GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru – 521 356.

Department of Electrical and Electronics Engineering

HANDOUT

On

NUMERICAL METHODS WITH COMPUTER APPLICATIONS

Vision

To be a pioneer in electrical and electronics engineering education and research, preparing students for higher levels of intellectual attainment, and making significant contributions to profession and society.

Mission

- To impart quality education in electrical and electronics engineering in dynamic learning environment and strive continuously for the interest of stake holders, industry and society.
- To create an environment conducive to student-centered learning and collaborative research.
- To provide students with knowledge, technical skills, and values to excel as engineers and leaders in their profession.

Program Educational Objectives

1. Graduates will have technical knowledge, skills and competence to identify, comprehend and solve problems of industry and society.
2. Graduates learn and adapt themselves to the constantly evolving technology to pursue higher studies and undertake research.
3. Graduates will engage in lifelong learning and work successfully in teams with professional, ethical and administrative acumen to handle critical situations.

HANDOUT ON NUMERICAL METHODS WITH COMPUTER APPLICATIONS

Class & Sem. : II B.Tech – I Semester Year: 2018-19
Branch : EEE Credits : 4

==

1. Brief History and Scope of the Subject

As the long history of numerical techniques indicates, numerical analysis does not require any particular computer resources. On the other hand, the scale and complexity of the problems that can be solved in a particular manner are strongly influenced by the availability of high-speed computers and efficient software implementation of numerical algorithms. Understanding the basic numerical methods and the issues involved in designing and analyzing them is a first step in the use of these techniques for real-world problems. Implementing the appropriate method, or using existing software, or a combination of the two approaches, is necessary to achieve the final goal.

MATLAB has been chosen as the programming environment for the presentation of numerical techniques for two reasons. First, MATLAB provides outstanding graphing and programming capabilities, together with the ability to solve many types of problems symbolically as well. Second, MATLAB's underlying matrix structure makes the software especially useful for focusing on the aspects of various numerical techniques that can be described conveniently in vector form. Because, vectorization is an important approach to parallel computing.

2. Pre-Requisites

Basic Knowledge of linear algebra, and calculus

3. Course Objectives:

- To introduce the various numerical techniques using MATLAB
- To be aware of different methods to solve first order differential equations
- To construct a curve for the given data.

4. Course Outcomes:

Upon successful completion of the course ,the students will be able to

CO1: Demonstrate various commands in MATLAB programming.

CO2: Analyse a mathematical problem and select a suitable numerical technique to implement it in MATLAB programming.

CO3: Construct an interpolating polynomial for the given data using MATLAB.

CO4: Find derivatives and integrals by using numerical techniques using MATLAB.

CO5: Utilize method of least squares to fit a curve for the given data using MATLAB.

5. Program Outcomes:

The graduates of electronics and communication engineering program will be able to

a) apply knowledge of mathematics, science, and engineering for solving intricate engineering problems.
b) identify, formulate and analyze multifaceted engineering problems.
c) design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
d) design and conduct experiments based on complex engineering problems, as well as to analyze and interpret data.
e) use the techniques, skills, and modern engineering tools necessary for engineering practice.
f) understand the impact of engineering solutions in a global, economic and societal context.
g) design and develop eco-friendly systems, making optimal utilization of available natural resources.
h) understand professional ethics and responsibilities.
i) work as a member and leader in a team in multidisciplinary environment.
j) communicate effectively.
k) manage the projects keeping in view the economical and societal considerations.
l) recognize the need for adapting to technological changes and engage in life-long learning.

6. Mapping of Course Outcomes with Program Outcomes:

	a	B	c	d	e	f	g	h	i	j	k	l
CO1	M	H		H	H							H
CO2	H	H		H	H							
CO3	H	H		H	H							
CO4	H	H		H	H							
CO5	H	H		H	H							

7. Prescribed Text Books

1. Laurene V. Fausett ,Applied numerical analysis using MATLAB: 2nd edition, Pearson publications, 2012, NewDelhi
2. B.S.Grewal, Higher Engineering Mathematics : 42nd edition, Khanna Publishers, 2012 , New Delhi.
3. B.V Ramana, Higher Engineering Mathematics, Tata-Mc Graw Hill Company Ltd.

8. Reference Text Books

1. Erwin Kreyszig, Advanced Engineering Mathematics : 8th edition, Maitrey Printech Pvt. Ltd, 2009, Noida.
2. Robert J.Schilling, SandrabL .Harries, Applied Numerical methods for engineers using MATLAB & C, Thomson books.
3. John.H.Mathews, kurtis D.Fink, Numerical methods using MATLAB, 4th edition-PHI.
4. Steven C.Chapra, Raymond P.Canale, Numerical methods for Engineers, 3rd Edition, TATA McGrawhill, 2000, NewDelhi.

9. URLs and Other E-Learning Resources.

Sonet CDs & IIT CDs on some of the topics are available in the digital library.

10. Digital Learning Materials:

https://youtu.be/R5eSBMP3XAM?list=PLEJxKK7AcSEFtiPAlDTJywUIAIF8FGYXA
https://youtu.be/6F89wOGgRf0?list=PLTwPa5Tfu7AULpYLzEGS5c2SFyuD5sOFt
https://www.youtube.com/watch?v=sZ_nCZjokQs
https://www.youtube.com/watch?v=fCKUOWiM-6s
https://www.youtube.com/watch?v=-QoZcEoGDEQ
https://www.youtube.com/watch?v=NZfd-EuBYyo

https://www.youtube.com/watch?v=ElEqbKICvEs

11. Lecture Schedule / Lesson Plan

S.No	Topics Covered	Periods	Tutorial
UNIT-I: Introduction to MATLAB			
1	Introduction and MATLAB Environment	1	1
2	Basic Commands	1	
3	Variables	1	
4	Arithmetic operations	1	
5	One dimensional array-vectors and operations on vectors	1	
6	Two dimensional array-matrices and operations on matrices	1	1
7	Scripts and Functions	1	
8	2D-Plotting	1	
UNIT-II: Algebraic and Transcendental Equations			
9	Introduction and interval calculation	1	1
10	Bisection Method	2	
11	Method of False Position	2	
12	Newton- Raphson Method.	2	1
UNIT-III : Interpolation			
13	Introduction	1	1
14	Finite differences	1	
15	Construction of difference tables & problems	2	
16	Newton's Forward Difference formula for interpolation	2	1
17	Newton's Backward difference formula for	2	

	interpolation		
18	Gauss Forward Difference formula for interpolation	2	1
19	Gauss Backward difference formula for interpolation	2	
20	Lagrange's Interpolation formula	2	
UNIT-IV: Numerical differentiation and integration			
21	Introduction to Numerical differentiation and formulae	1	1
22	Numerical differentiation by Newton's Forward differences	1	
23	Numerical differentiation by Newton's Backward differences	1	
24	Numerical Integration by Trapezoidal rule	1	1
25	Numerical Integration by Simpson's $1/3^{rd}$ rule	1	
26	Numerical Integration by Simpson's $3/8^{th}$ rule	1	
UNIT-V: Numerical solution of Ordinary Differential equations			
27	Introduction	1	1
28	Taylor's series Method	2	
29	Euler's Method	1	
30	Modified Euler's Method	1	1
31	Runge-Kutta Methods of 4^{th} order	2	
UNIT-VI: Curve Fitting			
32	Introduction and Method of least squares	1	1
33	Fitting of a straight line	1	
34	Fitting of a Parabolic curve	1	
35	Fitting of an exponential curve $y = ae^{bx}$	1	1
36	Fitting of an exponential curve $y = ab^{x}$	1	
37	Fitting of a power curve	1	
TOTAL		**48**	**13**

12. Seminar Topics:

1. Newton – Raphson method
2. Lagranges Interpolation formula
3. Trapezoidal, Simpson 1/3 rule.
4. Derive Normal equations to fit a parabola by Method of least squares.

13. List of Lab Experiments

S.No	Title of Lab Experiments	Periods
1	MATLAB Programs - function programs – Expressions - Array Operations.	2
2	To find out the root of the Algebraic and Transcendental equations using (i).Bisection, (ii). Regula – falsi method	2
3	To find out the root of the Algebraic and Transcendental equations using Newton – Raphson method.	2
4	To implement Newton's Forward and Backward Interpolation formula.	2
5	To implement Gauss Forward and Backward interpolation formula.	2
6	To implement Lagranges Interpolation formula.	2
7	To implement Numerical Differentiations Newton's interpolation formulae.	2
8	To implement Numerical Integration using Trapezoidal, Simpson 1/3 rule.	2
9	To find out the Numerical solution of ordinary differential equations using Euler's Method	2
10	To find out the Numerical solution of ordinary differential equations using R-K Method of fourth order.	2
11	To implement Least Square Method to fit a Straight line and Parabolic curve.	2
12	To implement Least Square Method to fit an exponential curve and power curve.	2
	TOTAL	24

NUMERICAL METHODS WITH COMPUTER APPLICATIONS

UNIT– I: Introduction to MATLAB

Pre-requisite : Basics of Linear algebra

Syllabus: Variables - Arrays: Vectors & Matrices - Array Operations – functions - plots in MATLAB

Course Objectives:

- To familiarize with the environment of MATLAB and its components.
- To introduce various data types and variables in MATLAB.
- To distinguish various aspects of MATLAB scalars, arrays-vectors, matrices.
- To get acquainted with the built-in functions on scalars, arrays-vectors, matrices.
- To learn how to create and execute script and function files in MATLAB.

- To learn about plot functions using MATLAB.

Course Outcomes:

The students are able to

- Launch a new MATLAB session, work on desktop environment and terminate the session
- Use various commands in MATLAB
- Create variables, scalars, arrays-vectors and matrices
- Use various built-in functions on different types of data
- Plot graphs using MATLAB commands with different style options
- Create and execute script and function files

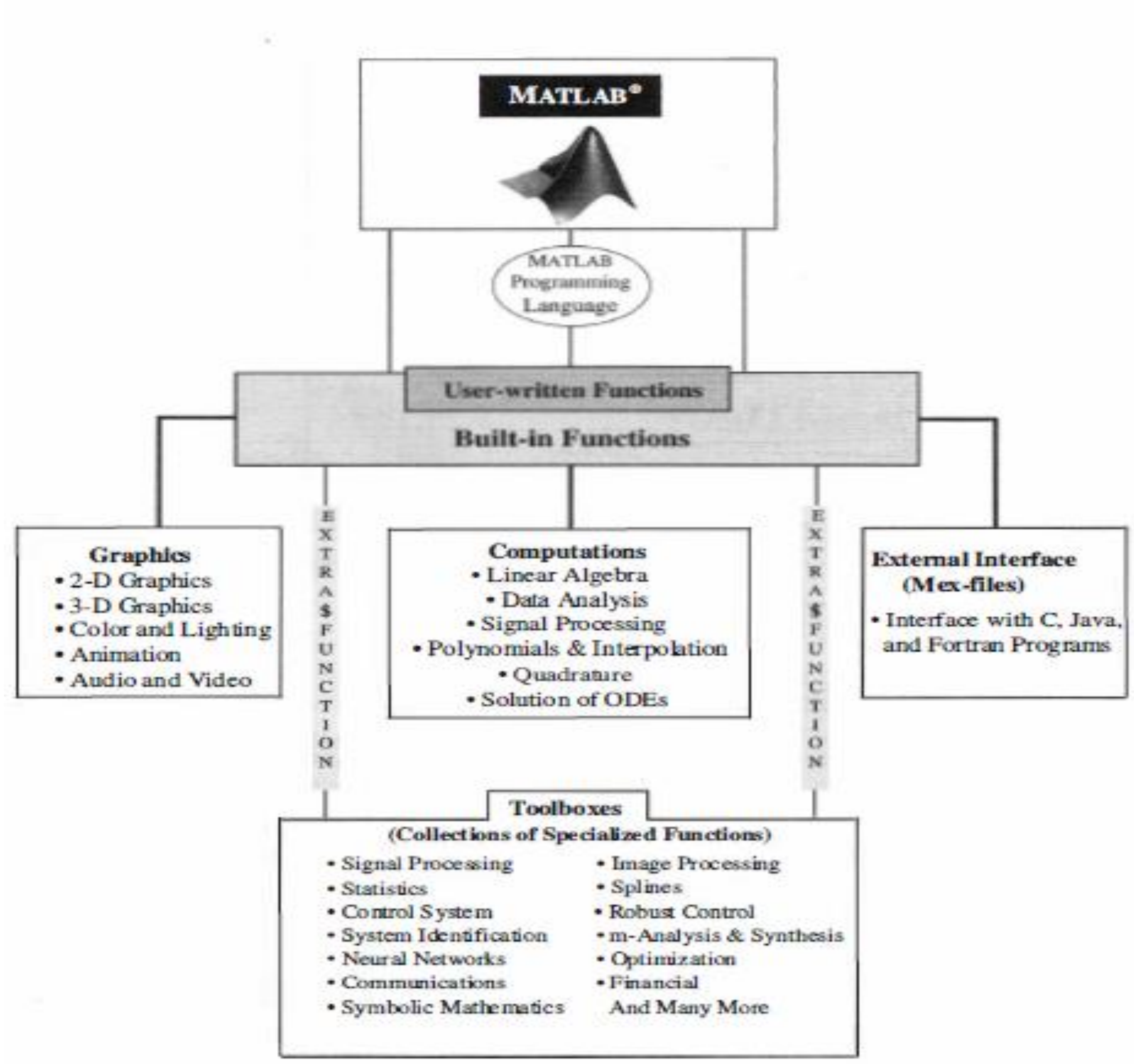
Introduction:

- The name **MATLAB** stands for **MAT**rix **LAB**oratory
- Developed primarily by **Cleve Moler**, Chairman of the Computer Science Department at the University of New Mexica in the 1970's
- Gained its popularity through word of mouth, because it was not officially distributed.
- The MathWorks Inc., Natick, Massachusetts, USA was created (1984) to market and continue development of MATLAB.

