}\, dt = e^{-8}\left[\frac{1}{4} - \frac{1}{4^3} + \frac{1.3}{4^5} - \frac{1.3.5}{4^7} + \frac{1.2.5.7}{4^9} - \frac{1.3.5.7.9}{4^{11}} + \frac{1.3.5.7.9.11}{4^{13}} - \frac{1.3.5.7.9.11.13}{4^{15}}\right]$$

The minimal bound on R_n is given by

$$R_3 < e^{-8}\left[\frac{1.3.5.7.9.11.13.15}{4^{17}}\right]$$

$$< e^{-8} \times .000027$$

Hence we write

$$\int_4^{\infty} e^{-t^2/2}\, dt = e^{-8}\left[\frac{1}{4} - \frac{1}{4^3} + \frac{1.3}{4^5} - \frac{1.3.5.7.9.11.13.15}{4^{17}}\right] + R_8$$

$$= e^{-8}\,[.236919] + R_8$$

$$\therefore \int_4^{\infty} e^{-t^2/2}\, dt \simeq .236919\, e^{-8}$$

Ex.2 : Show that

$$\int_x^{\infty} \frac{e^{x-t}}{t}\, dt = \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \ldots + \frac{(-1)^{n-1}\,(n-1)!}{x^n} + (-1)^n R_n$$

where $R_n < \dfrac{n!}{x^{n+1}}$.

Solution

$$\int_x^\infty \frac{e^{x-t}}{t}dt = e^x \int_x^\infty \frac{1}{t}\cdot e^{-t}\,dt$$

$$= e^{-x}\left[-\frac{1}{t}e^{-t} - \int \frac{1}{t^2}e^{-t}\,dt\right]_x^\infty$$

$$= \frac{e^x \cdot e^{-x}}{x} - e^x \int_x^\infty \frac{1}{t^2}e^{-t}\,dt$$

$$= \frac{1}{x} - e^x\left[-\frac{1}{t^2}e^{-t} - \int \frac{2}{t^3}e^{-t}\,dt\right]_x^\infty$$

$$= \frac{1}{x} - \frac{1}{x^2} + e^x \int_x^\infty \frac{2!}{t^3}e^{-t}\,dt$$

Proceeding similarly we get

$$= \frac{1}{x} - \frac{1}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \ldots + \frac{(-1)^{n-1}(n-1)!}{x^n} + (-1)^n R_n$$

where $R_n = \frac{n!}{x^{n+1}} - R_{n+1}$

R_n and R_{n+1} being integrals of positive functions are positive

$$\therefore R_n < \frac{n!}{x_{n+1}}$$

11.6 LAGRANGE SERIES

Suppose $f(x)$ is a function which can be expanded as a convergent series in x in the form $f(x) = a_0 + a_1x + a_2x^2 + \ldots \left(a_0 \neq 0\right)$. ... (3)

It may sometimes be necessary to express x as a power series in y from a given implicit relation of the form $x = yf(x)$. ... (4)

(i.e.,) we want to find constants $b_0, b_1, b_2, \ldots$ such that

$$x = b_0 + b_1y + b_2y^2 + \ldots \quad \ldots (5)$$

From (4), $x = 0$ when $y = 0$. Hence $b_0 = 0$.

Constants b_i can be obtained by differentiating (5) w.r.t. 'x'

$$1 = \left[b_1 + 2b_2y + 3\,b_3y^2 + \ldots\right]\frac{dy}{dx}$$

$$= \sum_{r=1}^{\infty} r\,b_r\,y^{r-1}\,\frac{dy}{dx} \qquad \ldots (6)$$

Multiplying (6) with $[f(x)]^n$, we have

$$[f(x)]^n = \sum_{r=1}^{\infty} r\,b_r\,y^{r-1}\,(f(x))^n\,\frac{dy}{dx} \qquad \ldots (7)$$

Let us evaluate $\frac{d^{n-1}}{dx^{n-1}}[f(x)]^n\Big|_{x=0}$ from the R.H.S. of (7)

one can write

$$[f(x)]^n = [f(0)]^n + (\,)\,x + (\,)\,x^2 + \ldots + (k)\,x^{n-1} + (\,)\,x^n + \ldots$$

$(n-1)$ th derivative of $[f(x)]^n$ at $x = 0$ will give $K \cdot (n-1)!$

Here K is the coefficient of x^{n-1} in $[f(x)]^n$.

Thus from the R.H.S. of (7) $\frac{d^{n-1}}{dx^{n-1}}[f(x)]^n\Big|_{x=0}$ will be

$$= (n-1)!\left[\text{coefficient of } x \text{ in } \sum_{r=1}^{\infty} r\,b_r\,y^{r-1}\,[f(x)]^n\,\frac{dy}{dx}\right]$$

from (4), $f(x) = \frac{x}{y}$

$$\therefore \frac{d^{n-1}}{dx^{n-1}}[f(x)]^n\Big|_{x=0},$$

$$= (n-1)!\left[\text{coefficient of } x^{n-1} \text{ in } \sum_{r=1}^{\infty} r\,b_r\,x^n y^{r-n-1}\,\frac{dy}{dx}\right]$$

Now $y = \frac{x}{f(x)} = x\,[f(x)]^{-1}$

$$= c_1x + c_2x^2 + c_3x^3 + \ldots \text{ say}$$

For $r = n$ the term

$$r\,b_r\,x^n\,y^{r-n-1}\,\frac{dy}{dx} = n\,b_n\,\frac{x^n}{y}\,\frac{dy}{dx}$$

$$= n\,b_n \cdot x^n\,\frac{c_1 + 2c_2x + 3c_3x^2 + \ldots}{c_1x + c_2x^2 + c_3x^3 + \ldots}$$

$$= n\,b_n \cdot x^n\,\frac{1}{x}\left[1 + d_1x + d_2x^2 + \ldots\right]$$

$\therefore$ Coefficient of x^{n-1} is $n\,b_n$ when $r = n$.

For $r \neq n$ the term

$$r\, b_r\, x^n\, y^{r-n-1}\, \frac{dy}{dx} = r\, b_r\, x^n \cdot \frac{1}{r-n}\, \frac{d}{dx}\left(y^{r-n}\right)$$

No power of x, either positive or negitive, has derivative x^{-1}.

$\therefore \frac{d}{dx}\left(y^{r-n}\right)$ does not contain x^{-1}.

Hence for $r \neq n$, the expression $r\, b_r\, x^n\, y^{r-n-1}\, \frac{dy}{dx}$ contains no term of x^{n-1}.

Thus $\sum_{r=1}^{n} r\, b_r\, x^n\, y^{r-n-1}\, \frac{dy}{dx} = \ldots + n\, b_n\, x^{n-1} + \ldots$

$$\text{or } k = n\, b_n$$

$$\therefore \frac{d^{n-1}}{dx^{n-1}} [f(x)]^n \Big|_{x=0} = (n-1)!\,(k) = (n-1)! \cdot n\, b_n = n! \cdot b_n \qquad \ldots (8)$$

Relation (7) enables us to determine the constants b_i which when substituted in (5) gives us x in powers of y.

Ex.3 : Find x in terms of y given that $x = y(1-x)$ using Lagrange formula.

Solution

In this problem $f(x) = 1 - x$

From (8) $b_n = \frac{1}{n!} \frac{d^{n-1}}{dx^{n-1}} [1 - x]^n \Big|_{x=0}$

$= \frac{1}{n!}(-1)^{n-1} \cdot n\,(n-1) \cdot 2\,(1-x)\Big|_{x=0}$

$= (-1)^{n-1}$

Thus $b_1 = 1, b_2 = -1, b_3 = 1, \ldots$

$$\therefore x = y - y^2 + y^3 - \ldots$$

This can be checked algebraically.

$$x = y - yx \text{ ot } x(1+y) = y$$

$$\text{or } x = \frac{y}{1+y} = y\left[1 - y + y^2 - \ldots\right] = y - y^2 + y^3 - \ldots$$

Ex.4 : Express x as a series in y, given that $x = y \cdot e^{-x^2}$

Solution

$[f(x)]^n = \left[e^{-x^2}\right]^n = e^{-nx^2}$ since $f(x) = e^{-x^2}$

Let $Y = e^{-nx^2}$ then

$$\frac{dY}{dx} = -2nx \cdot Y \Rightarrow \frac{dY}{dx} + 2nx\, Y = 0$$

Differentiating this differential equation $n-2$ times

$$Y_{n-1} + 2n\left[x \cdot Y_{n-2} + (n-2)\, Y_{n-3}\right] = 0$$

Put $x = 0$ then we get $Y_{n-1}(0) = -2n(n-2)\, Y_{n-3}(0)$

when n is even clearly the derivatives vanish. When n is odd

$n = 3$ $\quad Y_2(0) = -2.3.1\, Y(0)$

$n = 5$ $\quad Y_4(0) = -2.5.3\, Y_2(0)$

... ... $\quad$ ... ... $\quad$ ... ...

$n = n$ $\quad Y_{n-1}(0) = -2n(n-2)\, Y_{n-3}(0)$

Multiplication yields

$$Y_{n-1}(0) = (-2)^{\frac{n-1}{2}} \left(3^2 - 3.2\right)\left(5^2 - 5.2\right) \ldots \left(n^2 - 2n\right)$$

$\therefore b_1 = 1, b_2 = 0, b_3 = -6, b_4 = 0, b_5 = 45, b_6 = 0, \ldots$

Thus $x = y - (y^3 + 15y^5 - \ldots$

11.7 SUMMARY

The Euler transformation formula and Euler-Maclaurin series formula are obtained to compute the sum of series which converges slowly. We made use of the differences in the computation of a series through Euler transformation. The Euler-Maclaurin formula gives a relation between the sum and integral of a given function. Special care should be taken in truncating an asymptotic series to get most accurate result and to minimise R_n. Lagrange series formula is useful for expressing x as a power series is y from a given implicit relation of the form $x = yf(x)$.

11.8 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following in detail*

1. (a) Explain Euler transformation for summing of a series

   (b) Assuming $\frac{1}{1^2} + \frac{1}{3^2} + \frac{1}{5^2} + \ldots = \frac{\pi^2}{8}$, compute the value of π^2.

2. (a) Derive Euler-Maclaurin formula

   (b) Prove that $\sum_{n=1}^{\infty} n^{-4} = \frac{\pi^4}{90}$ (numerically)

3. (a) Use Euler transformation to evaluate

$1 - \frac{1}{2^3} + \frac{1}{3^3} - \frac{1}{4^3} + \ldots$ upto six decimal places.

(b) Use Euler Maclaurin formula to prove that

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

4. (a) Integrating $\int_x^\infty e^{-t^2/2}\, dt$ by parts show that an asymptotic series can be generated.

(b) Estimate the value of $\int_3^\infty e^{-t^2/2}\, dt$

5. (a) Derive Lagrange formula

(b) Given that $x = y\, e^{-x}$, express x as a power series in y.

II. *Briefly answer the following*

1. Using Euler transformation find $\sum_{n=1}^{1000} \frac{(-1)^{x+1}}{x}$

2. Use Euler-Maclaurin formula to evaluate $\sum_{n=1}^{\infty}$

3. Apply Euler transformation to find $\sum_{x=1}^{\infty} \frac{(-1)^{x+1}}{\sqrt{x}}$

4. Apply Euler-Maclaurin formula to evaluate $\sum_{n=1}^{10^8} \frac{1}{r}$ –8 log 10.

5. Apply Lagrange formula to $x = y - y\, e^{-x}$

6. Apply Lagrange formula to $x = y\left(1 + x^2\right)$

7. Apply Lagrange formula to $x = y \cos x$.

8. Show that $\int_x^\infty \frac{e^{x-t}}{t} dt \simeq \frac{1}{x} - \frac{1}{x^2} + \frac{2}{x^3} - \ldots$

9. Using (4) above compute $\int_4^\infty \frac{dt}{te^t}$.

BLOCK - 4 : NUMERICAL SOLUTIONS OF ORDINARY DIFFERENTIAL EQUATIONS

Introduction

Equations that involve the derivatives of a function of a single variable occur in many branches of applied mathematics. Any situation that concerns the rate of change of one variable with respect to another leads to a differntial equation.

Many classical techniques exist to solve differential equations in terms of elementary functions or in terms of special functions. In practice one encounters number of differential equations which are not easily amenable for obtaining exact solution. In such situations we employ two basic categories of methods :

1. One-step methods, in which the information about the solution curve at one point is used but the solution is not iterated. Euler, Taylor series and Runge-Kutta methods fall under this category.

2. Multi-step methods, in which the next point on the solution curve is estimated using number of iterations to arrive at a sufficiently accurate value. Predictor-corrector methods fall under this category.

UNIT–12 : NUMERICAL SOLUTIONS OF ORDINARY DIFFERENTIAL EQUATIONS

Contents

12.1 AIMS AND OBJECTIVES

After going through this unit you will be able to derive (i) Euler's and its modified methods, (ii) Runge-Kutta methods, (iii) Predictor-corrector methods for solving a first order first degree differential equations and verify the solutions obtained with the known solution (actual solutions, if available) and (2) discuss the merits and demerits of various methods and compare the accuracy of the methods.

12.2 INTRODUCTION

A general equation of first order and first degree is

$$\frac{dy}{dx} = f(x, y).$$

Many classical techniques are available for finding the solution of such equations. Besides all these techniques, some times it happens that a differential equation can not be solved at all or lead to slutions which are difficult to obtain. But, just as there are methods for finding to any desired degree of accuracy of the roots of any algebraic or transcendental equation having numerical coefficients, so like wise there are methods for finding to any desired degree of accuracy the numerical solution of any ordinary differential equation having numerical coefficient's and given intial conditions. In numerical methods we do not proceed in the hope of getting a relation between x and y, but we find the numerical values of the dependent variable for certin values of the independent variable starting with the initial values. In this unit we explain some of the important methods for solving differential eqautions numerically.

12.3 EULER'S METHOD

It is one of the oldest and simplest method for solving an initial value problem

$$\left.\begin{aligned} \frac{dy}{dx} &= f(x, y) \\ y\left(x = x_0\right) &= y_0 \end{aligned}\right\} \quad \ldots (1)$$

Suppose we are required to find the solutions $y_1, y_2 \ldots, y_n$ successively corresponding to $x = x_1, x_2 \ldots, x_n$. Here $x_n = x_0 + nh$, $n = 1, 2, \ldots$.

Let $y = F(x)$ be the actual solution of $\frac{dy}{dx} = f(x, y)$, the graph of which is a curve in the xy-plane, and since a smooth curve is practically a straight line for a short distance from any point on it, we approximate the curve by the tangent at the point $\left(x_0, y_0\right)$. The equation of the tangent at $\left(x_0, y_0\right)$ is

$$y - y_0 = \left(\frac{dy}{dx}\right)_{\left(x_0, y_0\right)} \left(x - x_0\right) = \left(x - x_0\right) f\left(x_0, y_0\right)$$

$$\text{or } y = y_0 + \left(x - x_0\right) f\left(x_0, y_0\right).$$

Hence the value of y corresponding to $x = x_1$ is given by

$$y_1 = y_0 + \left(x_1 - x_0\right) f\left(x_0, y_0\right).$$

$$\text{or } y_1 = y_0 + h f\left(x_0, y_0\right). \quad \ldots (2)$$

Similarly, approximating the curve in the next interval $\left[x_1, x_2\right]$ by a line through $\left(x_1, y_1\right)$ with slope $f\left(x_1, y_1\right)$ we get

$$y_2 = y_1 + h f\left(x_1, y_1\right).$$

In general, we can obtain

$$y_n = y_{n-1} + h f\left(x_{n-1}, y_{n-1}\right)$$

Thus in Euler's method, the actual curve of solution is apporoximated by a sequence of short straight lines.

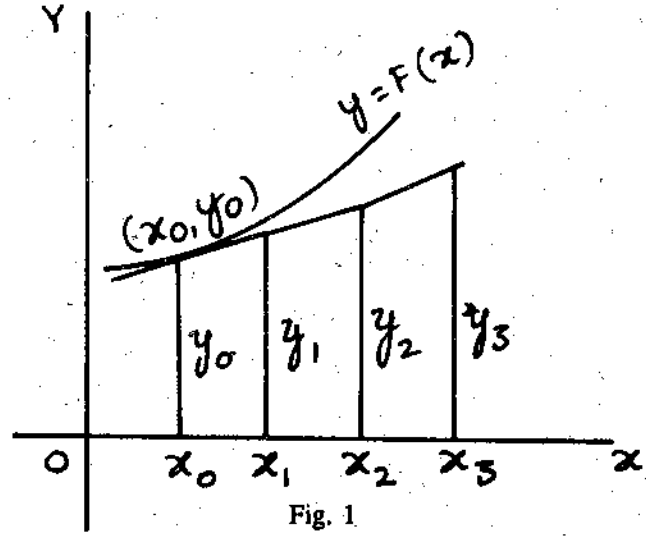


Fig. 1

Ex. 1 : Using Euler's method find $y(1)$ from the differential equation $\frac{dy}{dx} = -\frac{y}{1+x}$, given that $y(0) = 2$.

Solution :

Take $h = 0.2$. It is given that $y_0 = 2$ at $x_0 = 0$.

Now $y_1 = y_0 + hf(x_0, y_0)$,

$$f(x_0, y_0) = \frac{-y_0}{1+x_0} = \frac{-2}{1+0} = -2$$

$$y_1 = y(0.2) = 2 + (0.2)(-2) = 1.6$$

$$y_2 = y(0.4) = y_1 + hf(x_1, y_1) = 1.6 + (.2)\left(-\frac{1.6}{1.2}\right) = 1.334$$

$$y_3 = y(0.6) = y_2 + hf(x_2, y_2) = 1.334 + (.2)\left(-\frac{1.334}{1.4}\right) = 1.143$$

$$y_4 = y(0.8) = 1.143 + (.2)\left(-\frac{1.143}{1.6}\right) = 1.000$$

$$y_5 = y(1.0) = 1.000 + (.2)\left(-\frac{1}{1.8}\right) = .899.$$

Thus the approximate value of y at $x = 1$ (i.e.,) $y(1) = 0.899$.

Remark : Euler's method is of little practical importance due to the fact that if $\frac{dy}{dx}$ changes rapidly over an interval, its value at the beginning of the interval may give a very poor approximation which may be in much error from its true value. In this method the actual solution curve is approximated by the sequence of short straight lines which may deviate from the actual curve significantly. Improved Euler's method avoids this difficulty.

12.4 TAYLOR SERIES METHOD

We make use of the Taylor series expansion in solving the intiail value problem $\frac{dy}{dx} = f(x, y)$, $y(x_0) = y_0$.

Expanding $y(x)$ in the neighbourhood of x_n in a Taylor series,

$$y(x) = y_n + (x - x_n)\, y'_n + \frac{(x - x_n)^2}{2!} y''_n + \ldots \qquad \ldots (3)$$

which converges in the range $x_n \le x \le x\, x_f$.

The coefficients y_n', y_n'', y_n''', ... can be obtaind by differentiating $y' = f(x, y)$ successively.

$$\left.\begin{aligned} y'' &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\cdot\frac{dy}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f \ (\text{here } f = f(x, y)) \\ &= g\ (x, y, y')\ (\text{say}) \\ y''' &= \frac{\partial g}{\partial x} + \frac{\partial g}{\partial y} y' + \frac{\partial g}{\partial y'} y'' = g_1\ (x, y, y', y')\ \text{say} \\ y^{iv} &= \frac{\partial g_1}{\partial x} + \frac{\partial g_1}{\partial y} y' + \frac{\partial g}{\partial y'} y'' + \frac{\partial g}{\partial y''} y''' \text{ etc.} \end{aligned}\right\} \qquad \ldots (4)$$

Substituting $x = x_n$ in (4) we get the derivatives at $x = x_n$. The derivatives are then substituted in (3) to get $y(x)$.

Thus
$$y_{n+1} = y(x_{n+1}) = y_n + h\left[f(x_n, y_n)\right] + \frac{h^2}{2}\left\{\left(\frac{\partial f}{\partial x}\right)_{(x_n, y_n)} + f(x_n, y_n)\left(\frac{\partial f}{\partial y}\right)_{(x_n, y_n)}\right\} \quad \ldots (5)$$

+ terms of order h^3.

As an approximation, we leave the terms of order h^3 and higher order in (5).

Practically, this method is not of much importance because of its need of partial derivatives. If we are interested in a better approximation, then the evaluation of higher order derivatives is needed, which are very much complicated. This formula is useful in judging the degree of accuracy of the approximation given by other methods. We determine the extent to which any other formula agree with the Taylor series expansion. Some methods may agree upto the terms of h and some upto the terms of h^4.

Ex. 2 : Solve $\frac{dy}{dx} = x - y^2$ using Taylor series method for $x = 0.2, 0.4$ given that $y(0) = 1$.

Solution :

To start with we have $x_0 = 0, y_0 = 1$

$$y' = x - y^2 \qquad \therefore y'(0) = 0 - 1 = -1$$

$$y'' = 1 - 2yy' \qquad \therefore y''(0) = 1 - 2(1)(-1) = 3$$

$$y''' = -2yy'' - 2y'^2 \qquad \therefore y'''(0) = -8$$

$$y^{iv} = -2yy''' - 6y'y'' \qquad \therefore y^{iv}(0) = 34$$

$$y(x) = y_0 + (x - x_0)y'_0 + \frac{(x - x_0)^2}{2!}y_0'' + \ldots$$

$$\therefore y(x) = 1 + x(-1) + \frac{x^2}{2!} \cdot 3 + \frac{x^3}{3!}(-8) + \frac{x^4}{4!} \cdot 34 + \ldots$$

$$= 1 - x + \frac{3x^2}{2} - \frac{4x^3}{3} + \frac{17}{12}x^4 + \ldots$$

Taking $x = 0.2$

$$y(0.2) = 1 - 0.2 + \frac{3}{2}(0.2)^2 - \frac{4}{3}(0.2)^3 + \frac{17}{12}(0.2)^4 + \ldots$$

$$= 0.8513$$

To obtain y at $x = 0.4$ we shall take $x_0 = 0.2$ and $y(0.2) = .8513$

$$y'(0.2) = 0.2 - (0.8513)^2 = 0.2 - .7247 = -0.5247$$

$$y''(0.2) = .1066$$

$$y'''(0.2) = -.7322$$

$$y^{iv}(0.2) = 1.5822$$

$$\therefore y(0.4) = 1 - (0.5247)(0.4) + \frac{(0.4)^2}{2!}(.1066) + \frac{(0.4)^3}{3!}(-.7322) + \frac{(0.4)^4}{4!}(1.5822) + \ldots$$

$$= 0.8693$$

Ex. 3 : Solve $y' = 1 + y^2$ given that $y = 0, x = 0$ using Taylor series method for $x = 0.2$ and $x = 0.4$.

Solution :

Initial condition is $x_0 = 0, y_0 = 0$.

$$y' = 1 + y^2 \qquad \therefore y'(0) = 1$$

$$y'' = 2yy' \qquad \therefore y''(0) = 0$$

$$y''' = 2y^2 + 2yy'' \qquad \therefore y'''(0) = 2$$

$$y^{iv} = 6y'y'' + 2yy''' \qquad \therefore y^{iv} = 0$$

$$y^{v} = 8y'y'' + 6y''^2 + 2yy^{iv} \qquad y^{v}(0) = 16$$

$$y(x) = 0 + \frac{x}{1!} \cdot 1 + \frac{x^2}{2!} \cdot 0 + \frac{x^3}{3!} \cdot 2 + \frac{x^4}{4!} \cdot 0 + \frac{x^5}{5!} \cdot 16 + \ldots$$

$$= x + \frac{x^3}{3} + \frac{2x^5}{15} + \ldots$$

$$y(0.2) = 0.2027.$$

For obtaining $y(0.4)$ we start with

$x_0 = 0.2$, and $y_0 = y(0.2) = .2027$

$$\text{Now } y'(0.2) = 1 + [(.2027)]^2 = 1.0411$$

$$y''(0.2) = 2 \times .2027 \times 1.0411 = .4221$$

$$y'''(0.2) = 2(1.0411)^2 + 2(.2027)(.4221) = 2.3389$$

$$y(x) = y_0 + \frac{(x - x_0)}{1!} y_0' + \frac{(x - x_0)^2}{2!} y_0'' + \frac{(x - x_0)^3}{3!} y''' + \ldots$$

$$\text{Here } x - x_0 = 0.4 - 0.2 = 0.2$$

$$\therefore y(0.4) = .2027 + (0.2)(1.0411) + \frac{(0.2)^2}{2!}(.4221) + \frac{(0.2)^3}{3!}(2.3389) + \ldots$$

$$= 0.4224.$$

12.5 HEUN'S METHOD OR IMPROVED EULER METHOD

In this method the average of the slopes at (x_n, y_n) and $(x_n + h, y_n + hy_n')$ is used to calculate y_{n+1} with the Euler's formula. Geometrically explained the method is as follows.

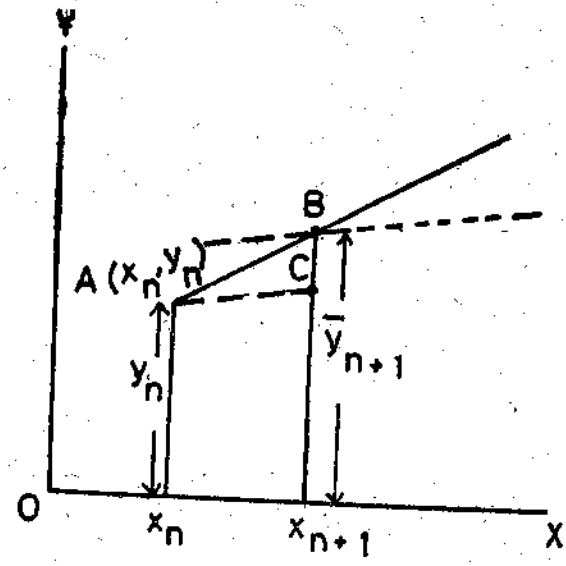


Fig. 2

The solution of $\frac{dy}{dx} = f(x, y)$, $y(x_0) = y_0$ is to be obtained at $x = x_{n+1}$ using the information at $x = x_n$. For this draw a straight line through A (x_n, y_n) with a solpe $f(x_n, y_n)$. Let it cut the ordinate $x = x_{n+1}$ at B $\left(x_{n+1}, \overline{y_{n+1}}\right)$.

Then $\overline{y_{n+1}} = y_n + hf(x_n, y_n)$ from Euler's method.

Now we determine the slope $\frac{dy}{dx}$ of the solution curve at $\left(x_{n+1}, \overline{y_{n+1}}\right)$ which is $f\left(x_{n+1}, \overline{y_{n+1}}\right)$. Let the straight line through A drawn with slope equal to the average of $f(x_n, y_n)$ and $f\left(x_{n+1}, \overline{y_{n+1}}\right)$ cut the ordinate $x = x_{n+1}$ at C. They y – coordinate of C gives the solution at $x = x_{n+1}$.

$$\text{Thus } y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f\left(x_{n+1}, \overline{y_{n+1}}\right)\right] \quad \ldots (6)$$

This method called (Heun's method) involves the computation of f at (x_n, y_n) and $\left(x_{n+1}, \overline{y_{n+1}}\right)$ only.

We shall show that this method is of 2nd order.

From (6), $y_{n+1} = y_n + \frac{h}{2} f(x_n, y_n)$

$$+ \frac{h}{2}\left[f(x_n, y_n) + h\frac{\partial f(x_n, y_n)}{\partial x} + hf(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right] + \ldots$$

(using Taylors expansion)

$$y_{n+1} = y_n + hf(x_n, y_n)$$

$$+ \frac{h^2}{2}\left(\frac{\partial f(x_n, y_n)}{\partial x} + f(x_n, y_n)\frac{\partial f(x_n, y_n)}{\partial y}\right) + \ldots$$

which shows that this formula (1) agrees upto terms of order h^2 compared with Taylor series method of formula.

12.6 MODIFIED EULER'S METHOD (OR IMPROVED POLYGON METHOD)

In the earlier method the average of slopes at two different points is used where as in this method we shall compute the slope at the middle of two points. Themethod is as follows.

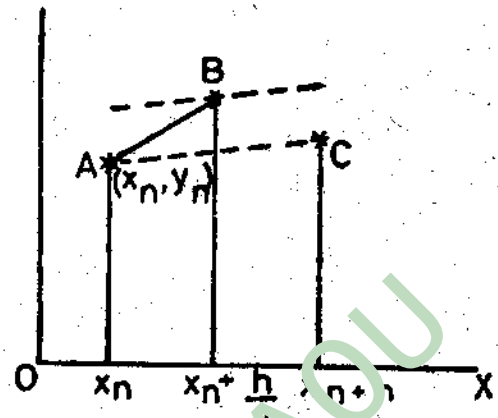


Fig. 3

Draw, through A (x_n, y_n), a line with slope $f(x_n, y_n)$ meeting the ordinate $x = x_n + \frac{h}{2}$ at B. Then the point B is

$$\left(x_n + \frac{h}{2},\ y_n + \frac{h}{2} f(x_n, y_n)\right)$$

Let the line through A with slope

$f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)\right)$ meet the ordinate

$$x = x_n + h \text{ at C.}$$

The ordinate y of C gives the solution of differential equation at $x = x_n + h$. Thus

(2) $$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)\right)$$

As in the case of the earlier method it can be shown that this method agrees upto terms of order h^2 with Taylor series method.

Ex. 1 : Given $\frac{dy}{dx} = \frac{y - x}{y + x}$, $y(0) = 1$ find $y(0.1)$ using

i) Improved Euler's method and ii) Modified Euler's method.

Solution :

i) $$f(x, y) = \frac{y - x}{y + x} \text{ and } y_0 = 1 \text{ when } x_0 = 0$$

Taking $h = 0.1$, we have

$$f(x_0, y_0) = \frac{1-0}{1+0} = 1 = \text{slope at } (x_0, y_0)$$

$$\overline{y_1} = y_0 + hf(x_0, y_0) = 1 + (0.1)\,1 = 1.1$$

$$f(x_1, \overline{y_1}) = f(0.1, 1.1) = \frac{1.1-0.1}{1.1+0.1} = \frac{1}{1.2} = .833$$

From Improved Euler's method

$$y_1 = y_0 + \frac{h}{2}\left[f(x_0, y_0) + f(x_1, \overline{y_1})\right]$$

$$= 1 + \frac{0.1}{2}(1 + .833) = 1.0916$$

ii) From Modified Euler's method

$$y_1 = y_0 + hf\left[x_0 + \frac{h}{2}, y_0 + \frac{h}{2}f(x_0, y_0)\right]$$

$$= 1 + (0.1)f\left(\frac{0.1}{2}, 1 + \frac{0.1}{2} \times 1\right) = 1 + (0.1)f(0.5, 1.05)$$

$$= 1.0909$$

Ex. 2 : **Solve the** differential equation $y' = \frac{1}{y}$ fo $x = 0.5, 1.0, 1.5$ and 2 given that $y = 1$ when $x = 0$.

Solution :

For improved polygon me.ic.l th. iterative formula is

$$y_{n+1} = y_n + hf\left[x_n + \frac{h}{2}, y_n + \frac{h}{2}f(x_n, y_n)\right]$$

The initial point is $x_0 = 0, y_0 = 1$. Take $h = 0.5$ then $x_1 = 0.5$ and $f(x_0, y_0) = 1$. From the above **formula**

$$y(0.5) = y_1 = y_0 + hf\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}f(x_0, y_0)\right)$$

$$= 1 + (0.5)f(0.25, 1 + (0.25) \times 1)$$

$$= 1 + 0.5 \times \frac{1}{1.5} = 1.333$$

$$f(x_1, y_1) = \frac{1}{0.5 + 1.333} = .5456$$

$$y(1.0) = y_2 = y_1 + hf\left[x_1 + \frac{h}{2}, y_1 + \frac{h}{2}f(x_1, y_1)\right]$$

$$= 1.333 + (0.5)f[0.75, 1.333 + (0.25)(.5456)]$$

$$= 1.333 + (0.5) \times \frac{1}{0.75 + 1.4694} = 1.558$$

$$y(1.5) = y_3 = y_2 + hf\left(x_2 + \frac{h}{2}, y_2 + \frac{h}{2}f(x_2, y_2)\right)$$

$$f(x_2, y_2) = \frac{1}{1.0 + 1.558} = .391$$

$$\therefore y(1.5) = 1.558 + (0.5) f(1.25, 1.6557) = 1.7301$$

$$y(2.0) = y_4 = y_3 + hf\left(x_3 + \frac{h}{2}, y_3 + \frac{h}{2} f\left(x_3, y_3\right)\right)$$

$$= 1.7301 + (0.5) f(1.75, 1.8075) = 1.8707$$

12.7 GENERAL SECOND ORDER RUNGE - KUTTA METHOD

The Runge-Kutta methods have the following useful properties :

(i) To evaluate y_{n+1}, the methods need information only at $\left(x_n, y_n\right)$.

(ii) The methods do not involve the derivatives of $f(x, y)$ such as in Taylor series.

(iii) The methods agree with Taylor series solution upto the terms of h^i, where i differens from method to method and it is known as the order of that Runga-Kutta method.

Since Euler's method and its improved and modified forms satisfy all the three properties they can be termed as Runge-Kutta methods of first and second order respectively.

The two second order Runge-Kutta Methods can be generalized in the following way.

Both the formulae are given by an expression of the form

(3) $$y_{n+1} = y_n + hg\left(x_n, y_n, h\right)$$

where $g\left(x_n, y_n, h\right) = a_1 f\left(x_n, y_n\right) + a_2 f\left(x_n + b_1 h, y_n + b_2 h\, y_n'\right)$ and

$$y_n' = f\left(x_n, y_n\right)$$

For Heun's method $a_1 = a_2 = \frac{1}{2}$ and $b_1 = b_2 = 1$.

and for the improved polygon method $a_1 = 0, a_2 = 1, b_1 = b_2 = \frac{1}{2}$.

If $f(x, y)$ has contineous second order partial derivatives, the Taylor series expansion for $f(x, y)$ is

$$f(x, y) = f\left(x_n, y_n\right) + \left(x - x_n\right) f_x\left(x_n, y_n\right) + \left(y - y_n\right) f_y\left(x_n, y_n\right)$$

(4) $$+ \frac{1}{2}\left[\left(x - x_n\right)^2 f_{xx}\left(\xi_0, \eta_0\right) + 2\left(x - x_n\right)\left(y - y_n\right) f_{xy}\left(\xi_0, \eta_0\right)\right.$$

$$\left. + \left(y - y_n\right)^2 f_{yy}\left(\xi_0, \eta_0\right)\right] + \ldots$$

where $\xi_0 \in \left(x_n, x\right)$ and $\eta_0 \in \left(y_n, y\right)$,

and $f_x, f_y \ldots$ are partial derivatives

Let $x = x_n + b_1 h$ and $y = y_n + b_2 hf$.

Then $f\left(x_n + b_1 h, y_n + b_2 hf\right) = f + b_1 hf_x + b_2 hff_y + O\left(h^2\right)$

Relation (3) can be expressed as

(5) $$y_{n+1} = y_n + h\left(a_1 f + a_2 f + h\left\{a_2 b_1 f_x + a_2 b_2 ff_y\right\} + O\left(h^3\right)\right)$$

comparing this with relation (5) of Taylor series method, we get

(6) $a_1 + a_2 = 1$ (if terms in hf are to agree) comparing terms in $h^2 f_x$ and $h^2 ff'_y$ respectively

(7) $a_2 b_1 = \frac{1}{2}$ and $a_2 b_2 = \frac{1}{2}$

As we have three equations in four parameters we may choose one of the parameters arbitrarily (except zero).