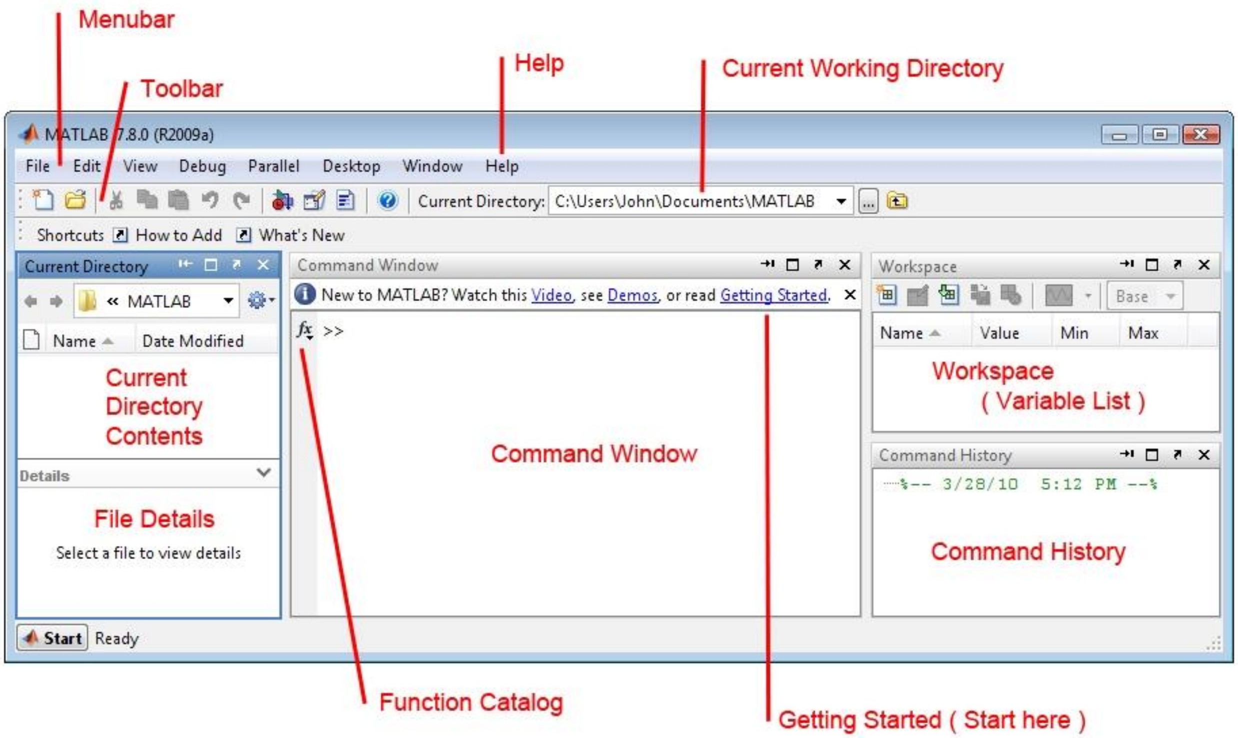
Features of MATLAB

- MATLAB is a high level interpreted programming language with interactive environment provides vast library of mathematical functions for numerical computation, visualization and application development.
- It is a case sensitive language.

Schematic diagram of MATLAB's main features



MATLAB Work Environment:



Four Important Windows

- The **Command Window** is where we type in Matlab commands.
- The **Command History Window** shows the commands we have entered in the past. We can repeat any of these commands by double-clicking on them, or by dragging them from the Command History Window into the Command Window. We can also scroll back to previous commands by using the up arrow in the Command Window.
- The **Workspace** shows the list of variables that are currently defined, and what type of variable each is. (i.e. a simple scalar, a vector, or a matrix, and the size of all arrays.) Depending on the size (i.e. type) of the variable, its value may also be shown.
- The **Current Directory Window** shows the contents of the current working directory.

Creating MATLAB variables

- Variable is a name made of a letter or a combination of several letters that is assigned a value or an expression.

- MATLAB variables are created with an assignment statement.

- The syntax of variable assignment is
  variable_name = a value (or an expression)

  For example,

  >> x =2

Where expression can involve:

- manual entry
- built-in functions
- user-defined functions

Rules for variable names

- ✓ A valid variable name starts with a letter, followed by letters, digits, or underscores.
- ✓ MATLAB is case sensitive, so, A and a are *not* the same variable.
- ✓ We cannot define variables with the same names as MATLAB keywords, such as if or end.
- ✓ For a complete list, run the ***iskeyword*** command.

Examples of valid names:	Invalid names:
x6	6x
lastValue	end
n_factorial	n!

MATLAB System Variables

MATLAB has certain variables that are recognized by MATLAB itself and not defined by the users.

Variable	Description
ans	This variable is automatically generated by MATLAB when there is no variable assigned to store the result of an expression
inf	It represents infinity, generated usually when a number is divided by zero
eps	This is a constant value representing the floating point relative accuracy uses in its calculations
NaN	It represents Not a Number. Resulting from operations like 0/0 and inf / inf
pi	This represents the constant value of π = 3.14159…
i	As the basic imaginary unit sqrt(-1), i is used to enter complex

	numbers Ex.: Z = 2+3i
j	Use the character j in place of the character i, if desired, as the imaginary unit.

MATLAB Data types

MATLAB, as a computing language, recognizes different types of data.

Scalars :

- Any number that represents magnitude (quantity or measure) is known as scalar.
- This includes integers, complex number and floating point numbers
- For example : 5, -7, 4 + 5i, -45.18 etc.

Characters :

- The character constant is an alphanumeric symbol enclosed in a single quote.
- The character that is not represented in single quote is numeric or sign.
- For example : 'H', '4', '*', '+' etc.

Arrays :

- An array is a list of homogenous data placed in rows and/or columns form
- The elements of an array can be numeric or character or strings, but not mixed up.
- An array is written using square brackets enclosing its elements separated by commas or spaces
- For Example : [4, 8, 3, -5], [cleve, moler] etc.

Strings :

- Any two or more alphanumeric symbols enclosed in a single quote is known as string data
- For example : 'INDIA', 'MATLAB' etc.

– A string is an array of characters, i.e., 'INDIA' is equivalent to ['I', 'N', 'D', 'I', 'A']

Cell arrays :

– A cell array is a special type of arrays, where the elements can be of different types

– The elements of a cell array are enclosed using braces

– For example : {'south', 544, '+', 'book'}

COMMAND HANDLING

Common System Commands

Command	**Description**	**Purpose / Action**
>> *clc*	Clear command window	Clears the command window. All the variables still appear in the workspace
>> *delete* filename or *delete* ('file name')	Delete any undesired files	The file is deleted from the current directory
>> *cd* pathname or *cd* ('pathname')	Change the directory	Used to the change the working directory
>> *copyfile* ('source file', 'destination file')	Copying file	Used to copy a file from source to destination
>> *dir* directory_name *dir* ('directory_name')	Directory	Used to list the name of subdirectories and files under a directory
>> *date*	Date	Displays current date
>> *save*		Saves workspace variables in a file
>> *load*		Loads workspace variables from file
>> *type*		Displays contents of a file

Workspace Commands

Command	Description	Purpose / Action
>> *clear*	Clears all variables	It deletes all variables from the current directory
>> *clear x y z*	Clears specific variables	It deletes specific variables from the current directory
>> *who*		Used to list out variables used in current workspace window.
>> *whos*		Used to list out variables along with their size
>> *help*		Used to get help for MATLAB related function. >> *help* sqrt SQRT(X) is the square root of the elements of X. Complex results are produced if X is not positive.
>> *quit/exit*		Stops MATLAB
>> *exist*		Checks for existence of a file or variable

SCALARS

- In MATLAB a scalar is a variable with one row and one column.
- Scalars are the simple variables that we use and manipulate in simple algebraic equations.

Creating scalars

To create a scalar you simply introduce it on the left hand side of an equal sign.

```
>> x = 1;
>> y = 2;
>> z = x + y;
```

Scalar operations

MATLAB supports the standard scalar operations like addition, subtraction, multiplication and division.

```
>> u = 5;
>> v = 3;
```

Operation	Description	Example
Addition	Performs addition on two scalar values	>> w=u+v w = 8
Subtraction	Performs subtraction on two scalar values	>> w=u-v w = 2
Multiplication	Performs multiplication on two scalar values	>> w=u*v w = 15
Division	Performs Division on two scalar values	>> w=u/v w = 1.6667
Exponentiation	Performs Exponentiation on two scalar values	>> w=u^v w =125

Some Built-in Scalar Functions:

Certain MATLAB functions are essentially used on scalars

Function	Syntax	Description	Example
Sin	sin(x)	trigonometric sine	>> sin(pi/2) ans = 1
Cos	cos(x)	trigonometric cosine	>> cos(pi/2) ans = 6.1230e-017
Tan	tan(x)	trigonometric tangent	>> tan(pi/4) ans = 1
Asin	asin(x)	trigonometric inverse sine (arcsine)	>>asin(pi/4) ans = 0.9033
Acos	acos(x)	trigonometric inverse cosine (arccosine)	>> acos(pi/4) ans = 0.6675
Atan	atan(x)	trigonometric inverse tangent (arctangent)	>> atan(pi/4) ans = 0.6658
Exp	exp(x)	Exponential (e^x)	>>exp(2) ans = 7.3891
Log	log(x)	natural logarithm	>> log(2) ans = 0.6931
Abs	abs(x)	absolute value	>> abs(-2) ans = 2
Sqrt	sqrt(x)	square root	>>sqrt(16) ans = 4
Rem	rem(x,y)	remainder	>> rem(12,5) ans = 2

Round	round(x)	round towards nearest integer	>> round(5.45) ans = 5 >> round(5.75) ans = 6
Floor	floor(x)	round towards negative infinity	>> floor(5.45) ans = 5 >> floor(5.75) ans = 5
Ceil	ceil(x)	round towards positive infinity	>> ceil(5.45) ans = 6 >> floor(5.75) ans = 6

VECTORS

- In MATLAB a vector is a matrix with either one row or one column.

Creating vectors

1. using the built-in functions *ones*, *zeros*, *linspace*, and *logspace*
2. assigning a mathematical expressions involving vectors
3. appending elements to a scalar
4. using colon operator

Creating vectors with *ones, zeros, linspace*, and *logspace*

Function	Syntax	Description	Example
ones	**ones(1,n)** **ones(n,1)**	**creates a row vector of length n, filled with ones** **creates a column vector of length n, filled with ones**	**>> x=ones(1,5)** **x = 1 1 1 1 1** **>>x=ones(3,1)** **x = 1** **1** **1**

zeros	**zeros(1,n)** **zeros(n,1)**	**creates a row vector of length n, filled with zeros** **creates a column vector of length n, filled with zeros**	**>> x=zeros(1,5)** **x = 0 0 0 0 0** **>>x=zeros(4,1)** **x = 0** **0** **0** **0**
linspace	**linspace (begin, end, no. of elements)**	**creates a vector with linearly spaced elements starting from *begin* to *end.***	**>> x = linspace(1,5,5)** **x = 1 2 3 4 5** **>> x = linspace(1,5,2)** **x = 1 5**
logspace	**logspace (begin, end, no. of elements)**	**creates a vector with logarithmically spaced elements starting from *begin* to *end.***	**>> y = logspace(1,4,4)** **y = 10 100 1000 10000** **>> y = logspace(1,4,2)** **y = 10 10000**

Creating vectors with *Colon* (:) operator:

MATLAB colon (:) operator is often used in creation of vectors.

>> x = initial_value : increment : final_value

- If no increment is specified, MATLAB uses the default increment of 1

Examples:

>> x = 0:10:100
x = [0 10 20 30 40 50 60 70 80 90 100]

>> a = 0:pi/50:2*pi
a = [0 pi/50 2*pi/50 . . . . . .2*pi]

>> u = 3:10
u = [3 4 5 6 7 8 9 10]

- To create a column vector, append the transpose operator to the end of the vector-creating expression

```
>> y = (1:5)'

y =  1
     2
     3
     4
     5
```

- **Using colon operator to create a vector requires you to specify the increment, whereas using the linspace command requires you to specify the total number of elements.**

Assigning vector expressions to a vector

- Once a vector is created, it may be assigned to another vector.
- **The following statements create a row vector, x, and then copies the third through seventh elements of x into y.**

```
>> x = linspace(31,40,10);
>> y = x(3:7)
   y =  33    34    35    36    37
>> y(3)
   ans =  35
```

Addressing vector elements

- Individual elements of a vector can be addressed with a subscript.

Example :

```
>> x = linspace(11,15,5);
>> x(2)
 ans =  12
```

Increasing the size of a vector (or scalar)

We can increase the size of a vector simply by assigning a value to an element that has not been previously used.

```
>> x = linspace(21,25,5)
            x =  21    22    23    24    25

>> x(7) = -9
```

x = 21 22 23 24 25 0 -9

Vector Operations:

>> x = [1 2 3]   >> y = [3 4 6]

Operation	Description	Example
+	Addition of two vectors	>> Z=X+Y Z = 4 6 9
-	Subtraction of two vectors	>>Z=X-Y Z = -2 -2 -3
*	Multiplication of two vectors	>> Z=X * Y' Z = 29
.*	Element-by-element multiplication	>> x.*y ans = 3 8 18
. /	Element-by-element left division	>> x./y ans = 0.3333 0.5000 0.5000
. \	Element-by-element right division	>> x.\y ans = 3 2 2
.^	Element-by-element exponentiation	>> x.^y ans = 1 16

Some Built-in vector functions:

The following MATLAB functions operate on vectors and return a scalar value.

Let *z* be the following row vector.

>> z = [34, 5, 11, 90]

Function	Description	Result
Max	Largest component	>>max(z) ans = 90
Min	Smallest component	>>min(z) ans = 5
Length	length of a vector	>>length(z) ans = 4

Sort	sort in ascending order	>>sort(z) ans = [5,11,34,90]
Sum	sum of elements	>>sum(z) ans = 140
Prod	product of elements	>>prod(z) ans = 168300
median	median value	>>median(z) ans = 22.5000
Mean	mean value	>>mean(z) ans = 35
Std	Standard deviation	>>std(z) ans = 38.7384

MATRICES

Creating Matrices

Function	Description	Example	Result
eye	Identity matrix	» eye(3) » eye(3,4)	1 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0
zeros	matrix of zeros	» zeros(2) » zeros(2,3)	0 0 0 0 0 0 0 0 0 0
Ones	matrix of ones	» ones(2) » ones(2,3)	1 1 1 1 1 1 1 1 1 1
diag	extract diagonal of a matrix or create diagonal matrices	x = 1 3 4 4 5 7 5 9 0 >> diag(x)	 1 5 0
triu	upper triangular part of a matrix	>> triu(x)	1 3 4 0 5 7

			0 0 0
tril	lower triangular part of a matrix	>> tril(x)	1 0 0 4 5 0 5 9 0
rand	randomly generated matrix	>> rand(3)	0.9501 0.4860 0.4565 0.2311 0.8913 0.0185 0.6068 0.7621 0.8214

- The above (vector) commands can also be applied to a matrix.
- In this case, they act in a column-by-column fashion to produce a row vector containing the results of their application to each column.

Matrix operations

Example:

A = [1 2 3 ; 4 5 6; 7 8 9];
B = [7 5 6 ; 2 0 8; 5 7 1];

Operation	Operator	Description	Example
Addition	+	$c_{ij} = a_{ij} + b_{ij}$	>> C=A+B C = 8 7 9 6 5 14 12 15 10
Subtraction	-	$c_{ij} = a_{ij} - b_{ij}$	>> C=A-B C = -6 -3 -3 2 5 -2 2 1 8
Multiplication	*	$c_{ij} = \sum_{k=1}^{n} a_{ik} * b_{kj}$	>> C=A*B C = 26 26 25 68 62 70 110 98 115
Right division	/	C = A/B => A*inv(B)	>> C=A/B C = -0.5254 0.6864 0.6610 -0.4237 0.9407 1.0169 -0.3220 1.1949 1.3729
Left division	\	C = A\B => inv(A)*B	A = [4,5,3; 2,1,4; 3,2,6] B = [1,2,3; 4,5,6; 7,8,9] >>C = A\B C = -4.2000 -2.4000 -0.6000 2.0000 1.0000 -0.0000 2.6000 2.2000 1.8000
Exponentiation	^	$c_{ij} = \sum^{n} a_{ik} * a_{kj}$	>> C=A^2 C = 30 36 42

		k = 1	66 81 96 102 126 150
Element-by-element multiplication	.*	$c_{ij} = a_{ij} * b_{ij}$	>>C = A .* B C = [10, 40, 90 ; 160, 250, 360; 490, 640, 810]
Element-by-element left division	. /	$c_{ij} = a_{ij} / b_{ij}$	>>C = A ./ B C = [10, 10, 10; 10, 10, 10; 10, 10, 10]
Element-by-element exponentiation	.^	$c_{ij} = a_{ij}$ ^ x	>>C = A .^ 2 C = [1, 4, 9; 16, 25, 36; 49, 64, 81]

Some Built-in Matrix functions:

A = [1 3; 2 4]

Function	Description	Example	Result
Size	size of a matrix	>>Size(A)	ans = 2 2
det	Determinant of a square matrix	>>det(A)	ans = -2
inv	inverse of a matrix	>>inv(A)	ans = -2.0000 1.5000 1.0000 -0.5000
rank	rank of a matrix	>>rank(A)	ans = 2
eig	Eigen values and eigen vectors Produces a diagonal matrix 'd' with the eigen values on the main diagonal, and a full matrix 'x' whose columns are the corresponding eigenvectors.	>>eig(A) >>[x d]=eig(A)	ans = -0.3723 5.3723 x = -0.9094 -0.5658 0.4160 -0.8246 d = -0.3723 0 0 5.3723
max	Produces a row vector with maximum element in each column Maximum of all elements of matrix	>>max(A) >>max(max(A))	ans = 2 4 ans = 4

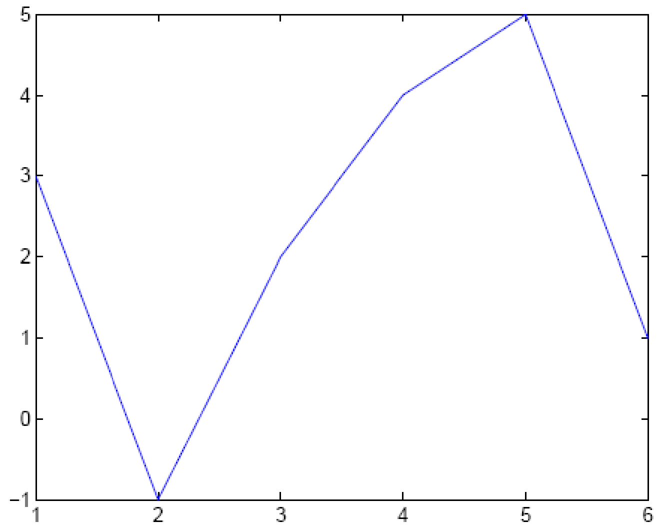
PLOTTING

- MATLAB has an excellent set of graphic tools.

- Plotting a given data set or the results of computation is possible with very few commands.

The MATLAB command to plot a graph is plot(x,y).

Example:
The vectors
x = (1; 2; 3; 4; 5; 6)
y = (3;-1; 2; 4; 5; 1)
```
>> x = [1 2 3 4 5 6];
>> y = [3 -1 2 4 5 1];
>> plot(x,y)
```

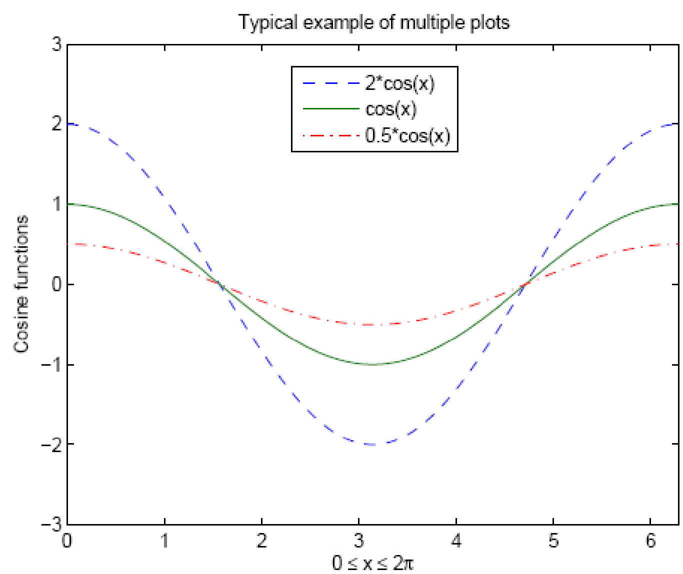


Adding titles, axis labels, and annotations:

Multiple data sets in one plot

Multiple (*x; y*) *pairs* arguments create *multiple* graphs with a single call to *plot*. For example,

```
>> x = 0:pi/100:2*pi;
>> y1 = 2*cos(x);
>> y2 = cos(x);
>> y3 = 0.5*cos(x);
>> plot(x,y1,'--',x,y2,'-',x,y3,':')
>> xlabel('0 \leq x \leq 2\pi')
>> ylabel('Cosine functions')
>>
legend('2*cos(x)','cos(x)','0.5*cos(x)')
>> title('Typical example of multiple
plots')
>> axis([0 2*pi -3 3])
```



Specifying line styles and colors

Using *plot* command, we can specify *line styles*, *colors*, and *markers* (e.g., circles, plus signs, . .)

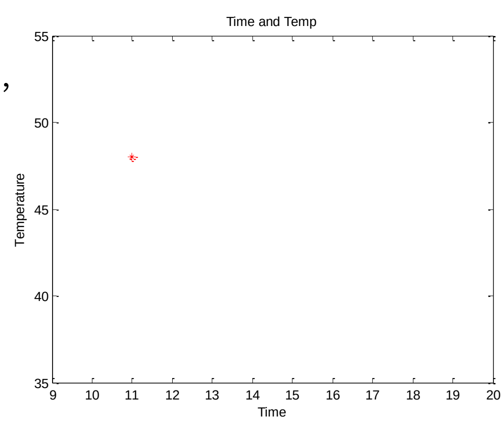
plot(x, y, 'style_color_marker')

where style_color_marker is a *triplet* of values

Symbol	Color	Symbol	Line Style	Symbol	Marker
k	Black	–	Solid	+	Plus sign
r	Red	– –	Dashed	o	Circle
b	Blue	:	Dotted	*	Asterisk
g	Green	–.	Dash-dot	.	Point
c	Cyan	none	No line	×	Cross
m	Magenta			s	Square
y	Yellow			d	Diamond

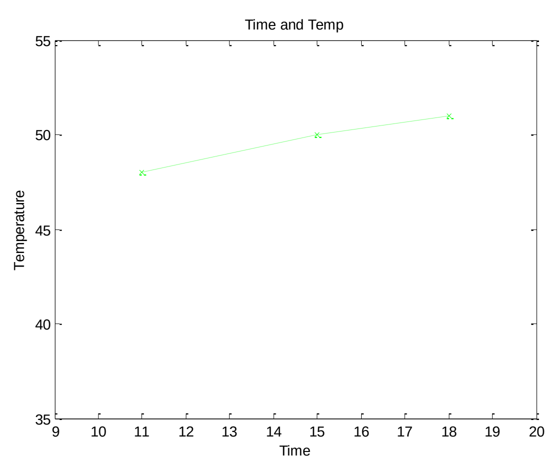
Example:

```
   % A simple plot of just one point!
   % Create coordinate variables and plot a red '*'
x = [11];
y = [48];
plot(x, y, 'r*')
        % Change the axes and label them
axis([9 20 35 55])
xlabel('Time')
ylabel('Temperature')
        % Put a title on the plot
title('Time and Temp')
```



Example:

```
x = [11,15,18];
y = [48,50,51];
plot(x, y, ':gx')
  % Change the axes and label them
axis([9 20 35 55])
```



```
xlabel('Time')
ylabel('Temperature')
  % Put a title on the plot
title('Time and Temp')
```

Introduction to programming in MATLAB

The commands entered in the Command Window cannot be saved and executed again for several times. To execute the commands repeatedly with MATLAB is:

1. to *create* a file with a list of commands,
2. *save* the file, and
3. *run* the file.

- These files are called script files or *scripts*
- They must have the file extension ".m"
- Corrections or changes can be made to the commands in the file
- There are two types of m-files: *script files* and *function files*.
- M-files can be *scripts* that simply execute a series of MATLAB statements, or they can be *functions* that can accept arguments and can produce one or more outputs.

Script files: A script file is an external file that contains a sequence of Matlab statements. By typing the filename, subsequent Matlab input is obtained from the file. Script files have a filename extension of .m and are often called M-files.

Example:

Consider the system of equations:

$$\begin{cases} x + 2y + 3z &= 1 \\ 3x + 3y + 4z &= 1 \\ 2x + 3y + 3z &= 2 \end{cases}$$

Find the solution *x* to the system of equations.

Solution:

- Use the MATLAB *editor* to create a file: File -> New -> M-file.
- Enter the following statements in the file:

```
A = [1 2 3; 3 3 4; 2 3 3];
b = [1; 1; 2];
x = A\b
```

- Save the file, for example, *example1.m*.

– Run the file, at the command prompt , by typing:

```
>> example1
    x =
       -0.5000
        1.5000
       -0.5000
```

Example:

Plot the following cosine functions, $y_1 = 2\cos(x)$, $y_2 = \cos(x)$, and $y_3 = 0.5 * \cos(x)$, in the interval $0 \leq x \leq 2\pi$. This example has been presented in previous Chapter. Here we put the commands in a file.

- Create a file, say `example2.m`, which contains the following commands:

```
x = 0:pi/100:2*pi;
y1 = 2*cos(x);
y2 = cos(x);
y3 = 0.5*cos(x);
plot(x,y1,'--',x,y2,'-',x,y3,':')
xlabel('0 \leq x \leq 2\pi')
ylabel('Cosine functions')
legend('2*cos(x)','cos(x)','0.5*cos(x)')
title('Typical example of multiple plots')
axis([0 2*pi -3 3])
```

- Run the file by typing `example2` in the Command Window.

Function files: Function file is a script file (M-file) that adds a function definition to Matlab's list of functions.

Syntax:

function [output variables] = function_name (input variables) ;

Example 1:

```
function y = myfunc (x)
        y = 2*x^2 - 3*x + 1;
end
```

Save this file as: myfunc.m in your working directory.