For example let $a_2 = \omega \neq 0$, then $a_1 = 1 - \omega$ and $b_1 = b_2 = \frac{1}{2\omega}$

Thus (3) reduces to

(8) $$y_{n+1} = y_n + h\left[(1-\omega) f\left(x_n, y_n\right)\right.$$

$$\left. + \omega f\left(x_n + \frac{h}{2\omega}, y_n + \frac{h}{2\omega} f\left(x_n, y_n\right)\right)\right] + O\left(h^3\right)$$

This is the most general Second order Runge-Kutta method

For $\omega = \frac{1}{2}$ we get Heun's method

and $\omega = 1$ gives the modified Euler method

12.8 RUNGE KUTTA FOURTH ORDER FORMULA

The Taylor series expansion, truncating beyond h^4 terms gives

$$y_{i+1} = y\left(x_i + h\right) = y_i + \frac{h}{1!} y_i' + \frac{h^2}{2!} y_i'' + \frac{h^3}{2!} y_i''' + \frac{h^4}{2!} y^{iv}$$

Now $y_i' = f\left(x_i, y_i\right)$

$$y_i'' = f_x\left(x_i, y_i\right) + f\left(x_i, y_i\right) f_y\left(x_i, y_i\right) = f_x + ff_y$$

$$y_i''' = f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_y\left(f_x + ff_y\right)$$

$$y_i^{iv} = f_{xxx} + 2ff_{xxy} + f^2 f_{xyy} + f^2 f_{yyy}$$
$$+ f_y\left(f_{xx} + 2ff_{xy} + f^2 f_{yy}\right) + 3\left(f_x + ff_y\right)\left(f_{xy} + ff_{yy}\right)$$
$$+ f_y^2\left(f_x + ff_y\right)$$

$$\therefore y_{i+1} = y_i + hf + \frac{h^2}{2}\left(f_x + ff_y\right) + \frac{h^2}{6}\left[f_{xx} + 2ff_{xy} + f^2 f_{yy}\right.$$

(1) $$\left. + f_y\left(f_x + ff_y\right)\right] + \frac{h^4}{24}\left[f_{xxx} + 3ff_{xxy} + 3f^2 f_{xyy}\right.$$

$$+ f^2 f_{yyy} + f_y \left(f_{xx} + 2ff_{xy} + f^2 f_{yy}\right) + 3\left(f_x + ff_y\right)$$
$$\left(f_{xy} + ff_{yy}\right) + f_y{}^2 \left(f_x + ff_y\right)\Big]$$

Let us define

(2) $$k_1 = hf(x, y), k_2 = hf\left(x + mh, y + mk_1\right),$$
$$k_3 = hf\left(x + nh, y + nk_2\right), k_4 = hf\left(x + ph, y + pk_3\right)$$

Now our aim is to express y_{i+1} in the form

(3) $$y_{i+1} = y_i + ak_1 + bk_2 + ck_3 + dk_4$$

Denoting $g_1 = f_x + ff_y$, $g_2 = f_{xx} + 2ff_{xy} + f^2 f_{yy}$

$$g_3 = f_{xxx} + 3ff_{xxy} + 3f^2 f_{xyy} + f^2 f_{yyy}$$

and using Taylor series (2) can be written as

(4)
$$\begin{cases} k_1 = hf \\ k_2 = h\left[f + mhg_1 + \dfrac{m^2h^2}{2} g_2 + \dfrac{m^3h^3}{6} g_3 + \ldots\right] \\ k_3 = h\left[f + nhg_1 + \dfrac{h^2}{2}\left(n^2 g_2 + 2mn f_y g_1\right)\right. \\ \quad \left. + \dfrac{h^3}{6}\left(n^2 g_3 + 3m^2 n f_y g_2 + 6mn^2 f_y' g_1 + \ldots\right)\right] \\ k_4 = h\left[f + phg_1 + \dfrac{h^2}{2}\left(p^2 g_2 + 2np f_y g_1\right)\right. \\ \quad \left. f\ \dfrac{h^3}{6}\left(p^3 g_2 + 3n^2 p f_y g_2 + 6np^2 f_y' g_1 + 6mnp f_y{}^2 g_1\right) + \ldots\right] \end{cases}$$

Substituting (4) in (3) and then comparing (1) and (3), we get on equating the coefficients of corresponding expressions

(5)
$$\begin{cases} a + b + c + d = 1 & cmn + dnp = \dfrac{1}{6} \\ m + cn + dp = \dfrac{1}{2} & cmn^2 + dnp^2 = \dfrac{1}{8} \\ bm^2 + cn^2 + dp^2 = \dfrac{1}{3} & cm^2 n + dn^2 p = \dfrac{1}{12} \\ bm^3 + cn^3 + dp^3 = \dfrac{1}{4} & dmnp = \dfrac{1}{24} \end{cases}$$

Any solution of (5) will work. Suppose we take $m = n = \frac{1}{2}$ and $p = 1$

Then $a = d = \frac{1}{6}$ and $b = c = \frac{1}{3}$

(2) then gives

$$k_1 = hf(x, y), k_2 = hf\left(x + \frac{h}{2}, y + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x + \frac{h}{2}, y + \frac{k_2}{2}\right), k_4 = hf\left(x + h, y + k_3\right)$$

Equation (3) gives

$$y_{i+1} = y_i + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \quad (6)$$

This is called the Runge-Kutta fourth order formula for solving the differential equation $\frac{dy}{dx} = f(x, y), y\left(x_0\right) = y_0$ which needs the calculation of f at four points and which agrees with (1) upto and including the terms in h^4.

Ex. 1 : Solve the differential equation $y' = x^2 + y^2, y(0) = 1$ for $x = 0.2$

Solution : Taking $h = 0.2$

$$y(0.2) = y_1 = y_0 + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

Now $k_1 = hf\left(x_0, y_0\right) = (0.2)\left(0^2 + 1^2\right) = 0.2$

$$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$$

$$= (0.2)\left[(.1)^2 + (1.1)^2\right] = 0.244$$

$$k_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right)$$

$$= (0.2)\left[(0.1)^2 + (1.122)^2\right] = 0.254$$

$$k_3 = hf\left(x_0 + h, y_0 + k_3\right)$$

$$= (0.2)\left[(0.2)^2 + (1.254)^2\right] = 0.323$$

$$\therefore y(0.2) = 1 + \frac{1}{6}(0.2 + .488 + .508 + .323) = 1.253.$$

12.9 PREDICTOR-CORRECTOR METHODS

First we shall discuss second order Predictor–corrector method. The simplist of this type is the Heun's method that has been already discussed

$$y_{n+1} = y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(x_{n+1}, y_{n+1}\right)\right] \quad (7)$$

From Euler's formula we have

$$y_{n+1}^{(0)} = y_n + hf\left(x_n, y_n\right) \quad (8)$$

the superscript (0) indicates the initial approximation.

Thus the value y_{n+1} is predicted from Euler's formula.

Substituting y_{n+1} value from (8) in (7) gives us the correction for y_{n+1}

The corrected value of y_{n+1} is given by

(9) $$y_{n+1}^{(1)} = y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(x_{n+1}, y_{n+1}^{(0)}\right)\right]$$

To obtain a better approximation we evalute $f\left(x_{n+1}, y_{n+1}^{(1)}\right)$ and substitute in (7). This process can be repeated till two successive iterative values agree to the desired accuracy.

Thus the iterative formula is

(1) $$y_{n+1}^{(m)} = y_n + \frac{h}{2}\left[f\left(x_n, y_n\right) + f\left(x_{n+1}, y_{n+1}^{(m-1)}\right)\right]$$

12.10 MILNE'S METHOD

In this method also we predict the value of y_{n+1} and then improve it by applying correction.

Newton's forward interpolation formula is

$$g(x) = g\left(x_0 + uh\right) = g\left(x_0\right) + \frac{u}{1}\Delta g\left(x_0\right) + \frac{u(u-1)}{2!}\Delta^2 g\left(x_0\right) + \frac{u(u-1)(u-2)}{3!}\Delta^3 g\left(x_0\right) + \ldots$$

Taking $g(x) = y'$ and $g\left(x_0\right) = y_0'$ this formula becomes

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

Integrating both sides from x_0 to $x_0 + h$ we get

$$\int_{x_0}^{x_0+4h} y'\,dx = h\int_0^4 \left(y_0' + u\,\Delta y_0' + \frac{u(u-1)}{2!}\Delta^2 y_0' + \frac{u(u-1)(u-2)}{3!}\Delta^3 y_0'\left(x_0\right) + \ldots\right) du$$

Here $dx = h\,du$

$$\therefore y_{x_0+4h} - y_0 = h\left[4y_0' + 8\Delta y_0' + \frac{20}{3}\Delta^2 y_0' + \frac{8}{3}\Delta^3 y_0'\right]$$

taking upto third order differences only

$$\text{(i.e.,)}\quad y_4 - y_0 = \frac{4h}{3}\left[2y_1' - y_2' + 2y_3'\right]$$

$$\text{or}\quad y_4 = y_0 + \frac{4h}{3}\left(2y_1' - y_2' + 2y_3'\right)$$

Thus we have

(12) $$y_{n+1} = y_{n-3} + \frac{4h}{3}\left[2y_{n-2}' - y_{n-1}' + 2y_n'\right]$$

This is Milne's Predictor formula.

Now let us integrate (11) from x_0 to $x_0 + 2h$

$$\int_{x_0}^{x_0+2h} y'\,dx = h\int_0^2 \left(y_0 + u\,\Delta y_0' + \frac{u(u-1)}{2!}\Delta^2 y_0' + \ldots\right) du$$

$$\text{(i.e.,)}\quad y_2 - y_0 = h\left[2y_0' + 2\Delta y_0' + \frac{1}{3}\Delta^2 y_0'\right]$$

taking only upto second order differences

$$\text{or}\quad y_2 = y_0 + \frac{h}{3}\left(y_0' + 4y_1' + y_2'\right)$$

Thus we have Milne's Corrector formula

$$y_{n+1} = y_{n-1} + \frac{h}{3}\left(y_{n-1}' + 4y_n' + y_{n+1}'\right) \tag{13}$$

In the right hand side of (13), y_{n+1} obtained from (12) is utilised. Writing in terms of f the Predictor-Corrector formulae are

$$\overline{y_{n+1}} = y_{n-3} + \frac{4h}{3}\left(2f_{n-2} - f_{n-1} + 2f_n\right)$$

$$\text{and}\quad y_{n+1} = y_{n-1} + \frac{h}{3}\left(f_{n-1} + 4f_n + \overline{f_{n+1}}\right)$$

the bar indicating that the value y_{n+1} from Predictor formula is used in the Corrector formula.

Note that this method can only be used after having the information at three earlier points.

Ex. 2 : Solve $\frac{dy}{dx} = x + y$, $y(0) = 1$ for $x = 2.0, 2.5$ using Milne's method.

Solution :

To use Milne's formula we must first obtain y and $\frac{dy}{dx}$ for $x = 0.5$, 1.0 and 1.5 by Euler's method (one can use Taylor series method or second order Runge-Kutta methods)

Taking $x_0 = 0$, $y_0 = 1$ and $h = 0.5$

$$y_1 = y(0.5) = y_0 + hf\left(x_0, y_0\right) = 1 + (0.5)(1) = 1.5$$

$$y_2 = y(1.0) = y_1 + hf\left(x_2, y_2\right) = 2.5 + (0.5)(3.5) = 4.25$$

Thus we have

x :	0	0.5	1.0	1.5
y :	1	1.5	2.5	4.25
$\frac{dy}{dx}$:	1	2.0	3.5	5.75

The Predictor formula is

$$y_{n+1} = y_{n-3} + \frac{4h}{3}\left(2y_{n-2}' - y_{n-1}' + 2y_n'\right)$$

Put $n = 3$, $h = 0.5$ we have

$$y(2.0) = y_4 = 1 + \frac{2}{3}\left[2(2.0) - 3.5 + 2(5.75)\right] = 9.0$$

The Corrector formula is

$$y_{n+1} = y_{n-1} + \frac{h}{3}\left[y_{n-1}' + 4y_n' + y_{n+1}'^{(p)}\right]$$

superscript p indicates the predicted value.

$$\therefore y_4 = y_2 + \frac{h}{3}\left(y_2' + 4y_3' + y_4'^{(p)}\right)$$

$$= 2.5 + \frac{0.5}{3}(3.5 + 4(5.75) + 11.0) = 8.75$$

$$\left(\because y_4'^{(p)} = f(x_4, y_4) = x_4 + y_4 = 2.0 + 9.0 = 11.0\right)$$

Thus the corrected value of $y(2.0) = 8.75$.

Taking $n = 4$,

$$y_5 = y_1 + \frac{4h}{3}\left(2y_2' - y_3' + 2y_4'\right)$$

$$y(2.5) = y_5 = 1.5 + \frac{2}{3}(7.0 - 5.75 + 22) = 17 \text{ (Predicted)}$$

$$y(2.5) = y_5 = y_3 + \frac{h}{3}\left(y_3' + 4y_4' + y_5'^{(p)}\right)$$

$$= 4.25 + \frac{0.5}{3}(5.75 + 44 + 19.5) = 15.79$$

$$= 15.79 \text{ (corrected value)}$$

Ex. 3 : In the above example, calculate $y(0.5)$ by reducing the step-size h to 0.1.

Solution : In this case $h = 0.1$. Using Euler's formula

$$y_1 = y(0.1) = 1 + (0.1)(1.1) = 1.01$$

$$y_2 = y(0.2) = 1.01 + (0.1)(1.11) = 1.21$$

$$y_3 = y(0.3) = 1.21 + (0.1)(1.41) = 1.351 \simeq 1.35$$

Thus we have

x :	0	0.1	0.2	0.3
y :	1	1.01	1.21	1.35
$\frac{dy}{dx}$:	1	1.11	1.41	1.65

Predicted value

$$y_4^{(p)} = y(0.4)^{(p)} = y_0 + \frac{4h}{3}\left(2y_1' - y_2' + 2y_3'\right)$$

$$= 1 + \frac{0.4}{3}(2.22 - 1.41 + 3.3)$$

$$= 1.548.$$

Corrected value

$$y_4^{(c)} = y_2 + \frac{h}{3}\left(y_2' + 4y_3' + y_4'^{(p)}\right)$$

$$= 1.21 + \frac{0.1}{3}(1.41 + 6.6 + 1.548)$$

$$= 1.5286.$$

$$y_5^{(p)} = y(0.5)^{(p)} = 1.01 + \frac{0.4}{3}(2.82 - 1.65 + 3.86)$$

$$= 1.681$$

$$y_5^{(c)} = y(0.5)^{(c)} = 1.35 + \frac{0.1}{3}(1.65 + 7.7144 + 2.181)$$

$$= 1.735$$

12.11 SUMMARY

We have discussed various methods for solving numerically, the first order first degree initial value problem. The simplest among them is the Euler's method but gives a crude approximation. The method has been modified by taking the average of slope at two points. The Taylor series method involves the calculation of partial derivatives and hence is not of much practical importance. The Taylor series formula is useful in judging the degree of accuracy of the approximations given by other methods. The improved and modified Euler's methods are termed as second order Runge-Kutta methods. The Runge-Kutta fourth order method is most widely used and this method coincides with the Taylor series solution upto terms of order h^4. In this method, the increments are calculated once for all by means of a definite set of formulas. In predictor-corrector formula, we first predict the value of y_{n+1} by some formula, known as predictor formula, and then recorrect this value by another formula, called corrector formula. Milne's method is also a predictor-corrector method. It requires past four points of the solution of predict the fifth value. The actual curve is approximated by a fourth degree polynomial.

12.12 SAMPLE EXAMINATION QUESTIONS

I. Answer the following questions in detail

i) a) Explain Euler's method of solving a first order first degree ordinary differential equation with a given initial condition.

b) Solve $y' = -2xy^2$, given that $y(0) = 1$, by Euler's method (in the interval $0 \le x \le 1$).

ii) a) Explain Taylor series method of solving an initial value problem.

b) Solve by Taylor series method $\frac{dy}{dx} = x + y$, $y(1) = 0$ for the range $1 \le x \le 1.2$ with $h = 0.1$

iii) a) Explain Heun's method and show that it is a Second-order Runge-Kutta method.

b) Solve the differential equation $y' = x + y$, $y(0) = 1$ for $x = .1, .2$ and.3.

iv) a) Discuss improved polygon method and show that it is a second order Runge-Kutta method.

b) Solve $\frac{dy}{dx} = x - y^2$, $y(0) = 1$ for $x = 0.2$ and compare the result with that using Taylor series method.

v) Discuss general second order Runge-Kutta method and show that Heun's method and modified Euler method are special cases of this.

vi) a) Derive Runge-Kutta fourth order formula for solving the differential equation $\frac{dy}{dx} = f(x, y)$, $y(x_0) = y_0$.

b) Solve $\frac{dy}{dx} = -2xy^2$, $y(0) = 1$ by Runge-Kutta fourth order method taking $h = 0.2$ for the interval [0, 1].

vii) a) Obtain Milne's Predictor-Corrector formula.

b) Apply it to solve the differential equation $\frac{dy}{dx} = x + y^2$, $y(0) = 1$ from $x = .20$ to .30 with $h = 0.05$.

II. Briefly answer the following.

i) Solve by Euler's method $\frac{dy}{dx} = 2xy$, $y(0) = 0.5$ for the range $1 \le x \le 0$.

ii) Find $y(0.5)$ from $y' = 2xy$, $y(0) = 0.5$ using Taylor series method.

iii) Solve $\frac{dy}{dx} = y$, $y(0) = 0$ for $0 < x \le 1.0$ with $h = 0.2$.

iv) Solve $y' = 1\,xy^2$, $y(0) = 1$ for $x = 0.3$

v) Explain Second order Predictor–Corrector method

vi) Solve by second order predictor–Corrector meth d

$\frac{dy}{dx} = -2xy^2$, $y(0) = 1$ for the interval [0, 1.

BLOCK - 5 : PRINCIPLES OF COMPUTER PROGRAMMING - FORTRAN - IV

Introduction

This is the age of Electronics. We are seeing the domination of electronics in all walks of life. One of the most powerful and useful outcome of electronics is the COMPUTER. A Computer is an electronic computational device, which brings out more accurate results in the shortest time. The growth and developments of Computers is one of the powerful and outstanding aspects of the technological revolution.

Now a dyas we are using Computers in almost all fields, in particular in data processing and for scientific and engineering problem solving. Many people have the wrong notion that Computer is a magic device and it will do every thing. Actually, a Computer cannot perform any new operation which cannot be performed by a human. A Computer of todays can perform arithmatic, logical, input and output operations. It can perform millions of computations in few minutes.

A Computer can neither think nor make judgements of its own. Therefore, it is necessary to give a set of detailed instructions which lead to a step by step procedure for solving a problem. To prepare the instruction set, first we have to analyse the problem. A systamatic and careful analysis will give better results. For the analysis of the problems we use flow charts and decision tables. In unit 13 we discuss about the historical development, the need and working principle of a Computer and we discuss in detail with sufficient number for examples the method of flow charting and how the construction of decision tables.

Computers do not have the capability of reading and understanding instructions written in a natural language like Telugu, English or Hindi. Thus it is necessary to express a set of instructions which we call Algorithm in a language understood by the Computer. These languages are called programming languages or Higer level languages or Machine independent languages. Some of the familier languages are FORTRAN, COBAL, PASCAL, BASIC etc. The recent versionof FORTRAN is FORTRAN-IV. In this block we discuss the FORTRAN-IV fundamentals.

Binary number system (Base 2) is important in data processing. Internally all computers store numbers in binary system. We discuss some important base systems and convertion from one system to another in unit 15.

UNIT–13 : THE WORKING PRINCIPLE OF A COMPUTER

Contents

13.1 AIMS AND OBJECTIVES

After going through this unit, you will be able to; (i) follow the working principle of a Computer with the help of a block diagram of a Computer. (ii) draw flow charts and decision tables for the given problems.

13.2 INTRODUCTION

A Computer is an electornic computational device. The growth and developments of coputational devices is one of the powerful and outstanding aspects of the technological revolution. In these days time plays an imprtant role in all walks of life. Every one would expect accurate and very fast response for any querry he makes. The computer has been causing a revolution in the field of data processing to bring out necessary results in the shortest time.

13.2.1 Uses and Need of a Computer

Computers are used in all fields, in particular we all know the applications of computer to scientific and engineering problems. Processing of data by manual methods is highly impossible as the data to be dealt with is increasing rapidly. Machanisation has brought accuracy and speed in calculatons.

To provide facilities like schools, colleges, hospitals or transport, government would require information on who are the possible users of these facilities in order to decide on what facilities are required, i.e., how many beds in a hospital, how many buses are required and at what times etc. A manager of a company would like to know the uptodate performence of each one of his products in terms of sales and profit within minutes. A person who is dealing with purchase and resale of stocks which fluctuates very fast within a day would like to know how a particular stock is behaying and whether it is worthwhile to invest on it during that day. This required necessary information to be

supplied to him in a fraction of a minute and this is possible only with the usage of computers. Similarly we can cite so many exmaples where we can use computers for accuracy and speed in decision making.

In scientific, engineering or research problem solving the amount of calculations required would be enormous. So we have no other go except using computers for quick results in scientific and engineering applications. For example, tracking of satellite is done by processing signals received from the satellite. The frequency at which the signals are received and the amount of calculations necessarily to be performed before determining the position of the satellite will be enormous. These calculations could not have been performed in a fraction of a second had they been done by any means other than by the use of computers.

13.2.2 Hostorical Background and Development

Better machines have been brought into picture such as electronically operated calculaters replacing the hand operated calculaters. Electronically operated calculaters with printing facilities avoid copying mistakes of the operator. As the need for speedy responses is increasing, faster machines were needed. This led to the use of electronic device called computer. Computers possessing memory, speed and accuracy in their performance have brought a lot of change in the human life.

There are two types of computers based on the way they represent information. The first one is Digital computer and the second type is Analog computer. A computer which represents all quantities as numbers is called digital computer. Here computer means digital computer only. A computer which represents quatities as a continuous analog is called an Analog computer. The slide rule is based on making use of continuous analog namely distance for performing calculations.

Computers built upto 1950 used vacuum tubes and are identified as first generation computers. Later on transistors were used in the place of vacuum tubes, which perform the same function as vacuum tube but is smaller, less expensive and requires little power. Computers built using transistors are identified as second generation computers. The later development is third generation computers which uses very small circuits. There have been improvements in third generation computers. A significant improvement is introduction of mini computers.

Many people have the wrong notion that a computer is a magic device and it will do every thing. But this is a wrong notion. A computer cannot perform any new operation which cannot be performed by a human, but it executes operations with high speed. Thus the computer can extend the power of man by performing works perfectly which man may perform imperfectly.

A computer of today can perfrom arithmetic, logical, input and output operations. That is, it can perform the basic operations of addition, subtraction, multiplication and division on finite sets of finite numbers, perform various logical operations such as comparing any two values, read the instructions and data from the user, store these for subsequent use, and produce the results in a suitable form. Not only perform millions of computations in few minutes, it can also handle a large number of significant digits in its arithmetic operations and provide extremely accurate results.

13.2.3 Plan to Write a Program

In order to solve a problem, first we have to analyse the problem. Since a computer can neither think nor make judgements on its own, it is necessary to give a set of detailed instructions to solve the problem, Such a set of instructions which leads to a step by step procedure for solving a problem is called an *Algorithm*. Computer does not have the capability of reading and understanding

instructions written in a natural language like Telugu or English. Thus it is necessary to express the algorithm in a language is called a *program* and the language used for coding is called *programming language*. The most important and difficult task in programming is not the preparation of instructions for the computer to solve a given problem, but a systematic and careful analysis of the whole problem. To analyse, first convert the physical model into an idealized mathematical one and based on this new model, formulate the related mathematical equations and then select a suitable numerical procedure to solve mathematical equations.

13.3 GENERAL STRUCTURE OF A COMPUTER

The computers depending on their hardware design, vary in size, speed and capacity but have the same functional organisation. In order to execute instructions a computer has a number of interconnected units. Any computer will have the following five essential units.

1. Input unit
2. Memory unit
3. Control unit
4. Arithmetic and logical unit
5. Output unit

Input and output units may assume varying forms in different computers. Also, some computers may have several different types of the same unit. Now let us study the working of the computer and the functioning of each of the above five units. For the structure of the computer see the block diagram of the computer (Fig. 1).

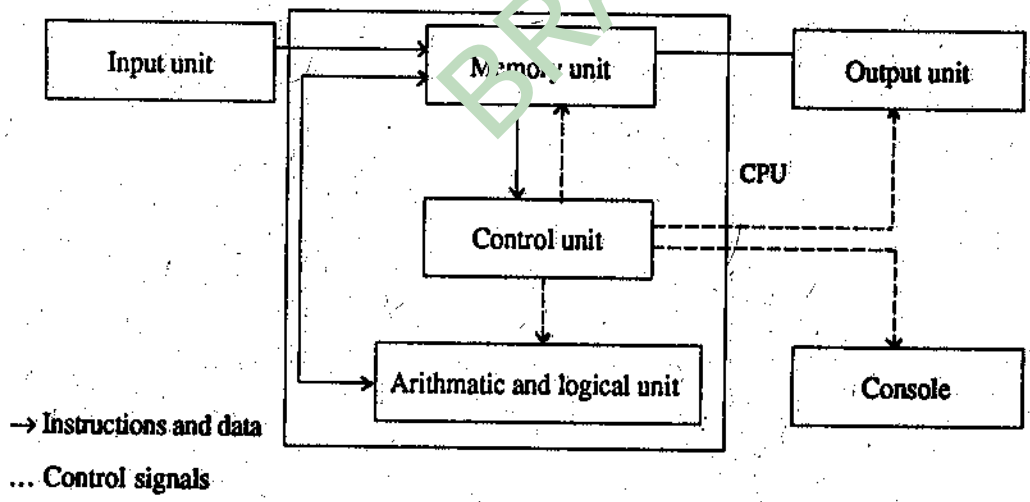


Fig. 1 : *Block Diagram of a Computer*

Through the input unit we communicate the set of instructions and data to the computer. First we feed the set of instructions and data to the input unit. From the input unit it is transferred to the memory unit of the computer. The memory of the computer may be thought of as a collection of labelled pigeon holes. A variable name used in the instruction set is the label of a pigeon hole in the memory. The value of the variable is the contents of the memory location with the corresponding label. When the value of a variable is read from memory for use in a computation it is preserved for future use. When a number is stored in memory it replaces the previous contents of that location. Execution is initiated after the entire program is stored in the memory. To begin execution the first instruction is decoded by the control unit. This unit activates the appropriate units and supervises the

execution of the instructions. After the first instruction is executed the control unit decodes the next instruction and so on. All arithmetic operations, namely, addition, subtraction, multiplication and division are performed in the arithmetic unit under the overall control of the control unit. The instructions are executed sequentially in the order in which they are written. The normal sequential order may be altered by a conditional branch instruction (to be discussed later). Likewise control unit decodes all the instructions and completes execution. The Output unit receives the stored result from the memory unit, converts it into a form the user can understand and through the aid of one or more output devices, prints or produces it in the desired format.

The CPU comprises memory unit, control unit and arithmetic unit. A digital computer also has a console which provides link for human intervention.

13.4 SOME KINDS OF INPUT MEDIA

Some of the many different kinds of input media and devices generally used for feeding data into a computer are

1. Punched card
2. Punched paper tape
3. Magnetic tape
4. Magnetic disks and Floppy disks
5. Magnetic ink character reader
6. Optical character reader
7. Console typewriter
8. Cathode ray tube terminal (CRT)

Now let us know briefly about the above mentioned input media.

Punched Card

The punched card was one of the earliest ways to introduce information into a computer system, and is obsolute now. The most serious drawback in using punched cards as an input/output medium for computers has been the relatively slow speed of the card reading and punching equipment compared to the internal speed of the computer itself.

The standard card measures 18.7 cms × 8.3 cm and has 80 columns. The most common method of representing data on 80-column punched cards in Hollerith code. Each character in this code is represented by a unique combination of punched holes. Figure 2 illustrates a card that has been punched with Hollerith code.

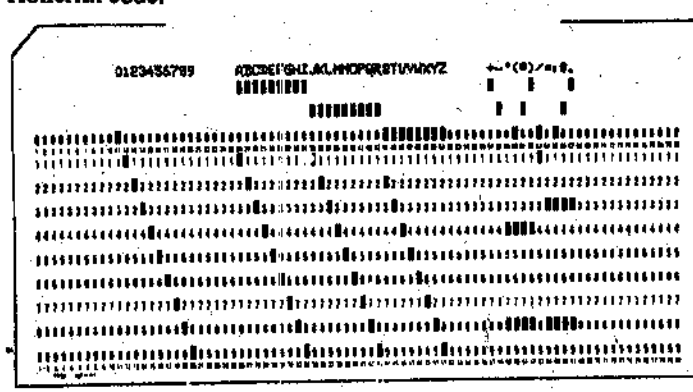


Fig. 2

Punched Paper Tape

The Punched paper tape is a continuous strip of paper. Each character is represented on it by a code made up of a combination of circular holes punched across the width of the tape. Several types of codes consisting of five to eight holes are used in computers.

Since the paper tape allows for easy and compact storage and is priced low, it has wide applications in data communication. The input device for reading punched paper tape is paper tape reader, and has a reading speed of 100-1000 characters per second.

Magnetic Tape

The magnetic tape used in computers is similar to that in audio tape recorders. It is made of this plastic, coated with minute particles of iron oxide which can be magnetized in either of two directions. One polarity is used to represent binary (base 2) bit 1 and the other, binary bit 0. Each character on magnetic tape is represented as a column of binary bits, in a code similar to that used on paper tape. The required data is then recorded in binary form in parallel channels or tracks along the length of the tape. Each channel can record 800-1600 or more binary bits on an inch which is called recording density. An 800 meter reel of magnetic tape can contain over 30 million characters equivalent to about 4,80,000 punched cards.

Reels of magnetic tape are mounted on an input device known as magnetic tape drive. The device can not only read the data from the tape but can also write on it. The rate of information transfer to and from the magnetic tape is 5000-2,40,000 characters per second. Since the magnetic tape records the data magnetically, it has the characteristic of being erasable and re-usable.

Magnetic disk packs and Floppy Disks

Magnetic disk packs and floppy disks (diskets) are popular media for computer storage. Disk packs containing millions of data storage positions are used on larger computer systems. Floppy disks are widely used in micro computer systems, small-scale business computer systems, word processing systems, remote terminals, and data entry systems. The floppy disk is a compact, flexible, magnetic-oxide-coated mylar disk that resembles a 45-rpm phonograph record. It comes in two popular sizes : approximately 20.3 cm (8 inch) and 13.3 cm ($5\frac{1}{4}$ inch) in diameter. The 13.3 cm diameter diskette is called minifloppy disk. The diskette is enclosed in a jacket for protection, with a slot for access by the disk reading and writing mechanism. Information is recorded in a digital fashion on the magnetic surface of the disk while the disk is rotating.

A diskette is easy to handle and easy to store. It is readily interchangable with other diskettes on the device used to process it. Like a magnetic tape, a diskette is reusable. Information can be read from a particular diskette location as often as necessary. When new information is written to a location, the information previously stored at that same location is lost. The flexible floppy disk is a converient low-cost information - recording medium for initial information capture and a popular form of low-cost storage.

Magnetic Ink Character Reader

A high speed input device, the magnitic ink character reader picks up the data directly from the source document. The reading heads of this device produce electrical signals when magnetically recorded characters are passed beneath them. These signals are analysed by special circuits to

determine the characters sensed. Such characters are then transmitted, in groups, to the memory unit. Magnetic ink character reader is widely used by the banking industry to process cheques and deposits. Its normal speed per minute is 750-1600 cheque sized cards or paper documents, each carrying not more than one printed line.

Optical Character Reader (OCR)

The Optical character reader is designed to read numeric and alphabetic characters from printed documents produced by type writters, cash registers, line printers etc. It reads a line of printed characters and converts each into an electrical signal which, when analysed, identifies the character. The reading speed of this device is generally 100-500 characters per second.

Console Type Writer

The external control centre of a computer, the console typewriter is mainly used for communication between the computer operator and the computer. It is similar to an electric type writer but contains, in addition, a series of switches, special keys, and lights. The operator can use it for entering the instructions into the control unit and for monitoring the computer. The console type writer also allow the computer to print and communicate information about a program inside the computer, ie., error messages and completion of operations. This device has a low speed of operation i.e., 10-20 characters per second.

Cathode Ray Tube Terminal (CRT)

Input to the computer is also possible by means of a cathode ray tube which is similar in arrangement and operation to a television picture. This process makes use of electronic pointers or light pens to quickly add to the memory unit, or erase, after any data displayed on the CRT. This is extermely useful aid.

The input unit, irrespective of the type of input medium used, convert the information into a series of signals which the computer can understand and store.

13.5 SOME KINDS OF OUTPUT MEDIA

Output receives the stored result from the memory unit, converts it into a form the user can understand and, through the aid of one or more output divices, prints or produces it in the desired form. Some of the output devices generally used are

1. Line printer
2. Graph plotter,
3. Visual display unit,
4. Card punch and paper tape punch,
5. Console type writer and teletypewriter,
6. Magnetic tape drive, magnetic disc drive and magnetic drum.

Not let us know briefly about the above mentioned Output media.

Line Printer

Line printer is a popular output device, it provides a printed copy of the result and program-listing which are easy to read and convenient for subsequent reference. It can print 300-1400 lines per minute, each line consisting of 96, 120, 132, 144 or 160 character positions. One line is called one record for output and the number of characters is known as Record Length. Some of the recently developed printers are capable of printing 6000 or more lines per miniute. Although the line printer is ahigh speed device, its speed does not match the speed at which the computer outputs the information. Therefore, output information is often recorded first on magnetic tape, magnetic disc or magnetic drum, then printed through the line printer.

Graph Plotter

A graph plotter is useful when a graphic or pictorial representation of the result is more meaningful and easier to interpert and use than an extensive alphabetic, numeric, or alphanumaric listing. A common form of this device makes use of a pen which moves on a graph paper, at a speed of 600-18,000 steps per minute, the record the result. The graph can be obtained in any desired format by (1) using a suitable combination of axes, lines, letters and symbols, and (2) selecting appropriate scale factors, letter and sybol sizes, and printing angles.

Visual Display Unit

Visual display unit is used when the output information is to be displayed on the CRT. For example, it may be used to display data, graphs, and all types of designs to a viewing manager, engineer or scientist. The device provides almost instantaneous response and can display 250-10,000 characters per second. The fact that it can be used to display the required output information even from a remote computer makes it useful especially in banks, airlines, hospitals etc.

Card Punch and Paper Tape Punch

When output information is to be recorded on cards or paper tape for use as input data, either a card punch or a paper tape punch may be used. The card punch has a set of punches which according to the instructions it receives, mechanically punch holes on a blank card. Once this operation is over, the card punch, to ensure accuracy, reads back the data recorded on the card. Because the punches operate mechanically, the card punch punches at a slow speed i.e., 100-500 cards per minute. In a paper tape punch, the punching operation is similar to that in a punch except that only one character at a time is punched. Therefore, its punching speed is only 20-500 per second.