This file can now be used in the command window just like any predefined Matlab function. In the command window enter:

```
x = -2:.1:2;     % Produces a vector of x values
y = myfunc(x); % Produces a vector of y values
plot (x,y)
```

Example 2:

Functions can have multiple inputs, which are separated by commas. For example:

```
function y = myfunc2d (x,p)
        y = 2*x.^p - 3*x + 1;
end
```

- Functions can have multiple outputs, which are collected into a vector

```
function [x2 x3 x4] = mypowers (x)
        x2 = x .^2;
        x3 = x .^3;
        x4 = x .^4;
end
```

We can use the results of the program to make graphs:

```
x = -1:.1:1
[x2 x3 x4] = mypowers (x);
plot (x,x,'black',x,x2,'blue',x,x3 ,'green',x,x4,'red')
```

Difference between Script files and Function files:

Script files	Function files
Do not accept input arguments or return output arguments	Can accept input arguments and return output arguments
Store variables in a workspace that is shared with other scripts	Store variables in a workspace internal to the function
Are useful for automating a series of commands	Are useful for extending the MATLAB language for your application

UNIT-II : Algebraic and Transcendental Equations

Pre-requisite : Commands of MATLAB

Syllabus : Solution of Algebraic and Transcendental Equations-Introduction - Bisection Method - Method of False Position - Newton-Raphson Method.

Course Objectives:

- To recongnize the algebraic and Transcendental Equations.
- To Understand the Bisection method, method of False Position and Newton Raphson Method.
- To Implement Bisection, False Position and Newton Raphson Methods in MATLAB

Course Outcomes:

Students will be able to

- Solve an Algebraic and Transcendental equation using Numerical Methods
- Find the roots of non-linear equations using MATLAB programs.

Solutions of Algebraic and Transcendental equations

<u>Introduction</u> : A problem of great importance in science and engineering is that of determining the roots/ zeros of an equation of the form f(x) = 0

- <u>Polynomial function:</u> A function f(x) is said to be a polynomial function if f(x) is a polynomial in x.

  i.e. f(x) $= a_0 x^n + a_1 x^{n-1} + \dots\dots\dots\dots + a^{n-1} x + a_n$ where $a_0 \neq 0$,

  the coefficients $a_0, a_1 \dots\dots\dots\dots a_n$ are real constants and n is a non-negative integer.

- <u>Algebraic function:</u> A function which is a sum (or) difference (or) product of two polynomials is called an algebraic function. Otherwise, the function is called a transcendental (or) non-algebraic function.

  Eg: $f(x) = c_1 e^{x} + c_2 e^{-x}$

  $f(x) = e^{5x} - \frac{x^3}{2} + 3$

- Algebraic Equation: If f(x) is an algebraic function, then the equation f(x) =0 is called an algebraic equation.
- Transcendental Equation: An equation which contains polynomials,

exponential functions, logarithmic functions and Trigonometric functions etc. is called a Transcendental equation.

Ex:- $xe^{2x} - 1 = 0$, $\cos x - x e^{x} = 0$, $\tan x = x$ are transcendental equations.

- Root of an equation: A number α is called a root of an equation f(x) = 0 if f(α) =0. We also say that α is a zero of the function.

  Note: (1) The roots of an equation are the abscissas of the points where the graph y = f(x) cuts the x-axis.
  (2) A polynomial equation of degree n will have exactly n roots, real or complex, simple or multiple. A transcendental equation may have one root or infinite number of roots depending on the form of f(x).

Methods for solving the equation

Direct method:
We know the solution of the polynomial equations such as linear equation $ax + b = 0$ and quadratic equation $ax^2 + bx + c = 0$, will be obtained using direct methods or analytical methods. Analytical methods for the solution of cubic and quadratic equations are also well known to us.

There are no direct methods for solving higher degree algebraic equations or equations involving transcendental functions. Such equations are solved by numerical methods. In these methods we find an interval in which the root lies.

We use the following theorem of calculus to determine an initial approximation. It is also called the Intermediate value theorem.

Intermediate value theorem : If f(x)is continuous on some interval [a, b]and f(a)f(b)< 0, then the equation f(x) = 0 has at least one real root in the interval (a, b).

In this unit we will study some important methods of solving algebraic and transcendental equations.

Bisection method:
Bisection method is a simple iteration method to solve an equation. This method is also known as "Bolzano method of successive bisection". Sometimes it is referred to as "Half-interval method". Suppose we know an equation of the form $f(x) = 0$ has exactly one real root between two real numbers x_0, x_1. The number is chosen such that $f(x_0)$ and $f(x_1)$ will have opposite sign. Let us

bisect the interval $\left[x_0, x_1\right]$ into two half intervals and find the midpoint $x_2 = \frac{x_0 + x_1}{2}$. If $f\left(x_2\right) = 0$ then x_2 is a root.

If $f\left(x_1\right)$ and $f\left(x_2\right)$ have same sign then the root lies between x_0 and x_2. The interval is taken as $\left(x_0, x_2\right)$ Otherwise the root lies in the interval $\left[x_2, x_1\right]$. Repeating the process of bisection , we obtain successive subintervals which are smaller. At each iteration, we get the mid-point as a better approximation of the root. This process is terminated when interval is smaller than the desired accuracy.

Problem:-Find a root of the equation $x^3 - 5x + 1 = 0$ using the bisection method in 5 – stages

Sol: Let $f\left(x\right) = x^3 - 5x + 1$

we note that f(0) > 0 and f(1)< 0

$\therefore$ Root lies between 0 and 1

Consider $x_0 = 0$ *and* $x_1 = 1$

By bisection method the next approximation is

$$x_2 = \frac{x_0 + x_1}{2} = \frac{1}{2}\left(0 + 1\right) = 0.5$$

$$\Rightarrow f\left(x_2\right) = f\left(0:5\right) = -1.375 < 0 \ and \ f\left(0\right) > 0$$

We have the root lies between 0 and 0.5

Now $x_3 = \frac{0 + 0.5}{2} = 0.25$

We find $f\left(x_3\right) = -0.234375 < 0$ *and* $f\left(0\right) > 0$

Since $f\left(0\right) > 0$, we conclude that root lies between x_0 *and* x_3

The third approximation of the root is

$$x_4 = \frac{x_0 + x_3}{4} = \frac{1}{2}\left(0 + 0.25\right)$$
$$= 0.125$$

We have $f\left(x_4\right) = 0.37495 > 0$

Since $f\left(x_4\right) > 0$ *and* $f\left(x_3\right) < 0$, the root lies between

$x_4 = 0.125$ *and* $x_3 = 0.25$

Considering the 4th approximation of the roots

$$x_5 = \frac{x_3 + x_4}{2} = \frac{1}{2}\left(0.125 + 0.25\right) = 0.1875$$

$f\left(x_5\right) = 0.06910 > 0$,

since $f(x_5) > 0$ *and* $f(x_3) < 0$ the root must lie between $x_5 = 0.18758$ and $x_3 = 0.25$

Here the fifth approximation of the root is

$$x_6 = \frac{1}{2}(x_5 + x_3)$$
$$= \frac{1}{2}(0.1875 + 0.25)$$
$$= 0.21875$$

We are asked to do up to 5 stages. We stop here and 0.21875 is taken as an approximate value of the root and it lies between 0 and 1

MATLAB Program for Bisection method

```
function c = bisection(f,a,b)

if f(a)*f(b)>0
    disp('Interval has no root')
else
   c = (a + b)/2;
   while abs(f(c)) > 1e-7
     if f(a)*f(c)> 0
       a = c;
     else
       b = c;
     end
     c = (a + b)/2;
   end
end
```

Output

```
>> f=@(x)x^3-5*x+1;
>> bisect(f,0,1)

ans =

    0.2016
```

False Position Method (Regula – Falsi Method)

In the false position method we will find the root of the equation $f(x)=0$. Consider two initial approximate values x_0 *and* x_1 near the required root so that $f(x_0)$ *and* $f(x_1)$ have different signs. This implies that a root lies between x_0 *and* x_1. The curve $f(x)$ crosses x- axis only once at the point x_2 lying between the points x_0 *and* x_1, Consider the point $A=(x_0, f(x_0))$ and $B=(x_1, f(x_1))$ on the graph and suppose they are connected by a straight line, Suppose this line cuts x-axis at x_2 , We calculate the values of $f(x_2)$ at the point. If $f(x_0)$ *and* $f(x_2)$ are of opposite sign, then the root lies between x_0 *and* x_2 and value x_1 is replaced by x_2

Otherwise the root lies between x_2 and x_1 and the value of x_0 is replaced by x_2. Another line is drawn by connecting the newly obtained pair of values. Again the point here the line cuts the x-axis is a closer approximation to the root. This process is repeated as many times as required to obtain the desired accuracy. It can be observed that the points x_2, x_3, x_4 .....obtained converge to the expected root of the equation $y = f(x)$

To obtain the equation to find the next approximation to the root

Let $A=(x_0, f(x_0))$ *and* $B=(x_1, f(x_1))$ be the points on the curve $y = f(x)$ Then the equation to the chord AB is $\frac{y-f(x_0)}{x-x_0} = \frac{f(x_1)-f(x_0)}{x-x_0} \rightarrow (1)$

At the point C where the line AB crosses the x – axis, we have f(x) =0 i.e. y =0

From (1), we get $x = x_0 - \frac{x_1 - x_0}{f(x_1)-f(x_0)} f(x_0) \rightarrow (2)$

x is given by (2) serves as an approximated value of the root, when the interval in which it lies is small. If the new values of x is taken as x_2 then (2) becomes

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

$$= \frac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1)-f(x_0)} \longrightarrow (3)$$

Now we decide whether the root lies between x_0 *and* x_2 $(or) x_2$ *and* x_1

We name that interval as (x_1, x_2). The line joining $(x_1, y_1)(x_2, y_2)$ meets x – axis at x_3 is given by $x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2)-f(x_1)}$

This will in general, be nearest to the exact root we continue this procedure till the root is found to the desired accuracy.

The iteration process based on (3) is known as the method of false position. The successive intervals where the root lies, in the above procedure are named as $(x_0, x_1), (x_1, x_2), (x_2, x_3)$ etc

Where $x_i < x_i + 1$ and $f(x_0), f(x_i + 1)$ are of opposite signs

Also $x_{i+1} = \dfrac{x_{i-1} f(x_i) - x_i f(x_{i-1})}{f(x_i) - f(x_{i-1})}$

Problem:-

Find out the roots of the equation $x^3 - x - 4 = 0$ using false position method

Sol: Let $f(x) = x^3 - x - 4 = 0$

$f(0) = -4, f(1) = -4, f(2) = 2$

Since $f(1)$ *and* $f(2)$ have opposite signs the root lies between 1 and 2

By false position method $x_2 = \dfrac{x_0 f(x_1) - x_1 f(x_0)}{f(x_1) - f(x_0)}$

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

$$= \frac{2+8}{6} = \frac{10}{6} = 1.666$$

$$f(1.666) = (1.666)^3 - 1.666 - 4$$
$$= -1.042$$

Now, the root lies between 1.666 and 2

$$x_3 = \frac{1.666\times 2 - 2\times(-1.042)}{2 - (-1.042)} = 1.780$$

$$f(1.780) = (1.780)^3 - 1.780 - 4$$
$$= -0.1402$$

Now, the root lies between 1.780 and 2

$$x_4 = \frac{1.780\times 2 - 2\times(-0.1402)}{2 - (-0.1402)} = 1.794$$

$$f(1.794) = (1.794)^3 - 1.794 - 4$$
$$= -0.0201$$

Now, the root lies between 1.794 and 2

$$x_5 = \frac{1.794\times 2 - 2\times(-0.0201)}{2 - (-0.0201)} = 1.796$$

$$f(1.796) = (1.796)^3 - 1.796 - 4 = -0.0027$$

Now, the root lies between 1.796 and 2

$$x_6 = \frac{1.796 \times 2 - 2 \times (-0.0027)}{2 - (-0.0027)} = 1.796$$

The root is 1.796

MATLAB Program for False Position method

```
function c=falsePos(f, a, b)
  %False Position method for nonlinear equation
 if f(a)*f(b) > 0
   disp('Interval has no root')
 else
    c = (a*f(b)-b*f(a))/(f(b)-f(a));
  while abs(f(c)) > 1e-7
     if f(a)*f(c) > 0
       a=c;
     else
       b=c;
     end
     c = (a*f(b)-b*f(a))/(f(b)-f(a));
  end
```

Output:

```
>> f=@(x)x^3-x-4;
>> falsePos(f,1,2)

ans =

    1.7963
```

<u>Newton- Raphson Method:</u>

The Newton-Raphson method is a powerful and elegant method to find the root of an equation. This method is generally used to improve the results obtained by 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 which implies that $f(x_1)=0$.

By Taylor's theorem neglecting second and higher order terms

$$f(x_1) = f(x_0+h) = 0$$

$$\Rightarrow f(x_0) + hf^1(x_0) = 0$$

$$\Rightarrow h = -\frac{f(x_0)}{f^1(x_0)}$$

Substituting this in x_1 we get

$$x_1 = x_0 + h$$

$$= x_0 - \frac{f(x_0)}{f^1(x_0)}$$

$\therefore x_1$ is a better approximation than x_0

Successive approximations are given by

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

Problem:-

Find by Newton's method, the real root of the equation $xe^x - 2 = 0$ Correct to three decimal places.