Magnetic Tape Drive

The primary function of this device is to move the magnetic tape past its read-write head to facilitate the reading or writing of data. It provides sequential access to the data stred on magnetic tapes but has a longer access time. For example, if the required information is stored at the end of the tape which happens to be positioned at the start, the entire tape lenght has to be wound over before the tape drive can retrieve the information The salient characteristics of this device are 1) slow access 2) low cost and 3) medium storage capacity (20-200 million characters per reel of tape).

Magnetic Disc Drive

It contains a stack of magnetic discs mounted on a rotating vertical shaft. The magnetic disc is similar to a phonograph record and is coated on both its surfaces with a magnetic recording material. Enough space is provided between the discs to allow the access arms, containing the read and write heads, of the disc drive to move in and read or record the data on either surface as minute magnetized spots arranged in binary form, in concentric circular tracks rather than spiral as in phongraph records. Magenetic disc drives with removable and replaceable disc packs are available. The disc drive provides a random access storage in the sense that any specific track on a prticular disc can be selected at random but, once this tract is identified, the information can be recorded or retrieved only sequentially. A pack of six 14 inch discs having 200 - 500 tracks on each of 10 recording surfaces provides a storage capacity of over 45 million characters, and a data transfer rate of more than 1,56,000 characters per second. The chief characteristics of magnetic disc drive are 1) fast access 2) low cost and 3) high storage capacity (160 - 200 million characters per disc pack).

Magnetic Drum

It is a cylinder, ranging from a few inches to a few feet in diameter with a magnetizable outer surface. Data is stored in this surface as minute magnetized spots arranged in binary form in a series of parallel circular tracks. The drum rotates at a constant speed (800 -12,000 revolutions per minute) and data is recorded or retrieved by the read/write head, one per each track, positioned close enough to the surface and to sense the magnetisation on it. A drum having 800 tracks can store over 4 million characters, transfer more than 1.5 million characters per second and have an average access time of 8.6 milliseconds. The prominent features of this device are 1) very fast access 2) high cost and 3) medium storage capacity (4 - 200 million characters per a drum.)

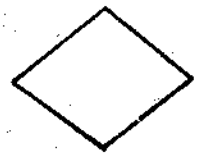
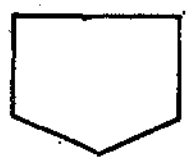
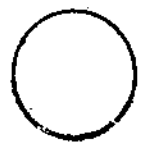
We already discussed the console typewriter and teletype writer. Irrespective of their configuration, modern computers can be broadly classified as 1) Special purpose and 2) General purpose computers. Special purpose computers are engineered for handling only a specific type of problem. For example, a computer designed only to compute pay roll information, or perform inventory control, or reserve passenger seats in international flights, falls within this category. On the other hand, general purpose computers can solve a variety of problems.

13.6 FLOW CHARTS

Now we study the construction and use of flow charts for different types of problems. A flow chart is a pictorial or graphical representation of a specific sequence of steps to be performed by the computer to produce the solution of a given probelm, i.e., the flow chart is a pictorial description of algorithm, represnting the points of decision, computations and the sequence in which the problem takes place to solve. Therefore flow chart is the most vital part of the programming of any problem.

It makes use of flow chart symbols to represent many of the basic operations in programming. These symbols are connected by directing line segments to indicate the flow of information and processing.

The following are some of the standard basic symbols commonly used in flow charts and their representations.

S.No.	Flow chart Symbol	Representation
1.		The entry or exit point of the flow chart i.e., start or stop
2.		An input/output operation
3.		A processing operation such as addition and movement of data within the memory unit
4.		A decision point at which a branch to one of two or more alternate paths are possible.
5.		Entry from or exit to another page of the flow chart, the symbol is so placed that it indicates the direction of flow.
6.		Entry from or exist to another part of the flow chart on the same page.

In the flow charts, the relation of the form P=Q or P←Q means P becomes Q i.e., assign the value Q to P. The "equal to" sign should always be interpreted as an assignemnt symbol. Thus R←R + 3 implies R becomes its present value plus 3. Now we shall learn the process of flow charting by considering some examples.

Example 1. Draw a flow chart to pick the largest of th given three numbers.

Solution

The procedure to be followed is, first read the three numbers into three locations (in memory) labelled as A,B,C. We then compare A with B. If A is larger, then it is compared with C. If A is again larger, then it is the largest number, otherwise C is the largest lumber. If in the first step B is found greater than A, then B is compared with C. The larger of these is the largest number. Now we express the above procedure in a concise easily understood flow chart.

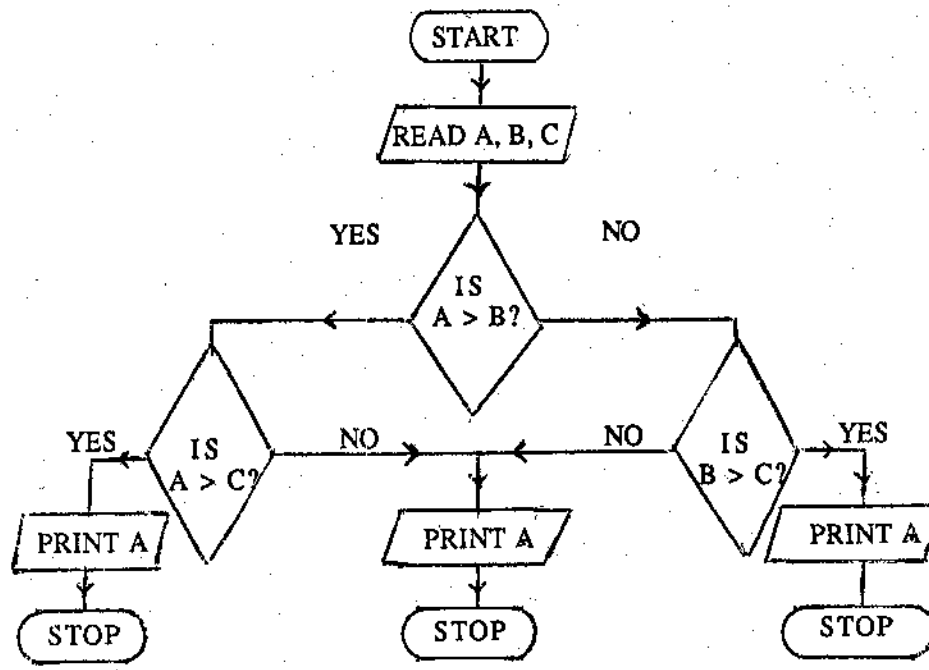


Fig.2 : *Flow Chart to pick largest of given three numbers.*

The flow chart rapidly becomes very long as the number of numbers to be compared is increased. We follow another method, if we want to find the largest of a large set of numbers.

Example 2. Develop a flow chart to find the largest of the given N numbers.

Solution

Here we will follow the following method. We call the first in the list as the largest number and store it in a memory location called BIG. We compare it with the next number and store and larger of the two in BIG. Thus BIG will contain the largest number in the list encountered so far. We continue this procedure till we exhaust all the numbers in the list.

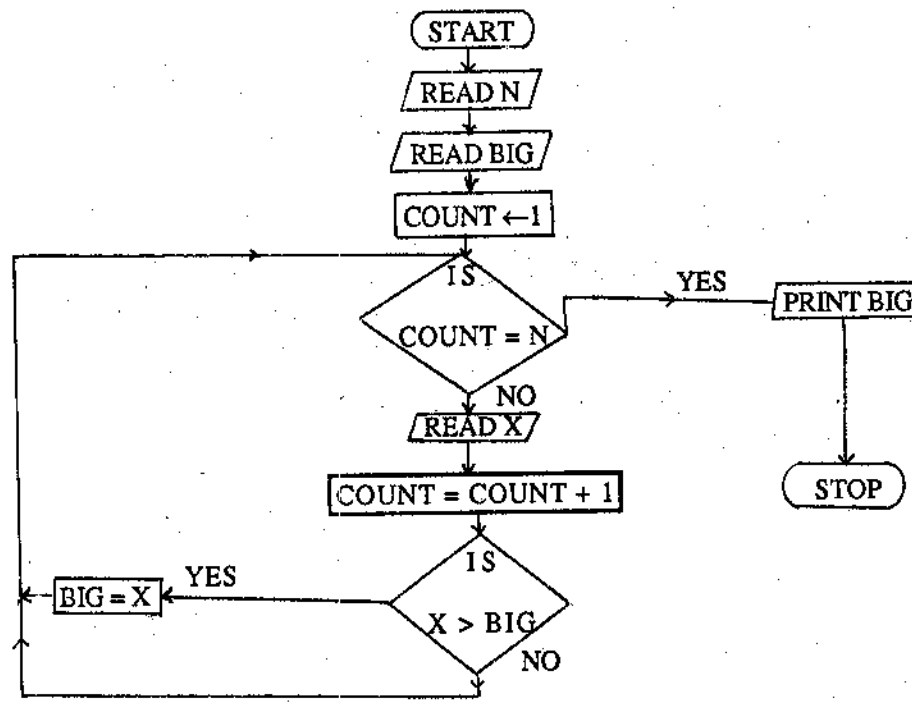


Fig. 3 : *Flow chart to pick the largest of N numbers*

Observe the way in which the numbers read in are counted and compared with the total number of items in the set of numbers. COUNT←COUNT + 1 is to be intepreted as "COUNT becomes the previous number in COUNT + 1". Such a technique of repeating a certain set of operations for a fixed number of times is very important in formulating flow charts.

Example 3. Draw a flow chart to compute the scalar product of two vectors A and B given by sum = $\sum_{i=1}^{10} a_i b_i$ where a_i and b_i are the components of vectors A and B for i = 1,2 .....10 are known

Solution

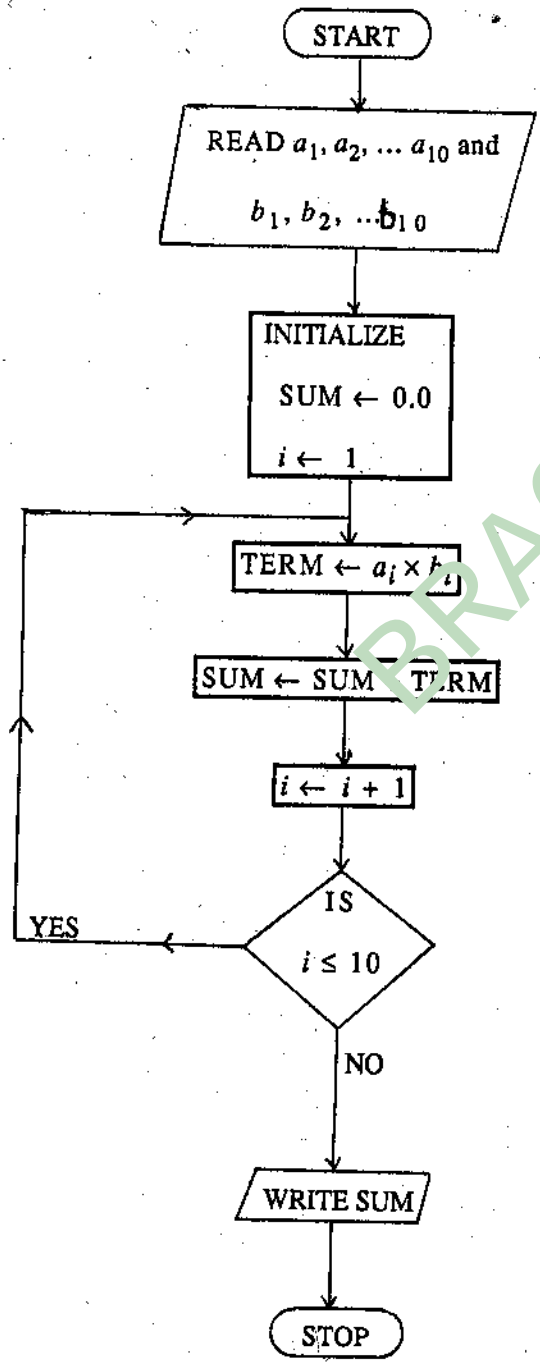


Fig. 4 : *Flow chart to compute scalar product.*

Example 4. Given N draw the flow chart to compute N!

Solution

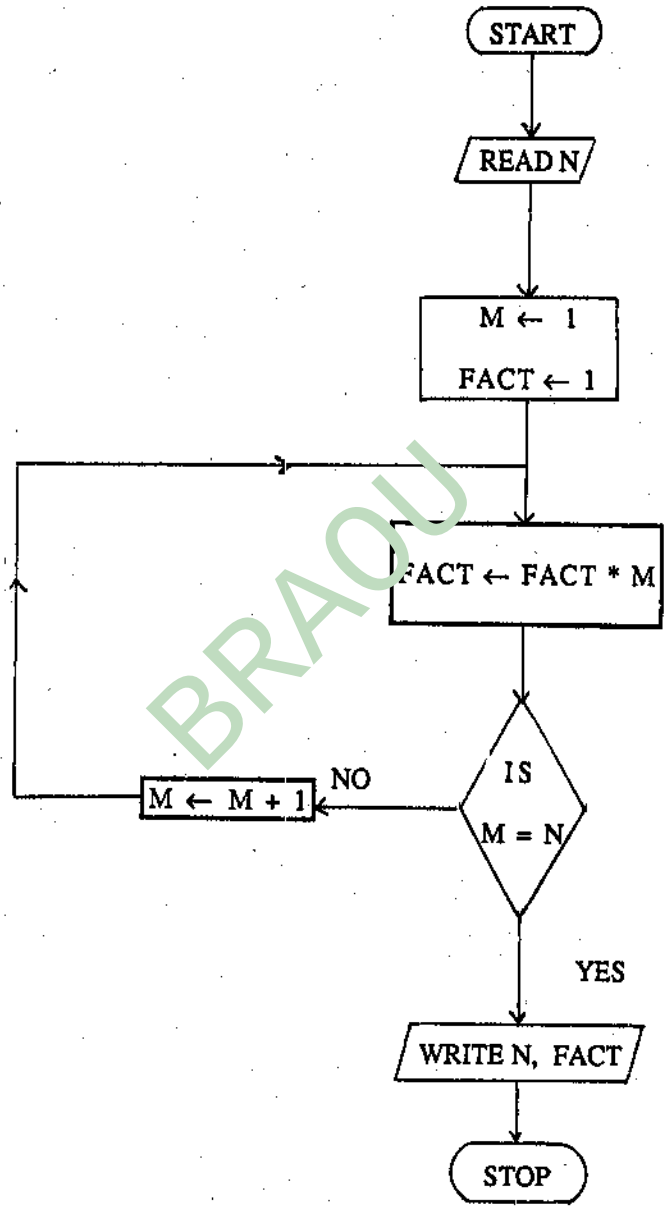


Fig. 5 : *Flow chart to compute N !*

Example 5. Given the coordinates of two points P (a,b,c) and Q (x,y,z). Draw a flow chart to calculate the distance PQ and direction cosines of the line PQ.

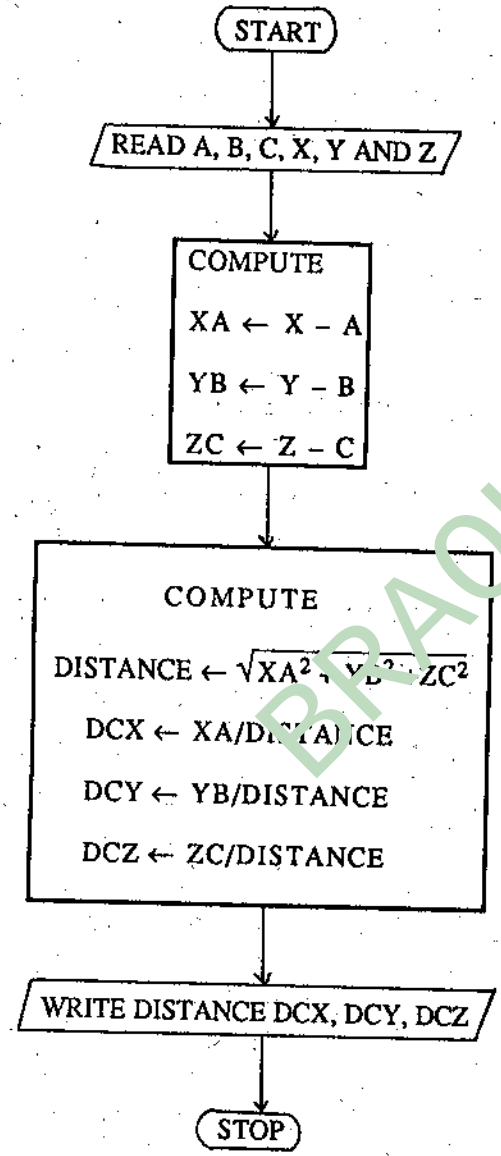


Fig. 6 : *Flow chart to compute distance and direction cosines*

Example 6. Draw a flow chart for solving a quadratic equation $ax^2 + bx + c = 0$ by considering all the possible cases.

Solution

Here, if $a = 0$ we get $bx + c = 0$, a linear equation, will have only one root. If $a \neq 0$ then depending upon the value of the discriminent $D = b^2 - 4ac$, the equation will have different roots. If $D \geq 0$, the equation will have two real roots x_1 and x_2, and if $D < 0$, then it will have complex conjugate roots whose real part is x_1 and x_2 imaginary part. All these cases are depicted in flow chart.

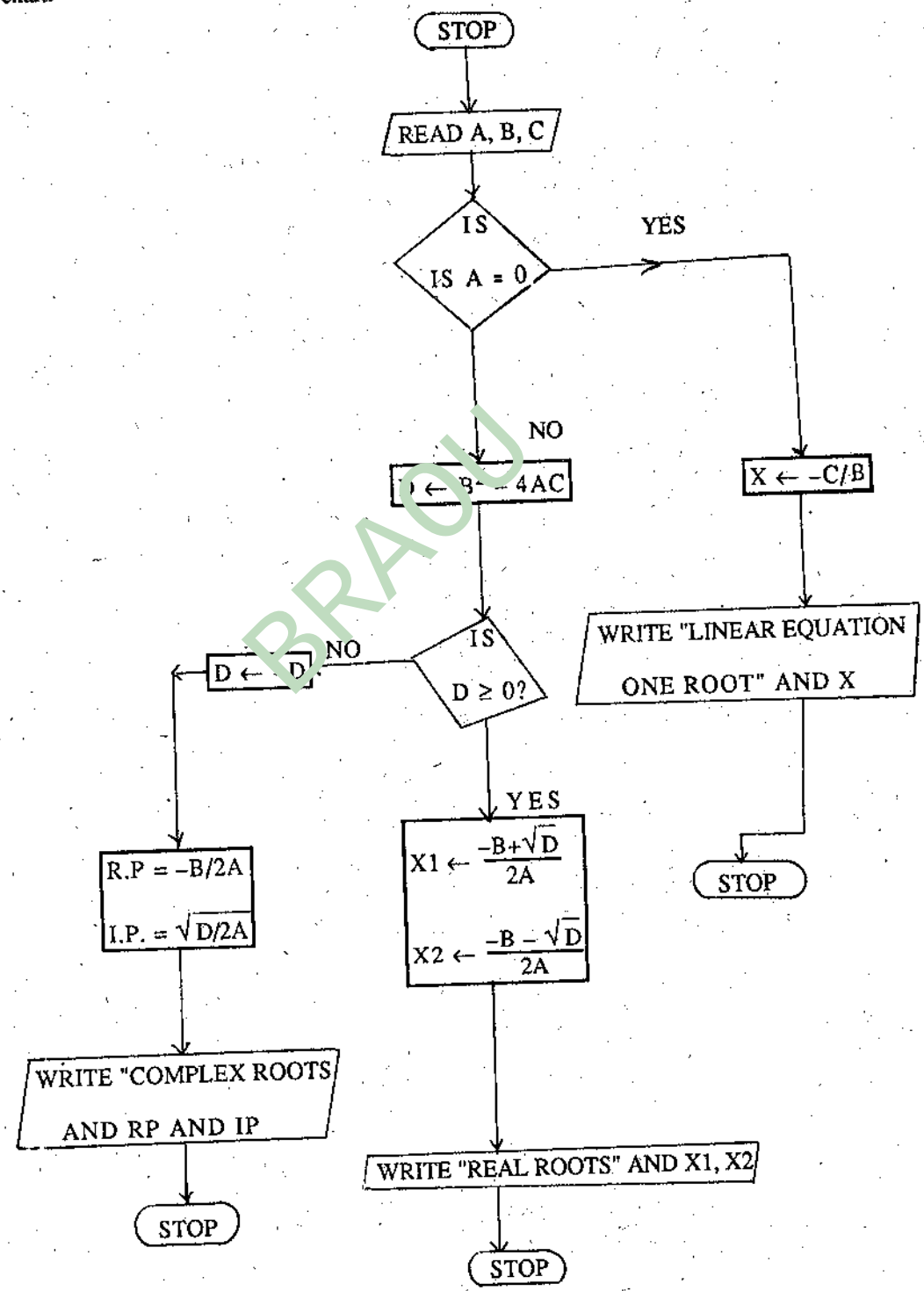


Fig. 7: *Flow chart to find the roots of a quadratic equation*

Example 7. Draw a flow chart to compute the sum of the first 100 natural numbers.

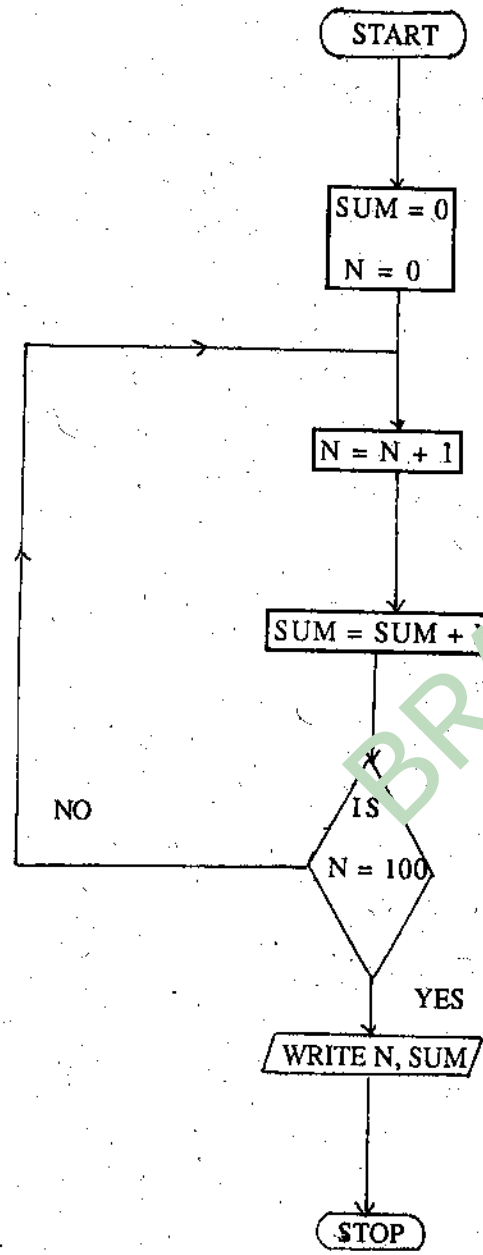


Fig. 8 : *Flow chart to calculate sum of first 100 natural numbers*

Example 8. Draw a flow chart to find the sum upto N terms of the following series

$S = 3x + 5x^2 + 7x^3 + ...$

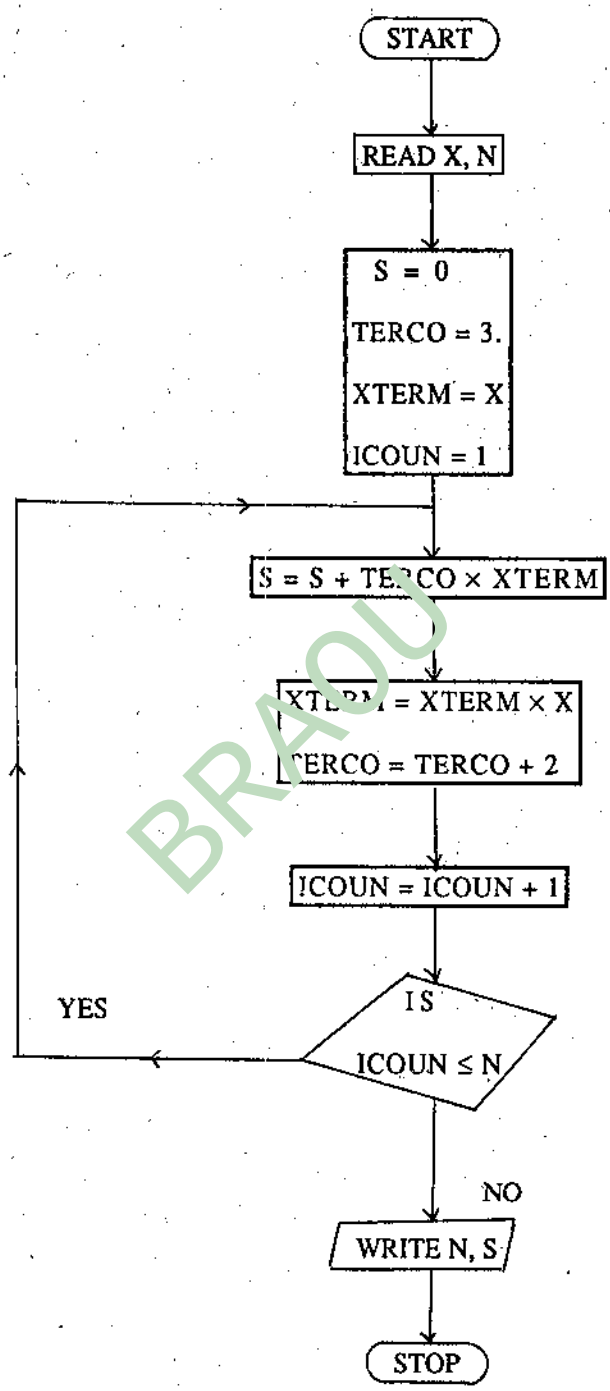


Fig.9 : *Flow chart to find sum of the series*

Therefore a flow chart offers an easy visual understanding of the logic applied to obtain the solution of a problem. After preparing flow chart, the accuracy of the procedure it represents can be tested by applying it to a probelm whose solution is already known. The ability to produce correct flow charts is of crucial improtance in computer programming.

13.7 DECISION TABLES

If a problem involves complex logic, we can analyse it by using a table called Decision Table rather than flow chart. A decision talbe is a tool for the data processor. A programmer can prepare a program without the necessity of preparing a flow chart.

A decision table is divided into four parts. i) condition stub ii) condition entry iii) action stub iv) action entry as shown in Fig. 10.

CONDITION STUB	CONDITION ENTRY
ACTION STUB	ACTION ENTY

Fig. 10 : *Elements of decision table*

In condition stub we enter all conditions that exist and all actions that exist in action stub. To see the concept of dicision table, consider the following simple inventory control applicaton in which it is necessary to maintain stock balances. We must consider appropriate action for Issue and Receipt transactions. The only conditions to be considered are whether or not the transaction is an issue or receipt. If the transaction is an issue, the necessary action is a reduction of the stock balances. If the transaction is a receipt, the action to be taken is an increasing of the stock balance

Transaction is ISSUE	X	N
Transaction is RECEIPT	N	Y
Add quantity to balance	—	X
Sub. quantity from balance	X	—

and 'X' in a row opposite an action means "do the action"

Y — Yes, N — No.

Fig. 11. *Decision Table*

Now let us consider some examples.

Example 1 : If the credit limit is satisfactory, the order is to be approved. If the credit limit is unsatisfactory, but the pay experience is good, the order is to be approved. If the credit limit is not O.K, the pay experience bad, but special clearance is obtained, then the order is to be approved. Finally, if the credit limit is not O.K, the pay experience is not good, and no special clearance is obtained, the order to be disapproved. Construct a decision table which depicts the steps required to determine whether or not a customer is to be permitted credit.

Solution

Here the condition stub includes the following : i) Credit limit O.K. ii) Pay experience good iii) Special clearance obtained. The action stub include either an approve or disapprove.

Rule No.	1	2	3	4
Credit limit is O.K.	Y	N	N	N
Pay experience is good	—	Y	N	N
Special clearence obtained	—	—	Y	N
Approve Order	X	X	X	—
Return to Sales	—	—	—	X

Fig. 12 : *Decision Table*

Example 2 : The rules for declaring the results in an examination are as follows. If a student gets 50% or more in the main subject and 40% or more in the ancillary subject, he passes. If he get less than 50% in the main subject he must get 50% or more in the ancillary subject to pass. However the minimum passing marks is 40% in the main subject. If a student gets 60% or more in the main subject, he is allowed to repeat the only ancillary subject if the ancillary marks fall below 40%. There are, however, a group of students in the class who are granted special consideration. Their pass percentage is 40% in the main and 40% in the ancillary. If they get less than 40% in the ancillary they are allowed to repeat that subject only. Obtain the decision table.

Solution

The conditions which determine the results are : 1) Marks in the main subject, 2) Marks in the ancillary subject, 3) Whether the student is granted a special consideration. The actions to be taken based on the conditions are 1) Pass a student 2) Fail a student 3) Ask a student to repeat the ancillary subject.

Main Marks %	≥50	≥40	≥60	≥40	≥40	ELSE
Ancillary Marks %	≥40	≥50	<40	≥40	<40	
Special consideration	No	No	No	Y	Y	—
Pass	X	X	—	X	—	—
Repeat Ancillary	—	—	X	—	X	—
Fail	—	—	—	—	—	X

Fig. 13 : *Decision Table*

13.8 SUMMARY

A computer is an electronic computational device which can perform millions of calculations in a few minutes and can provide extremely accurate results. Any computer will have input unit,

memory unit, control unit, arithmetic–logical unit and output unit. The information is fed to the computer through input device, then it is stored in the memory and the execution will be started. Then, the instructions are decoded by the control unit. All the arithmetic operations are performed in the arithmetic–logical unit. The instructions are executed sequentially in the order in which they are written. The normal sequential order may be altered by a conditional branch instruction.

There are two types of Computers based on the way they represent information. They are digital and analog computers. A set of detailed instructions which leads to a step by step procedure for solving a problem is called an algorithm. An algorithm coded in a computer language called a program and the language used is called programming language. Flow charts will guide us in writing a program. A flow chart is a graphical represntation of a specific sequence of steps to be performed by the computer to solve a give problem. Another way of analysing a given problem is by using a decision table.

13.9 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail.*

1. Expalin the importance of computers in problem solving by citing some examples.
2. What are the five essential units of a computer? Explain the structure of a computer by means of a block diagram.
3. Mention any three kinds of input media and explain briefly about them.
4. Explain the importance of flow charts and decision tables.
5. Draw a flow chart to calculate S for R = 10, 20, ..., 200 using the formula.

$$S = \begin{cases} 17000 - 0.48\ R^2 & \text{for } R < 100 \\ \dfrac{18000}{\left(1 + R^2/18000\right)} & \text{for } R > 100 \end{cases}$$

6. An insurance company follows the following rules to calculate the premium.
   1. If a person's health is excellent and the person is below 35 and live in a city and is a male then the premius is Rs. 2 per thousand the maximum amount of policy is Rs. 22 lakhs.
   2. If a person is a female and satisfies all the other conditions above then the premium is Rs. 1.50 per thousand and the policy is Rs. 1.5 lakhs.
   3. If a person's health is poor, and age is below 35 and the person live in a village, then the premium is Rs. 3 per thousand and the policy may not be written for more than Rs. 10,000.
   4. In all other cases the person is not insured. Obtain a decision table summerising the above results.

II. *Briefly answer the following*

1. What is a computer? What is the difference between Digital and Analog computer?
2. Explain briefly about punched card.
3. Expalin the difference between special purpose computers and general purpose computers.
4. Mention some of standard basic symbols commonly used in flow charts and their representations.
5. Given a set of n numbers, $A_1, A_2, A_3, \ldots A_n$. Draw a flow chart to arrange them in ascending sequence.
6. Draw a flow chart to count negative numbers from a set of given numbers.

UNIT–14 : FORTRAN PROGRAMMING PRELIMINARIES

Contents

14.1 AIMS AND OBJECTIVES

After going through this unit, you will be able to, (i) Identify various symbols used in Fortran and Write the expressions and Statements in Fortran language. (ii) Identify the real variable names and integer variable names, real constants and integer constants and use them appropriately in program writing.

14.2 INTRODUCTION

14.2.1 Machine Language Instructions

A computer cannot directly interpret a flow chart. It can interpret and execute only a set of coded instructions called Machine language instructions. The machine language is the computer's native language which forms the sequence of instructions. During the early days of the computers, all programs had to be written in machine language, the only language that any computer can understand directly. This consists of combination of zeros (0) and ones (1). The machine language also differs from computer to computer. A machine language instruction for a well known computer is, for example 21 00500 09400. In this instruction 21 is the code for add, 00500 and 09400 are the 'labels' of two memeory boxes where the two operands are stored. This instruction commands that the numbers in locations 500 and 9400 be added and the answer put back in 9400. If we remember the codes and know all the available memory boxes and their addresses we can write a series of instructions corresponding to a flow chart to solve a problem.

There are two difficulties in writing machine language instructions. (1) We have to memorise all the codes and keep detailed count of memory locations used. This is difficult since in any computer, there would be many different instructions and so many locations in memory. (2) The

codes will differ from one computer to another. Thus one has to rewrite instructions again and again as new computers are introduced. A series of machine language instructions for solving of a problem is known as a machine language program.

14.2.2 Higher Level Languages

Now-a-days it is not necessary to write programs using machine language instructions. Computer programs may be written in one of the Assembly language or higher level langugaes. They are also called programming languages. The instructions in a higher level language are not coded numbers, they are similar to ordinary formulae of algebra. We need not refer to the numerical lables of memory locations, instead we can use symbolic names to label memory locations. Therefore programming languages are easy to learn and use. FORTRAN, COBOL, PASCAL, BASIC, ECOBOL are some familiar programming languages.

Associated with each higher level language is an elaborate computer program which translates it into a machine language. This translation program is called a compiler. The original program written in the higher level language is called source program and the resulting machine language program is called the object program. Compilers are written by professional programmers. Since the same higher level language may be used with different computers, these languages are also called as machine independent languages. The compilers are different leading to different machine language equivalents of the source program. This is illustrated below.

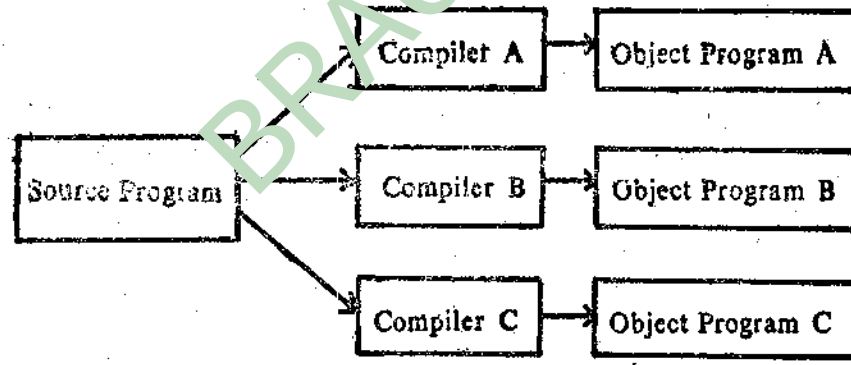


Fig. 1

Since the compiler has to translate a higher level language,it is essential to define the rules for constructing instructions in that language. These are called the Syntax rules of the language. When a program written in any higher level language is fed to a computer the compiler analyze each instruction and determines if any syntax rules are violated. If no syntax rules are violated the program is successfully compiled and stored in the memory of the computer for execution. If there are any syntax errors then the compiler prints out the list of statements which are wrong and indicate the mistakes made by the programmer. It does not execute the program.