Sol: Let $f(x) = xe^x - 2 \rightarrow (1)$

Then $f(0) = -2$ and $f(1) = e - 2 = 0.7183$

So root of $f(x)$ lies between 0 and 1

It is near to 1. so we take $x_0 = 1$ and $f^1(x) = xe^x + e^x$ *and* $f^1(1) = e + e = 5.4366$

$\therefore$ By Newton's Rule

First approximation $x_1 = x_0 - \frac{f(x_0)}{f^1(x_0)}$

$$= 1 - \frac{0.7183}{5.4366} = 0.8679$$

$\therefore f(x_1) = 0.0672 \qquad f^1(x_1) = 4.4491$

The second approximation $x_2 = x_1 - \frac{f(x_1)}{f^1(x_1)}$

$$= 0.8679 - \frac{0.0672}{4.4491}$$

$$= 0.8528$$

$\therefore$ Required root is 0.853 correct to 3 decimal places.

MATLAB Program for Newton-Raphson method

```
function c=newtonR(f,df,a)

 if(df(a)==0) disp('differentiation of a is zero');
  else
    c=a-f(a)/df(a);
    while abs(f(c)) > 1e-7
       a=c;
       c=a-f(a)/df(a);
    end
 end
```

Output:

```
>> f=@(x)x^3-x-4;
>> df=@(x)3*x^2-1;
>> newtonR(f,df,1)

ans =

    1.7963
```

UNIT-III

INTERPOLATION

Objectives:

- Develop an understanding of the use of numerical methods in modern scientific computing.
- To gain the knowledge of Interpolation

Syllabus:

Pre-requisite : Commands of MATLab

Interpolation – Introduction - Finite differences - Forward Differences - Backward differences - Central differences - Newton Interpolation formulae - Gauss Interpolation formulae - Lagrange's interpolation.

Learning Outcomes:

Student should be able to

- Know about the Interpolation, and Finite Differences.
- Utilize the Newton's formulae for interpolation.
- Utilize the Gauss formulae for interpolation.
- Operate Lagrange's Interpolation formula.

Introduction:

Consider the statement $y = f(x)$, $x_0 \le x \le x_n$ we understand that we can find the value of y, corresponding to every value of x in the range $x_0 \le x \le x_n$. If the function *f(x)* is single valued and continuous and is known explicitly then the values of *f(x)* for certain values of x like $x_0, x_1, \ldots\ldots x_n$ can be calculated.

Now the problem is, if we are given the set of tabular values

$$x : x_0 \quad x_1 \quad x_2 \ldots\ldots\ldots x_n$$

$$y : y_0 \quad y_1 \quad y_2 \ldots\ldots\ldots y_n$$

Satisfying the relation $y = f(x)$ and the explicit definition of *f(x)* is not known, is it possible to find a simple function say $\phi(x)$ such that *f(x)* and $\phi(x)$ agree at the set of tabulated points. This process of finding $\phi(x)$ is called interpolation. If $\phi(x)$ is a polynomial then the process is called polynomial interpolation and $\phi(x)$ is called interpolating polynomial. In our study we are concerned with polynomial interpolation.

Interpolation is the process of deriving a simple function from a set of discrete data points so that the function passes through all the given data points (i.e. reproduces the data points exactly) and can be used to estimate data points in-between the given ones.

Finite Differences:

Here we introduce forward, backward and central differences of a function *y = f(x).* These differences play a fundamental role in the study of differential calculus, which is an essential part of numerical applied mathematics.

1. Forward Differences:

Consider a function *y = f(x)* of an independent variable x. let $y_0, y_1, y_2 \ldots\ldots y_n$ be the values of y corresponding to the values $x_0, x_1, x_2 \ldots\ldots x_n$ of x respectively. Then the differences $y_1 - y_0$, $y_2 - y_1 \ldots\ldots\ldots\ldots$ are called the first forward differences of y, and we denote them by $\Delta y_0, \Delta y_1, \ldots\ldots$

that is $\Delta y_0 = y_1 - y_0,\ \Delta y_1 = y_2 - y_1,\ \Delta y_2 = y_3 - y_2, \ldots\ldots\ldots$

In general $\Delta y_r = y_{r+1} - y_r \quad where\ r = 0,1,2,3,\ldots\ldots n-1$

Here the symbol Δ is called the forward difference operator.

The second forward differences and are denoted by $\Delta^2 y_0, \Delta^2 y_1, \Delta^2 y_2, \ldots\ldots\ldots$

that is

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0$$

$$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$$

In general $\Delta^2 y_r = \Delta y_{r+1} - \Delta y_r \quad where\ r = 0,1,2,3,\ldots\ldots n-2$

similarly, the k[th] forward differences are defined by the formula.

$\Delta^k y_r = \Delta^{k-1} y_{r+1} - \Delta^{k-1} y_r, where\ r = 0,1,2,\ldots\ldots n-k\ and\ k = 1,2,\ldots\ldots n-1$

The symbol Δ^k is referred as the k[th] forward difference operator.

If *f(x)* is a function of x and h be the increment in x then forward difference of *f(x)* is defined as $\Delta f(x) = f(x+h) - f(x)$

Forward Difference Table:

The forward differences are usually arranged in tabular columns as shown in the following table called a forward difference table

Values of x	Values of y	First order differences	Second order differences	Third order differences
x_o	y_0			
		$\Delta y_0 = y_1 - y_0$		
x_1	y_1		$\Delta^2 y_0 = \Delta y_1 - y_0$	
		$\Delta y_1 = y_2 - y_1$		$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0$
x_2	y_2		$\Delta^2 y_1 = \Delta y_2 - \Delta y_1$	
		$\Delta y_2 = y_3 - y_2$		
x_3	y_3			

2. Backward Differences:

Consider a function *y = f(x)* of an independent variable x. let $y_0, y_1, y_2 \ldots\ldots y_n$ be the values of y corresponding to the values $x_0, x_1, x_2 \ldots\ldots x_n$ of x respectively. Then the differences y_1-y_0, y_2 – y_1………… are called the first backward differences of y, and we denote them by $\nabla y_1, \nabla y_2, \ldots\ldots$

that is $\nabla y_1 = y_1 - y_0,\ \nabla y_2 = y_2 - y_1,\ \nabla y_3 = y_3 - y_2, \ldots\ldots\ldots$

In general $\Delta y_r = y_r - y_{r-1}$ *where* $r = 1, 2, 3, \ldots\ldots n$

Here the symbol ∇ is called the backward difference operator.

The second backward differences and are denoted by $\nabla^2 y_2, \nabla^2 y_3, \nabla^2 y_4, \ldots\ldots\ldots$

That is $\nabla^2 y_2 = \nabla y_2 - \nabla y_1, \nabla^2 y_3 = \nabla y_3 - \nabla y_2$

In general $\nabla^2 y_r = \nabla y_r - \nabla y_{r-1}$ *where* $r = 2, 3, \ldots\ldots n$

similarly, the k^th^ backward differences are defined by the formula as

$\nabla^k y_r = \nabla^{k-1} y_r - \nabla^{k-1} y_{r-1}, where\ r = k, k+1, \ldots.. n\ and\ k = 1, 2, \ldots\ldots n-1$

The symbol ∇^k is referred as the k^th^ backward difference operator.

If *f(x)* is a function of x and h be the increment in x then bckward difference of *f(x)* is defined as $\nabla f(x) = f(x) - f(x-h)$

Backward Difference Table:-

X	Y	∇y	$\nabla^2 y$	$\nabla^3 y$
x_0	y_0			
		∇y_1		
x_1	y_1		$\nabla^2 y_2$	
		∇y_2		$\nabla^3 y_3$
x_2	y_2		$\nabla^2 y_3$	
		∇y_3		
x_3	y_3			

3.Central Differences:

Consider a function y = *f(x)* of an independent variable x. let $y_0, y_1, y_2 \ldots\ldots y_n$ be the values of y corresponding to the values $x_0, x_1, x_2 \ldots\ldots x_n$ of x respectively. We define the first central differences

$\delta y_{1/2}, \delta y_{3/2}, \delta y_{5/2} ----$ as follows

$$\delta y_{1/2} = y_1 - y_0, \delta y_{3/2} = y_2 - y_1, \delta y_{5/2} = y_3 - y_2 ----$$

$$\delta y_{r-1/2} = y_r - y_{r-1} \rightarrow (1)$$

The symbol δ is called the central differences operator. This operator is a linear operator

Comparing expressions (1) above with expressions earlier used on forward and backward differences we get

$\delta y_{1/2} = \Delta y_0 = \nabla y_1, \delta y_{3/2} = \Delta y_1 = \nabla y_2 \ldots..$

In general $\delta y_{n+1/2} = \Delta y_n = \nabla y_{n+1}, n = 0,1,2...... \to (2)$

The first central differences of the first central differences are called the second central differences and are denoted by $\delta^2 y_1, \delta^2 y_2 ...$

Thus $\delta^2 y_1 = \delta_{3/2} - \delta y_{1/2}, \delta^2 y_2 = \delta_{5/2} - \delta_{\ 3/2} ....$

$\delta^2 y_n = \delta y_{n+1/2} - \delta y_{n-1/2} \to (3)$

Higher order central differences are similarly defined. In general the nth central differences are given by

for odd $n : \delta^n y_{r-1/2} = \delta^{n-1} y_r - \delta^{n-1} y_{r-1}, r = 1,2.... \to (4)$

for even $n : \delta^n y_r = \delta^{n-1} y_{r+1/2} - \delta^{n-1} y_{r-1/2}, r = 1,2.... \to (5)$

while employing for formula (4) for $n = 1$, we use the notation $\delta^0 y_r = y_r$

If y is a constant function, that is if $y = c$ a constant, then $\delta^n y_r = 0 \ \ for \ \ all \ \ n \geq 1$

If *f(x)* is a function of x and h be the increment in x then forward difference of *f(x)* is defined as $\delta f(x) = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right)$

Central Difference Table:

x_0	y_0	δy	$\delta^2 y$	$\delta^3 y$	$\delta^4 y$
		$\delta y_{1/2}$			
x_1	y_1		$\delta^2 y_1$		
		$\delta y_{2/2}$		$\delta^3 y_{3/2}$	
x_2	y_2		$\delta^2 y_2$		$\delta^4 y_2$
		$\delta y_{5/2}$		$\delta^3 y_{5/2}$	
x_3	y_3		$\delta^2 y_3$		
		$\delta y_{7/2}$			
x_4	y_4				

Note: The forward difference operator Δ, backward difference operator ∇ and central difference operator δ are linear operators.

E-operator: The shift operator E is defined by the equation $Ey_r = y_{r+1}$. This shows that the effect of E is to shift the functional value y_r to the next higher value y_{r+1}. This is also called forward shift operator.

A second operation with E gives $E^2 y_r = E(Ey_r) = Ey_{r+1} = y_{r+2}$

Generalizing $E^k y_r = y_{k+r}$

Inverse operator E^-1^: Inverse operator E^{-1} is defined as $E^{-1}y_r = y_{r-1}$. This is also called backward operator.

In general $E^{-k}y_r = y_{r-k}$

Relationship Between Δ *and* E

We have

$$\Delta y_0 = y_1 - y_0$$
$$= Ey_0 - y_0 = \left(E-1\right)y_0$$
$$\Rightarrow \Delta = E - y\left(or\right)E = 1 + \Delta$$

Some more relations

$$\Delta^3 y_0 = \left(E-1\right)^3 y_0 = \left(E^3 - 3E^2 + 3E - 1\right)y_0$$
$$= y_3 - 3y_2 + 3y_1 - y_0$$

Note: We can easily establish the following relations

$$\nabla \equiv 1 - E^{-1}$$

Theorem: If *f(x)* is a polynomial of degree n and the values of x are equally spaced then $\Delta^n f(x)$ is constant.

Note: As $\Delta^n f(x)$ is a constant, it follows that $\Delta^{n+1} f(x) = 0,\ \Delta^{n+2} f(x) = 0,........$ The converse of above result is also true that is, if $\Delta^n f(x)$ is tabulated at equal spaced intervals and is a constant, then the function *f(x)* is a polynomial of degree n.

Q.

Estimate the missing figure in the following table:

x	1	2	3	4	5
$y = f(x)$	2	5	7	---	32

Solution

Since we are given four entries in the table, the function $y = f(x)$ can be represented by a polynomial of degree three.

$$\Delta^3 f(x) = \text{Constant}$$
$$\text{and } \Delta^4 f(x) = 0, \quad \forall x$$

In particular,

$$\Delta^4 f(x_0) = 0$$

Equivalently,

$$(E-1)^4 f(x_0) = 0$$

Expanding, we have

$$(E^4 - 4E^3 + 6E^2 - 4E + 1)f(x_0) = 0$$

That is,

$$f(x_4) - 4f(x_3) + 6f(x_2) - 4f(x_1) + f(x_0) = 0$$

Using the values given in the table, we obtain

$$32 - 4f(x_3) + 6\times 7 - 4\times 5 + 2 = 0$$

which gives $f(x_3)$, the missing value equal to 14.