One important point we have to remember is that the compilers cannot point out errors in logic. They don't know one's intension. For example if you read in - 15 as the age of Mr. X and used it in computations, the compiler cannot diagonise this. The computer will execute such programs but we get wrong results. So one should be careful while preparing programs.

14.2.3 Fortran Language

We study now the essential features of one of the most popular higher level language FORTRAN. This language was developed in 1957 by IBM (International Business Machines). This language is a hybrid of English and Mathematics which is suitable for scientific and engineering computations. Our aim is to know how to solve probelms in a computer using FORTRAN as a vehicle of communication. FORTRAN stand for FORmula TRANslation. Due to the practical difficulty of writing compilers, there are slight variations in implementation of FORTRAN languages for different computers.

If we want to learn a spoken language, first we have to learn the various characters used in the language. Then we learn how to combine the characters to form words, words to form sentences and sentences to express a thought. Similarly we can learn any higher level language. First a set of legal characters in the language are defined. Then a precise syntactic definition on how to combine the characters to form syntactic unit is given and then about grammatical forms, namely, the correct relationship between various syntactic structures. Now we describe the rules of FORTRAN-IV language. All subsequent references to FORTRAN are to FORTRAN-IV.

14.3 CHARACTERS AND OPERATIONS IN FORTRAN

The following are the set of allowed characters in FORTRAN and is known as FORTRAN character set.

English capital letters : A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y and Z.

Decimal digits : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Special characters

Equal to	=
Plus	+
Minus	–
Asterisk	*
Slash	/
Left Parenthesis	(
Right Parenthesis	)
Comma	,
Decimal point	.
Currency symbol	$
Blank	b

This set of characters together with a set of syntax and semantic rules (which we know in the next articles) constitutes FORTRAN IV language. These syntactic rules describe the methods for constructing the various syntactic units (i.e, constants, variables, expressions and statements) and the semantic rules specify the operational meanings associated with them.

14.3.1 Arithmetic Operations

FORTRAN Permits five arithmetic operations, i.e., those of addition, subtraction, multiplication, division and exponentiation, all others must be built up from these five operations. The following symbols are used to represent the above arithmetic operations.

Operation	Symbol
Addition	+
Subtraction	–
Multiplication	*
Division	/
Exponentiation	**

Now let us consider the following table to study the meaning of arithmetic operations in FORTRAN.

Sl. No	Example	Meaning
1.	A = B	The value of B replaces the existing value of A.
2.	A = B + C	Add the value of B and C, and that value replaces the existing value of A.
3.	L = M * N	Compute the product of M and N and put that value in place called L.
4.	R = P ** Q	Compute PQ and call the result R.
5.	Z = (W)	Compute the value of W in parentheses and call the result Z.

14.4 FORTRAN CONSTANTS

In ordinary algebra we deal with two types of quantities known as (i) constants and (ii) variables. If a quantity retains a fixed specified numerical value, then it is called a constant and if it is unknown or takes on different numerical values, then the quantitiy is called a variable. For example in $a = 5 + 2b$, a & b are variables and 5 and 2 are constants.

FORTRAN, a scientific oriented language, permits us to denote and use both constants and variables. A constant in FORTRAN is a quantity that does not change its numerical value during the execution of the program and a variable in FORTRAN is a quantity that changes its value several

times during the execution of a program i.e., any quantity referred by name rather than by its numerical value is a variable. Here we introduce the basic elements of the programming language FORTRAN such as constants, variables functions, expressions and arithmetic statements.

Two types of constants are used in FORTRAN, i) Integer constants ii) Real constants. Integer constants are also known as Fixed point constants and Real constants are also known as Floating point constants.

14.4.1 Integer Constants

Integer constants are whole numbers without any fractional part. The following rules indicate the method of writing Integer constants in the FORTRAN language. When a rule is given while defining the language it must be exactly follwed. If we violate the rules, the computer will reject the computer program.

Rule : An Integer FORTRAN constant is a string of decimal digits such that

i) It must be written without a decimal point, comma or an exponent symbol.

ii) It may have either of the signs + or –

iii) If neither sign precedes the constant, then the constant is assumed to be positive.

The maximum and minimum size of an integer constant depends upon the type of the computer in use, i.e., the largest magnitude of an integer depends upon the particular computer used. For TDC-312, the range of integer constants is ±2407, for TDC-516 it is ±32768 and for IBM 360 FORTRAN, an integer constant is any whole number in the range $-\left(2^{31}-1\right)$ to $\left(21^{31}-1\right)$.

Examples

1. 560 — valid integer constant.
2. –10 — valid integer constant.
3. 2,352 — invalid integer constant since comma is there in between.
4. 11. — invalid integer constant due to the presence of decimal point.
5. 100 — valid integer constant.

14.4.2 Real Constant (Floating Point Constant)

A FORTRAN real constant is a string of decimal digits with a decimal point. The decimal point may be placed at the beginning, at the end or between any two digits. A minus sign must precede if the number is negative, if it is positive, a preceding plus sign is optional. The maximum number of allowed digits varies from one computer to other.

Examples

1. 123.456 — valid real constant.
2. 0. — valid real constant.
3. 1540 — Invalid since decimal point is missing.

4. 4,522.7 — Invalid since comma is there in between.

5. –17.56 — valid real constant.

There is a second way in which real constants may be represented. It is called the exponential form. In this form a real number be written as a mantissa consisting of a string of decimal digits including a decimal point followed by the letter E and a string of digits with no decimal point which is known as exponent. Both the mantissa and exponent may have their independent sign. For a negative exponent the minus sign must be used, for positive exponent '+' is optional. For exmaple, in the number 10.321 E – 19, the number 10.321 is mantissa and –19 is exponent and the value of the number is 10.321×10^{-19}

The ranges of mantissa and exponent varies from computer to computer. For example in TDC-312 the exponent range is ± 610 and it allows 7 digits for mantissa. TDC-315 allows an exponent range of ± 38 and for mantissa 7 digits.

Examples

1. .36 E – 17 — valid real constant.

2. 34.786 E 25 — valid real constant

3. 16.0 E 5 — invalid due to the presence of decimal point in exponent.

4. –1.492 E + 03 — valid real constant.

5. 175.67 E 00 — valid real constant.

14.5 FORTRAN VARIABLES

14.5.1 Integer Variables

A quantity which may vary during program execution is called a FORTRAN variable. Each variable is given a name and it has a specific storage location in memory where its numerical value is stored. A variable is said to be defined if a number is stored in the storage location identified by the name of the variable. A variable must be defined before it is used in computation.

When the value of a variable is read from memory, the original value still remain in memory. When a new number is stored in a memory location, the number which was here previously is erased. We know that numbers are two types. Integer constants and real constants. Similarly there are two types of variables i) Integer variables ii) Real variables. A number stored in a memory location with an integer variable name is an integer and one stored in a location with a real variable name is a real constant.

Rules for forming Integer variable

1. Integer variable is a combination of one to five letters or digits.
2. The first character in the name must be one of the following letters I, J, K, L, M or N.
3. The name must not contain any special character.

The first character distinguishes an integer variable from real. Some computers accept the combinaltion of one to six letters or digits.

Examples

1.	JOHN	—	valid integer variable.
2.	K12	—	valid integer variable.
3.	COUNT	—	invalid integer variable since it is not starting with I, J, K, L, M or N.
4.	KOUNT	—	valid integer variable.
5.	IA * B	—	Invalid integer variables since special characters are not allowed.

14.5.2 Real Variables

Real variables are also called Floating point variables. A number stored in a memory location with a real variable name is a real constant.

Rules for forming Real Variables

1. Real variable name is a combination of one to five letters or digits (some computers accept one to six letters)
2. The first character must be a letter other than I, J, K, L, M or N.
3. The name must not contain any special character.

Examples

1.	BETA	—	valid real variable
2.	TAX	—	valid real variable
3.	AB.DE	—	invalid, due to the presence of decimal
4.	D 156	—	valid real constant
5.	IMAT	—	invalid, since it is starting with I.

It is convenient, to choose variable names which suggest the quantities involved such as TEMP, KOUNT, ALIFE for temparature, count and life respectively. Note that 'A' in front of LIFE, which serves to make the variable name real. Similarly in an integer calculations if the technical term for the physical variable starts with a letter other than I, J, K, L, M or N, we can add one of these at the beginning of the name to make it integer variable. Using such names simplifies programming and also simplifies the search for errors.

14.5.3 Type Declaration for Integer and Real Variables

The above restriction to use a special letter as the first letter of a variable name to determine whether it is integer or real is a restriction in FORTRAN II. In FORTRAN IV the user is allowed to

declare the nature of the variable before it is used in a program by means of statements which are called Type delcaration statements.

A variable name may be specified as an Integer variable name or a Real variable name through the use of type declaration statements as follows :

INTEGER list of variable names seperated by commas and full stop is not allowed at the end.

REAL list of variable names seperated by commas and full stop is not allowed at the end.

For example, the type statements

INTEGER LIFE, X, GAMA, ALPHA

REAL I, SUM, LENG, KOUNT

inform the complier about the type of values that are to be stored with the variable names.

In a program one may have more than one declaration statements. The declarations must appear before the first use of the variable name. It is advisable to have these declaration statements at the beginning of a program.

14.6 FORTRAN EXPRESSIONS

We know that FORTRAN permits five arithmetic operations i.e., addition, subtraction, multiplication, division and exponentiation. There are two modes of arithmatic operations i) Integer mode and ii) Real mode. Integer mode arithmetic is performed on only integer constants and/or integer variables and the results are always restricted to the type integer, i.e.., whole number. i.e., 7/4 would give the result 1 and not 1.75, 1/2 would give the result 0 and not 0.5. Real mode arithmetic is performed on only real constants and/or real variables. Therefore the results include fractions also i.e., 1./2. would give .5 not 0.

A FORTRAN arithmetic expression a series of variables and constants connected by arithmetic operation symbols. It represents a quantity. Thus a signed or an unsigned integer or real variable name or integer or real constant is a valid arithmetic expression. For example —I, –15, 17.4, A + 1.5, I * J – 5, KON * 5—I * K are acceptable arithmetic expressions. While evaluating an expression all quantities appearing in it must be in the same mode. Thus expressions are of two types.

i) Integer expressions ii) Real expressions.

For forming and evaluating an expression we have to note the following points.

1) Two operation symbols must never appear consecutively.

Ex. : i) A * – B is invalid since * and – are appearing consecutively, but A * (–B) is valid.

ii) I * * 6 is valid, since '* *' is one arithmetic operation symbol for exponentiation.

2) Only the operators + and – must be followed by an operand. All others should be preceded and followed by an operand.

Ex. : i) B + 7.4 – invalid

ii) –I * J + K/L valid

iii) –C + 17.7 + invalid

3) Parenthesis may be used to indicate the required groupings. For every opening parenthesis, there must be a corresponding closing paranthesis.

Example

i) $\frac{A}{5C}$ should be written as A/(5. * C)

ii) IA/–5 is wrong, where as IA/(–5) is valid.

iii) $\left(\frac{C}{D}\right)^{(A-B)}$ may be written as (C/D) * * (A – B)

4) In a program the value of any expression is calculated by executing one arithmetic operation at a time. The order in which the arithmetic operations are executed in an expression is called the hierarchy of operations.

When no parenthesis occur in an expression, the operators have an implict order of priority.

i) All exponentiations first.

ii) All multiplications and divisions are carried out second.

iii) All additions and substractions are done last.

Ex. 1 : A * * B/C+D

Here the exponentiation is done first the result is divided by C and then D is added which is equivalent to

$$\frac{a^b}{c} + d$$

Ex.2 : Write mathematical expression equivalent to the following FORTRAN expression.

A * * B/C+D * * E * F – H/P*R+Q.

Sol : Following hierarchy, exponentiations first,

$$\therefore A^B, D^E$$

next multiplications and divisions from left to right.

$$\frac{A^B}{C}, D^E F, \frac{H}{P} R$$

In the third pass,

$$\frac{A^B}{C} + D^E F - \frac{H}{P} R + Q$$

This is the required equivalent mathematical expression.

Ex.3 : A/B/C means *a/bc* and P/Q*R means $\frac{p}{q}r$ but not $\frac{p}{qr}$

5) Parentheses are used if the order of operations governed by the hierarchial rules are to be altered. If there are parentheses in the expression, then these must be cleared first. The expression inside the parentheses is evaluated using the same hierarchial rules. When one or more pairs of parentheses are used, the parentheses are cleared from the innermost pair to outermost pair. Now let us consider some examples.

Ex.1 : A * B – C/(5. * D – 6.2)

Sol. : (5. * D – 6.2) will be evaluated first. Let it be valued as e_1

(A * B) and C/e_1 will be evaluated in the second pause and let them be denoted by e_2 and e_3 respectively.

$e_2 - e_3$ will be evaluated last from left to right.

Ex.2 : Evaluate (P * Q/R) + (X * (X * (A * X + B) + C) + D) + W

Sol. : The subexpression P*Q/R will be evaluated first. Let its value be e_1.

The nested subexpression A*X+B will be evaluated second. Let it be e_2.

X * e_2 + C will be evaluated next and let its value be e_3.

X * e_3 + D will be evaluated next and let its value be e_4.

The expression $e_1 + e_4$ + W will be evaluated finally.

6) If in an expression all the quantities are integer quantities (i.e., integer variables and integer constants) then that expression is integer Arithmetic expression. On the other hand if all the quantities are real, then that expression is called Real Arithmetic expression. If it is mixture of both (i.e., integer and real), then it is an arithmatic expression in mixed mode. The mode of each term in any such expression will depend on the mode of the terms of constituent elements. If all the elements in a term are of the integer type, the term is treated as an integer sub-expression and is evaluated in integer mode. If the elements are of both the types, the elements of the type integer are converted into real mode and then the term evaluated in real mode. FORTRAN-IV compliers accept arithmetic expression in mixed mode.

Ex.1 : Consider the expression I/J+A/ (I + J) – K * * J

Solution

Here i) J divide I in integer mode

ii) (I + J) evaluated in integer mode

iii) The result of (I + J) is converted into real mode and A/(I + J) is evaluated in real mode.

iv) (K * * J) evaluated in integer mode

v) Finally the result evaluated in real mode.

7) A real or an integer expression may be raised to a power that is a real or an integer expression but a negative value should not be reised to a real power, not a zero quantity to a zero power. If R, S are real expressions, and I and J are integer expression, then

i) I * * J will be evaluated in integer mode by multiplying I by itself J times.

ii) R * * S will be evaluated in real mode as $10^{(S \log_{10} R)}$

iii) I * * S will be evaluated in real mode by converting I into real mode

iv) R * * I will be evaluated in real mode by multiplying R by itself I times.

Now let us consider few examples.

Ex. i : Find the equivalent FORTRAN expression for $\frac{a-b}{a+b}$

Solution

Normally, one may write A – B/A + B. This is wrong since this will be evaluated as A – (B/A) + B, which is not what the programmer intended. A – B/(A + B) is also wrong translation as the expression evaluated would be $A - \frac{B}{(A+B)}$. The correct translation is (A – B)/(A + B).

Therefore proper care should be taken while translating.

Ex. ii : Write the equivalent FORTRAN expression for the polynomial

$$6x^4 + 5.2x^3 + 17.5x^2 + 12x + 5$$

Solution

Equivalent FORTRAN Expression is

6. * X * * 4 + 5.2 * X * * 3 + 17.5 * X * * 2 + 12. * X + 5.

Each operation takes a finite time for execution on the computer and reducing the number of operations reduces the time needed for the calculations. The above FORTRAN expression involves a total of 4 additions and 10 multiplication operations. By using parenthesis we can reduce the arithmetic operations as follows :

$$\begin{aligned} 6x^4 + 5.2x^3 + 17.5x^2 + 12x + 5 &= 5 + 12x + 17.5x^2 + 5.2x^3 + 6x^4 \\ &= 5 + x\left(12 + 17.5x + 5.2x^2 + 6x^3\right) \\ &= 5 + x\left(12 + x\left(17.5 + 5.2x + 6x^2\right)\right) \\ &= 5 + x\left(12 + x\left(17.5 + x\left(5.2 + 6x\right)\right)\right) \end{aligned}$$

∴ The correct translation with minimum number of operations (i.e., 4 multiplications and 4 additions) is

$$5. + X * \left(12. + X * \left(17.5 + X * (5.2 + 6. * X)\right)\right)$$

14.7 FORTRAN STATEMENTS

The equality sign (=) in FORTRAN means "is to be replaced" rather than as "is equal to". For example K = K + 5 means "replace K by its previous value plus 5".

The general form of a FORTRAN arithmetic statement is

Variable name = Expression i.e., V = *e*

Examples

i)	I = 10 – (J/10) * 100 + K * * 2	valid arithmetic statement
ii)	S = (A + B + C) * 0.5	valid arithmetic statement
iii)	N = N + 5	valid arithmetic statement
iv)	W = X/0.0	invalid arithmetic statement, since division by zero is not permitted.
v)	S = (–15) * * 6.5	invalid, since negative quantity cannot be raised to a real power.

Real and integer quantities may be mixed in an arithmatic statement. The following ponts should be remembered for evaluation of statements.

1. If V and *e* are of the same mode, the numerical value of *e*, without change, is transmitted to V.

2. If V is real and *e* is in integer mode, then the numerical value of *e* is converted into real mode and the value is assigned to V. For example in Z = 17/7, 17/7 will be evaluated in integer mode, which gives 2, and which will be converted into real value 2.0 before it is assigned to Z.

3. If V is integer and *e* is in real mode, then the fractional part of the numerical value of *e* is ingored and the resulting integer value is assigned to V. For example in J = 18.6/3., 18.6/3 will be evaluated in real mode, giving the numerical value 6.2 which will after truncation, be converted into the integer value 6 before it is assigned to J.

Example

Find the value of I calculated in the following arithmetic statement for J = 2 and K = 5.

$$I = J * 2/3 + K/4 + 6 - J * * 3/8$$

Solution

$$I = 2 * 2/3 + 5/4 + 6 - 2 * * 3/8$$

$$= 4/3 + 5/4 + 6 - 8/8$$

$= 1 + 1 + 6 - 1$

$= 7$

(using Hierarchy of arithmatic operations)

14.8 SPECIAL FUNCTIONS

In addition to the simple arithmetic operations, we use several common functions, such as square root, sine, cosine, exponential and absolute value in mathematical expressions. FORTRAN allows the use of may such functions. The exact list of functions available varies from one computer to other. The method of writing such function is Function name (variable name of expression). For example, if square root of a real Z is to be evaluated, we write it is SQRT (Z)

Some important common functions and their spellings in FORTRAN are given below.

Function	Type of Argument	Symbolic form
$\sin x$	Real	SIN (X)
$\cos x$ (x must be in radians)	Real	COS (X)
e^x	Real	EXP (X)
$\|x\|$	Real	ABS (X)
$\|k\|$	Integer	IABS (K)
$\log_e x$	Real	ALOG (X)
$\log_{10} x$	Real	ALOG10 (X)
Arc tan x	Real	ATAN (X)
$\sqrt{x}$	Real	SQRT (X)

14.9 SUMMARY

A series of machine language instructions for solving a problem is known as machine language program or object program. It is very difficult to write a program in machine language. Hence languages, in which one can easily write a program were developed. These languages are called higher level languages or machine independent languages. Associated with each higher level language, there will be a compiler (a program) in the computer which will translate the higher level language into machine language so that the machine (computer) can easily understand our instructions written in higher level language. Fortran language, a hybrid of English and Mathematics, allows all the English alphabets in capitals and the usual arithmatic operations + and –. The division is denoted by / (slash) and multiplication by *. Fortran stands for Formula Translation. This language was developed by IBM. For writing an arithmetic statement in Fortran, we need to identify the integer and real constants and variables. Integer variable is a combination of one to five letters or digits, the first character in the name must be one of the following letters I, J, K, L, M and N. The name must not

contain special characters. The real variable is a combination of one to five letters or digits, which must not contain any special character and the first character must be a letter other than I, J, K, L, M and N. A Fortran arithmetic expression is a series of variables and constants connected by arithmetic operation symbols. The general form of an arithmetic statement is variable = expression.

14.10 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following in detail.*

1. Explain "Machine language instructions". What are the difficulties we come across while writing instructions in Machine language.

2. List out the allowable characters in FORTRAN IV.

3. a) Identify each of the following as a Real constant or integer constant.

   1. 560 2. –1690 3. 1680 4. 10.753

   5. .008 6. –13.5 E – 15 7. 0.0 E – 10.

   b) Which of the following are valid integer variable names, valid real varialbe names, invalid integer variable names specifying reasons (in the absence of type declaration statement)

   1. SUM 2. AT* B 3. 5.AB 4. INJ

   5. X + Y 6. POJ 7. NGRI 8. BETA

4. Define FORTRAN arithmetic expression and give the important points concerning the evaluation of arithmetic expression.

5. Write FORTRAN expressions corresponding to each of the following mathematical expressions.

   i) $\frac{x}{a} + \frac{y}{b} + \frac{z}{c}$

   ii) $ab + \frac{c}{d} + \frac{ef}{gh} + \frac{d^3}{x^2 + y^2}$

   iii) $\sqrt{S\ (S - a)\ (S - b)\ (S - c)}$

   iv) $\frac{(\alpha + \beta^2 + \alpha\beta)^2}{\sqrt{\alpha + \beta + 1}}$

II. *Briefly anser the following*

1. Explain (a) Complier (b) Higher level languages.

2. Mention the arithmetic operations that are permitted by FORTRAN and give the symbols that are used to represnt in FORTRAN language.

3. Define FORTRAN real variable and integer variable and give two examples to each.

4. Each of the following arithmetic statements contains atleast one error. Point out errors.

   i) Z = X * Y – 17,210.0

   ii) X * 5 = 5. * Z – 12.74Y

   iii) 15 = I + J/5 + 7 * J

   iv) VEC = 5 (T3 – T2 + T1)/7

   v) VEL = 12.74 E – (5 * I) – T

5. Write equivalent mathematical expressions for the following expressions.

   i) –B + SQRT (B * B – 4.0 * A * C)

   ii) U + V/(R + S) * * 7

   iii) A + B/(A – B)

   iv) ALOG (COS (X) + 5. * SIN (X))

   v) ATAN (X + Y) + ALOG (SQRT (X))

6. Write FORTRAN arithmetic statements, using the letters in the formula as variable names.

   i) $\theta = \tan^{-1}\left(2xy \Big/ \left(x^2 + y^2\right)\right)$

   ii) $r = \frac{16pr}{\pi d^3}\left(1 + \frac{d}{4r}\right)$

   iii) $x = R \cos\theta \cos\phi$

   iv) $\lambda = 9.118 \Big/ \left(\frac{1}{N^2} - \frac{1}{M^2}\right)$

   v) $V = \tan x + \log_e (\cos x + \sin y)$

7. What is the value of P in the following :

   i) P = 5/7 + 6 + 1.5

   ii) P = 10/3 * (4/6) + 7 – 2 * 5

   iii) P = (1.0/5.0) + 50 + 5.0

8. What is the value of I calculated in the following arithmetic statements.

   i) I = B/2. + B * 4./A – B + A * * 3 for A = 1.5, B = 3

   ii) I = (I + J)/K – L, for I = J = 2. K = L = 4

   iii) I = J * 2/3 + K/4 + 6 – J * * 3/8 for J = 2, K = 5

Answers

I. 3. (a) 1, 2 – Integer Constrants, all others Real Constansts.

(b) 4, 7 – Integer variables since first letters are I and N.

1, 6, 8 – Real variables since first letter is other tan I, J, K, L, M, N

2, 3, 5 – Not variables due to presence of special characters.

5. i) X/A + Y/B + Z/C

ii) A * B + C/D + E * F/(G * H) + D * * 3/ (X * X + Y * Y)

iii) SQRT (S * (S – A) * (S – B) * (S – C))

iv) (ALFA + BETA + * BETA + ALFA * BETA) * * 5/SQRT

(ALFTA + BETA + 1)

II. 4. i) Comma not allowed (ii) Left side there must be variable name must not expression and also * is missing in between 12.74 and Y (iii) 15 not allowed on left side (iv) "*" is missing (v) After E, expression not allowed.

5. i) $-b + \sqrt{b^2 - 4ac}$

ii) $U + \frac{V}{(R + S)^7}$

iii) $A + \frac{B}{A - B}$

iv) $\log (\cos x + 5 \sin x)$

v) $\tan^{-1} (x + y) + \log \left(\sqrt{x}\right)$

6. i) THETA = ATAN (2. * X * Y/(X * X + Y * Y))

ii) R = 16. * P * R /(PI * D * * 3) * (1 + D/4. * R)

iii) X = R * COS (THETA) * COS (PHI)

iv) LAMDA = 9.118/(1/N * * 2 – 1/M * * 2)

v) V = SIN (X)/COS (X) + ALOG (COS (X) + SIN (Y))

7. i) 7.5 ii) –3 iii) 55.2

8. i) 9 ii) –3 iii) 7

UNIT–15 : CONVERSION OF NUMBERS

Contents

15.1 AIMS AND OBJECTIVES

After going through this unit you will be able to : (i) convert a given number in one base system to anumber in another base system.

15.2 INTRODUCTION

We use nubers every day. For example consider the number 395. What does it mean? It tells how many there are certain things, they may be books, cars, rupees or students or some other items. The number tells that there are three hundred and ninety five certain things. (3 hundreds plus 9 tens, plus 5 units). i.e., $3 \times 100 + 9 \times 10 + 5 = 395$ or $3 \times 10^2 + 9 \times 10^1 + 5 \times 10^0 = 395$. Therefore a number is made up of digits having positional values. The positional values can be illustrated by the chart as given below.

10^4	10^3	10^2	10^1	10^0

Fig. 1

Observe that, as we move to the left, each position is worth 10 times the value to its right. Consider the number 1561. Place this on the right most divisions of the chart.

1	5	6	1
10^3	10^2	10^1	10^0

Fig. 2

$\therefore$ 1561 means

$1 \times 1000 = 1000$

$5 \times 100 = 500$

$6 \times 10 = 60$

$1 \times 1 = 1$

Sum $= 1561$

The number system we have been discussing which is used commonly is called "Decimal system or base 10 system". The word decimal is derived from the Latin word meaning 'ten'. The decimal system employs ten digits. They are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. We may express values as large or small as we please using decimal numbers. Probably, since human being have ten fingers, we use ten digits in our number system (i.e., we use decimal system).

Base 10 is not the only base that may be employed to tell how many there are of various things. We may employ numbers based upon two digits. (base 2 or binary number system) three digits (base 3 system), four digits (base 4 system) etc. In this unit, we discuss some important base systems and conversion of numbers from one number system to another.

15.3 BASE CHARTS

When we want to convert a number from any base to base 10, we have to construct a chart, that shows the decimal values for each of the divisions. The right most digits of the chart is always 1. The next digit to the left is the same as the base being converted. Suppose the base is 3, so the value placed in this division is the multiplier to use for all remaining divisions to the left i.e.,

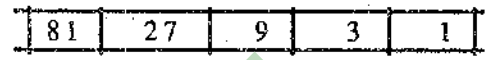


Fig. 3

So if base is b, the base b chart is as follows :

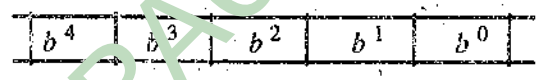


Fig. 4

15.4 BASE 8 SYSTEM OR OCTAL SYSTEM

In a particular country, let us imagine all are having only four fingers on each hand. Instead of employing the base 10 system they may prefer a system based upon 8 digits viz., 0, 1, 2, 3, 4, 5, 6 and 7, which is called base 8 system or octal system. When they write numbers they use the positional values as shown in the chart given below.

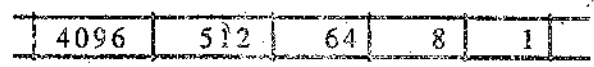


Fig. 5

Excepting the rightmost position, each position is 8 times the value of the position to its right.

Consider a number 564 in base 8 system. Placing the digits of the number above the rightmost divisions of the chart, as given below, it indicates that the value is made up of the sum of five 64s, 6 eights and 4 units.

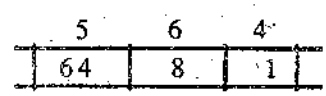


Fig. 6

So, 564 in base 8 gives $5 \times 64 = 320$

$6 \times 8 = 48$

$4 \times 1 = 61$

Sum $= 372$

When discussing numbers to various bases we have to specify which base we mean, to avoid confusion. We do it by indicating the base by writing a subscript to the right of the number. Thus 564_8 means that the number is written in the base 8 system. Therefore from the above example, $564_8 = 372_{10}$. From here onwards, assume that if there is no subscript, it is written in base 10 system.

$\therefore 564_8 = 372.$

Ex. 1 : Convert 1046_8 to base 10 system.

Solution : Placing the base 8 number on base 8 chart, we get

1	0	4	6	
512	64	8	1	

Fig. 7

$1 \times 512 = 512$

$0 \times 64 = 0$

$4 \times 8 = 32$

$6 \times 1 = 6$

Sum $= 550$

$\therefore 1046_8 = 550$

Ex. 2 : Find the decimal equivalent of 456_8.

Solution :

Placing the base 8 number on base 8 chart, we get

4	5	6
64	8	1

Fig. 8

$\therefore 4 \times 64 = 256$

$5 \times 8 = 40$

$6 \times 1 = 6$

Sum $= 302$

$\therefore 456_8 = 302$

Now we consider the case of converting a number expressed in decimal system to a number in base 8 system. the procedure is, divide the number in decimal system repeatedly by 8. Then the

remainders in each division in downward direction give the base 8 equivalent. For example, if we want base 8 equivalent of 258, then divide 258 repeatedly by 8 as follows.

		Remainders
	0	4
8	4	0
8	32	2
8	258	

$\therefore 258_{10} = 402_8$

Ex. 3 : Find the base 8 equivalent of 2000_{10}.

Solution :

Divided 2000 repeatedly by 8, we get

		Remainders
	0	3
8	3	7
8	31	2
8	250	0
8	2000	

$\therefore 2000 = 3720_8$

So far we dealt with integers (whole numbers). Sometimes we may wish to express a value that has an integer part and a fractional part or we may wish to express a pure fraction. Let us take up pure fraction first. Consider the question of finding decimal equivalent of $.462_8$.

The position of the first digit to the right of the decimal point is worth 8^{-1} or .125, the second digit of 8^{-2}, the third is of 8^{-3} and so on. The base chart may therefore be extended to the right end therefore it looks like

				4	6	2	
	64	8	1	.125	.015625	.001953125	
	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}	

Fig. 9

$\therefore$ The value of the given number is

$4 \times .125 = .5$

$6 \times .015625 = .093750$

$2 \times .001953125 = .003906250$

$.597656250$

$\therefore .462_8 = 0.597656250$

Now given any number in base 8, including those that contain integer part and fraction, we can convert them easily.

Ex. 4 : Find the decimal equivalent of 156.21_8.

Solution :

Place the number over base 8 chart.

1	5	6	2	1
64	8	1	.125	.015625

Fig. 10

The decimal equivalent of the given number is

$1 \times 64 = 64$

$5 \times 8 = 40$

$6 \times 1 = 6$

$2 \times .125 = 0.250$

$1 \times 0.15625 = 0.015625$

110.265625

$\therefore 156.21_8 = 110.265625$

Now consider the problem of converting base 10 fraction to base 8. We know if the number is pure integer, we get base 8 equivalent by rpeatedly dividing by 8. Here if the number is a fraction, then we have to multiply repeatedly by 8 as shown below, then the spill over digits give base 8 equivalent. Suppose we want to find base 8 equivalent of .833, then

.	833 × 8
6.	664 × 8
5.	312 × 8
2.	496

$\therefore 833_{10} = .652_8$

Observe that a verticle line has been placed extending downward from the decimal point location in the value .833. Now, every thing to the right of the line is multiplied by 8. The result is 6.664, the digits to the right of the decimal point are written at the right of the vertical line, while 6 is written to the left of the line, i.e., 6 spills over the line. Repeat the process. Note that this time we have to multiply only .664 but not 6.664, continue this process. The spill over digits will give the base 8 equivalent. This answer is not complete. The value of $.833_{10}$ does not exactly equal to $.652_8$ since we terminated multiplying by 8 after 3 multiplications. We can continue this multiplication if we wish to get more accuracy. In many problems exact base 8 equivalents to decimal fractions cannot be obtained, but we may make the conversion as close as we please by continuing to multiply by 8.

Ex. 5 : Find the Octal equivalent of .750.

Solution : Multiply repeatedly by 8, we get

.	750
	× 8
6.	000

$\therefore .750_{10} = .6_8$

Ex. 6 : Convert .453 to base 8

Solution : Multiply repeatedly by 8, we get

.	453
	× 8
3.	624
	× 8
4.	992
	× 8
7.	936

$\therefore .453_{10} = .347_8$

Ex. 7 : Find the Octal equivalent of 916.78

Solution : First we convert the integer portion and then the fractional portion.

Integer portion : Divide 916 by 8 repeatedly

	0	1
8	1	6
8	14	2
8	114	4
8	916	

$\therefore 916 = 1624_8$

Fracrional Part : Multiply .78 repeatedly by 8

.	78
	× 8
6.	24
	× 8
1.	92
	× 8
7.	36
	× 8
2.	88

$\therefore .78 = .6172_8$

So. 916.78 equals approximately to 1624.6172_8.

Ex. 8 : Convert 634.64025 to base 8.

Integer Part :

	0	1
8	1	1
8	9	7
8	79	2
8	634	

$\therefore 634 = 1172_8$

Fracrional Part :

.	640625
	× 8
5.	125000
	× 8
1.	000000

$\therefore 0.640625 = 0.51_8$

So, Octal equivalent of 634.640625 is 1172.51

i.e., $634.640625_{10} = 1172.51_8$.

15.5 BASE 6 SYSTEM

In the last section we discussed base 8 system. If we want we may write numbers in base 6 system also. In this we have six symbols to represent the numbers and those are 0, 1, 2, 3, 4, 5. The same rules, which we used in the previous section for the conversion from base 8 to decimal anddecimal to base 8, will hold good for the conversion from base 6 to decimal and decimal to base 6. The only difference is that there the base is 8 and here the base is 6. For illustration let us consider few examples.

Ex. 9 : Find the decimal equivalent of 35044.34_6.

Solution :

Place the digits of the number above the base 6 chart.

3	5	0	4	4	3	4
1296	216	36	6	1	.1667	.0278
6^4	6^3	6^2	6^1	6^0	6^{-1}	6^{-1}

Fig. 11

$\therefore$ $3 \times 1296 = 3888$

$5 \times 216 = 1080$

$0 \times 36 = 0$

$4 \times 6 = 24$

$4 \times 1 = 4$

$3 \times .1667 = 0.5001$

$4 \times .0278 = 0.1112$

4996.6113

So, $35044.34_6 = 4996.6113$

Ex. 10 : Find the base 6 equivalent of 790.674

Solution : Integer Portion : Divide 790 repeatedly by 6.

	0	3
6	3	3
6	21	5
6	131	4
6	790	

$\therefore 790 = 3354_6$

Fracrional Part : Multiply .674 repeatedly by 6

.	674 × 6
4.	044 × 6
0.	264 × 6
1.	584 × 6
3.	504

$\therefore .674 = 4013_6$

So, $790.674 = 3354.4013_6$

15.6 BASE 16 SYSTEM OR HEXA DECIMAL SYSTEM

It is also possible to express numbers in bases greater than 10. In this case we are faced with the problem of inventing symbols to stand for values greater than 9. Base 16, which is also called Hexa decimal is useful in data processing. We require 16 symbols to express numbers in this. The 16 symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Here we are using the symbols A, B, C, D, E and F to represent 10, 11, 12, 13, 14, 15 to avoid confusion. Therefore the numbers in base 16 look like ABC_{16}, $95CD_{16}$ etc. The same rules, which we used in the previous sections to convert from one base to decimal and from decimal to another base, will hold good here also. For illustration let us consider few examples.

Ex. 11 : Find the decimal equivalent of $4B3F5.A9_{16}$.

Solution :

Place the digits of the number above the base 16 chart.

4	B	3	F	5	A	9
65536	4096	256	6	1	.0625	.00390625

Fig. 12

$$
\begin{aligned}
\therefore \quad 4 \times \quad 65536 &= 262144 \\
B \times \quad 4026 &= 45056 \\
3 \times \quad 256 &= 768 \\
F \times \quad 16 &= 240 \\
5 \times \quad 1 &= 5 \\
A \times \quad .0625 &= .625 \\
9 \times .00390625 &= .03515625 \\
&\underline{308213.66015625}
\end{aligned}
$$

$$4B3F5.A9_{16} = 308213.66015625$$

Ex. 12 ; Find the Hexa decimal equivalent of 2754.984.

Solution :

	0	10 = A
16	10	12 = C
16	172	2
16	2754	

$$\therefore 2754 = AC2_{16}$$

Fracrional Part : Multiply .984 repeatedly by 16

.	984 × 16
15. (F)	744 × 16
11. (B)	904 × 16
14. (E)	464

$$\therefore .984 = FBE_{16}$$

$$\therefore 2754.984 = AC2.FBE_{16}$$

15.7 CONVERSION OF NUMBERS FROM ONE BASE SYSTEM TO ANOTHER BASE SYSTEM

So far we came to know about conversion from any base to decimal and decimal to any base. In this section we consider the case of converting a number in any base to its equivalent in any other base. The method is easy and straight forward. According to this, first convert the number to base 10 then convert it to new base i.e.,

A number in any base → Base 10 → The number in other base

Let us consider few examples for illustration.

Ex. 13 : Find the base 5 equivalent of 546_7.

Solution : First we find the base 10 equivalent by using base 7 chart.

5	4	6
49	7	1

Fig. 14

$$\therefore 5 \times 49 = 245$$
$$4 \times 7 = 28$$
$$6 \times 1 = 6$$
$$\text{Sum} \quad 279$$
$$\therefore 546_7 = 279$$

Now we get the base 5 equivalent by dividing 279 repeatedly by 5

	0	2
5	2	1
5	11	0
5	55	4
5	279	

$$\therefore 279 = 2104_5$$
$$\therefore 546_7 = 2104_5$$

So base 5 equivalent of 546_7 is 2104_5.

Ex. 14 : Convert $AB6.C5_{16}$ to base 8.

Solution :

First we find decimal equivalent of $AB6.C5_{16}$ using base 16 chart.

A	B	6	.	C	5
256	16	1		.0625	.00390625

Fig. 15

$$A \times 256 = 2560$$
$$B \times 16 = 176$$
$$6 \times 1 = 6$$
$$C \times .0625 = .7500$$
$$5 \times .00390625 = .01953125$$
$$2742.76953125$$

$$\therefore AB6.C5_{16} = 2742.76953125$$

Now convert 2742.7695 (rounded) to base 8

	0	5
8	5	2
8	42	6
8	342	6
8	2742	

$\therefore 2742 = 5266_8$

Fractional Part :

.	7695 × 8
6.	1560 × 8
1.	2480 × 8
1.	9840

$\therefore .7695 = .611_8$

So, $AB6.C5_{16} = 5266.611_8$

15.8 BINARY (BASE 2) NUMBER SYSTEM

So far we considered representation of numbers in the base 8, hexadecimal system etc. Base 2 is important in Data processing. Internally most computers store numbers in binary system. There are only two digits in binary system and they are '0' and '1'. The same rules, which we used in the previous sections to convert from one base to decimal and from decimal to another base, will hold good here also. Let us consider few examples for illustration.

Ex. 15 : Convert 483 to its binary equivalent

	0	1
2	1	1
2	3	1
2	7	1
2	15	0
2	30	0
2	60	0
2	120	1
2	241	1
2	483	

$\therefore 483 = 111100011_2$

Ex. 16 : What is the decimal equivalent of 101.1011_2.

Solution : Using base 2 chart we can find the decimal equivalent

1	0	1	.	1	0	1	1
4	2	1		5	.25	.125	.00625

Fig. 16 Base 2 chart

$1 \times 4 + 0 \times 2 + 1 \times 1 = 5$

$1 \times 5 + 0 \times .25 + 1 \times .125 + 1 \times .0625 = 5.6875$

Hence $101.1011_2 = 5.6875$.

Ex. 17 : Find the base 2 equivalent of .83

Solution :

Multiply .83 repeatedy by 2 then spillovers give the binary equivalent.

.	83
	× 2
1.	66
	× 2
1.	32
	× 2
0.	64
	× 2
1.	28
	× 2
0.	56

$\therefore .83 = .11010_2$

.83 does not exactly equal to .11010. In may problems exact binary to decimal fractions cannot be obtained, but we may make the conversion as close as we please by continuing to multiply by 2.

For the numbers which contain more number of zeros and ones, using base chart and converting is a tedious job. In such cases we use base 8 of base 16 systems as stepping stones to convert base 2 numbers to base 10.

For example consider the problem of converting 1010100101.1110101_2 to decimal. We use the base 8 as steping stone. Break up the number into groups of three digits working right and left from the decimal point and write the equivalent base 8 digits, using base 2 to base 8 convertion chart. Then use base 8 chart to find the decimal equivalent,

base 8 units	binary equivalent
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Now we solve the given problem by using above chart.

$$\underset{1}{\underline{001}}\ \underset{2}{\underline{010}}\ \underset{4}{\underline{100}}\ \underset{5}{\underline{101}} \cdot \underset{7}{\underline{111}}\ \underset{2}{\underline{010}}\ \underset{4}{\underline{100}}$$

$\therefore 1010100101.1110101_2 = 1245.724_8$

Now use base 8 chart to find decimal equivalent.

1	2	4	5	.	7	2	4
512	64	8	1		.125	.015625	.001953125

Base 8 chart

$1 \times 512 + 2 \times 64 + 4 \times 8 + 5 \times 1 = 677$

$7 \times .125 + 2 \times .015625 + 4 \times .001953125 = .9140625$

$\therefore 1110010101.1110101_2 = 677.9140625$

Ex. 18 : find the decimal equivalent of

11100100101.0011011_2

Solution :

Here we use the base 16 as the stepping stone. We consider the base 2 to base 16 conversion chart.

base 2	base 16
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Base 2 to base 16 conversion chart

Now breakup the number into groups of four digits and write the corresponding base digits.

$$\therefore \underset{7}{\underline{0111}}\ \underset{2}{\underline{0010}}\ \underset{5}{\underline{0101}} \cdot \underset{3}{\underline{0011}}\ \underset{6}{\underline{0110}}$$

Now we use base 16 chart to find decimal equivalent.

71	2	5	.	3	6
256	16	1		0.0625	.00390625

Base 16 chart

$7 \times 256 + 2 \times 16 \times 5 + 1 = 1829$

$3 \times .0625 + 6 \times .0039625 = .21093750$

$\therefore 11100100101.0011011_2 = 1829.21093750$

Ex. 19 : Find the binary equivalent of 0.064

Solution :

Here we can use base 8 or base 16 as stepping stone. now in this problem, let us use base 16.

.	064 × 16
1.	024 × 16
0.	384 × 16
6.	144 × 16
2.	304

$\therefore .064 = .1062_{16}$

Now using base 16 to base 2 conversion chart (Fig. 20) we get

$.064 = .1062_{16} = .0001000001100010_2$

Ex. 20 : Convert the number 474.326 to binary.

Solution :

First convert the integer portion and then the fractional portion. Here let us use base 8 system as the stepping stone.

	0	7
8	7	3
8	59	2
8	474	

$\therefore 474 = 732_8$

$= 111011010_2$

Fracrional Part :

.	326 × 8
2.	608 × 8
4.	864 × 8
6.	912 × 8
7.	296

$\therefore .326 = .2467_8$

$= .010100110111_2$

$\therefore 474.326 = 732.2467_8 = 111011010.010100110111_2$

15.9 SUMMARY

We are familiar with the numbers with base 10. The number is made of digits having positional values. When disucssing numbers to various bases, we have to specify which base system we mean to avoid confusion and we do it by indicating the base by writing a subscript to the right of the number. We have converted numbers from one base system to another base system.

15.10 SAMPLE EXAMINATION QUESTIONS

I. Answer the following questions in detail

1) a) What are the different symbols employed in (i) binary system (ii) Octal system and (iii) Hexa decimal system.

b) Find the Octal and Hexa decimal equivalents of .833

2. Find the decimal equivalent of

(i) 256_6 (ii) 742_8 (iii) 120_3 (iv) 444_5

3. What are the decimal equivalent of

(i) $C4D_{16}$ (ii) ABC_{16} (iii) $7AE_{16}$

II. Briefly answer the following.

1. Convert (a) 3074 to base 8, (b) 960 to base 16

2. Express the following numbers in binary form

(i) 15.8 (ii) 6.02 (iii) .833

3. What is the decimal equivalent of

(i) 0.1001 (ii) 10101.001 (iii) 01001.00101

Answers

I 1. (a) (i) 0,1 (ii) 0, 1, 2, 3, 4, 5, 6, 7 (iii) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

(b) $.652_8$ and $D53_{16}$

2. (i) 106 (ii) 482 (iii) 15 (iv) 124

3. (i) 3149 (ii) 2748 (iii) 1966

II. 1. (a) 6002_8 (b) $3C0_{16}$

2. (i) 1111.110011_2 (ii) 110.000001 (ii) .1101

3. (i) .2812 (ii) 21.1250 (iii) 9.152

4. 0011101110100110

UNIT–16 : INPUT - OUTPUT STATEMENTS

Contents

16.1 AIMS AND OBJECTIVES

After going through this unit you will be able to write a program with the complete format specifications for input and output statements.

16.2 INTRODUCTION

The aim of a programmer should be to write a program in such a manner that it not only solves the problem on hand but should be capable of solving similar types of problems for different sets of data. To achieve this the data for the problem and the procedure should be separate. Whenever the data is not given as a part of the program, the data can be transmitted into the computer by means of input statement. The input statement written in to the source program causes the object program to transmit the input data from the input device into the memory of the computer. The input statement indicates the type of input device, from which the data is to be transmitted, into the memory, the number of variables and the order in which their values are to be read. The input statement must be followed by a statement which is called FORMAT statement, which specifies about the format i.e., the form size of the value of the variables in which the data will be supplied.

Another important part of a program is to output the results of the problem, from the memory of the computer, in a required form. This is achieved by means of "OUTPUT" statement, and the corresponding FORMAT statement which indicates the form in which the results should appear.

Therefore, Input-Output statements consist of two parts, an executable part and declarative part. The executable part commands certain items to be read into the memory or read out of it. The declarative part gives information on the exact form in which data will be found in the data cards or is

to be printed out. This declarative part is called FORMAT statement. Some compilers accept Input - Output Statements without FORMAT statements also. These statements are called FORMAT free Input - Output statements. When FORMAT is not specified the input data can be supplied in a flexible form and the output is supplied in a standard form.

We limit our discussion to one form of input, namely, information punched on cards. This is performed by the READ statement. Similarly we consider one type of output statement, namely the WRITE statement, which causes the results to be printed.

Each READ or WRITE statements specifies 1) The unit number of the input/output device used. 2) The statement number of the associated FORMAT statement. 3) The list of variable names to which their values are to be assigned or from which the results are to be output.

We may choose any of the input/output devices. We identify them by the numerical codes which we can know from the manual of the computer used. These code numbers vary from one computer to other. For simplification, let us use in all the examples that follow, a card reader as an input device and a line printer as an output device. Let unit number 5 designate a card reader and unit number 9, a line printer, in a particular computer. First we know about Format - free input and output statements.

16.3 FORMAT – FREE INPUT AND OUTPUT STATEMENTS

The general form of the input statement is *READ, list* where list represents a set of variable names, each separated from the other by a comma. The execution of this statement will cause the computer to read in the data from an input device for all the variables in the list. For example consider the following statement.

READ, X, Y, Z

By the above statement, the computer reads the numerical values of X, Y and Z from data cards (i.e., cards where data is punched) and store these values in the memory locations whose addresses are X, Y and Z. The values of the variables should be punched in the same sequence as in the list and must be seperated by comma or a blank. The values may be punched starting from first column upto and including column 80, where as FORTRAN statements must be punched only from column 7. Columns 1 to 5 are reserved for statement numbers of FORTRAN statements. If some values remain, they may be punched on another card but a single value should not be split between two cards.

For example in the following statement READ, A, B, C the values of A, B and C are 11.75, 0.125 and 150.75 say, then they are punched on data card as follows where b represents a blank column.

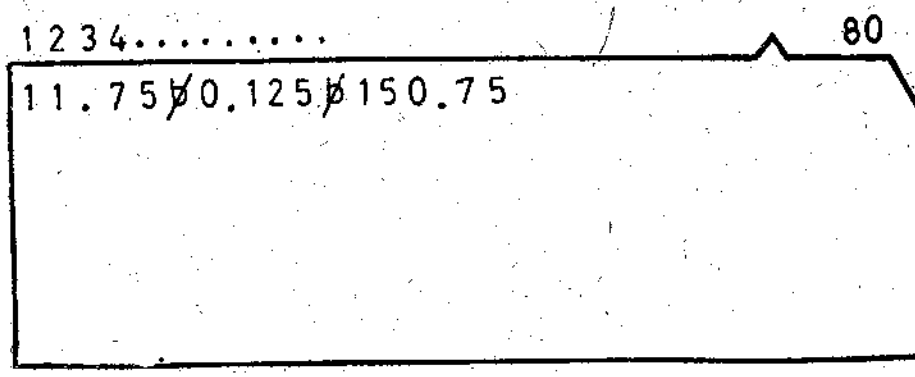


Fig. 1 : *Data Card Containing values of A, B, C*

A READ statement always causes a new data card to the read. Therefore the value corresponding to the lists of two different READ statements should not be punched on the same data card.

The general form of the output statement in PRINT, *list* where list represents a set of variable names, each sepdrated by a comma. This statements causes the values of the variables specified in the list to be printed out.

For example, the values of I, J and K stored in the memory unit are 150,2700 and 192 say. Then if we want to ask the computer to print out the values of I, J and K, we have to use the statment PRINT, I, J, K.

Some examples of valid FORMAT free input/output statement are

1. READ, L, M, N, A, B
2. READ, ALP, KOU, BETA
3. PRINT, PHI, GAMA, X, Y
4. PRINT, P, Q, R

Before we discuss about formatted input - output statements, we try to know about statement number and FORMAT specifications.

16.4 STATEMENT NUMBER

We know that each READ or WRITE statement specifies the statement number of the associated FORMAT statement. A statement number is usually used when we want to refer to a statement. This number may be 5 digits long. It may be punched any where betwen columns 1 and 5. A statement number should be an unsigned non zero integer.

Some valid statement numbers :

1) 50 2) 250 3) 95 4) 1540 5) 2345

Some invalid statement numbers.

1. –75 due to minus sign.
2. .9KI because this is not an integer.
3. 15, 35 due to the presence of comma.

4. 2532167 since it exceeds the range (i.e., more than 5 digits)

5. 0 since statement number should be an unsigned non zero integer.

Remember the following points

1. Statement numbers in a program need not be in ascending order, they can be in any order.
2. If a statement extends over two or more cards, only the first card should have statement number.
3. The same statement number cannot be assigned to two different statements.

16.5 FORMAT SPECIFICATIONS FOR NUMERALS

FORMAT statement is a non executable statement. Formatted input - output statement must be followed by FORMAT statement which gives information to the complier 1) in what form are the data values, punched or printed. 2) How many card columns are allocated to each input value? 3) How many printing positions are allocated to each output value? 4) Is each value fixed or floating point?

The contents of one 80 column card to be read in is called a Record for input. For paper tapes columns 1 to 72 constitute a record. A record for printing is normally a 132 column line. A record consits of a number of fields. A field is normally the set of columns on a card reserved for one variable. The number of columns in a field is called the field width. For example the number 1.724 has a field width 5 and the number –15.25 has a field width 6. The decimal point is to be counted as a character and also negative sign to be counted as a character.

In the FORMAT statement there is field specification for each variable in the list of the READ or WRITE statement. The complete set of field specifications in a FORMAT statement is enclosed in parenthesis and the individual field specifications are separated by commas.

The general form of the FORMAT statement is

$$n \text{ FORMAT } (F_1, F_2, \ldots\ldots, F_m)$$

where n is the statement number and $F_1, F_2, \ldots\ldots F_m$ are field specifications of m variables in the list of READ or WRITE statement.

Here we discuss the FORMAT specifications for numerical quantities. There are three FORMAT specificatins. They are

1. I FORMAT specification.
2. F FORMAT specification.
3. E FORMAT specification.

16.5.1 I - FORMAT Specification or Integer Field Specification

If a data field on the card represents an integer quantity, it is specified by using code I to indicate that it is an integer. Its form is 'Iw', where *w* stands for field width.

For exmaple, the value of variable J is 516 say, then the field specification is I3 and the corresponding FORMAT specification is 5 FORMAT (I3) where 5 is the statement number.

Suppose if the values of the variables K, LN, LM and J are punched on the card as follows.

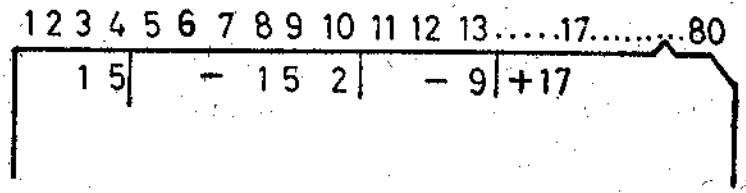


Fig. 2 : *Data card containing values of K, LN, LM, J*

then, the field specifications of K, LN, LM and J are I4, I6, I3 and I4 respectively and the corresponding FORMAT statement is

15 FORMAT (I4, I6, I3, I4)

where 15 is statement number.

Note : The values in the data fields must be right adjusted, i.e., the integer value must be written with the units appearing in the right most column of the field. Spaces in the data fields are interpreted as zeros.

16.5.2 F - FORMAT Specification or Real Field Specification

If the data to be transmitted is in the decimal form without exponent, then this specification may be used. The general form of this specification is F*w* .*d* were *w* indicates the total number of characters in the field (including sign, decimal point and blanks if any) and *d* represents the number of decimal digits after the decimal point.

For example the value of X is –50.123, then the field specification if F7.3.

For example, if the values of X, Y and Z are punched on card as follows,

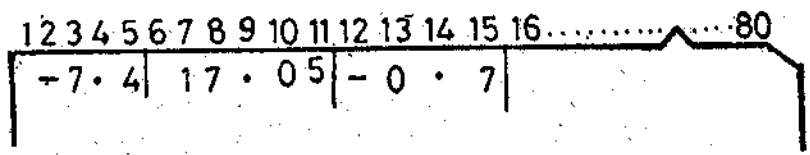


Fig. 3 : *Data card containing values of X, Y and Z.*

then the format specifications are F5.1, F6.2 and F4.1 and the corresponding FORMAT statement is

10 FORMAT (F5.1, F6.2, F4.1)

16.5.3 E - FORMAT Specification or Real Field Specification - E

For real numbers involving exponent notation, we use E field Specification. The general form is E*w* .*d* where E indicates the presence of exponent in the real number, *w* indicates the total number of characters (including sign, decimal point, exponent and blanks if any) and *d* represents the number of decimal digits after the decimal point (excluding the exponent).

For example, if the values of P, Q, R are 12.72 E – 15, – 7.5 E – 05 and 17.325 E +15 then the corresponding FORMAT specifications are E9.2, E8.1 and E10.3 respectively the FORMAT statemtnt is

25 FORMAT (E9.2, E8.1, E10.3)

Note : If several consecutive field specifications are the same, we may write a repetition number in front of the first of these specificatins in FORMAT statement.

For example, if the values of L,M, N, P and Q are punched on data card as follows,

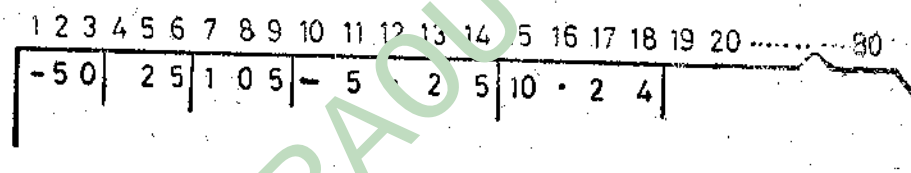


Fig. : *Data card containing values of L, M, N, P and Q.*

then the FORMAT specifications L, M, N, P and Q are I3, I3, I3, I3, F5.2 and F5.2 respectively. Corresponding FORMAT statement is

40 FORMAT (I3, I3, I3, F5.2, F5.2)

Here FORMAT specifications of L, M, N, are same and Format specification of P and Q are same. Therefore using repetition number, above FORMAT statement may also be written as

40 FORMAT (3I3, 2F5.2)

16.5.4 Blank Fields

Blank columns may be provided in an input data card or output record by using the specification *w*X, where *w* is the number of columns to be skipped while reading or the number of columns to be left blank on output.

For example, the Format Statement

15 FORMAT (I6, 5X, 17)

will cause 5 blank columns between the two numbers on the data card specified by 16 and 17.

For example, if the values of A, B, C are punched on data card as follows,

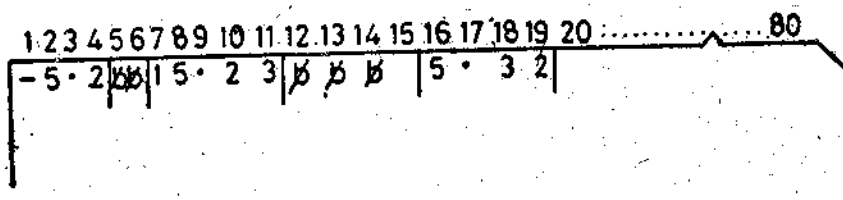


Fig. 5 : *Data card containing the values of A, B, C.*

the corresponding FORMAT statement is

60 FORMAT (F4.1, 2X, F5.2, 4X, F4.2)

16.6 FORMATTED INPUT AND OUTPUT STATEMENTS

Formatted statements allow us to organize the data in several ways, since we can specify separate formats for each variable name in the I/O (Input/Output) list. We discuss in detail the formatted READ and WRITE statements.

The general from of the READ statement is READ (i, n) List

$$n \text{ FORMAT } (S_1, S_2, \ldots, S_m)$$

when n is the FORMAT statement number, i is the input unit code number which vary from one computer to other and from one unit to another unit and $S_1, S_2, \ldots, S_m$ are Format specifications.

As mentioned early, we use code number 5 for card reader in all examples that follow.

For example consider the following READ statement

READ (5, 10) X, Y, Z

10 FORMAT (F5.2, F10.2, F7.3)

The above statement causes the computer to read in the values of X, Y, Z by card reader from data card which were punched according to the format specification F5.2, E10.2 and F7.3.

Few more examples of READ statement

1. READ (5, 75) L, M, N, A, B

   75 FORMAT (I4, I7, I3, F 7.3, E 15.6)

2. READ (5, 105) I, J, X, Y, Z

   105 FORMAT (I6, 2X, I4, 2X, F 7.2, 2X, F 5.1)

3. READ (5, 17) BETA, GLAM, GAMA, K

   17 FORMAT (2 F 8.4, F 5.2, I6)

The general form of the WRITE statement is

WRITE (i, n) List

n FORMAT $(S_1, S_2, \ldots, S_m)$

where i is output unit code number which vary from one computer to other and from one unit to another, n is Format statment number and $S_1, S_2, \ldots, S_m$ are FORMAT specifications.

As mentioned earlier, we use code number 9 for line printer in the examples that follow. One additional point to be noted for outputting the results through a line printer is the carriage control character and is essential. This character must appear as the first character in the FORMAT statement. IX or IHb is used to indicate single spacing, 1H0 for double spacing of lines on printer and 1H1 for skipping to next page.

Now few examples of WRITE statement

1) WRITE (9, 15) L, M, N

15 FORMAT (1Hb, I5, 5X, I4, 5X, I8)

2) WRITE (9, 2) X, Y, Z, A, B

2 FORMAT (1H1, 3F 7.2, E10.3, F5.2)

3) WRITE (9, 75) A, B, C, L

75 FORMAT (1X, F 8.6, 2X, F11.3, 3X, E 10.3, 3X, I5)

16.7 PAUSE, STOP AND END STATEMENTS

PAUSE statement is used to halt a program temporarily, so that operator can restart the program from the PAUSE statement through console type writer. This is an executable statement. Its general from is

PAUSE

or PAUSE n

when n is an unsigned Octal integer constant consisting of 1 to 5 Octal digits. 'n' is printed by the console type writer whenever the statement is executed.

STOP statement is used to stop the running of a program, and the operator cannot restart. This is also an executable FORTRAN statement. Its general form is

STOP

or STOP n

where n is an unsigned Octal integer constant of 1 to 5 Octal digits.

Examples

1. PAUSE
2. PAUSE 27
3. PAUSE 532
4. STOP
5. STOP 10
6. STOP 430

END statement is a non executable statement and is used to inform the compiler, that it has reached to the end of the problem.

Therefore END statement indicates the physical end of a FORTRAN program and this statement should be present in every program and it should always be the very last statement in each program, physically.

Ex.1 : Write the READ statement to read in the values of A, B, C which are punched on the data card as follows.

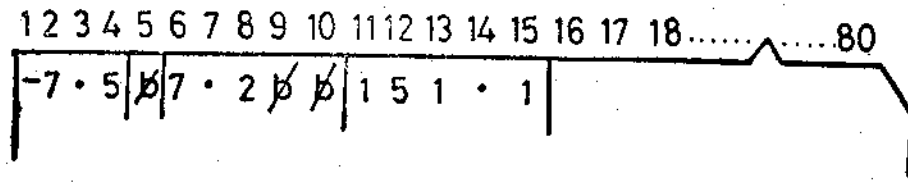


Fig. 6 : *Data card containing values of A, B, C.*

Solution

READ (5, 10) A, B, C

10 FORMAT (F 4.1, 1X, F 3.1, 2X, F 5.1)

Ex.2 : Two real numbers X and Y are to be printed out in the exponent notation and an integer J of 4 digit length is also to be printedout with 5 column gap in between each variable. Write corresponding Output statement.

Solution

WRITE (9, 24) X, Y, J

24 FORMAT (1Hb, E15.8, 5X, E15.8, 5X, I4)

Notes : 1. While writing E - FORMAT specification for output statement we must be careful to see that $w \geq d + 6$ i.e., the total width of the field must exceed the number of digits required to the right of the decimal point by atleast six. The six punch positions are required for i) Two digits for exponent ii) One position for exponent symbol E iii) One position for the sign of the exponent iv) One position for the decimal point v) One position for the sign of the mantissa.

2. To avoid accidental loss of digits in output (if we do not know what will be the value of the variable to be output) it is good practise to allow 15 places for all E specifications, that is, write all E specifications in the form E15.X where X is the number of decimal places required to the right of the decimal part.

16.8 FORMAT SPECIFICATIONS FOR ALPHA NUMERALS

The formatted input - output statement must be followed by FORMAT statement which gives information to the compliers i) in what form are the values punched or printed? ii) How many card columns are allocated to each input value? iii) How may printing positions are allocated to each output value? iv) Is each value fixed or floating point? We came to know about 3 format specifications i) Integer field specification for integer vlues (i.e., I - specification) ii) Real field specification F, for fractional numbers iii) Real field specification E, for real numbers in exponent form and also about column skip specification. All the above three format specifications are for numerical quantities. When the input or output is to contain words and sentences along with the numerical values of variables (i.e., heading, messages, comments etc.) we use Hollerith field specification (H – specification) Similarly there are few more format specifications available in FORTRAN - IV. Here we discuss i) H - field specification ii) use of slash iii) A - specification iv) multiple use of single specification.

16.8.1 Hollerith Field Specification or H - Field Specification

Some times we want to output descriptive information to provide headings, comments and idenfication of printed results. To read and write alphameric information (which contains alphabets and numerals) for such purposes we use H - field specification. The general form of Hollerith field specification is

w H XXXX .....

where '*w*' is the number expressing the hollerith field width and XXX ... represents any *w* alphameric characters, which includes any blank columns in between the 'X's. 'H' is the symbol indicating that the field is a Hollerith field. The specification *w* H followed by *w* alphameric characters may be used in FORMAT statements to provide descriptive information.

For example if we want to print the message 'TWO ␣ REAL ␣ ROOTS', we can achieve this by writing

```
      WRITE (9, 35)
   35 FORMAT (1H␣, 14HTWO␣ REAL␣ROOTS)
```

For example if we want to print the values of variables I, J and K whose Format specifications are I5, I5 and I8 respectively with headings as follows

I = XXXXX ␣ ␣ ␣ ␣ ␣ J = XXXXX ␣ ␣ ␣ ␣ ␣ K = XXXXXXXX

then the corresponding output statement is

```
      WRITE (9, 45) I, J, K
   45 FORMAT (1X, 2HI =, I5, 6X, 2HJ =, I5, 6X, 2HK =, I8)
```

Consider the following output statement

```
WRITE (9, 10) N, ROOT
10  FORMAT (1Hb, 6HbAFTER, I2, 23Hb ITE –
                RATIONSbTHEbROOTbIS, F3.1)
```

if the value of N is 10 and the value of root is 1.8, then the output corresponding to the above output statement will be as follows.

AFTERb10bITERATIONSbTHEbROOTbISb 1.8

16.8.2 Use of Slash

The symbol '/' (slash) is used in the FORMAT statement to instruct the computer to skipover a card during input or output. If the output is in a printed form, a slash will instruct the computer to skipover one line.

To have clear idea consider the following examples.

1)
```
READ (5, 16) A, B, C, D
16 FORMAT (F 8.3, F 7.5/F 5.3, F 7.2)
```

By these statements, the computer will read the values of A and B from the first card according to the format specifications F 8.3 and F 7.5, then the computer senses the slash. Now it does not continue to scan the same card but goes over to the next card and read the values of C and D from second card according to the format specifications F 5.3 and F 7.2.

2)
```
WRITE (9, 25) X, Y, I
25  FORMAT (1Hb, F 8.3/1Hb, F7.5/1Hb, I5)
```

The above statement causes the computer to print the value of X on one line with format specification F 8.3, Y on the second line with the format specification F 7.5 and I on the third line with the format specification. I5,

3)
```
WRITE (9, 45) X, Y, L, M
45  FORMAT (1Hb, F7.3, E10.3//1Hb , 2I5)
```

This statement causes the computer to print out the values of X and Y on first line with format specification F 7.3 and E 10.3 and then it skip the second line and print the values of L and M on the third line with the format specification I5 (∵ format specifications of L and M are same).

16.8.3 Group Repeat Specification (Multiple Use of Single Specification)

If input/output lists are to be obtained in more than one record (one card or one printed line) then the multi record format or group repeat specification is to be specified. To understand this, first let us know how the complier interprets FORMAT statements when the object program encounters an Input/Output statement. It scans the corresponding FORMAT specification from left to right and

simultaneously scans the variables in the list. Each element in the list is read or printed according to the given format, when it reaches the final parenthesis, it checks if all the elements in the list are exhausted. If not, it scans the FORMAT specifications in reverse order i.e., from right to left and stops at the first left parenthesis. Nothing is done during this scan. It again starts scaning from left to right from this left parenthesis and reads or prints the rest of the list elements in the next card or line. This scaning will go on until all the elements in the list of the input/output statement are exhausted.

For example consider the following statement

```
WRITE (9, 16) I, J, K, L
16  FORMAT (1Hb, I5)
```

By this statement the computer will print the value of I on the first line with FORMAT specification I5 and the value of J on second line with the same FORMAT specification I5 and the values of K and L on the next two lines with the same FORMAT specification.

Ex. 1 : Consider the following statement

```
WRITE (9, 27) A, B, C, X, Y, Z, P, Q, R
27  FORMAT (1Hb, F5.2, (F8.2, F8.3))
```

In this example, the computer will print the values of A, B and C on the first line with the specifications F5.2, F8.2, and F8.3 respectively. The values of X, Y on the second line with the FORMAT specifications F8.2 and F8.3, the values of Z and P on the third line, Q and R on the fourth line with the same FORMAT specification F8.2 and F8.3.

Ex.2 : Consider the example

```
WRITE (9, 45) A, B, C, I, J, K, L, M, N, KOU, INT, MON
45  FORMAT (1Hb, 3F8.4/(3I4/2I3))
```

This statement causes the computer to print the three values of A, B and C on the first line with FORMAT specification 3F8.4 and the values of I, J and K on the second line with FORMAT specification 3I4 and the values of L and M on third line with FORMAT specifications 2I3 and alternately 3 and 2 integer variable values on all succeeding lines with the FORMAT specifications 3I4 and 2I3 until all the variables are exhausted.

16.8.4 A – FORMAT specification

The Hollerith Format specification deals with character strings. But the characters read using the Hollerith Format are not stored in the computer's memory and therefore cannot be manipualted. Actually in printing headings etc. there is no need to read in these into the memory and thus H – Format is more appropriate. For some problems like arranging a set of names in alphabetical order, to find how may times a particular character is appearing in a set of names etc., it is necessary to read in strings of characters into the memory of the computer. It is possible to store strings of characters in the memory of the computer and can be manipulated by using A – Format specification.

General form of the A – specification is A*w*

where *w* is the width of the field. Corresponding to each 'A*w*' specification in the FORMAT statement, there should be one varaible in the I/O list. The specification 'A*w*' will cause '*w*' characters to be I/O as the value of the variable int he I/O list.

For example consider the following input statement

```
      READ (5, 100) X
100   FORMAT (A4)
      Data Card HEMA
```

The above statements cause the character string HEMA to be read into the variable name X. The number of characters which may be read into a variable name depends on whether it is an integer or a real and on the computer being used. For example TDC-316 (Manufactured by ECIL, Hyderabad) allocates 4 characters per real variable name and 2 characters per integer variable name. Therefore if the number of characters in a string exceeds 4 it should be read into more than one real variable name. In such we use subscripted variables (To be discussed in next unit).

Ex. 1 : Consider

```
      READ (5, 15) A, B, C, D
15    FORMAT (4A4)
      DATA : VENKATANARAYANA
```

These statements will store 'VENK' in A, 'ATAN' in B, 'ARAY in C and 'ANA ' in D.

Ex.2 : Consider

```
      READ (5, 25) (K(I), I = 1, 7)
25    FORMAT
      DATA : OPENUNIVERSITY
```

By the above input statement, the computer will store 'OP' in K(1), 'EN' in K(2), 'UN' in K(3), 'IV' in K(4), 'ER' in K(5), 'SI" in K(6). 'TY' in K(7).