Q.Find the missing term in the following data

X	0	1	2	3	4
Y	1	3	9	-	81

Why this value is not equal to 3^3 . Explain

Sol. Consider $\Delta^4 y_0 = 0$

$\Rightarrow 4y_0 - 4y_3 + 5y_2 - 4y_1 + y_0 = 0$

Substitute given values we get

$81 - 4y_3 + 54 - 12 + 1 = 0 \Rightarrow y_3 = 31$

From the given data we can conclude that the given function is $y = 3^x$. To find y_3, we have to assume that y is a polynomial function, which is not so. Thus we are not getting $y = 3^3 = 27$

Q.Evaluate

$(i)\Delta \cos x$

$(ii)\Delta^2 \sin(px+q)$

$(iii)\Delta^n e^{ax+b}$

Sol. Let h be the interval of differencing

$$(i)\,\Delta\cos x=\cos(x+h)-\cos x$$
$$=-2\sin\left(x+\frac{h}{2}\right)\sin\frac{\text{h}}{2}$$
$$(ii)\,\Delta\sin(px+q)=\sin\left[p(x+h)+q\right]-\sin(px+q)$$
$$=2\cos\left(px+q+\frac{ph}{2}\right)\sin\frac{ph}{2}$$
$$=2\sin\frac{ph}{2}\sin\left(\frac{\pi}{2}+px+q+\frac{ph}{2}\right)$$
$$\Delta^2\sin(px+q)=2\sin\frac{ph}{2}\Delta\left[\sin(px+q)+\frac{1}{2}(\pi+ph)\right]$$
$$=\left[2\sin\frac{ph}{2}\right]^2\sin\left[px+q+\frac{1}{2}(\pi+ph)\right]$$

$$(iii)\,\Delta e^{ax+b}=e^{a(x+h)+b}-e^{ax+b}$$
$$=e^{(ax+b)}\left(e^{ah-1}\right)$$
$$\Delta^2e^{ax+b}=\Delta\left[\Delta\left(e^{ax+b}\right)\right]-\Delta\left[\left(e^{ah}-1\right)\left(e^{ax+b}\right)\right]$$
$$=\left(e^{ah}-1\right)^2\Delta\left(e^{ax+h}\right)$$
$$=\left(e^{ah}-1\right)^2e^{ax+b}$$

Proceeding on, we get $\Delta^n\left(e^{ax+b}\right)=\left(e^{ah}-1\right)^ne^{ax+b}$

Newton's Forward Interpolation Formula:

Let *y=f(x)* be a polynomial of degree n and taken in the following form

$$y=f(x)=b_0+b_1(x-x_0)+b_2(x-x_0)(x-x_1)+b_3(x-x_0)(x-x_1)(x-x_2)+---$$
$$+b_n(x-x_0)(x-x_1)----(x-x_{n-1})\rightarrow(1)$$

This polynomial passes through all the points (for i = 0 to n. Therefore, we can obtain the $y_i's$ by substituting the corresponding $x_i's$ as

$$at\ \ x=x_0,y_0=b_0$$
$$at\ \ x=x_1,y_1=b_0+b_1(x_1-x_0)$$
$$at\ \ x=x_2,y_2=b_0+b_1(x_2-x_0)+b_2(x_2-x_0)(x_2-x_1)\rightarrow(1)$$

Let 'h' be the length of interval such that $x_i's$ represent

$$x_0,x_0+h,x_0+2h,x_0+3h----x_0+xh$$

This implies $x_1-x_0=h,x_2-x_0-2h,x_3-x_0=3h----x_n-x_0=nh\rightarrow(2)$

From (1) and (2), we get

$$y_0 = b_0$$
$$y_1 = b_0 + b_1 h$$
$$y_2 = b_0 + b_1 2h + b_2(2h)h$$
$$y_3 = b_0 + b_1 3h + b_2(3h)(2h) + b_3(3h)(2h)h$$
....

....

$$y_n = b_0 + b_1(nh) + b_2(nh)(n-1)h + ---- + b_n(nh)[(n-1)h][(n-2)h] \rightarrow (3)$$

Solving the above equations for $b_0, b_{11}, b_2.....b_n$, we get $b_0 = y_0$

$$b_1 = \frac{y_1 - b_0}{h} = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h}$$

$$b_2 = \frac{y_2 - b_0 - b_1 2h}{2h^2} = y_2 - y_0 - \frac{(y_1 - y_0)}{h}2h$$

$$= \frac{y_2 - y_0 - 2y_1 - 2y_0}{2h^2} = \frac{y_2 - 2y_1 + y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2}$$

$$\therefore b_2 = \frac{\Delta^2 y_0}{2!h^2}$$

Similarly, we can see that

$$b_3 = \frac{\Delta^3 y_0}{3!h^3}, b_4 = \frac{\Delta^4 y_0}{4!h^4} ----- b_n = \frac{\Delta^n y_0}{n!h^n}$$

$$\therefore y = f(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_1)$$
$$+ \frac{\Delta^3 y_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + ---- +$$
$$+ \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1) --- (x - x_{n-1}) \rightarrow (3)$$

If we use the relationship $x = x_0 + ph \Rightarrow x - x_0 = ph, \text{ where } p = 0,1,2,.....n$

Then

$$x - x_1 = x - (x_0 + h) = (x - x_0) - h$$
$$= ph - h = (p-1)h$$
$$x - x_2 = x - (x_1 + h) = (x - x_1) - h$$
$$= (p-1)h - h = (p-2)h$$
....

$$x - x_i = (p - i)h$$
....

$$x - x_{n-1} = [p - (n-1)]h$$

Equation (3) becomes

$$y = f(x) = f(x_0 + ph) = y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{3!}\Delta^3 y_0 + - - - - - +$$
$$\frac{p(p-1)(p-2) - - - - (p-(n-1))}{n!}\Delta^n y_0 \to (4)$$

Q. Find the melting point of the alloy containing 54% of lead, using appropriate interpolation formula

Percentage of lead(p)	50	60	70	80
Temperature $(Q°c)$	205	225	248	274

Sol. The difference table is

x	Y	Δ	Δ^2	Δ^3
50	205			
		20		
60	225		3	
		23		0
70	248		3	
		26		
80	274			

Let temperature = $f(x)$

$$x_0 + ph = 24, x_0 = 50, h = 10$$
$$50 + p(10) = 54 \ (or)\ p = 0.4$$

By Newton's forward interpolation formula

$$f(x_0 + ph) = y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{n!}\Delta^3 y_0 + - - -$$
$$f(54) = 205 + 0.4(20) + \frac{0.4(0.4-1)}{2!}(3) + \frac{(0.4)(0.4-1)(0.4-2)}{3!}(0)$$
$$= 205 + 8 - 0.36$$
$$= 212.64$$

Melting point = 212.6

Q. Consider the following table of values

Consider the following table of values

x	.2	.3	.4	.5	.6
F(x)	.2304	.2788	.3222	.3617	.3979

Find f (.36) using Newton's Forward Difference Formula.

Sol.

x	$y=f(x)$	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0.2	0.2304	0.0484	-0.005	0.0011	-0.0005
0.3	0.2788	0.0434	-0.0039	0.0006	
0.4	0.3222	0.0395	-0.0033		
0.5	0.3617	0.0362			
0.6	0.3979				

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

$$+\frac{p(p-1)(p-2)(p-3)}{4!}\Delta^4 y_0 + \cdots + \frac{p(p-1)(p-2)........(p-n+1)}{n!}\Delta^n y_0$$

Where

$x_0 = 0.2,\quad y_0 = 0.2304,\quad \Delta y_0 = 0.0484,$

$\Delta^2 y_0 = -0.005,\quad \Delta^3 y_0 = 0.0011,\ \Delta^4 y_0 = -.0005$

$$p = \frac{x - x_0}{h} = \frac{0.36 - 0.2}{0.1} = 1.6$$

$$y_x = 0.2304 + 1.6(0.0484) + \frac{1.6(1.6-1)}{2!}(-0.005) + \frac{1.6(1.6-1)(1.6-2)}{3!}(0.0011) + \frac{1.6(1.6-1)(1.6-2)(1.6-3)}{4!}(-.0005)$$

$$= 0.2304 + .077441 - .0024 + \frac{1.6(.6)(-.4)}{6}(.0011) + \frac{1.6(.6)(-.4)(-1.4)}{24}(-.0005)$$

$$= 0.3078 - .0024 - .00007 - .00001$$

$$= .3053$$

MatLab Code for Newton's Forward Interpolation Formula:

```
function fp = newtonint( x,y,p )
%To find a function value y at certain value of x when we have %x values and
their corresponding y values
% By using Newton's Forward Interpolation Method
% x is data for x
% y is data for y
% p is a point where we have to calculate y value
n=length(x);
for i= 1:n
```

```
    diff(i,1)=y(i);
end
for j=2:n
    for i=1:n-j+1
        diff(i,j)=diff(i+1,j-1)-diff(i,j-1);
    end
end
fp=y(1);
h=x(2)-x(1);
u=(p-x(1))/h;
for i=1:n-1
    factor=1;
    for j=0:i-1
        factor=factor*(u-j);
    end
    fp=fp+factor*diff(1,i+1)/factorial(i);
end
end
```

Newton's Backward Interpolation Formula:

If we consider

$y_n(x) = a_0 + a_1(x-x_n) + a_2(x-x_n)(x-x_{n-1}) + a_3(x-x_n)(x-x_{n-1})(x-x_{n-2}) + ----(x-x_i)$

and impose the condition that y and $y_n(x)$ should agree at the tabulated points $x_n, x_n - 1, ...x_2, x_1, x_0$

We obtain

$$y_x = y_n + p\nabla y_n + \frac{p(p+1)}{2!}\nabla^2 y_n + \frac{p(p+1)(p+2)}{3!}\nabla^3 y_n + \cdots$$
$$+ \frac{p(p+1)(p+2)\cdots(p+n-1)}{n!}\nabla^n y_n + \text{Error}$$

Where $p = \frac{x - x_n}{h}$

This uses tabular values of the left of y_n. Thus this formula is useful formula is for interpolation near the end of the tabular values.

Q. The sales for the last five years is given in the table below. Estimate the sales for the year 1979

Year	1974	1976	1978	1980	1982
Sales (in lakhs)	40	43	48	52	57

Sol.

Newton's backward difference Table

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
1974	40				
1976	43	3			
1978	48	5	2		
1980	52	4	-1	3	
1982	57	5	1	2	5

In this example,

$$p = \frac{1979 - 1982}{2} = -1.5$$

and

$$\nabla y_n = 5, \quad \nabla^2 y_n = 1,$$
$$\nabla^3 y_n = 2, \quad \nabla^4 y_n = 5$$

Newton's interpolation formula gives

$$y_{1979} = 57 + (-1.5)5 + \frac{(-1.5)(-0.5)}{2}(1) + \frac{(-1.5)(-0.5)(0.5)}{6}(2)$$
$$+ \frac{(-1.5)(-0.5)(0.5)(1.5)}{24}(5)$$
$$= 57 - 7.5 + 0.375 + 0.125 + 0.1172$$

Therefore,

$$y_{1979} = 50.1172$$

Q. Consider the following table of values

x	1	1.1	1.2	1.3	1.4	1.5
F(x)	2	2.1	2.3	2.7	3.5	4.5

Use Newton's Backward Difference Formula to estimate the value of f(1.45).

Sol.

x	y=F(x)	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$	$\nabla^5 y$
1	2					
1.1	2.1	0.1				
1.2	2.3	0.2	0.1			
1.3	2.7	0.4	0.2	0.1		
1.4	3.5	0.8	0.4	0.2	0.1	
1.5	4.5	1	0.2	-0.2	-0.4	-0.5

$p = \frac{x - x_n}{h} = \frac{1.45 - 1.5}{0.1} = -0.5$, $\nabla y_n = 1$, $\nabla^2 y_n = .2$, $\nabla^3 y_n = -.2$, $\nabla^4 y_n = -.4$, $\nabla^5 y_n = -.5$

As we know that

$$y_x = y_n + p\nabla y_n + \frac{p(p+1)}{2!}\nabla^2 y_n + \frac{p(p+1)(p+2)}{3!}\nabla^3 y_n$$
$$+ \frac{p(p+1)(p+2)(p+3)}{4!}\nabla^4 y_n + \frac{p(p+1)(p+2)(p+3)(p+4)}{5!}\nabla^5 y_n$$

$$y_x = 4.5 + (-0.5)(1) + \frac{(-0.5)(-0.5+1)}{2!}(0.2) + \frac{(-0.5)(-0.5+1)(-0.5+2)}{3!}(-0.2)$$
$$+ \frac{(-0.5)(-0.5+1)(-0.5+2)(-0.5+3)}{4!}(-0.4) + \frac{(-0.5)(-0.5+1)(-0.5+2)(-0.5+3)(-0.5+4)}{5!}(-0.5)$$

$$= 4.5 - 0.5 - 0.025 + 0.0125 + 0.015625 + 0.068359$$
$$= 4.07148$$

MatLab Code for Newton's Backward Interpolation Formula:

```
function yval=nbdi(xd,yd,xval)
n=length(xd);
bdt(:,1)=xd';
bdt(:,2)=yd';
for j=3:n+1
    for i=n:-1:j-1
        bdt(i,j)=bdt(i,j-1)-bdt(i-1,j-1);
    end
end
bdt
h=xd(2)-xd(1)
p=(xval-xd(n))/h
c=ones(1,n);
for r=1:n-1
    c(r+1)=prod(p:1:p+r-1)/factorial(r);
end
terms=bdt(n,2:n+1).*c;
```

```
yval=sum(terms);
fprintf('\n The approximated y value at given x=%f is: %f',xval,yval)
end
```

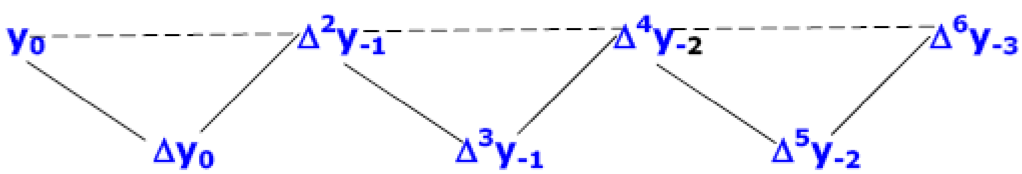
Gauss Forward Interpolation Formula:

Gauss forward interpolation formula is given by

$$y_p = y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_{-1} + \frac{(p+1)p(p-1)}{3!}\Delta^3 y_{-1} + \frac{(p+1)p(p-1)(p-2)}{4!}\Delta^4 y_{-2} + \frac{(p+2)(p+1)p(p-1)(p-2)}{4!}\Delta^5 y_{-2} + \ldots..$$

Where $p = \frac{x-x_0}{h}$

- The value p is measured forwardly from the origin and 0<p<1
- The above formula involves odd differences below the central horizontal line and even differences on the line. This is explained in the following figure.