Now we summarise the important points to be noted while writing FORMAT statements.

1. The order in which data is punched on a card and the order in which the variables in the input list are specified must be the same (for the output statement also).

2. Input data (i.e., in I, F, E format) should be right adjusted with in their respective fields.

3. The type of variable name and the corresponding Format declaration should tally, i.e., for integer variables we have to use I Format and for reals we have to use E and F format specifications.

4. The elements in the lisst of READ and WRITE statements should be variable names but not expressions.

5. In output statements one extra column should be left for the sign of the number. If the range of values of a number to be printed is not known, E format should be used. A safe E format is E15.8 in such cases.

6. In H – field specifications the number of characters (including blanks) should be accurately counted.

7. In WRITE statements the first specification is carrige control character (ie., 1H *b*, 1H1 or 1H0) and the appropriate character should be used.

8. The number of characters to be printed per line (one record) should not exceeds the capacity of the printer. Most printers have 132 characters per line.

16.8.5 Some Simple Computer Programs

We know how to read in and take out data from a computer. Therefore we can solve some simple problems by using computer. In this section we will develope FORTRAN programes to solve some simple problems, which can be run on a computer. Before that, let us know how to prepare a FORTRAN PROGRAM. First we have to analyze the problem by means of flow chart or decision table and then we translate each instruction using FORTRAN language.

1. The capital letter 'C' punched in column 1 indicates the statement is a comment statement and is to be ignored by the compiler during compilation. When the listing of the source program is requested, the commet statement is also to be printed.

2. Statement numbers of FORTRAN statements may be punched in columns 1-5.

3. All FORTRAN Statements must be punched in columns 7-72 only (including both 7 and 72). One statement is punched on one card. But if a statement is too long to fit in a single card, it may be continued on one or more additional cards. If this is the case, the continuation cards should contain a FORTRAN character other than the digit '0' or the character blank in column 6. By convention, numbers in sequence are used. The maximum number of continuation cards varies from one computer to other.

4. Any number of blanks in a FORTRAN statement (except in the case of data cards) will be ignored by the compiler. These blanks may be used to improve the readability of the program.

5. Columns 73-80 of the punch cards may be used for the purpose of identification by the programmer. In order to run the program on a computer, we have to instruct the computer. To instruct the computer a set of system control cards is added to the source program. These cards act in a supervisory capacity, directing the computer to take appropriate action with respect to a specific program. The arrangement and their nature vary from computer to computer, but functionally all are similar.

The complete deck of cards, i.e., control cards, program cards and data cards, constitute a job. Which can be submitted to a computer for processing.

Now let us consider few examples.

Ex. 1 : Write a computer program to solve the linear equation

$$ax + b = c\ (a \neq 0)$$

Solution

To find the values of x, first we have to read in the values a, b, c into computer's memory by means of input statement and stored in locations identified by the variable names A, B and C. Then the value of X is calculated using the formula $x = (c - b)/a$ (execution part) and then the value of x is output by means of output statement.

The flow chart is as follows

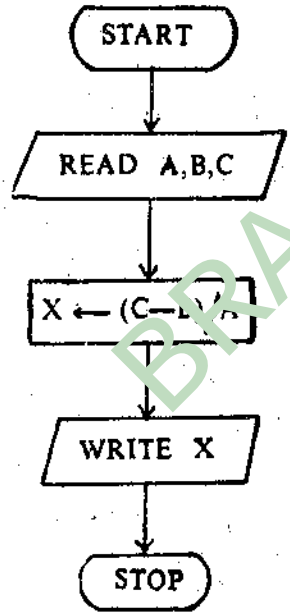


Fig. 7 : *Flow chart to solve linear equation* $ax + b = c$,

First we write the FORTRAN program with FORMAT free input and output statements.

Program 1 :

```
C   PROGRAME TO SOLVE LINEAR EQUATION
    READ, A, B, C
    X = (C – B)/A
    PRINT, X
    STOP
    END
```

Now we have to punch the above set of instructions on punch cards. The other card needed is a data card. In this problem the data consists of three numbers, namely the values of A, B and C. The values of A, B, C are punched starting from column 1. To this set of cards, if we add the control cards and give to the computer, we get the solution of the problem.

Note : If another linear equation is to be solved the program reamains the same. Only another data card with new data is to be used in the place of the present data card. Therefore if a program is written for a problem a whole class of similar problems where only the data is different can also be solved with the help of it.

Now write the program with FORMAT statement. Let the values of A, B and C are 0.5, 1.25 and 10.0 respectively. Therefore the FORMAT specifications of A, B and C are F3.1, F4.2 and F4.0 respectively.

Program 2 :

```
C       PROGRAM TO SOLVE LINEAR EQUATION
        READ (5, 15) A, B, C
15      FORMAT (F3.1, 2X, F4.2, 2X, F4.0)
        X = (C – B)/A
        WRITE (9, 20) X
20      FORMAT (1H , E15.8)
        STOP
        END
```


(Since we don't know the value of X, we had taken the E format specification E15.8 for output). The data card in this case will be as follows.

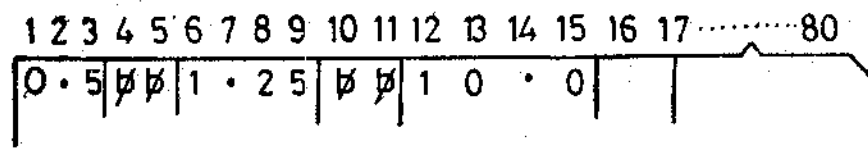


Fig. 8 : *Data card containing values of A, B and C.*

Ex. 2 : Write a computer program to read in the values of A, X and S with FORMAT specification F10.4 to calculate Y and Z using the formulae

$$Y = \sqrt{X^2 - A^2},\ Z = \frac{X \cdot S}{2} - \frac{A^2}{2} \log |X + S|$$

and to print the values of Y and Z along with A, X and S.

Solution

Here in this problem the procedure is clear. First we have to read in the values of A, X and S, then we have to calculate Y and Z and then finally we ask the computer to print out the values of A, X, S, Y and Z. For simple problems, wher there is no logic, and if the procedure to solve the problem is simple, no need to draw flow chart. An experienced programmer won't go for flow charts for simple problems and he can wirte the program deirectly.

So the program is as given below.

Program 3 :

```
C       PROGRAM TO CALCULATE Y AND Z
        READ (5, 25) A, X, S
   25   FORMAT (3F10.4)
        Y = SQRT (X * X - A * A)
        Z = X * S/2 – A *  A/2 * ALOG (ABS (X + S))
        WRITE (9,35) A, X, S, Y, Z
   34   FORMAT (1Hb, 3F10.4, 2X, 2HY =, E15.8, 2X, 2HZ = E15.8)
        STOP
        END
```


Ex. 3 : Write a program, using suitable Format specification, to evaluate E and V using the following formulae and output the results.

$$E = \frac{1}{2} CQ^2, \text{ for } C = 0.00001 \text{ and } Q = 0.0025$$

$$V = RT/P, \text{ or } R = 15.7, T = 12.87, \text{ and } P = 156.05$$

Program 4 :

```
C       TO CALCULATE AND V
        READ (5, 8) C, Q, R, T, P
    8   FORMAT (2X, F7.5, 2X, F6.4, 2X, F4.1, 2X, F5.2, 2X, F6.2)
        E = 1./2. * C * Q * * 2
        V = R * T/P
        WRITE (9, 12) E, V
```

```
12    FORMAT (1Hb, E15.8/1Hb, E15.8)
      STOP
      END
```

Here we used '/' in the FORMAT statement of Output statement. Therefore on the first line, it print the value of E with E15.8 specification and on the second line the value of V with specification E15.8.

Ex. 4 : Given the two points A (4, 5, 7), B (32., 5.4, 7.6) write a program to compute and print i) distance between the two points A and B ii) the direction cosines of the vector AB.

Solution

Let us denote the co-ordinates of A by A_1, A_2, A_3 and the co-ordinates of B by B_1, B_2, B_3. Let us use the variable name 'DIST' for distance and DCX, DCY and DCZ for direction cosines. We know the formulae

$$DIST = \sqrt{(B_1 - A_1)^2 + (B_2 - A_2)^2 + (B_3 - A_3)^2}$$

$$DCX = (B_1 - A_1)/DIST$$

$$DCY = (B_2 - A_2)/DIST$$

$$DCZ = (B_3 - A_3)/DIST$$

The following is the required flow chart.

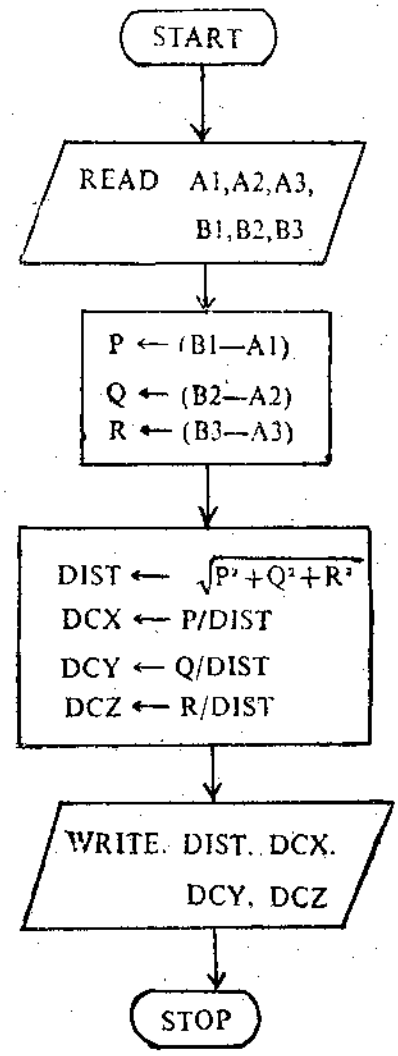


Fig. 9: *Flow chart to find distance and direction cosines.*

Program 5 :

```
C       TO CALCULATE DISTANCE AND DCS
        READ (5, 3) A1, A2, A3, B1, B2, B3
    3   FORMAT (3F2.0, 2X, 3F3.1)
          P  =  B1 – A1
          Q  =  B2 – A2
          R  =  B3 – A3
        DIST  =  SQRT (P * P + Q * Q + R * R)
        DCX  =  P/DIST
        DCY  =  Q/DIST
        DCZ  =  R/DIST
        WRITE (9, 7) DIST, DCX, DCY, DCZ.
    7   FORMAT (1Hb, 5HDIST =, E15.8/1Hb, 4HDCX =
    1   E15.8, 2X, 4HDCY =, E15.8, 2X, 4HDCZ =, E15.8)
        STOP
        END
```

Ex. 5 : Write a program to find the sum of the series

$$S = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} + \frac{x^{10}}{10!} \text{ for } x = 3.4$$

Solution

While solving any problem using computer, one's intention must be to reduce the number of arithmetic operations as far as possible. Here in this problem, the use of intermediate variable reduces the number of multiplication operations performed by the computer. Now consider the program.

Program 6 :

```
C       TO FIND THE SUM
        READ (5, 15) X
   15   FORMAT (F3.1)
        X2  =  X * X
        X4  =  X2 * X2/24.
        X6  =  X4 * X2/30.
        X8  =  X6 * X2/56.
        X10 =  X8 *X2/90.
```

```
      SUM = 1. – X2/2. + X4 – X6 + X8 – X10
      WRITE (9, 20) X, SUM
20    FARMAT (1Hb, F3.1, 5X, E15.8)
      STOP
      END
```

16.9 SUMMARY

The input statements are useful for feeding the computer with the information to be processed and the output statements are useful for giving us the results after processing. While feeding, the computer needs the information about the data like the form and the size in magnitude of each of the variables to be processed and the processed information has to be printed in a required form. For this purpose the format specifications are needed. An input/output statement should always be followed by its format statement. A format statement is a non-executable statement and will give information to the compiler about the form, the space allocated to each value of the variables to be fed/printed by the compter. There are three types of format specifications for numerical quantities namely I – format, E – format and F – format specifications. When the input/output statements contains words and sentences along with numerical values of the variables we use Hollerith field specifications and A – specifications.

16.10 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following in detail.*

1. Explain the importance of Input - Output statements.
2. a) Exaplain statement number.

   b) State whether the following numbers are the allowable statement numbers or not by giving resons

      1) 150 2) 10. 3) K 15 4) 200 5) 1235784

3. a) Define i) Record ii) Field iii) Field width

   b) Explain FORMAT statement.

4. Explain i) F – FORMAT specification ii) E – FORMAT specification by giving two examples each.
5. Explain Hollerith field specification with two examples.

6. If the values of A, B and C are 15.85, 101.23 and 1050.25, write the output statement with corresponding FORMAT statement, to print the values of A, B and C in the following way,

   A = 15.85

   B = 101.23

   C = 1050.25

7. Write a computer program to read in the values of P, Q and R as 12.0, 7.25 and 5.75 respectively, to calculate

   $S = P^2 + Q^2 + R^2$ and to print the value of S.

8. Read a card containing the values of A, B and C, in this order each punched with F7.0 specification. Write a program to compute S and Area and to print A, B and C on one line S and Area on the next line where S = (A + B + C)/2

   $$\text{Area} = \sqrt{S\,(S-A)\,(S-B)\,(S-C)}$$

9. Write a program to calculate and print F, given by the formula

   $$F = \frac{\cos\left(e^{-x^2}\right) - \sin\,(\theta + \pi/4)}{\tan\,\phi + z^{2.5}}$$

   where X = 2.2 θ = .5 radians, φ = .2 and Z = .0034.

II. *Briefly answer the following*

1. How many FORMAT specifications are there for numerical quantities.

2. If the values of the variable A, B and C are –15.752, 153.175 and 10.72E –15 write the corresponding FORMAT specifications.

3. Write the READ statement to read in the values of X, Y, Z as 15.75, 40.17413 and

   0.575E –01.

4. Write the WRITE statement to print out the values of A, B, C, K and L as 10.75, –75.752, 100.02,452 and –1750 respectively.

5. Write the FORMAT statement to print the message "THE b ROOTS b ARE b COMPLEX"

6. If the values of X, Y, Z, P, Q, R, are 10.18, 1.75, 3.75, 150.0, 100.75 and 108.5 and if they are to be printed 2 values per line, then write the corresponding output statement.

7. Write the corresponding input statement to store the following "BALAPARAMESHWARA RAO" in computer's memory using A – FORMAT specification.

8. Write a program to calculate and to print Q and R, given by

Q = H (T – U) and R = V (T – U) where

H = 352000, T = 815, U = 850, and V = 0.05×10^{-15}

Answers

I. 2 (b) 1) Yes 2) No – Due to presence of decimal point 3) No – Not a number 4) Yes 5) No – More than five digits.

6. WRITE (9, 15) A, B, C

15 FORMAT (1Hb, 2HA =, F5.2/1Hb, 2HB =, F6.2/1Hb, 2HZ =, F7.2)

II 2. F7.3, F7.3, E9.2

3. READ (5, 15) X, Y, Z

15 FORMAT (F5.2, F8.5, E9.3)

4. WRITE (9, 17) A, B, C, K, L

II 6. WRITE (9, 20) X, Y, Z, P, Q, R

20 FORMAT (1Hb, F5.2, F4.2/1Hb, F4.2, F4.2/1Hb, F5.1, F6.2)

7. READ (5, 13) A, B, C, D, E

13 FORMAT (5 A4)

DATA : BALAPARAMESHAWARA RAO

UNIT–17 : CONTROL STATEMENTS

Contents

17.1 AIMS AND OBJECTIVES

After going through this unit you will be able to write : (i) a fortran program using the appropriate control statements, (ii) identify the errors involved in a programme, (iii) use the DO statements in writing programmes involving looping.

17.2 INTRODUCTION

In this unit we shall deal with the need and use of "CONTROL STATEMENTS". We know that the computer executes the statements in the order of their appearance in the program. That is, when all the operations specified in a particular statement are executed, the statement appearing on the next line in the program is taken up for execution. This is known as the normal flow of control.

Some times we may have to alter this order of execution. For example to repeat a particular series of steps, or to repeat the entire program a given number of times with different sets of data, or to omit some statements in the program due to constraints in the algorithm or depending upon the result of a logical decision, to go to an intermediate point in the program so as to begin a new series of calculations, or to go to the end of the program in order to terminate its execution. To achieve this, we need the transfer of control. So we need, logical decision making and/or transfer of control facilities. FORTRAN provides both these facilities.

In this unit, we shall discuss the facilities for decision making and the transfer statements available in FORTRAN-IV and their usage in programming. Control statements can be classified into two categories.

i) Unconditional Control statements,

ii) Conditional Control statements.

17.3 UNCONDITIONAL CONTROL STATEMENTS

When the computer executes an unconditional control statement the control is transferred straight away to the executable statement referred by the control statement by statement number of that executable statement. This statement is also known a unconditional GO TO statement.

The unconditional GO TO statement is written as

GO TO *n*

where '*n*' is the statement number of an executable statement. Blanks between GO and TO are optional. This statement is used to indicate that the next statement is not the one following it, but the statement that is labelled *n*. This statement may be located anywhere in the program, that is, either before or after the GO TO statement itself.

Some valid examples of GO TO statements

1. GO TO 5
2. GO TO 150
3. GO TO 40

Some invalid examples of GO TO statements

1. GO TO 0
2. GO TO I+5
3. GO TO, 35
4. GO TO 154362

17.4 CONDITIONAL CONTROL STATEMENTS

The conditional control statements available in FORTRAN-IV are

i. Arithmetic IF statement

ii. Logical IF statement

iii. Computed GO TO statement

17.4.1 Arithmetic IF statement

A widely used conditional control statement is "Arithmetic IF" statement. Here a decision is made by the computer and then the control is transferred according to the decision. The general form of Arithmetic statement is

IF (*e*) n_1, n_2, n_3 where *e* is any arithmetic expression and n_1, n_2, n_3 are statement numbers of executable statements.

The meaning of this statement is as follows. If the value of the arithmetic expression '*e*' is less than zero, the control is transferred to the statement number n_1. If the value of '*e*' is equal to zero, the control is transferred to the statement number n_2. If the value of '*e* is greater than zero, then the control is transferred to the statement number n_3.

For example, consider the following statement

IF (2.*A–B) 5, 10, 16

Here, computer first evaluate the arithmetic expression '*e*' i.e., (2.*A–B), if

(2.*A–B) < 0 Control goes to execute the statement numbered 5 in the programme

(2.*A–B) = 0 Control goes to execute the statement numbered 10 in the programme

(2.*A–B) > 0 Control goes to execute the statement numbered 16 in the programme

Therefore the Arithmetic IF statement acts as a three way switch depending upon the value of an arithmetic expression. To get clear idea see fig. 1 which shows the effect of execution of Arithmetic IF statement.

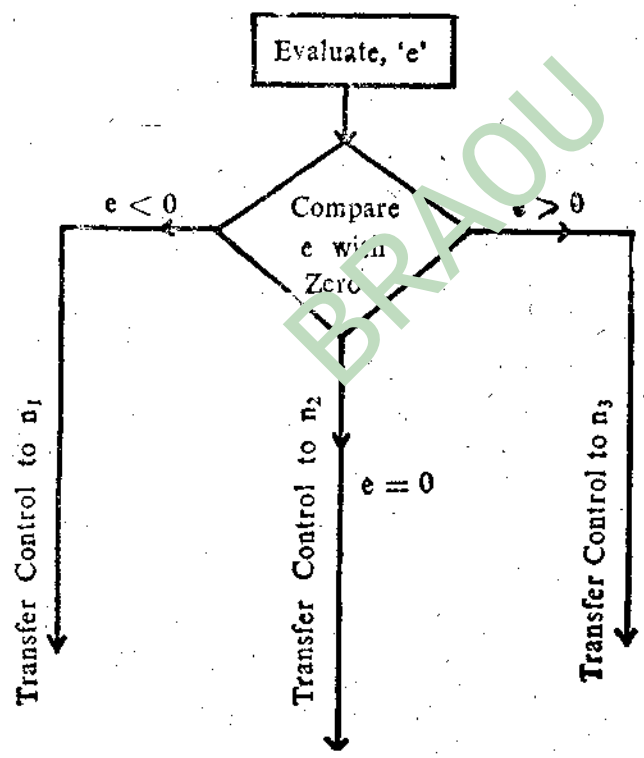


Fig. 1 : Execution of Arithmetic IF Statement

Note : All the three n_1, n_2, n_3 in the arithmetic statement need not always be diffeent. Any two of them may be equal. For example, consider If (*e*) 5, 5, 10 Here $n_1 = n_2 = 5$, $n_3 = 10$. i.e., negative branch and zero branch are same.

If all three branch paths are equal, for example IF (*e*) 25, 25, 25 an unconditional branch is implied. Therefore the above statement will have the same effect as the unconditional GO TO statement GO TO 25. Now let us consider few examples.

Ex. 1 : IF P = 3.0, q = 1.5, what statement the computer would execute after following IF statement, IF (P–Q) 11, 111, 1111.

Solution

First, computer will evaluate the arithmetic expression

$$P - Q = 3.0 - 1.5 = 1.5$$

$$\therefore e = P - Q > 0.$$

$\therefore$ the computer will execute the statement numbered 1111, since $e > 0$.

Ex. 2 : Write the arithmetic If statement for the c omputer to execute statement number 10, If $B^2 - 4AC$ has a value zero, and the statement number 15, if the value is other than zero.

Soltuion

IF (B * B – 4. * A * C) 15, 10, 15

Ex. 3 : Write a computer program to find the largest of given three numbers A, B and C using Arithmetic IF.

Solution

We had drawn flow chart for this problem in Unit – 13. It is reproduced here for reference. By seeing it we canwrite the programe easily.

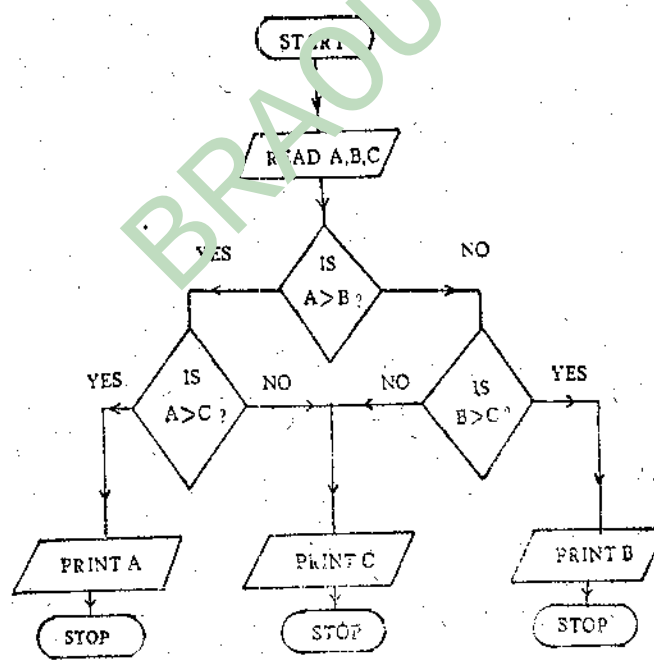


Fig. 2 : Flow chart to pick largest of A, B and C

Here first we are comparing A with B, so take arithmetic expression '*e*' as A–B, similarly when comparing B with C, take '*e*' as B–C and finally when comparing A with C, take A–C as '*e*'. Following the flow chart carefully, we can write the program as follows (we take the FORMAT specification of A, B and C as F5.3).

Program 1 :

```
C          PICKING LARGEST OF A, B, C USING
C          ARITHMETIC IF
```

```
      READ (5, 10) A, B, C
10    FORMAT (3F5.3)
      IF (A–B) 6, 6, 12
 6    IF (B–C) 7, 7, 15
 7    WRITE (9. 25) C
      STOP
12    IF (A–C) 13, 13, 20
13    WRITE (9, 25) C
      STOP
20    WRITE (9, 25) A
      STOP
15    WRITE (9, 25) B
25    FORMAT (1Hϕ, F5.3)
      STOP
      END
```

Ex. 4 : Write a program to find the roots of the quadratic equation $ax^2 + bx + c = 0$, taking the format specification of A, B and C as F6.2, F6.3 and F7.3 respectively.

Solution

We know the procedure to find the roots of given quadratic equation. First, we draw the flow chart and then we translate it into program.

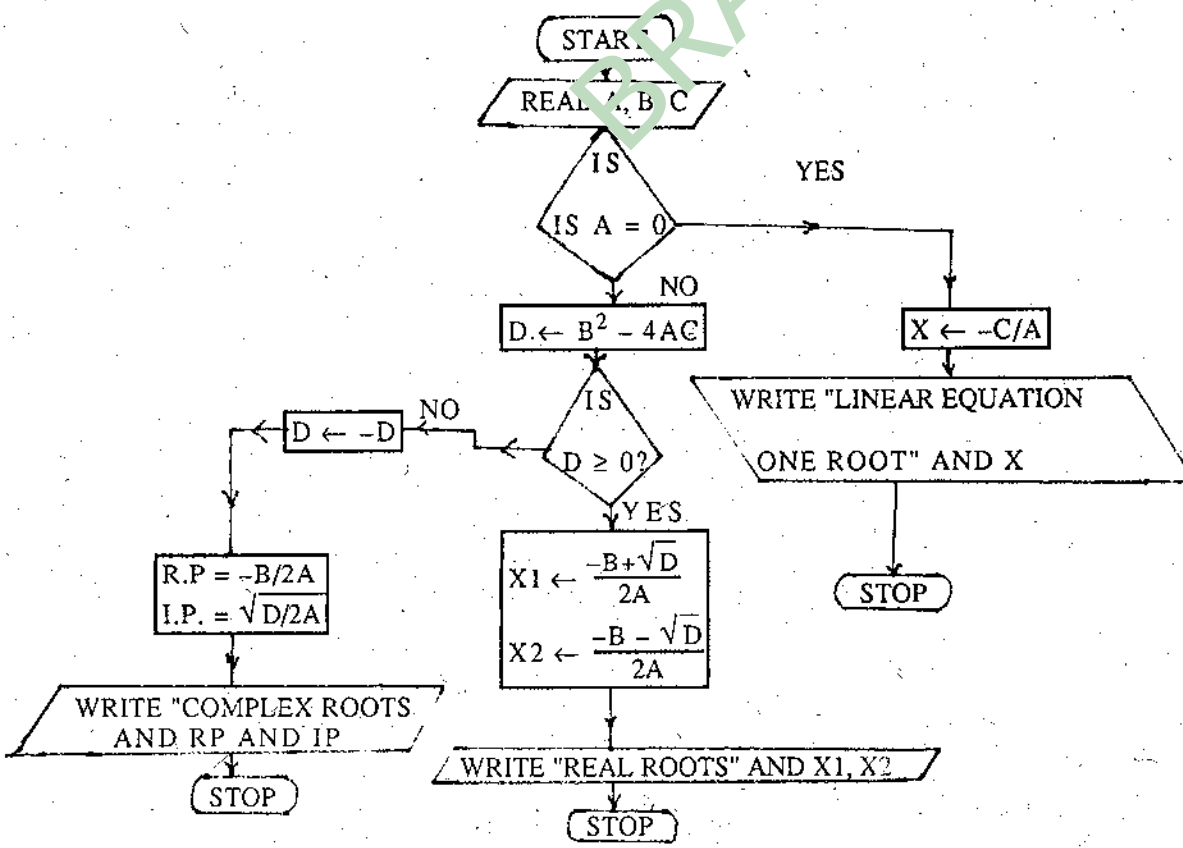


Fig. 7: *Flow chart to find the roots of a quadratic equation*

Program 2 :

```
C   ROOTS OF QUADRATIC EQUATION
    READ (5, 7) A, B, C
7   FORMAT (2X, F6.2, 2X, F6.3, 2X, F7.3)
    IF (A) 8, 18, 8
8   D = B*B-4.*A.C
    IF (D) 9, 13, 13
9   D = -D
    A2 = 2.*A
    X1 = -B/A2
    X2 = SQRT (D)/A2
    WRITE (9, 10) X1, X2
10  FORMAT (1X, 13HREAL PART X1 = , E15.8/1X,
    118HIMAGINARY PART X2 =, E15.8)
    STOP
13  ROOTD = SQRT (D)
    X1 = (-B+ROOTD)/A2
    X2 = (-B-ROOTD)/A2
    WRITE (9, 15)
15  FORMAT (1H b, 3HX1 =, E15.8, 8X, 4HX2 =, E15.8)
    WRITE (9, 16) X1, X2
16  FORMAT (1H b, 3HX1 =, E15.8, 5X, 4HX2 =, E15.8)
    STOP
18  X = C/B
    WRITE (9, 19)
19  FORMAT (1X, 28HLINEAR EQUATION ONE ROOT ONLY)
    WRITE (9, 20) X
20  FORMAT (1H b, 3HX =, E15.8)
    STOP
    END
```

(notice the use of H – specification and arithmetic If)

FORTRAN–IV allow the use of another type of IF statement which is called LOGICAL IF statement. Besides integer and real constants, variables and expressions, FORTRAN – IV permits the use of Logical constants, variables and expressions.

17.5 LOGICAL STATEMENTS

17.5.1 Logical constants

There are only two logical constants. They are i) .TRUE, ii) FALSE.

The full stops preceding and following these symbols are essential. A logical quantity can have only one of the two logical values that is, it can be either .TRUE or .FALSE.

17.5.2 Logical Variables

Logical variables represent logical values that may change during the course of computation. Logical variables are assgined names in the same manner as integer or real variables but the name of each logical variables must always be declared logical by means of Type declaration statement.

For example, if KOUN, GAMA, YES, BIG are selected as name of logical variables in a particular program, then these must be declared by the following type declaration statement before the first use of these variables in the program.

LOGICAL KOUN, GAMA, YES, BIG

17.5.3 Relational Operators and Logical Operators

The available relational operators and their FORTRAN equivalents are given below.

Sl. No.	Mathematical symbol	Meaning	Fortran equivalent
1.	<	less than	.LT.
2.	=	equal to	.EQ.
3.	>	greater than	.GT.
4.	≤	less than or equal to	.LE.
5.	≠	not equal to	.NE.
6.	≥	greater than or equal to	.GE.

Table 1 : Relational Operators

The logical operators available are (i) .AND. (ii) .OR. (iii) .NOT.

For example consider X.EQ.Y has the value .TRU. if $x = y$ and the value .FALSE. otherwise.

Similarly, P.AND.Q has the value .TRUE. if both P and Q have the value .TRUE., it has the value .FALSE. if either P or Q is .FALSE. or if both P and Q are .FALSE., where P and Q are logical quantities.

Consider .NOT.X This has the value .TRUE. if X is .FALSE. and the value .FALSE. if X is .TRUE.

17.5.4 Logical Expressions

A logical expression is any valid combination of

i) Logical constants, variables or functions separated by logical opeators.

or

ii) Arithmetic constants, variables, functions separated by relational operators.

or

iii) a combination of (i) and (ii)

while forming logical expression, the following points should be remembered.

a) The operator .NOT. must be followed by, but not preceded by an element (i.e., like .NOT.B etc.)

b) Two logical opeators must not be adjacent unless the first is .AND. or .OR. and the second is .NOT.

c) The operators .AND. and .OR. must be preceded by an element and followed by an element or .NOT.

Some valid logical expression are

i) .FALSE.

ii) (A .GT. 5.2) .AND. (B .LE. 15.0)

iii) I .GT. KOUN*J

iv) (WET .LE. 50) .OR. (HT .GE. 5.7)

17.5.5 Logical IF statement

The general form of a logical IF statement is

IF (e) n_1
n_2

where e is a logical expression, n_1 may be any valid executable FORTRAN statement except another logical If or a DO statement.

n_2 may be any valid executable FORTRAN statement and is the statement immediately next to the logical IF statement.

Now let us consider the meaning of this logical IF statement. If the expression 'e is true then 'n_1' is executed and then control passes to 'n_2'. If the expression 'e' is false then 'n_1' is not executed and 'n_2' is executed. A flow chart describing the meaning of Logical IF statement is given below.

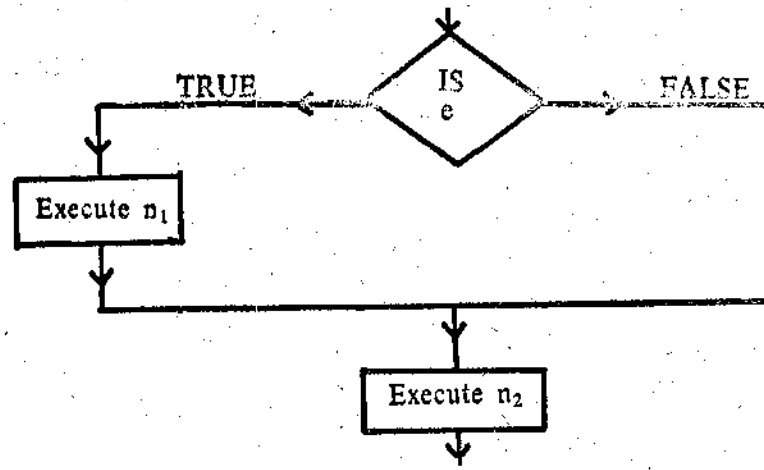


Fig. 4 : Flow chart representing Logical IF

When the expression 'e in the logical IF statement is true and if we want the computer to execute only 'n_1' and not 'n_2'. In such cases we may use a GO TO statement as follows.

IF (*e*) GO TO *n*
n_2
⋮
$n\ n_1$

Now let us consider few examples.

Ex. 1 : Write logical IF statements to Compute F defined as

$$F = x^2 + 5x + 7, \text{ if } x \leq 10.5$$
$$= x^2 + 15, \quad \text{if } x > 10.5$$

Solution

The logical statements that will do this are

IF (X .LE. 10.5) F = X*X + 5.*X+7.
IF (X .GT. 10.5) F = X*X+15.

Ex. 2 : Write a program to compute and print the values of J for I = 10 to 100 in steps of 10 where

$$J = 100 - 5I^2, \text{ for } I \leq 40$$
$$= 50I - 10, \text{ for } I > 40$$

Solution

Here we use logical IF statement. Consider the flow chart.