Q. Find f(30) from the following table values using Gauss forward difference formula:

x:	21	25	29	33	37
F(x):	18.4708	17.8144	17.1070	16.3432	15.5154

Sol:

The difference table is

x	f	Δf	Δ²f	Δ³f	Δ⁴f
21	18.4708				
		-0.6564			
25	17.8144		-0.0510		
		-0.7074		-0.0054	
29	17.1070		0.0564		-0.0022
		-0.7638		-0.0076	

33	16.3432		-0.0640		
		-0.8278			
37	15.5154				

Let $x_0 = 29$ given x=30 and h=4 then $p = \frac{x-x_0}{h} = \frac{30-29}{4} = 0.25$

Gauss forward interpolation formula is given by

$$y_p = y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_{-1} + \frac{(p+1)p(p-1)}{3!}\Delta^3 y_{-1} + \frac{(p+1)p(p-1)(p-2)}{4!}\Delta^4 y_{-2} + ...$$

Therefore f (30) = 16.9217

MatLab Code for Gauss Forward Interpolation Formula:

```
function [ yval ] = gauss_p( xd,yd,xp )
n=length(xd);
if(length(yd)==n)
tbl=yd';
for j=2:n
    for i=1:n-j+1
        tbl(i,j)=tbl(i+1,j-1)-tbl(i,j-1);
    end
end
tbl
h=xd(2)-xd(1);
if rem(n,2)==0
    k=n/2+1;
else
    k=n/2+0.5;
end

p=(xp-xd(k))/h;

pt=cumprod([1,p-(0:n-3)]);

dt=[tbl(k,1),tbl(k,2),tbl(k-1,3:n-1)+tbl(k-1,4:n)]./factorial(0:n-2);

yval=sum(pt.*dt);

end
```

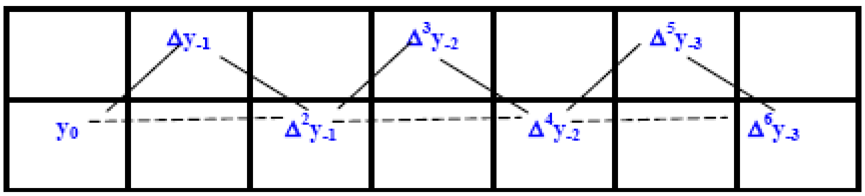
Gauss Backward Interpolation Formula:
Gauss backward interpolation formula is given by

$$y_p = y_0 + p\Delta y_0 + \frac{(p+1)p}{2!}\nabla^2 y_1 + \frac{(p+1)p(p-1)}{3!}\nabla^3 y_1 + \frac{(p+2)(p+1)p(p-1)}{4!}\nabla^4 y_2 + \frac{(p+2)(p+1)p(p-1)(p-2)}{4!}\nabla^5 y_2 + ...$$

Where $p = \frac{x-x_0}{h}$

- The value p is measured forwardly from the origin and $-1<p<0$.

- The above formula involves odd differences above the central horizontal line and even differences on the line.



Q. From the following data find y when x=38 by using Gauss backward interpolation formula

x	30	35	40	45	50
y	15.9	14.9	14.1	13.3	12.5

Sol: The difference table is

x	**y**	∇y	$\nabla^2 y$	$\nabla^3 y$	$\nabla^4 y$
30	15.9				
		-1			
35	14.9		0.2		
		-0.8		-0.2	
40	14.1		0		0.2
		-0.8		0	
45	13.3		0		
		-0.8			
50	12.5				

Let $x_0 = 40$ given x=38 and h=5 then $p = \frac{x-x_0}{h} = \frac{38-40}{5} = -0.4$

Gauss forward interpolation formula is given by

$$y_p = y_0 + p\Delta y_0 + \frac{(p+1)p}{2!}\nabla^2 y_1 + \frac{(p+1)p(p-1)}{3!}\nabla^3 y_1 + \frac{(p+2)(p+1)p(p-1)}{4!}\nabla^4 y_2 + \frac{(p+2)(p+1)p(p-1)(p-2)}{4!}\nabla^5 y_2 + ...$$

Therefore y(38)=14.4245

Lagrange’s Interpolation Formula:

Let $x_0, x_1, x_2, \ldots.\ x_n$ be the $(n+1)$ values of x which are not necessarily equally spaced. Let $y_0, y_1, y_2 \ldots\ldots.. y_n$ be the corresponding values of $y = f(x)$ let the polynomial of degree n for the function $y = f(x)$ passing through the $(n+1)$ points $(x_0, f(x_0)), (x_1, f(x_1)) ---- (x_n, f(x_n))$ be in the following form

$$y = f(x) = a_0(x-x_1)(x-x_2)\ldots\ldots(x-x_n) + a_1(x-x_0)(x-x_2)\ldots\ldots\ldots(x-x_n) +$$
$$a_2(x-x_0)(x-x_1)\ldots\ldots.(x-x_n) + \ldots\ldots.. + a_n(x-x_0)(x-x_1)\ldots\ldots(x-x_{n-1}) \rightarrow (1)$$

Where $a_0, a_1, a_2 \ldots.$ a^n are constants

Since the polynomial passes through $(x_0, f(x_0))$, $(x_1, f(x_1))\ldots\ldots(x_n, f(x_n))$. The constants can be determined by substituting one of the values of $x_0, x_1, \ldots\ldots x_n\ for\ x$ in the above equation

Putting $x = x_0$ in (1) we get, $f(x_0) = a_0(x-x_1)(x_0-x_2)(x_0-x_n)$

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

Putting $x = x_1$ in (1) we get, $f(x_1) = a_1(x-x_0)(x_1-x_2) ---- (x_1-x_n)$

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

Similarly substituting $x = x_2$ in (1), we get

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

Continuing in this manner and putting $x = x_n$ in (1) we get

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

Substituting the values of $a_0, a_1, a_2 \ldots. a_n$, we get

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

Q . Using Lagrange's formula calculate $f(3)$ from the following table

x	0	1	2	4	5	6

$f(x)$	1	14	15	5	6	19

Sol. Given $x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 4, x_5 = 6, x_4 = 5$

$f(x_0)=1, f(x_1)=14, f(x_2)=15, f(x_3)=5, f(x_4)=6, f(x_5)=19$

From lagrange's interpolation formula

$$f(x)=\frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)(x_0-x_5)}f(x_0)$$
$$+\frac{(x-x_0)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)(x_1-x_4)(x_1-x_5)}f(x_1)$$
$$+\frac{(x-x_0)(x-x_1)(x-x_3)(x-x_4)(x-x_5)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)(x_2-x_5)}f(x_2)$$

$$\frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_5-x_0)(x_5-x_1)(x_5-x_2)(x_5-x_3)(x_5-x_4)}f(x_5)$$

Here $x=3$ then

$$f(3)=\frac{(3-1)(3-2)(3-4)(3-5)(3-6)}{(0-1)(0-2)(0-4)(0-5)(0-6)}\times 1+$$
$$\frac{(3-0)(3-2)(3-4)(3-5)(3-6)}{(1-0)(1-2)(1-4)(1-5)(1-6)}\times 14+$$
$$\frac{(3-0)(3-1)(3-4)(3-5)(3-6)}{(2-0)(2-1)(2-4)(2-5)(2-6)}\times 15+$$
$$\frac{(3-0)(3-1)(3-2)(3-5)(3-6)}{(4-0)(4-1)(4-2)(4-5)(4-6)}\times 5+$$
$$\frac{(3-0)(3-1)(3-2)(3-4)(3-6)}{(5-0)(5-1)(5-2)(5-4)(5-6)}\times 6+$$
$$\frac{(3-0)(3-1)(3-2)(3-4)(3-5)}{(6-0)(6-1)(6-2)(6-4)(6-5)}\times 19$$

$$=\frac{12}{240}-\frac{18}{60}\times 14+\frac{36}{48}\times 15+\frac{36}{48}\times 5-\frac{18}{60}\times 6+\frac{12}{40}\times 19$$
$$=0.05-4.2+11.25+3.75-1.8+0.95$$
$$=10$$
$$f(x_3)=10$$

MatLab Code for Lagrange's Interpolation Formula:

```
function [ yval ] = lagrange(xd,yd,x)
%To find a function value y at certain value of x when we have x values and
```

```
%their corresponding y values by using Lagranges Interpolation Method
% xd is data for x
% yd is data for y
% x is a point where we have to calculate y value
n=length(xd);
if(length(yd)==n)
   p=zeros(1,n);
   for i=1:n
      temp=xd;
      temp(i)=[];
      p(i)=prod((x-temp)./(xd(i)-temp));
   end
   yval=sum(p.*yd);
   fprintf('\n The value of y at x=%f is %f',x,yval)
else
   error('xd and yd must be of same size');
end
```

Assignment-Cum-Tutorial Questions

Section-A

Objective Questions:

1. A linear version of the Lagrange's interpolation formula for f(x) is [ ]

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

c) $\left(\frac{x-x_1}{x_0-x_1}\right)f(x_0)+\frac{(x-x_0)(x_2-x_0)}{(x_1-x_0)}f(x_1)$ d) $\frac{(x-x_0)}{(x_0-x_1)}f(x_0)+\frac{(x_1-x_0)}{(x-x_0)}f(x_1)$

2. The following is used for unequal interval of x values [ ]
   a) Lagrange's formula
   b) Newton's forward interpolation formula
   c) Newton's backward interpolation formula
   d) Gauss forward interpolation formula

3. The $(n+1)^{th}$ order difference a polynomial of n^{th} degree is [ ]
   a) polynomial of n^{th} degree b) zero
   c) polynomial on first degree d) constant

X	1	2	3	4

4.

f(x)	1	4	27	64

If x=2.5 then p= [ ]

a) 1.5 b) 1 c) 2.5 d) 2

5.

X	0.1	0.2	0.3	0.4
f(x)	1.005	1.02	1.045	1.081

When p=0.6, x= [ ]

a) 0.16 b) 0.26 c) 0.1 d) 3.0 2.

6. Relation between Backward and Shifting operator is ________________.

7. When do we apply Lagrange's interpolation?

8. If $y = x^2 + 2x$ then $\Delta^3 y =$ [ ]

a) 1 b) 2 c) 0 d) 3

9. $\frac{\Delta^2}{E}\left(e^x\right) =$ _____ [ ]

a) $e^x\left(e^h - 1\right)^2$ b) $e^x\left(e^h - 1\right)$ c) $e^{x-h}\left(e^h - 1\right)^2$ d) $e^x\left(e^{h-1}\right)$

10. $\left(E^{1/2} + E^{-1/2}\right)\left(1+\Delta\right)^{1/2}$ [ ]

a) $1+\Delta$ b) $2+\Delta$ c) $1-\Delta$ d) Δ

11.

X	0	1	2
F(x)	7	10	13

By Newton's forward formula f(2.5)= [ ]

a) 15.25 b) 16.75 c) 16.25 d)16.108.

12. Using Lagrange's interpolation, find the polynomial through (0,0),(1,1) and (2,2).

13. $\Delta(\cos x) =$ ________________________.

Section-B

Subjective Questions

1. Prepare a MATLAB code to construct a forward difference table for the given data.
2. Prepare a MATLAB code to construct a backward difference table for the given data.

3. Certain corresponding values of x and log x are (300, 2.4771), (304, 2.4829), (305, 2.4843) and (307, 2.4871). Find log 301.
4. If the interval of differencing unity prove that

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

5. Find a cubic polynomial in x which takes on the values -3, 3, 11, 27, 57 and 107, when x=0, 1, 2, 3, 4 and 5 respectively.
6. Using Newton's forward interpolation formula, for the given table of values

X	1.1	1.3	1.5	1.7	1.9
$f(x)$	0.21	0.69	1.25	1.89	2.61

Obtain the value of f(x) when x= 1.4.

7. Develop a MATLAB code to find y value corresponding to some x value by using Newton's forward difference interpolation formula.
8. The population of a town in the decimal census was given below. Estimate the population for the 1895

Year x	1891	1901	1911	1921	1931
Population of y	46	66	81	93	101

9. Find the cubic polynomial which takes the values y(0)=1,y(1)=0,y(2)=1,y(3)=10
10. Using Newton's backward formula find the value of sin38 ?

x:	0	10	20	30	40
sin x:	0	.17365	.34202	.50000	.64279

11. Develop a MATLAB code to find y value corresponding to some x value by using Newton's backward difference interpolation formula.
12. Fit a polynomial of degree three which takes the following values

x:	3	4	5	6
y:	6	24	60	120

13. Using Newton's forward formula, find the value of f(1.6) if

X	1	1.4	1.8	2.2	2.6
y	3.49	4.82	5.96	6.5	8.4

14. Find log 58.75 from the following data:

X	40	45	50	55	60	65
Log x	1.60206	1.65321	1.69897	1.74036	1.77815	1.81291

Using Newton's Backward Interpolation formula.