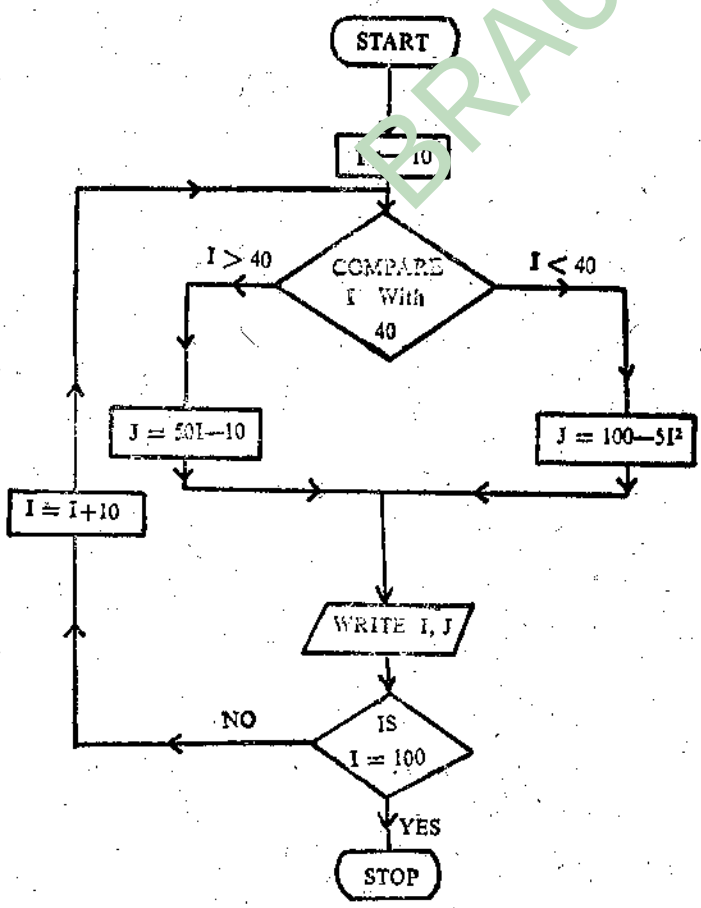


Fig. 5 : Flow chart to calculate J

Program 3 :

```
C                    TO CALCULATE J
                     I = 10
  15                 IF (I .LE. 40) J = 100–5*I*I
                     IF (I .GT. 40) J = 50*I–10
                     WRITE (9, 50) I, J
  50                 FORMAT (1Hb, I3,5x,E15.8)
                     IF (I .EQ. 100) STOP
                     I = I+10
                     GO TO 15
                     END
```

17.5.6 Computed GO TO statement

This is another conditional control statement. This statement transfers control unconditionally to a specified statement. If we want to branch to one of set of a statements based on the value of an integer variable we use this computed GO TO statement. The general form of this statement is

$$\text{GO TO } (n_1, n_2, \ldots, n_k), i$$

where $n_1, n_2, \ldots, n_k$ are statement numbers and 'i' is a non-subscripted integer variable name. The parentheses enclosing the statement numbers, the commas separating the statement numbers, and the comma following the right parenthesis are all required punctuations.

The meaning of this statement is as follows. When executed, this causes transfer of control to the first, second, third, ... statements listed within the parenthesis depending on whether the value of i is 1, 2, 3, ... etc. The value of 'i' at any time must never exceed the total number of statements with in the parenthesis. A value of 'i' less than or equal to 0 is invalid. This statement may be punched any where between columns 7 and 72 of a card.

For example consider GO TO (10, 13, 30, 20, 5), I

Here if I = 1, this statement transfers the control to statement 10, if I = 2 to statement 13, ... etc. and finally if I = 5, to statement 5.

Now let us consider few examples.

Ex. 1 : If it is required to transfer the control to statement 20, if the value of the variable M is 2, 3 or 4 or to transfer the control to statement 40, if the value of the variable M is 1, or to transfer the control to 60, if the value of M is 5. Write the corresponding GO TO statement.

Solution

```
GO TO (40, 20, 20, 20, 60), M
```

Ex. 2 : Write a program to find the sum upto N terms of the series

$$S = 3x + 5x^2 + 7x^3 + \ldots$$

Extended this to find the sum upto 20, 30, 40 terms of the series using computed GO TO statement.

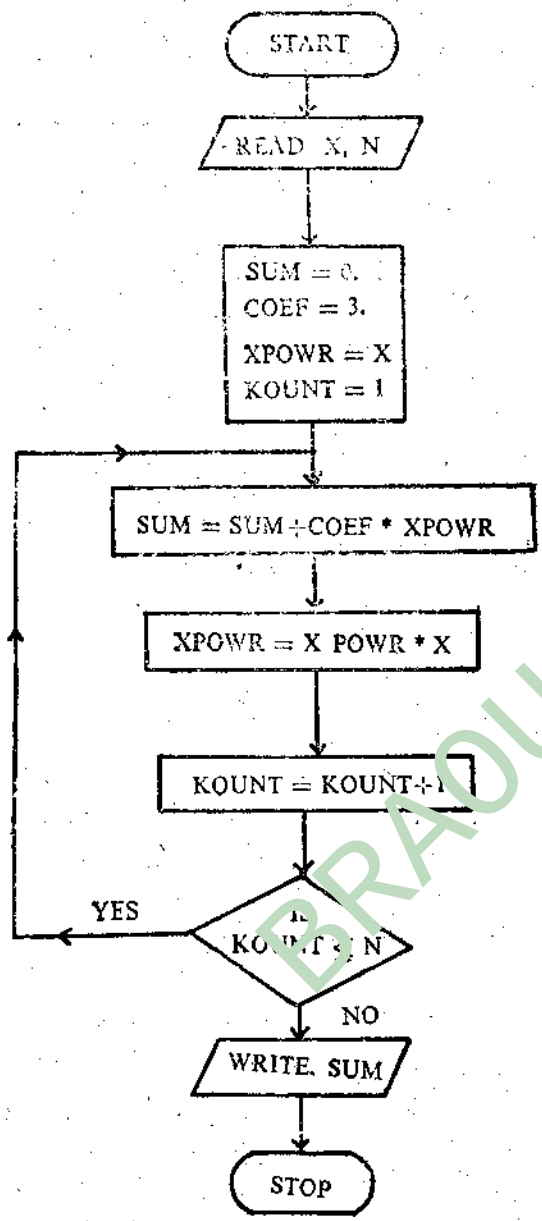


Fig.6 : Flow chart for summing series

Program 4 :

```
C       SUMMING SERIES
        READ (5, 6) X, N
6   FORMAT (F6.2, 5X, I3)
        SUM = 0.
        COEF = 3.
        XPOWR = X
        KOUNT = 1
        NDEX = 0
```

```
12  SUM = SUM + COEF * XPOWR
    XPOWR = XPOWR * X
    KOUNT = KOUNT+1
    IF (KOUNT .LE. N) GO TO 12
    WRITE (9, 15) N, SUM
15  FORMAT (1HØ, I3, 5X, E15.8)
    NDEX = NDEX + 1
    GO TO (16, 17, 18, 19), NDEX
16  N = 20
    GO TO 12
17  N = 30
    GO TO 12
18  N = 40
    GO TO 12
19  STOP
    END
```

Note : Here we used the formats of X and N as F6.2 and I3 respectively.

17.6 SUBSCRIPTED VARIABLES

So far we came to know about Real vairable and Integer variables only. They are single entities. They are also called 'non-subscripted' variables. FORTRAN allows another type of variables which are called subscripted variables.

A subscripted variable refers to an Array or a group of quantities. All the elements in a column or a table can be referred to by using one name only. This collection of elements is called an Array, and Each element with in the array is called a subscripted variable. Therefore we can say that an array is a set of variables having the same name. We refer to each member of the group by its position in the group.

It is advantageous to represent a group of values by a common variable name. For example, the students in a class may be represented by a common variable name, say STUD, and different students in the same class may be distinguished by writing different subscripts in parethesis after the variable name STUD. Thus, the variables STUD (1), STUD(2), ... STUD(50) represent 50 different students in a class. In mathematics, the co-ordinates of a point in three dimensional space, we represent them as X_1, X_2, X_3. In FORTRAN these quanties are represented by an array X, whose elements would be X(1), X(2), X(3). This array is said to be one dimensional array i.e., only one subscript is associated with each element. The individual elements of array are indicated by writing a subcript (or subscripts) in parenthesis after the common variable name. If a subscripted variable name in FORTRAN has one subscript, then it will represent one dimensional array of data. If it has two subscripts, then it will represent two dimensional array of data. If the subscripted variable name has three subscripts, then it will represent three dimensional array.

Subscripted variable may have more than three subscripts, in that case it will represent multi dimensional array of data. The subscripts must be enclosed in parenthesis and in the case of two or more dimensional arrays, the individual subscripts should be separated by commas. The number of subscripts permitted vary from one computer to other.

An example of a two dimensional array is the matrix L given by

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \end{bmatrix}$$

The elements of matrix L can be represented in FORTRAN as

L (1,1), L (1,2), L (1,3)
L (2,1), L (2,2), L (2,3)

A subscripted variable name is an integer subscripted variable name if it starts with I, J, K, L, M, or N and the elements of this array must be integers. Otherwise if it starts with other than I, J, K, L, M, or N if the elements are real, then it is called real subscripted variable name. Therefore the name of a subscripted variable in FORTRAN is formed in the same way as a non subscripted variable name. for example STUD is real subscripted variable, L is integer subscripted variable name. Therefore, the general form of subscripted variable is an integer or a real variable name followed by parenthesis enclosing the subscripts. The subscripts are separated by commas

i.e., W (*i*, *j*, ...)

where 'W' is the variable name which may be either integer or real *i*, *j*, ... etc., are subscripts.

Rules for writing subscripts

A subscript can only be a non-zero positive integer constant or variable or expression. If the subscript is an expression it should be in one of the following forms only

1. variable ± constant
2. constant * variable ± constant

(both vairable and constant or non-subscripted variables) i.e., if L denotes a non subscripted integer variable, M and N are integer constants then a subscript may be in any of the following forms

L
M or N
L + M or L – M
M * L
M * L + N or M * L – N

a) some acceptable examples of subscripts

J, L, 15, 5, K + 5, L – 10, 10* KOUN, 5 * INT, 25 * I – 6, 15 * LAMB – 7 etc.

b) some illegal subscripts

–5, 4.2, – I, 10 + I, 17 – M, J * 5, K*2–5, IN * 10 – 15 * I etc.

c) some examples of valid subscripted variables.

VEL (J), K (5), KON (2*I, 5*I–15), AB (L+2, M + 5, KN),

NUM (4*N–3), MARKS (I), ARRAY (I, J, K) etc.

d) some illegial subscripted variables

VEL (–J*3), INK (A*5), J (0), MAT (0, B), Z (2.9, X, 5*J), W (–5, I, K * J – I * J) etc.

17.6.1 The Dimension Statement

When we use the subscripted variable in a program, the information about these variable names must be supplied by a special specification statement. It is known as the DIMENSION statement. The Dimension statement which is a non-executable statement specifies the name, dimension, and size of those subscriped variables that are used by the programmer in the source program. i.e., this statement provides information to the compiler for reserving a specified number of memory locations for each subscripted variable appearing in the program.

The general form of the DIMENSION statement is
DIMENSION $q_1, q_2, \ldots$

where q's stand for subscripted variable names followed by parenthesis, enclosing one, two or three unsigned integer constants which give the maximum size of each subscript. The Dimension statement may be punched anywhere between the seventh and seventy second column for a card.

Examples

1. DIMENSION STUD 95, MARKS (50)
2. DIMENSION I(5, 10), J (10, 5), X (10)
3. DIMENSION KOUN (4, 4, 4), INT (10)
4. DIMENSION VEL (15), Z (5, 5), X (10)

Note the following points regarding DIMENSION statement.

1. Any number of subscripted variables may be dimensioned by a single DIMENSION statement.
2. the DIMENSION statement must appear before the first appearance of the subscripted variable in the program and it is a common practise to place the DIMENSION statement at the beginning of the program.
3. More than one DIMENSION statement may appear in the same program.
4. All the subscripted variable names occuring in aprogram should be different from all the names of standard library functions, ordinary variable names appearing in a program, and the name of the program itself.
5. DIMENSION I (10, 5), means that I is a two dimensional array for which the subscripts will never exceed 10 and 5. The DIMENSION statement thus reserve $10 \times 5 = 50$ storage locations for the array I. When writing a DIMENSION statement the number of storage locations asked by DIMENSION statement must not exceed the actual storage locations available in the computer's memory.

6. The dimension information may be specified either by DIMENSION statement or by a Type declaration statement (which we discussed earlier) such as

INTEGER MARKS (50), STUD (50)

REAL I (10, 5), KAT (50)

but never by both DIMENSION statement and Type declaration statement.

Subscripted variables are particularly useful in repetitive calculations. We will take up few more problems after DO statement, where we can notice the importance of subscripted variables.

Consider the following examples.

Ex. Write a program to compute the scalar product of the vecotrs A and b given by the formula

$$\text{SUM} = \text{SUM} + \sum_{i=1}^{5} a_i\, b_i$$

where $a_1, a_2, \ldots a_5$ and $b_1, b_2, \ldots, b_5$ are punched 5 values on each card with 5F6.2 format.

Solution

Here first we initialize SUM by zero, then each time we calculate $a_i\, b_i$ and add to the SUM until $i \leq 5$. If $i > 5$, we ask the computer to print the value of SUM. Consider the flow chart.

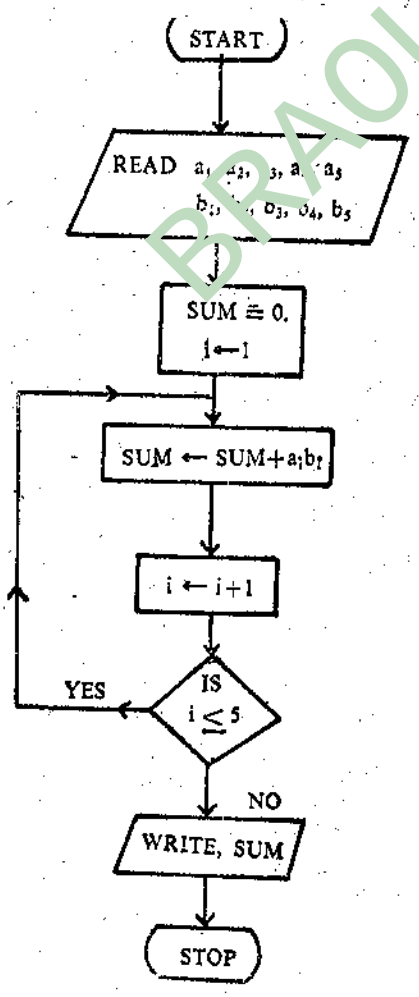


Fig. 7 : Flow chart to compute scalar product

Now the programe is as follows.

```
C       TO CALCULATE SCALAR PRODUCT
        DIMENSION A (5), B (5)
        READ (5, 10) A (1), A (2), A (3), A (4), A (5)
    1   B (1), B (2), B (3), B (4), B (5).
   10   FORMAT (5F6.2)
        SUM = 0.0
        I = 1
   15   SUM = SUM + A (I) * B (I)
        I = I + 1
        IF (I. LE. 5) GO TO 15
        WRITE (9, 20) SUM
   20   FORMAT (1Hb, 4HUSM =, 415.8)
        STOP
        END
```

17.7 PROGRAMMING ERRORS

Some times a source program deck may contain programming errors such as incorrect arithmetic expressions and improper FORMAT statements. Care should be taken to remove these errors before a program is submitted for run on the computer.

There are three types of errors. 1. compilation errors, 2. execution errors, 3. logical errors. In case we miss to detect compilation errors such as syntax errors and typographical errors, the compiler will detect and produce a list of diagnostic messages at the time of compilation which will assist us in locating them. Execution error such as division by a nearly zero quantity and error in FORMAT statement, cause the program to be terminated at the time of execution. Locating execution errors requires considerable experience in programming. In order to locate them, the programmer must select appropriate data sets to test each phase of the program and produce intermediate and final results which can be easily checked out. Computer cannot detect logical errors. Logical errors, such as, improperly sequenced statements, omission of some statements, an incorrect formula for the problem to be solved, and omission of some possibilities in the flow chart of a program cannot be detected by the computer. It is the programmer's skill that detects all types of errors, and this art of detecting errors is called Debugging comes through experience only.

Computer will take certain time for each arithmetic operation reducing the number of arithmetic operation (if possible) will reduce the computer time. The efficiency of the programmer lies in reducing computer time. The following are some simple guidelines which help to reduce computer time on most computers.

1. Minimize the number of machine operations (if possible). For example consider, the evaluation of the polynomial

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \text{ for a given value of } x.$$

For this, we have to evaluate each term and add it to a variable. We need $n(n + 1)/2$ multiplications and n additions. Instead of above direct method, if we use Horner's nesting method (using parenthesis) we can write the above polynomial as

$$P_n(x) = a_0 + x\left(a_1 + x\left(a_2 + x\left(a_3 + \ldots x\left(a_{n-1} + xa_n\right)\right)\right)\right)$$

The number of multiplications in this case is only n (less than $n(n+1)/2$) and the number of additions is still 'n'.

2. Whereever possible, both division and exponentiation should be kept to minimum (since, most computers will take more time for division and exponentiation than addition and subtraction),

For example the expression $\left(b^2-4c\right)/2a$ should be written as

(B * B–4.0 * A * C)/(A + A) and not as (B **2 – 4. * A * C)/(2.*A)

If we want to calculate z^3, write as Z * Z * Z but not as Z ** 3

3. Repeated evaluation of the same expression or sub-expression should be avoided. For example

ROOT 1 = (–B + SQRT (B * B – 4.0 * A * C)/(A + A)

should be written as G = A + A, H = –B/G

DISC = SQRT (B * B – 4.0 * A * C)/G

ROOT 1 = H + DISC

ROOT 2 = H – DISC

4. Repeated calls for standard functions should be avoided. For example

$$T = 15 \sin^4(x) + 7 \sin^3 x - 12 \sin^2(x)$$

should be written as

S = SIN (X)

T = ((15.0 * S + 7.) *S – 12.) *S*S.

5. Standard library functions should be used whenever feasible. For example SQRT (X) is faster than X ** 0.5.

6. Whenever possible, the exponent should be of type integer not eal. For example X ** 5 is faster than A ** 5.

7. Data type should be matched whenever feasible, for example the statement R = S * 4.0* P/(B + 5) will be executed faster than R = S * 4 * P/(B + 5) (since computer time is consumed in making conversions while evaluating expressions)

Only by experience in programming we can reduce the computer time whenever possible, following the guidelines given above and similar other simple guidelines.

17.8 TO DO STATEMENT

Earlier we have dealt with control statements which alter the normal sequential order of execution to repeat a particular series of steps, or to repeat the entire program a given number of times with different sets of data on each occasion etc. There, we came to know that GO TO statement provides unconditional branching and IF statement provides conditional branching.

Another important control statement is 'DO' statement. We can solve many problems on a computer where we involve with a small number of computations which are repeated a specific number of times whenever we come across a repetitive calculation, we use DO statement which makes the program very compact.

The general form of the DO statement is

DO n $i = m_1, m_2, m_3$
.....
.....
n .....

where n is the statement number of an executable statement, 'i' is an unsigned non-subscripted integer variable and m_1, m_2, m_3 are non-subscripted integer constants or variables which must have values greater than zero. Here m_3 is optional and is assumed to be 1 if it is omitted. Therefore another valid form of the DO statements is

DO n $i = m_1, m_2$
.....
.....
n .....

The DO statement is punched between columns 7 and 72 of a card.

Now consider the meaning of the DO statement. The DO statement is a command to execute all the statements following it upto and including the statement numbered 'n'. This set of statements is called the Domain or Range of DO. DO statement first initialises the index 'i' and sets it equal to an intial value m_1. The domain of DO is executed. Now the index is increased by m_3 and compared with the final value of m_2, control passes out of the domain to the statement next to 'n', if the value is greater than m_2.

Therefore the domain of the DO is executed with all the values of the index 'i', less than or equal to the test value m_2. In the second form of the DO statement where m_3 is absent, 'i' is incremented by 1 after each execution of the domain. To get clear idea see the following flow chart which shows the effect of a DO statement.

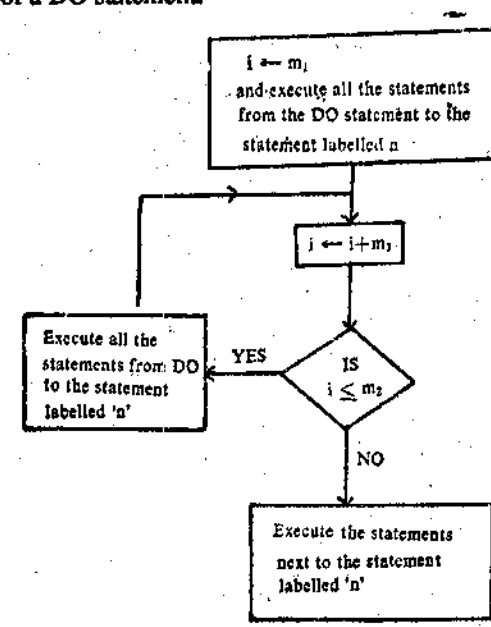


Fig. 8 Flow chart of the effect of DO statement

A loop in a program controlled by a DO statement is called a DO loop.

For example, consider the following DO statement

```
DO 10   J  =  1, 25, 5
```

Here the computer initialize J by 1 and starts with the very next statement after the DO statement and executes all the statements down to and including the statement numbered 10. As soon as the statement numberd 10 is executed for the first time, the control goes back to the DO statement. Now it increases the value of J by 5. So now the value of J is 6. Now it compares with 25. Since the value of J is less then 25, the computer repeats the execution of the statements down to and including the statement labelled 10. It goes on doing this. Finally after doing the calculations with the value J = 21 the computer goes to execute statement immediately after the statement numbered 10 (∵ next value of J willbe 21 + 5 = 26 greater than 25).

Some examples of allowable DO statements

1. DO 50 I = 1, 155, 3
2. DO 5 IR = L, M, N
3. DO 44 KOUN = 1, 12
4. DO 25 NUM = N, M
5. DO 125 IRPT = 5, MN, ML

Now let us consider few examples.

Ex. 1 : Write a program to calculate D, given by the formula

$$D = \sum_{i=1}^{N} \left(x_i^2 + y_i^2 + z_i^2 \right)$$

Solution

Here we have to calculate $x_i^2 + y_i^2 + z_i^2$ for i = 1, 2, ... N and then we have to sum up all these. So we use DO statement. The programme appears as follows,

Programe 1 :

```
C  TO CALCULATE   D USING DO STATEMENT
   D = 0.
   DO 10 I = 1, N
   READ (5, 2) X, Y, Z
 2 FORMAT (3F8.3)
   TERM = SQRT (X*X + Y*Y + Z*Z)
10 D = D + TERM
   WRITE (9, 12) N, D
12 FORMAT (1Hb, I5, 5X, E15.8)
   STOP
   END
```

Ex. 2 : A subscripted vairable C_i is defined as $C_i = \sqrt{A_i^2 + B_i^2}$ for even values of i, and $C_i = \sqrt{4A_i + 10B_i}$ for odd values of i. Write a program to compute C_i for 'i' in the range from 1 to 10.

Solution

Here we have to use subscripted variable A (I), B (I), C (I) and therefore remember that must be dimensioned by DIMENSION statement. Here in this problem we use two DO Statements one for finding even C_i and another for finding odd C_i. The corresponding program is as follows.

Program 2 :

```
C       TO CALCULATE C(I)
        DIMENSION A (10), B (10), C (10)
        READ (5, 2) A(1), A(2), A(3), A(4), A(5), A(6), A(7), A(8), A(9), A(10)
    2   FORMAT (5F7.2)
        READ (5, 2) B (1), B (2), B (3), B (4), B (5), B (6), B (7), B (8), B (9), B (10)
C       TO GET EVEN C (I)
        DO 4  I = 2, 10, 2
    4   C (I) = SQRT (A (I) * A (I) + B (I) * B (I))
C       TO GET ODD C (I)
        DO 6  I = 1, 10, 2
    6   C (I) = SQRT (4. * A (I) + 10. * B (I))
        WRITE (9, 8) C (1), C (2), C (3), C (4), C (5), C (6), C (7), C (8), C (9), C(10)
    8   FORMAT (1H0, 5E15.8)
        STOP
        END
```

Ex. 3 : Write a program to calculate F for values of X varying from 1.0, 10.0 in steps of 0.05 where

$$F = \frac{\sin x}{e^x - \cos x}$$

Solution

We know that the DO loop index 'i' must be non subscripted integer variable. Here index is X and indexing parameters are 1.0, 10.0 and 0.05. So computer will not accept if we write

```
DO 10   X  =  1.0, 10.0, 0.05
```

instead of X, we take IX and define X = IX/100.0 and if we take m_1 as 100, m_2 as 1000 and m_3 as 5, and if we write

```
DO 10 IX  =  100, 1000, 5
      X   =  IX/100.0
```

we get the required values of X, i.e., 1.0, 1.05, 1.10 etc. Therefore required program is as follows.

Program 3 :

```
C       CALCULATION OF F
        DO 10 IX = 100, 1000, 5
        X = IX/100.0
        F = SIN (X)/(EXP (X) - COS (X))
   10   WRITE (9, 15) X, F
```

```
15  FORMAT (1H$, 5X, F6.2, 5X, E15.8)
    STOP
    END
```

There are certain restrictions in writing DO statements. Now let us know about these restrictions.

17.8.1 Restrictions in Writing DO Statements

1. The first and last statements with in the range of a DO (Range of DO is the set of statements beginning with the one immediately following the DO statement and continuing upto and including the statement number n, must not be a non executable statement, such as DIMENSION or FORMAT statements etc.

2. The last statement in the range of a DO should not be a GO TO, STOP, PAUSE, arithmetic IF, DO, or a logical IF.

3. If it is unavoidable, FORTRAN provides a dummy statement, which is called CONTINUE statement. The general form of this executable statement which most often placed at the end of the DO loop is

```
CONTINUE
```

For example, consider a one dimensional array X containing 100 elements. Suppose we want to find out the product of all the elements of X. To do this, first we have to check up if any of these elements is zero. If any of elements is zero, then we know the product is zero. Therefore to do this, we write

```
    DO 10 I = 100
10  IF (A (I) .EQ. 0.0) GO TO 15
15  PROD = 0.
```

So here the last statement in the range of DO is logical IF statement. Therefore computer will not accept it. So we use CONTINUE statement as last statement as follows.

```
    DO 10 I = 1, 100
    IF (A(I).EQ. 0.0) GO TO 15
10  CONTINUE
15  PROD = 0
```

This continue statement may be punched any where between columns 7 and 72 of a card.

3. The index 'i' and the indexing parameters m_1, m_2, m_3 cannot be changed by another statement with in the range of a DO. Throughout the range of a DO statement the index 'i' is available for computations and it is usually used as a subscript.

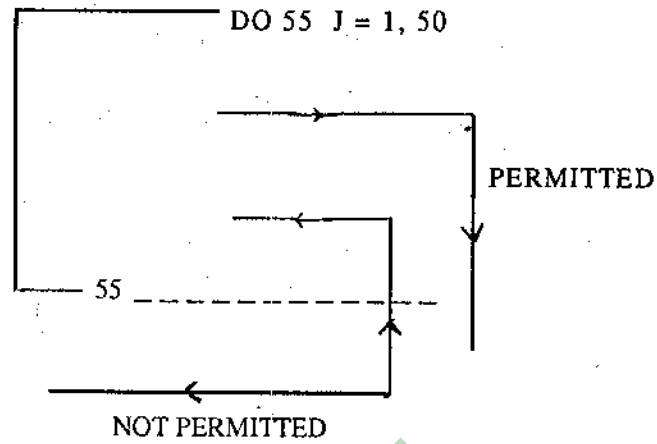
4. The number of repetitions r specified by a DO statement, with indexing parameters $m1$, $m2$ and $m3$ is given by

$$r = 1 + \left(m_2 - m_1\right)/m_3.$$

where $\left(m_2 - m_1\right)/m_3$ is calculated in integer mode (i.e., neglect fractional part). For example if $m_1 = 1$, $m_2 = 10$ and $m_3 = 2$ then $r = 1 + 9/2 = 1 + 4 = 5$.

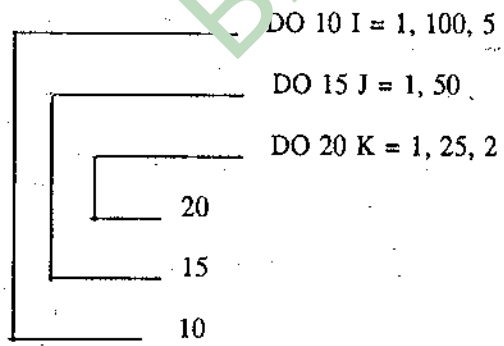
5. After the statements with in the range of the DO have been executed required number of times (r), control passes to the statement immediately following the statement labelled n. This exit is

known as 'normal exist' and DO is said to be satisfied. A non-normal exit occurs when a transfer of control statement (GO TO statement etc.) with in the range of a DO transfers control to a statement outside the range of the DO. When a normal exit occurs, the value of index '*i*' is undefined and this index should be redefined if we want to use it in further computations. Where as when a non normal exit occurs, the index '*i*' is available for computation and is equal to the last value it attained in the DO loop. Transfer from inside to outside of the DO loop is allowed whereas transfer from outside to inside is not allowed.

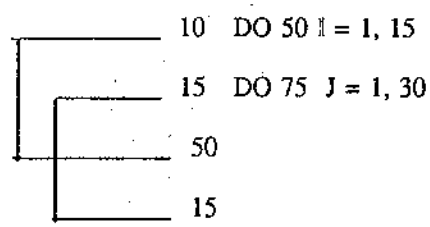


6) Enclosed with in the domain of DO loop there may be other DO loops. That is, the domain of the latter DOs must be within the domain of the previous DOs. A set of DOs satisfying this rule is called nested DOs.

Consider the following example of nested DOs.



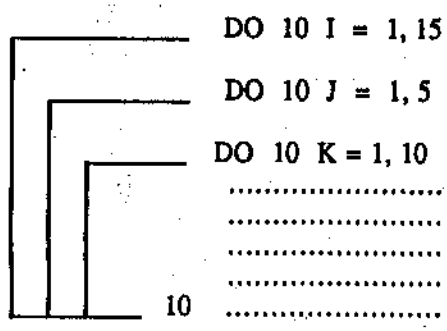
The domain of a DO may not cross the domain of another DO. Consider the following situation.



Here the first DO statement (whose statement number is 10) commands the program to execute all statements following it upto statement number 50 fifteen times. While executing this,

control reaches statement 15 which is another DO statement commands the program to execute all statements upto 75 thirty times. Thus when control reaches statement 50, the first do commands it to loop back to 10 and the second DO commands it to proceed sequentially to statement 75. This ambiguity is not allowed.

The last statement defining the domain of the two or more DOs may be same. For example the following is valid.



Now let us consider few examples.

Ex. 1 : Two one dementional arrays A and B have 10 elements each. Write a program to find the sum 'C' of two arrays A and B, assuming the elements of A and B are each punched with field specification F15.6.

Solution

Here we are using subsctipted variables A, B and C. So we have to use DIMENSION statement. We will read in the elements of A and B by using DO statement.

Program 4 :

```
C          TO FIND THE SUM OF A AND B
           DIMENSION A (10), B (10), C (10)
           DO 10  I = 1, 10
10         READ (5, 15) A (I)
15         FORMAT (F15.6)
           DO 20  I = 1, 10
20         READ (5, 15) B (I)
           DO 25 I = 1, 10
           C (I) = A (I) + B (I)
25         WRITE (9, 35) I, C (I)
35         FORMAT (1Hb, I3, 5X, E15.8)
           STOP
           END
```

Ex.2 : Given that

$$W = \frac{911.8}{\frac{1}{n^2} - \frac{1}{m^2}}$$

write a program to produce a table of values of W for all combinations of

m = 2, 3, 4, 5, ... 50

n = 1, 2, 3, 4, ... $(m - 1)$

Solution

```
C       CALCULATION OF W
        DO 15 M = 2, 50
        K = M - 1
        DO 15 N = 1, K
        W = 911.8/(1./ (N * N) - 1./(M * M))
15      WRITE (9, 20) M, N, W
20      FORMAT (1Hb, I5, 5X, I5, 5X, E15.8)
        STOP
        END
```

17.9 DO TYPE NOTATION IN INPUT/OUTPUT STATEMENT

When arrays are to be read in or to be printed out DO type Input/Output lists will be helpful in writting the source program. For example, in the previous artical (Example 1) to read in the elements A (I) and B (I) we used DO statement as

```
        DO 10 I = 1, 10
10      READ (5, 15) A (I)
15      FORMAT (F15.6)
```

Here we need 10 data cards each with one value of A (I) because the READ statement is executed 10 times as command by DO. Time taken to input the information on 10 data cards will be about 10 times that needed for inputting the information from one card.

If we want to read all the 10 values with one READ statement (one data card) we may write

```
        READ (5, 25) A (1), A (2), A (3), A (4), A (5), A (6), A (7), A (8), A (9), A (10)
25      FORMAT (10F15.6)
```

It is tedious to write all the elements in the READ statement if an array has more number of elements. FORTRAN allows the use of another simpler READ statement which does what DO statement presented earlier would accomplish but without requiring 10 data cards.

The corresponding statement is

```
        READ (5, 15) (A (I), I = 1, 10)
15      FORMAT (10F15.6)
```

The above statement will read in 10 values punched in one card and assign them respectively to A (1), A (2), A (3), ..., A (10). If all the ten values cannot be accomodated in a single card, they may be punched on a succeeding card. Cards will be read till 10 values are found. The above READ statement is called READ statement with an implied DO input list.

For example, if the values of the even component of the vector A are to be read in, then the corresponding statement is

```
READ (5, 15) (A (I), I = 2, 10, 2)
15  FORMAT (5F15.6)
```

The above statement will assign succeeding numbers on one card to A (2), A (4), A (6), A (8), A (10). Similarly to read in odd components, the corresponding statement is

```
READ (5, 15) (A (I), I = 1, 10, 2)
15  FORMAT (5F15.6)
```

More than one array may be specified in the list of a READ statement. For example, consider

```
READ (5, 30) (A (I), B (I), I = 1, 10)
30  FORMAT (20F15.6)
```

The above statement assign the numbers punched on cards successively to A (1), B (1), A (2), B (2), ..., A (10), B (10).

Therefore the implied DO list for subscripted variables using only one index (One dimensional array) takes the form

$$(q_1, q_2, \ldots, q_k, i = m_1, m_2, m_3)$$

where q's are subscribed or non subsribed variables, 'i' is a non subscribed integer variable and m_1, m_2 and m_3 are unsigned integer constans or non-subscribed integer variables.

Some Examples

1) READ (5, 60) (A (I), B (I), C (I), I = 1, 50)
2) READ (5, 25) M, (X (I), Z (I), I = 1, M)
3) READ (5, 100) L, (R (M), M = 1, L) N, (S (I), I = 1, N)

Two dimensional arrays may also be input by a similar READ statement. The general form of the implied DO list in this case is

$$\left((q_1, q_2, \ldots q_k, i = m_1, m_2, m_3), q_1', q_2', \ldots q_n', i = l_1, l_2, l_3\right)$$

where qs and q's are subscripted or non subscripted variables and $i, m_1, m_2, m_3, j, l_1, l_2, l_3$ imply same as the symbols in previous case.

For example, the statement

```
READ (5, 50) ((A (I, J), I = 1, 10), J = 1, 5)
50  FORMAT (10F8.3)
```

This will read in successive numbers punched on cards and assign them respectively to

A (1, 1) A (2, 1) A (3, 1) ..... A (10, 1)

A (1, 2) A (2, 2) A (3, 2) ..... A (10, 2)

⋮

A (1, 5) A (2, 5) A (3, 5) ..... A (10, 5)

For example consider the statement

```
      WRITE (9, 15) (((A, J), B (I, J), I = 1, 5), J = 1, 2)
15    FORMAT (1H h, (10F15.8))
```

This will print 10 values on each line in the following order

A (1, 1) B (1, 1) A (2, 1) B (2, 1) ..... A (5, 1) B (5, 1)

A (1, 2) B (1, 2) A (2, 2) B (2, 2) ..... A (5, 2) B (5, 2)

Three deimensional arrays can also be read in using implied DO notation in a similar way.

For example consider

```
      READ (5, 100) (((A (I, J, K), I = 1, 10), J = 1, 2), K = 1, 2)
100   FORMAT (10F8.5)
```

This statement will read in successive number punched on cards and assign them respectively to

A (1, 1, 1) A (2, 1, 1) ..... A (10, 1, 1)

A (1, 2, 1) A (2, 2, 1) ..... A (10, 2, 1)

A (1, 1, 2) A (2, 1, 2) ..... A (10, 1, 2)

A (1, 2, 2) A (2, 2, 2) ..... A (10, 2, 2)

Note : The index used in the list of the READ statement may itself be an integer variable provided it is defined before it is used.

For example, consider the statement

Read (5, 16) (A (I), I = J, K) is valid provided J and K are already defined. All the other restrictions on indicies of a DO statement also apply to the DO type READ and WRITE statements.

Now let us consider few examples.

Ex. 1 : A and B are two one-dimensional arrays each containing N (less than 50) elements, write a program to compute and printing C where

$$C = \sum_{I=1}^{N} A(I)\, B(I).$$

Solution

To read in the elements of A and B, we use DO type notation. Since A and B are subscripted variable names, they must be dimensionalized. To calculate C, we use DO statement. Therefore the program appears as follows.

Program : 6

```
C       TO CALCULATE C
        DIMENSION A (50), B (50)
        READ (5, 10) N, (A (I), B (I), I = 1, N)
10      FORMAT (I3/(10F7.3))
        C = 0
        DO 15 I = 1, N
        C = C + A (I) * B (I)
15      CONTINUE
        WRITE (9, 20) N, C
20      FORMAT (4Hb, I3, 5X, E15.8)
        STOP
        END
```

Ex. 2 : X and Y are two dimensional arrays of order N × K and K × M where each M, N and K is less than or equal to 10. Write a program to find the product (Z) of the two matices X and Y.

Solution

Here we use DO type notation in READ statement and also in WRITE statement. We know the procedure how to find the product of two matrices. We use DO statement to find the product. So the program appears as follows.

Program : 7

```
C       PRODUCT OF TWO MATRICES
        DIMENSION X (10, 10), Y (10, 10), Z (10, 10)
        READ (5, 10) N, K, M
10      FORMAT (3I5)
        READ (5, 15) (X (I, J), I = 1,  N), J = 1, K)
        READ (5, 15) (Y (I, J), I = 1, K), J = 1, N)
15      FORMAT (10F6.2)
        DO 20  I = 1, N
        DO 20  J = 1, M
        Z (I, J) = 0.0
        DO 20 L = 1, K
```

```
20  Z (I, J) = Z (I, J) + X (I, L ) * Y (L, J)
    WRITE (9, 25) ((Z (I, J), I = 1, N), J = 1, M)
25  FORMAT (1Hb, (5E15.8))
    STOP
    END.
```

17.10 SUMMARY

Sometimes we may have to alter the normal flow of control of execution of a program. In such cases we will use the control statements. The most important control statements are GO TO, Arithmetic IF, Logical IF, computed GO TO and DO statements. When we come across the subscripted variables, we make use of DIMENSION statement. The dimension statement must appear beofe the first appearance of the subscripted variable in a program. The dimension statement is a non-executable statement and will specify the name, dimension and size of the subcripted variables. There are three types of errors possible in a program (i) compilation errors, (ii) execution errors and (iii) logical errors. The art of detecting errors is called debugging. One will be considered as an efficient programmer if one writes the program in such a way that it will consume least computer time for the execution. One will achieve it by experience. When a small number of computations are to be repeated a specified number of times, we use the DO statement. Also, when arrays are to be read in or to be printed out, we make use of DO type input/output statements.

17.11 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail*

1. Explain a) unconditional GO TO statement b) Arithmetic IF statement
2. Draw flow chart and write program to find the largest of given N numbers using Arithmetic IF statement.
3. If $g = 32.17\ (4390/(4390 + h))^2$, for $h > 0$

   $= 32.17\ (1 + h/4390)$, for $h \leq 0$

   write a program to compute and print the values of g for $h = 20$ to 200 in steps of 10.
4. Draw the flow chart and write a program to find the largest of given three numbers using Logical IF statement.
5. The sum of the squares of the first 'n' natural numbers is given by $S = n\ (n + 1)\ (2n + 1)/6$, write a program that will find S for $n = 5$ to 50 in steps of 5.
6. Verify whether each one of the following are valid subscript or not. If it is not a valid subscript, specify the reason.

   (1) –1500, (2) 150*V – 1, (3) J * 5, (4) 0, (5) J + 1

7. Satate whether the following are valid subscripted variables or not. If not give reason.

   (1) A (L + 2, M + 5, N + 10) (2) MATX (5, 5, 5) (3) IN (K * 10)

   (4) X (-15, 0, 5) (5) GAMA (X + 5, Y + 5, Z + 5).

8. Explain the importance and use of DO statement in FORTRAN programming.

9. Write a program to find the sum of the following series $S = 3x + 5x^2 + 7x^3 + 9x^4 + \ldots\ldots$ upto N terms using DO statement.

10. If X and Y are 2, two dimentional arrays of order M × N, where each M, N is less than or equal to 10. Write a program to compute and print

    T = Transpose of X

    S = X + Y

11. Two one-dimensional arrays X and Y are 20 elements each. Write a program to compute and print the quantities

$$A = \sum_{k=1}^{20} (x_k + y_k)^2, \quad B = \sum_{k=1}^{20} |x_k - y_k|$$

assuming the input data is punched in a pairwise manner (x_1, y_1 and y_1 on the 1st card, x_2 and y_2 on second card and so on) with 2F8.3 format.

II. *Briefly answer the following*

1. Given the statement IF (X * X – Y * Y) 10, 15, 20 where X = –3.0, Y = 4.0, what statement would the computer execute after the above IF statement in the program?

2. If $T = \dfrac{150}{\dfrac{R - 100}{15} + \dfrac{15}{R - 100}}$

   write a program to read 5 data cards containing the values of R (one value on each card with format specification F4.0) and compute and print the value of T in each case.

3. Write a program to read in the value of x with F10.3 format and compute and print the value of F (x) defined by

$$F(x) = x^2 + 5x, \text{ for } x < 0$$

$$= 0, x = 0$$

$$= 1 + \frac{x}{\sqrt{x^2 + 5}}, x > 0$$

4. The scattering cross section of an electron is given by the formula

$$A = \frac{0.0666\, r^4}{(r^2 - 1)^2}$$

where 'r' is the ratio of the radiation frequency of the electron. Write a program to calculate and output the values of A for r = 10, 20, ... 100.

5. Using a DO loop, write a program to compute and print Z for each value of X, X = 1.5, 1.55, 1.60, ..., 10.0, where

$$Z = \log(1 + X) / \left(e^{x} - 1 + x^{3}\right)$$

6. Give that E = log V + log T/(S – 1) using a nest of three DO loops, write a program that will produce a table for all combinations of

   V varying from 1 to 10 in steps of 0.5

   T varying from 100 to 1000 in steps of 10 and

   S varying from 1.1 to 1.5 in steps of 0.1

7. Write an input statement using DO type list to read in the elements of two one-dimensional arrays P and Q having 20 elements each, if the values are punched 5 on each card with FORMAT specifications F10.3.

UNIT–18 : SUBPROGRAMS AND SUBROUTINES

Contents

18.1 AIMS AND OBJECTIVES

After going through this unit you will be able to divide a large program into several subprograms by making use of arithmetic statement function function subprogram and subroutine subprogram and will be able to call them to the main program.

18.2 INTRODUCTION

In Fortran language the programmer has the facility to segment large program into several subprograms which on reassembly constitute a complete Fortran program. For a host of problems like finding inverse of a matrix, calculation of zeros of a polynomial etc. the subprograms can be prepared. These subprograms, once written, can be executed any number of times at different points within the main program. The time and effect in writing long and complex source program can be minimized by properly utilising the available subprograms.

If the values of a single-valued function is to be computed by using i) a single arithmetic statement we make use of arithmetic function statement and ii) several Fortran statements we make use of FUNCTION subprogram.

18.3 ARITHMETIC FUNCTION STATEMENT

The general form of an arithmetic function statement is,

Name $(v_1, v_2, \ldots, v_n)$ = expression

"Name" is the name of the function statement, $v_1, v_2, \ldots, v_n$ are distinct non-subscripted variable names.

The Rules to be observed while making use of an Arithmetic function statement in a program are :

1. All arithmetic function statement definitions should appear before the first executable statement of the program.
2. The name of the function is to be formed according to the rules that are applicable to a variable name.
3. The name of the function must be different from the names of library functions and other subprograms occuring in the main program.
4. An arithmetic function statement should be defined by a single arithmetic statement.
5. The arguments $v_1, v_2, \ldots, v_n$ are dummy variables. They will be replaced later by the actual variables or values when the function is used in the program.
6. The "expression" should not contain any subscripted variable. It may cantain variables not specified as arguments.
7. Any other arithmetic statement function occuring in the "expression" should have been previously defined.
8. The arithmetic statement function is called by the program whenever the name of the function occurs in an arithmetic expression in the program
9. The list of actual arguments must agree in order, number and type with the list of dummy arguments in the function statement definition.

Let us consider the following example

```
      RSUM (X, Y, Z) = SQRT (X * X + Y * Y + Z * Z)
      A  =  4.892
      B  =  5.613
      C  =  6.194
10    D  =  4.167 + RSUM (A, B, C)
      E  =  D * 4.0
      WRITE (9, 100) E
100   FORMAT (5Hb E = b b b, F15.4)
      STOP
      END
```

In this example the functin RSUM is defined before the first executable statement A = 4.892. The occurance of the function RSUM together with its actual arguments A, B, C in the statement

with label 10 makes the object program evaluate the function replacing X, Y and Z by the values of A, B and C respectively and add the result of 4.167 and store it as E.

The actual arguments of a function can be members of the list of a subscripted variable though no subscripted variable can be used as an argument in the definition of the function. For example the statement with label 10 may be

```
10   D = S (4) + RSUM (S (1), S (2), S (3))
```

18.4 FUNCTION SUBPROGRAM

The general form of a FUNCTION subprogram is

type FUNCTION name $(v_1, v_2, \ldots, v_n)$

.....

.....

name = expression

.....

.....

RETURN

.....

.....

name = expression

.....

.....

RETURN

END.

Here type indicates the type of function (i.e.,) REAL or INTEGER. Type indication is optional. Name is the symbolic name of the function and $v_1, v_2, \ldots, v_n$ ar dummy arguments.

Let us write the program for calculating *n*!

```
      INTEGER FUNCTION FACT (N)
      IF (N) 11, 12, 13
11    FACT = 0
      RETURN
12    FACT = 1.
      RETURN
13    P = 1.
      DO 100 J = 2, N
      100  P = P * J
      FACT = P
       RETURN
      END
```

In this subprogram the function name is FACT. It is declared that the type of the function FACT is INTEGER. The argument N is a dummy argument

The following program calls the FUNCTION sub-program FACT.

```
      READ (3, 10) K, L, M
      L = I * M + I
      F = FACT (I) + FACT (L)/FACT (K)
      PRINT (9, 20) K, L, I, M, F
10    FORMAT (3I5)
20    FORMAT (4I5, E14.6)
      STOP
      END
```

Let us consider another example. Suppose we want to evaluate the value of a third order determinant.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

```
C     EVALUATING THIRD ORDER DETERMINANT
      DIMENSION A (3, 3)
      READ (3, 10) ((A (I, J), I = 1, 3), J = 1, 3)
10    FORMAT (9F6.2)
      C1 = DET (A (2, 2), A (2, 3), A (3, 2), A (3, 3))
      C2 = DET (A (2, 1), A (2, 3), A (3, 1), A (3, 3))
      C3 = DET (A (2, 1), A (2, 2), A (3, 1), A (3, 2))
      VALUE =  A (1, 1) * C1 – A (1, 2) * C2 + A (1, 3) * C3
      WRITE (4, 20) VALUE
20    FORMAT (E14.6)
      STOP
      END
C     SUBPROGRAM TO EVALUATE SECOND ORDER DET
      FUNCTION DET (P, Q, R, S)
      DET = P * S – Q * R
      RETURN
      END
```

It can be observed that the body of the subprogram follows the function declaration and specifies the computation to be performed on the arguments. The following points are to be kept in mind while writing a sub-program.

i) Every function to be evaluated by the FUNCTION sub-program must be single valued.

ii) If the type of the FUNCTION is not indicated then it will be treated as REAL or INTEGER in the same way as in the case of variable names.

iii) The rule for writing the name of the FUNCTION sub-program is the same as that for variable names.

iv) The name of the FUNCTION should be different from those of the standard library functions, the name of the main program and the name of other subprograms used in the main program.

v) The dummy arguments must be non-subscripted variables, or names or arrays (without subscripts) or names of subprograms or library functions.

vi) Whenever a dummy argument is an array name, it must appear in a DIMENSION statement in the subprogram and its maximum size must be the size of the actual argument in the calling program.

vii) The name of the function must appear, atleast once, as a variable name on the left hand side of an assignment statement.

viii) The RETURN statement must appear, atleast once, in the body of the subprogram.

ix) The last statement in a subprogram is the END statement which signifies the physical end of the subprogram.

x) The actual arguments in the calling function must agree in numbers, order and mode with the dummy arguments in the FUNCTION statement. They can be constants, variable names, subscripted variables or expressions.

xi) When the name of a FUNCTION subprogram is encountered in the main program, the control is transfered to the subprogram. The dummy arguments are replaced by the actual arguments in the body of the FUNCTION subprogram and the execution of subprogram is carried out.

xii) The RETURN statement returns control to the particular referencing point in the main program.

Ex. Write a program to add the scalar products of 3 pairs of 30 element vectors.

Solution

Let the three pairs of vectors be A_1, B_1 ; A_2, B_2; A_3, B_3 each of which consists of 30 elements.

We want to find S $= \sum_{i=1}^{30} \left(a_{1i}\, b_{1i} + a_{2i}\, b_{2i} + a_{3i}\, b_{3i} \right)$

Where $a_{11}, a_{12}, \ldots\ldots a_{1,30}$ are the elements of A_1 etc.

```
C       MAIN PROGRAM TO FIND SUM OF SCALAR PRODUCTS
        DIMENSION A1 (30), B1 (30), A2 (30), B2 (30), A3 (30), B3 (30)
        READ (3, 10) (A1 (I), B1 (I), A2 (I), B2 (I), A3 (I), B3 (I), I = 1, 30)
   10   FORMAT (5F14.4)
        SUM = SCPDT (A1, B1) + SCPDT (A2, B2) + SCPDT (A3, B3)
        WRITE (4, 20) SUM
   20   FORMAT (F16.6)
        STOP
        END
C       FUNCTION SUBPROGRAM TO CALCULATE SCPDT
        FUNCTION SCPDT (SJ, SK)
        DIMENSION SJ (30), SK (30)
        SCPDT = 0.
        DO 50 I = 1, 30
   50   SCPDT = SCPDT + SJ (I) * SK (I)
        RETURN
        END
```


18.5 SUBROUTINE SUBPROGRAM

SUBROUTINE name $(v_1, v_2, \ldots, v_n)$

.....

.....

RETURN

.....

.....

END

Name is the name of the SUBROUTINE and $v_1, v_2, \ldots, v_n$ $(n \geq 0)$ are the arguments of the SUBROUTINE and are to be enclosed within parantheses following the name. Commas seperating the arguments are essential.

FUNCTION Subprogram returns a single value to the main program whereas SUBROUTINE can return any number of values to the main program.

Rules to be observed while writing a Subroutine subprogram

1. The name of a SUBROUTINE can have 1 to 6 alphanumeric characters with the first character an alphabet.
2. The arguments $v_1, v_2, \ldots, v_n$ are dummy variables. These can be non-subscripted array names, or dummy names of SUBROUTINE or FUNCTION subprograms. When an array name is used, its dimension should be specified in the subroutine. The dimensions of corresponding arguments must be same in the subroutine and the main program.
3. Subroutine should contain atleast one RETURN statement (which is the last executable statement)
4. A Subroutine doesnot store a value in its 'name'. The name serves as a device for reference for a CALL Statement.
5. A Subroutine can call another Subroutine or Function subprogram
6. A Subroutine is called by the main program by a CALL statement.

18.6 CALL STATEMENT

The appearance of CALL statement in the main program makes the control shift to the Subroutine. The subroutine is executed and the results are returned to the main program along with the control by the appearance of RETURN in subroutine.

The general form of a CALL statement is

$$\text{CALL name}\left(e_1, e_2, \ldots, e_n\right)$$

where *name* is the name of the subroutine and $e_1, e_2, \ldots, e_n$ are the actual arguments.

The actual arguments should agree with the dummy arguments in number, order, mode and dimension.

If an argument of a subroutine is a dummy name of a FUNCTION or SUBROUTINE subprogram, the corresponding actual argument in the CALL statement should be the name of the actual FUNCTION or SUBROUTINE to be used.

Note : i) SUBROUTINE can be written even without arguments.

ii) The 'name' of the subroutine does not decide the modes of the results.

iii) The 'name' of the subroutine should not occur anywhere in the subprogram except in the first statement.

Ex. 1 : Write a SUBROUTINE subprogram to evaluate the product of 10×10 matrics. Write a main program which calls this subroutine and prints the product matrix.

Solution

Let A (L × M) and B (M × N) be the matrices whose product is written as C (L × N).

Note : Usually when a program is prepared it is better to write it in a general way applicable to any special case.

```
C       SUBROUTINE MATPDT (A, B, C, L, M, N)
        DIMENSION A (10, 10), B (10, 10), C (10, 10)
        DO 100  I = 1, L
        DO 100  J = 1, N
        C (I, J) = 0.
        DO 100  K = 1, M
100     C (I, J) = C (I, J) + A (I, K) * B (K, J)
        RETURN
      END
```

The main program to call this subroutine and print the matrix is given below.

```
C       MAIN PROGRAM
        DIMENSION P (10, 10), Q (10, 10), R (10, 10)
        READ (5, 10) ((P (I, J), I = 1, 5) , J = 1, 6), ((Q (I, J), I = 1, 6), J = 1, 10)
        CALL MATPDT (P, Q, R, 5, 6, 10)
        WRITE (9, 50) ((C (I, J), J = 1, 10), I = 1, 5)
10      FORMAT (10F6.2)
50      FORMAT (5F14.4)
        STOP
        END
```

Ex. 2 : Write a subroutine to calculate cos x for all angles from 0° to 90° in 10° increments. When x is in radians take

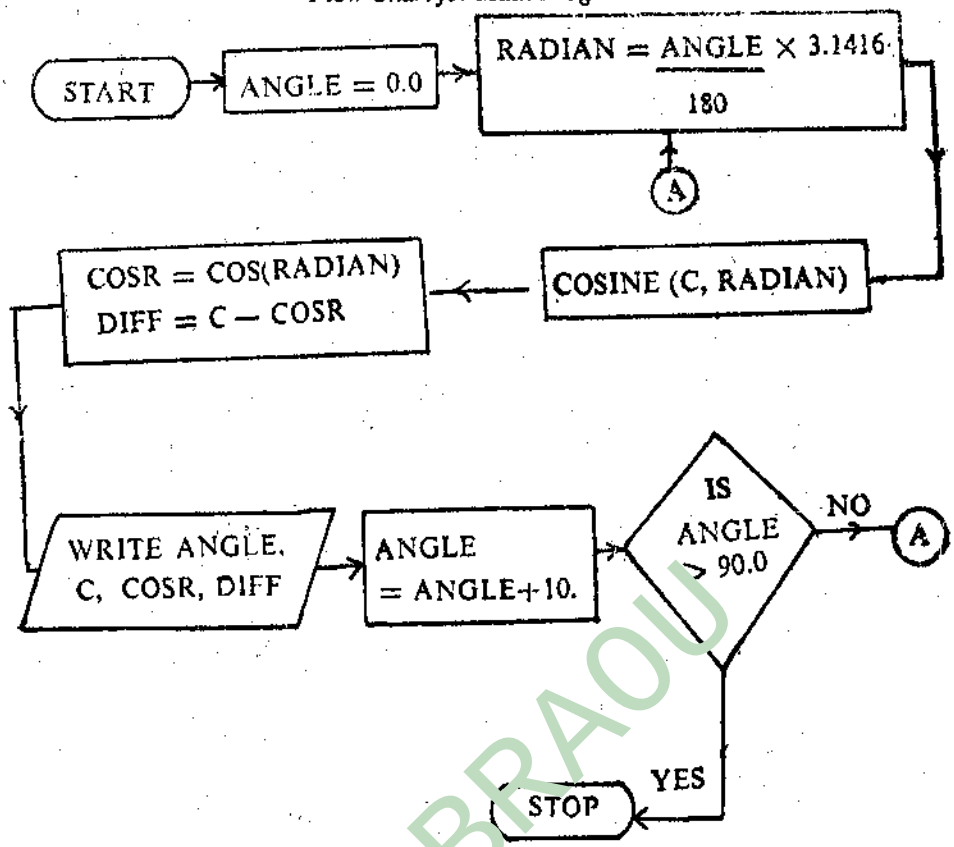
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots - \frac{x^{10}}{10!}$$

Write a program to obtain the difference between the cosine value obtained from the above formula and the value from library function and print the results.

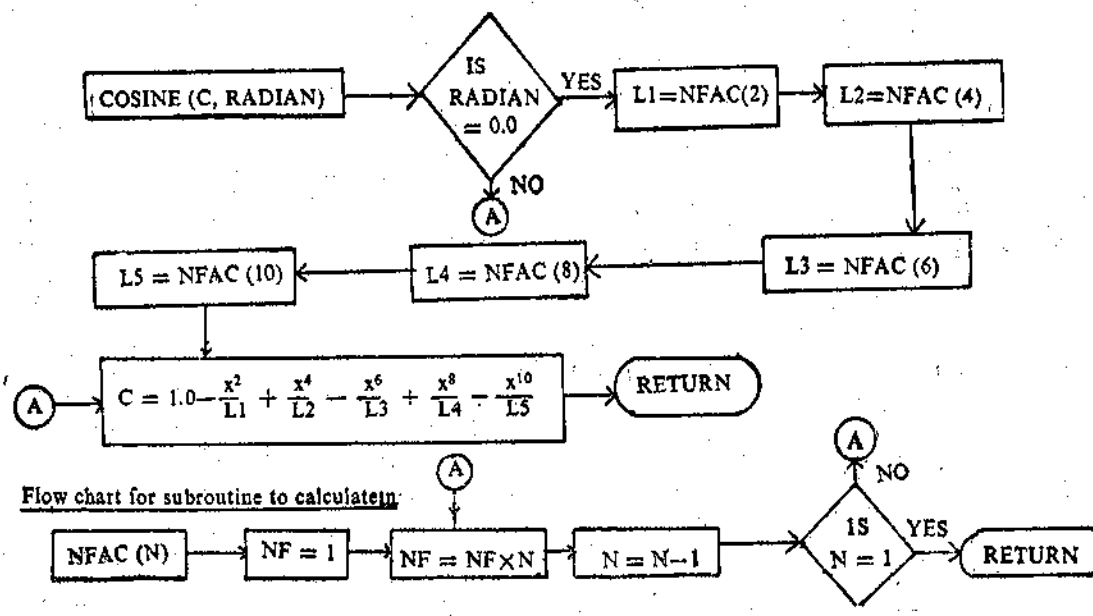
Solution

First we shall prepare the flow charts for i) main program ii) subprogram to calculate cos x and iii) subroutine to calculate the factorial of the integer.

Flow Chart for Main Program



Flow Chart for COSINE FUNCTION Subprogram



Flow chart for subroutine to calculatein

COSINE (C, RADIAN) is the FUNCTION subprogram to calculate cos x using the given expansion and NFAC (N) is the SUBROUTINE to calculate $n!$ It can be observed that the COSINE subprogram uses the Fractional subroutine.

Now we shall write the Main program and the two subprograms.

```
C       MAIN PROGRAM TO CALCULATE COSINE OF ANGLE
        ANGLE = 0.0
    5   RADIAN = ANGLE * 3.1416/180.
        COSR =  COS (RADIAN)
        DIFF = COS (C, RADIAN) – COSR
        WRITE (4, 100) ANGLE, COSR, C, DIFF
  100   FORMAT (F10.2, 3F15.4)
        ANGLE = ANGLE + 10.0
        IF (ANGLE . GT . 90.0) GO TO 200
        GO TO 5
  300   STOP
        END
C       COSINE FUNCTION SUBPROGRAM
        DIMENSION  L(5)
        IF (RADIAN . NE. 0.0) GO TO 40
        CALL NFAC (L)
   40   X = RADIAN
        X 2 = X * X
        X 4 = X 2 * X 2
        X 6 = X 4 * X 2
        X 8 = X 4 * X 4
        X 10 = X6 * X 4
        C = 1.0 –X 2/L (1) + X 4/L (2) – X6/L (3) + X8/L (4) – X10/L (5)
        RETURN
        END
```

```
C       SUB ROUTINE NFAC (NR)
        DIMENSION NR (5)
        DO 6 I = 1, 5
        NR(I) = 1
        N = I * 2
8       NR (I) = NR (I) * N
        N = N – 1
        IF (N – 1) 6, 6, 8
6       CONTINUE
        RETURN
        END
```

18.7 SUMMARY

We have introduced the concept of dividing a large program into several subprograms. Depending upon the problem we make use of arithmetic statement function, subprogram or subroutine. If the values of a single valued function is to be computed by a single arithmetic statement, we make use of an arithmetic statement function. If the values of a single valued functions to be computed by using several arithmetic statements we make use of a function subprogram. In the case of a multivalued function to be evaluated we make use of subroutine. All the arithmetic function statements have to be defined prior to any executable statement in a function. The Function subprogram will be called into the main program as and when the name of the subprogram appears. A subroutine is called by the main program by a CALL statement.

18.8 SAMPLE EXAMINATION QUESTIONS

I. *Answer the following questions in detail.*

i) a) Define a FUNCTION subprogram. How is the result calculated by the subprogram is tranferred to the main program?

b) Write a program to compute

$$A = 10 + f(x - y)$$

$$B = \frac{f(x) + f\left(x^2 + y^2\right)}{6.72\, f\left(x^2\right)}$$

$$C = \frac{f(\sin x) - \cos x}{x^3 + 2x \cos x + \sqrt{1 + f(\sin x)}}$$

ii) a) What are the rules to be observed while writing a subprogram?

b) Write a FUNCTION subprogram to compute

$$f(x) = 2 + \frac{x}{\sqrt{1 + x^2}} \quad \text{for } x < 0, f(0) = 2$$

$$f(x) = 2 - \frac{x}{\sqrt{1 + x^2}} \quad \text{for } x > 0$$

iii) a) Explain Subroutine subprogram and give the rules to be observed while using it.

b) $P(x_1, y_1, z_1)$ and $Q(x_2, y_2, z_2)$ are two points. Write a subroutine to compute distance PQ and direction cosines of PQ.

iv) a) Explain the common points and differences between FUNCTION and SUBROUTINE subprograms.

b) Write a subroutine to compute the inverse of a given matrix.

II. *Briefly answer the following*

i) Explain 'Arithmetic function statement'. Give an example and use it in a program.

ii) Write an arithmatic function statement to compute $f(x) = x^2 + \sqrt{2x + 3x^2 + 4x^3}$ and write a program which uses this function to compute and write

$$A = \frac{8.6 + y^2}{y^4 \sqrt{2y + 3y^2 + 4y^3}}$$

iii) Write a Subroutine to multiply $A_{n \times m}$ and $B_{m \times n}$ matrices.

iv) $A_{m \times n}$ and $B_{m \times n}$ are two matrices. Write a Subroutine to compute A + B and A – B.

v) Write a subroutine to compute A, B and C where

$$A = x^2 + y^2 + \sin x$$

$$B = x \sin y + \cos x^2$$

$$C = y^3 + x^2 \cos x$$

REFERRENCES

1. **V. Rajaraman**

   Principles of Computer Programming
   Prentice - Hall of India Private Limited, New Delhi

2. **V. Rajaraman**

   Computer Oriented Numerical Methods
   Prentice - Hall of India Private Limite, New Delhi.

3. **K.D. Sarma**

   Programming in Fortran – IV
   Affiliated East – West Press Pvt. Ltd., New Delhi

4. **D.D. Mc. Cracken, W.S. Dorn**

   Numerical methods and Fortran Programming
   John Wiley & Son's, New York.

5. **G.B. Davis**

   Computer Data Processing
   Mc. Graw Hill.

MATHEMATICS - COURSE - IV

ASSIGNMENT - I

SECTION - A

1. Define interpolation. Derive Newton's divided difference formula for interpolation and obtain a polynomial approximaton for the data $f(0) = 1, f(1) = 3$ and $f(3) = 55$.

2. Define the operators δ and μ. Derive the Bessel's formula for interpolating to halves. Obtain y_{25} from the data $y_{20} = 2854$, $y_{24} = 3162$, $y_{28} = 3544$ and $y_{32} = 3992$.

3. Explain the method of Least squares approximation, and obtain the normal equations for a parabola. Fit a parabola for the following data.

x :	2	4	6	8	10
y :	3.07	12.85	31.47	57.38	91.29

SECTION – B

1. Find the Function whose first difference is $9x^2 + 11x + 5$

2. Use Lagrange interpolation formula to express $\frac{3x^2 + x + 1}{(x-1)(x-2)(x-3)}$ in partial fractions.

3. Determine the accuracy of interpolation by Lagrange's formula for $\log_{10} 47$ give that

x :	40	42	45	48	49	50
$\log_{10} x$:	1.620600	1.6232493	1.6532126	1.6812413	1.6901960	1.9897000

MATHEMATICS - COURSE - IV

ASSIGNMENT - II

SECTION - A

1. Explain Newton's Raphson method for finding a real root of the equation $f(x) = 0$. Apply this method to find a real root of $2x - 3\sin x - 5 = 0$.

2. Derive Simpson's 3/8th rule for Numerical Integration obtain an approximation to π by applying the rule to $\int_0^1 \frac{dx}{1+x^2}$.

3. Explain the method to solve the initial value problem $\frac{dy}{dx} = f(x, y)$; $y = y_0$ at $x = x_0$, using Taylor series method. Using this method solve $\frac{dy}{dx} = x - y^2$ given that $y(0) = 1$.

SECTION - B

1. Explain inverse interpolation. Obtain the cube root of 10, from the following data, by using the method of itegration

x :	2	3	4	5
y :	8	27	64	125

2. Solve the difference equation $u_{n+1} - 6u_{n+1} - 8u_n = 2^n + 6n$.

3. Use Euler transformation to find the sum of series $1 - \frac{1}{2^3} + \frac{1}{3^3} - \frac{1}{4^3} + \ldots\ldots$ upto 6 decimal places.

MATHEMATICS - COURSE - IV

ASSIGNMENT - III

SECTION - A

1. Explain the importance of flow charts in problem solving. Write a flow chart to compute $n!$.

2. Write a program, using suitable format specifications to evaluate E and V using the following formulae and output the results.

   $E = \frac{1}{2} CQ^2$ for C = 0.0000 1 and Q = 0.0025

   V = RT/P for R = 15.7, T = 12.87 and P = 156.05.

3. Explain the difference between function subprogram and subroutine subprogram.

   Write a subroutine to compute A, B, C where $A = x^2 + y^2 + \sin x$, $B = x \sin y + \cos x^2$;

   $C = y^2 + x^2 \cos x$.

SECTION – B

1. Explain the hierachy of operations in Forfran. Write the fortran statements for the following :

   (i) $v = \tan \theta + \log_e (\cos x + \sin y)$

   (ii) $x = r \cos \theta \sin \phi$

   (iii) $r = \frac{16\pi r}{\pi d^3} \left(1 + \frac{d}{4r}\right)$

2. What is the decimal equivalent of 35044.34_6 ?

3. Using Do statements write a program to obtain the product of the matrices $A_{3\times4} \times B_{4\times5}$.

FACULTY OF SCIENCE

B.SC. III YEAR (3 YEAR DEGREE COURSE) EXAMINATION - 1992

MATHEMATICS - COURSE - IV

(NUMERICAL ANALYSIS AND PRINCIPLES OF COMPUTER PROGRAMMING - FORTRAN IV)

Time : 3 hrs

Max. Marks : 100
Min Marks : 35

SECTION – A

Answer any Four Questions. $4 \times 15 = 60$

1. Explain interpolation and derive the Newton's formual for forward interpolation. Using this formula, find a cubic polynomial $f(x)$ which takes on the values $f(0) = -5, f(1) = 1, f(2) = 9, f(3) = 25, f(4) = 55, f(5) = 105$ and estimate $f(3.2)$

2. Obtain the Bessel's interpolation formula for central differences. The following table contain values of the function $y = x^4 + 10x^5$ for certain values of x. Find y when $x = 2.27$ by using Bessel's formula.

x :	2.0	2.1	2.2	2.3	2.4	2.5
y :	336.0000	427.8582	5[illegible]88	671.6184	824.4400	1015.6250

3. Solve the following system of equations by Jacobi method :

$13x_1 + 5x_2 - 3x_3 + x_4 = 18$

$2x_1 + 12x_2 + x_3 + 4x_4 = 13$

$3x_1 - 4x_2 + 10x_3 + x_4 = 29$

$2x_1 + x_2 - 3x_3 + 9x_4 = 31.$

4. Derive the general quadrature formula and hence the trapezoidal rule. Use this rule to evaluate $\int_0^2 (1 + x^2)\, dx$ numerically.

5. Given $\frac{dy}{dx} = \frac{y - x}{y + x}$, $y(0) = 1$, find $y(0.1)$ using (i) Imported Euler's method and (ii) Modified Euler's method.

6. What is the working principle of a computer? Explain the importance of computer in problem solving by citing some examples. Write a flow chart for picking the largest of three given numbers.

7. Mention three important control statements in fortran. Write their general forms and explain. Wirte a program to read and obtain the sum of the matrices $A_{m\times n}$, $B_{m\times n}$ and $C_{m\times n}$.

8. Explain the difference between a function subprogram and a subroutine subprogram. Write a subroutine to evaluate the product of two 20×20 matrices. Write a main program which calls this subroutine and prints the product of matrices.

SECTION – B

Answer any five of the following.

9. Represent the function $f(x) = x^4 - 12x^3 + 24x^2 - 30x + 9$ and its successive differences in factorial notation. Find the function whose first difference is $9x^2 + 11x + 5$.

10. Define the divided difference operator and apply Newton's divided difference formula to obtain a polynomial $f(x)$ such that $f(0) = 1$ $f(1) = 3$ and $f(3) = 55$.

11. Use subtabulation to compute u_x for $x = 30\ (1)\ 35$ from the following data :

x :	30	35	40	45	50
u_x :	0.86603	0.81915	0.76604	0.70711	0.64279

12. Find the least squares approximation of second degree for the data :

x :	–2	–1	0	1	2
$f(x)$:	15	1	1	3	19

13. Use the method of interation to find a real root of $2x - \log_{10} x = 7$.

14. Solve $y_{n+2} - 2 \cos \alpha\ y_{n+1} + y_n = \cos \alpha_n$.

15. Use Euler – Maclaurin formula to evaluate $\sum_{x=1}^{n} x^2$.

16. Solve $\frac{dy}{dx} = x + y$, $y(0) = 1$ for $x = 2.0, 2.5$ using Milne's method.

17. Explain the hierarchy of operations in Fortran language.

Each of the following statements contain atleast one error. Idetify them :

i) Z = X * Y – 15, 200 ii) X * Y = Y * X

iii) 24 = I + J/5 – K iv) VEL = 3 (I – J)

v) 2R = PNR/100

18. i) Convert 634.640625 to base 8.

ii) Find the decimal equivalent of $4B3F5.A9_{16}$.