15. Write a MATLAB program to implementation of Gauss backward difference interpolation.
16. Using Gauss forward formula find y(3.3) from the following data

X	1	2	3	4	5
Y	15.30	15.10	15.0	14.5	14.0

17. Develop a MATLAB code to find y value corresponding to some x value by using Gauss backward difference interpolation formula.
18. Write a MATLAB program to implementation of Lagrange's interpolation.
19. Find the Lagrange's interpolating polynomial and using it find y when x = 10, if the values of x and y are given as follows:

x	5	6	9	11
y	12	13	14	16

20. Find the number of students who got marks between 40 and 45

Marks	: 30-40	40-50	50-60	60-70	70-80
No. of students	: 31	42	51	35	31

21. The area A of a circle of diameter d is given below:

d:	80	85	90	95	100
A:	5026	5674	6362	7088	7854

Find approximately the areas of the circles of diameters 82 and 91.

Section-C

GATE/IES/Placement Tests/Other competitive examinations

1. Evaluate $\Delta^{10}(1-x)(1-2x)(1-3x)..........(1-10x)$ taking h=1

UNIT-IV

NUMERICAL DIFFERENTIATION AND INTEGRATION

Objectives:

- To understand the concepts of numerical differentiation and integration.

Syllabus:

Pre-requisite : Commands of MATLab

Approximation of derivative using Newton's forward and backward formulae - Integration using Trapezoidal and Simpson's rules.

Learning Outcomes:

At the end of the unit, Students will be able to

- Utilize numerical techniques to evaluate derivatives at a point.
- Utilize numerical techniques to evaluate definite integrals.
- Calculate the area and slope of a given curve.

Introduction:

Suppose a function y = f(x) is given by a table of values (x_i, y_i). The process of computing the derivative $\frac{dy}{dx}$ for some particular value of x is called Numerical differentiation.

Derivatives using Newton's forward difference formula:

Suppose that we are given a set of values (x_i, y_i), i = 0,1,2,...., n.

We want to find the derivative of y = f (x) passing through the (n + 1)points, at a point nearer to the starting value at x = x_0.

Newton's Forward Difference Interpolation Formula is

$$y = y0 + p\,\Delta y0 + \frac{p(p-1)}{2!}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{3!}\Delta^3 y_0 + \ldots\ldots\ldots\ldots \qquad \text{------------------ (1)}$$

$$\text{Where } p = \frac{x - x_0}{h} \qquad \text{------------------ (2)}$$

On differentiation (1) w.r.t., p we have

On differentiation (2) w.r.t. x we have, $\frac{dp}{dx} \approx \frac{1}{h}$

$$\frac{dy}{dx} = \frac{dy}{dp} \cdot \frac{dp}{dx} = \frac{1}{h}\left[\begin{array}{l} \Delta y0 + \frac{2p-1}{2}\Delta^2 y_0 + \frac{3p^2 - 6p + 2}{6}\Delta^3 y_0 \\ + \frac{4p^3 - 18p^2 + 22p - 6}{24}\Delta^4 y_0 + \ldots \end{array}\right] \qquad \ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Equation (3) gives the value of $\frac{dy}{dx}$ at any point x which may be anywhere in the interval.

At x = x_0 and p = 0, hence putting p = 0, equation (3) gives

$$\left(\frac{dy}{dx}\right)_{x \approx x_1} = \left(\frac{dy}{dp}\right)_{p \approx 1} = \frac{1}{h}\left|\begin{array}{l} \Delta y0 + \frac{1}{2}\Delta^2 y_0 + \frac{1}{6}\Delta^3 y_0 \\ + \frac{4p^3 - 18p^2 + 22p - 6}{24}\Delta^4 y_0 + \ldots \end{array}\right] \qquad \ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Again on differentiation (3) we get

$$\frac{d^2 y}{dx^2} = \frac{d\left(\frac{dy}{dx}\right)}{dx} = \frac{d}{dp}\left(\frac{dy}{dx}\right) \cdot \frac{dp}{dx} = \frac{d}{dp}\left(\frac{dy}{dx}\right) \cdot \frac{dp}{dx} = \frac{1}{h^2}\left[\Delta^2 y_0 + \frac{(p-1)}{}\Delta^3 y_0 + \frac{6p^2 - 18p + 11}{12}\Delta^4 y_0 + \ldots\right]$$

From which we obtain

$$\left(\frac{d^2 y}{dx^2}\right) x \approx x_0 = \frac{1}{h^2}\left[\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12}\Delta^4 y_0 - \frac{5}{6}\Delta^5 y_0 + ..\right] \text{ at } x = x_0 \text{ and } p = 0 \ \ldots\ldots\ (5)$$

$$\text{Similarly, } \left(\frac{d^3 y}{dx^3}\right) x \approx x_0 = \frac{1}{h^3}\left[\Delta^3 y_0 - \frac{3}{2}\Delta^4 y_0 + \ldots\ldots\right] \qquad \ldots\ldots\ (6)$$

Derivatives using Newton's Backward Difference Formula:

Newton's Backward Difference Interpolation Formula is

$$y(x) = y_n + p\,\Delta y_n + \frac{p(p+1)}{2!}\Delta^2 y_n + \frac{p(p+1)(p+2)}{3!}\Delta^3 y_n + \ldots\ldots\ldots\ldots \qquad \text{------------------ } (7)$$

$$\text{Where } p = \frac{x - x_n}{h} \qquad \text{------------------ } (8)$$

On differentiation (7) w.r.t., p we have

$$\frac{dy}{dp} = \left[\Delta y_n + \frac{2p+1}{2}\Delta^2 y_n + \frac{3p^2+6p+2}{6}\Delta^3 y_n + \frac{4p^3+18p^2+22p+6}{24}\Delta^4 y_n + \ldots \right] .$$

On differentiation (8) w.r.t. x we have, $\frac{dp}{dx} \approx \frac{1}{h}$ Now

$$\frac{dy}{dx} = \frac{dy}{dp}.\frac{dp}{dx} = \frac{1}{\mathrm{h}}\left[\nabla y_n + \frac{2p+1}{2}\Delta^2 y_n + \frac{3p^2+6p+2}{6}\Delta^3 y_n + \frac{4p^3+18p^2+22p+6}{24}\Delta^4 y_n + .. \right] \qquad \ldots\ldots\ldots\ldots\ldots(9)$$

Equation (9) gives the value of $\frac{dy}{dx}$ at any point x which may be anywhere in the interval.

At x = x_n and p = 0, hence putting p = 0, equation (9) gives

$$\left(\frac{dy}{dx}\right)x \approx x_{n_1} = \left(\frac{dy}{dx}\right)x_n = \frac{1}{\mathrm{h}}\left[\Delta y n + \frac{1}{2}\Delta^2 y_n + \frac{1}{3}\Delta^3 y_n + \frac{1}{4}\Delta^4 y_n + \ldots \right] \ldots\ldots\ldots\ldots\ldots(10)$$

Again on differentiation (09) we obtain

$$\frac{d^2 y}{dx^2} = \frac{d\left(\frac{dy}{dx}\right)}{dx}.\frac{dp}{dx} = \frac{\mathrm{d}}{\mathrm{dp}}\left(\frac{dy}{dx}\right).\frac{dp}{dx} = \frac{\mathrm{d}}{\mathrm{dp}}\left(\frac{dy}{dx}\right).\frac{dp}{dx}$$

$$= \frac{1}{h^2}\left[\Delta^2 y_n + (p+1)\Delta^3 y_n + \frac{6p^2+18p+11}{12}\Delta^4 y_n + . \right]$$

From which we obtain

$$\left(\frac{d^2 y}{dx^2}\right) x \approx x_n = \frac{1}{h^2}\left[\Delta^2 y_n + \Delta^3 y_n + \frac{11}{12}\Delta^4 y_n + \frac{5}{6}\Delta^5 y_n + .. \right]$$ at x= x_n and p = 0

Similarly, $$\left(\frac{d^3 y}{dx^3}\right) x \approx x_n = \frac{1}{h^3}\left[\Delta^3 y_n - \frac{3}{2}\Delta^4 y_0 + \ldots\ldots \right] \qquad \ldots\ldots. (12)$$

Q. Find $\frac{dy}{dx}$ *and* $\frac{d^2y}{dx^2}$ *at* $x = 51$ from the following data.

x	50	60	70	80	90
y	19.96	36.65	58.81	77.21	94.61

Solution: Here h = 10. To find the derivatives of y at x = 51 we use Newton's Forward difference formula taking the origin at a = 50.

We have $p = \frac{x - x_0}{h} = \frac{51 - 50}{10} 0.1$

$$\left(\frac{dy}{dx}\right)_{x=51} = \left(\frac{dy}{dx}\right)_{p=0.1} = \frac{1}{h}\left[\Delta y_0 + \frac{(2p-1)}{2!}\Delta^2 y_0 + \frac{(3p^2 - 6p + 2)}{3!}\Delta^3 y_0 + \frac{(4p^3 - 18p^2 + 22p - 6)}{4!}\Delta^4 y_0 + \ldots\right]$$

The difference table is given by

x	$p = \frac{x-50}{10}$	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
50	0	**19.96**				
			16.69			
60	1	36.65		**5.47**		
			22.16		**-9.23**	
70	2	58.81		-3.76		**11.99**
			18.40		2.76	
80	3	77.21		-1.00		
			17.40			
90	4	94.61				

$$\therefore \left(\frac{dy}{dx}\right)_{p=0.1} = \frac{1}{10}\left[16.69 + \frac{(0.2-1)}{2}(5.47) + \left[\frac{3(0.1)^2 - 6(0.1) + 2}{6}\right](-9.23) + \frac{\left[4(0.1)^3 - 18(0.1)^2 + 22(0.1) - 6\right]}{24} \times 11.99 + \ldots\right]$$

$$= \frac{1}{10}[16.69 - 2.188 - 2.1998 - 1.9863] = 1.0316$$

$$\left(\frac{d^2y}{dx^2}\right)_{p=0.1} = \frac{1}{h^2}\left[\Delta^2 y_0 + (p-1)\Delta^3 y_0 + \frac{(6p^2 - 18p + 11)}{12}\Delta^4 y_0 + \ldots\right]$$

$$= \frac{1}{100}\left[5.47 + (0.1) - 1(-9 - 23) + \frac{\left[6(.1)^2 - 18(.1) + 11\right]}{12}\right] \times 11.99$$

$$= \frac{1}{100}[5.47 + 8.307 + 9.2523]$$

=0.2303.

Q. The population of a certain town is shown in the following table

Year x	1931	1941	1951	1961	1971
Population y	40.62	60.80	79.95	103.56	132.65

Find the rate of growth of the population in 1961.

Solution. Here h = 10 Since the rate of growth of population is $\frac{dy}{dx}$ we have to find $\frac{dy}{dx}$ at x = 1961, which lies nearer to the end value of the table. Hence we choose the origin at x = 1971 and we use Newton's backward interpolation formula for derivative.

$$\frac{dy}{dx}=\frac{1}{h}\left[\nabla y_4+\frac{(2p+1)}{2}\nabla^2 y_4+\frac{(3p^2+6p+2)}{6}\nabla^3 y_4+\frac{(2p^3+9p^2+11p+3)}{12}\nabla^4 y_4+\ldots\right]$$

Where $p=\frac{x-x_0}{10}=\frac{1961-1971}{10}=-1$

The backward difference table

x Year	y Population	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1931	40.62				
		20.18			
1941	60.80		-1.03		
		19.15		5.49	**-4.47**
1951	79.95		4.46		
		23.61		**1.02**	
1961	103.56		**5.48**		
		29.09			
1971	132.65				

$$\left(\frac{dy}{dx}\right)_{p=-1}=\frac{1}{10}\left[29.09+-\left(\frac{1}{2}\right)(5.48)+\frac{\left[3(-1)^2+6(-1)+2\right]}{6}\times 1.02+\frac{\left[2(-1)^3+9(-1)^2+11(-1)+3\right]}{12}(-4.47)\right]$$

$$=\frac{1}{10}[29.09-2.74-0.17+0.3725]$$

$$=\frac{1}{10}[26.5525]=2.6553$$

$\therefore$ The rate of growth of the population in the year 1961 is 2.6553.

MATLAB Code to numerical differentiation by using Newton's forward difference Interpolation:

```
function[d1val,d2val]=derbynfdi(xd,yd,xval)
n=length(xd);
syms p;
fdt=zeros(n,n+1);
fdt(:,1)=xd';
fdt(:,2)=yd';
for j=3:n+1
    for i=1:n-j+2
        fdt(i,j)=fdt(i+1,j-1)-fdt(i,j-1);
    end
end
fdt
temp=[1,cumprod(p-(0:n-2))];
a=diff(temp);
b=diff(a);
h=xd(2)-xd(1)
p=(xval-xd(1))/h;
ap=eval(a);
bp=eval(b);
c=fdt(1,2:n+1)./factorial(0:n-1);
d1val=sum(ap.*c)/h;
d2val=sum(bp.*c)/(h*h);
fprintf('\n the approximated value of first derivative of y at given x value %f is %f ',xval,d1val)
fprintf('\n the approximated value of second derivative of y at given x value %f is %f ', xval,d2val)
```

MATLAB Code to numerical differentiation by using Newton's backward difference Interpolation:

```
function[d1val,d2val]=derbynbdi(xd,yd,xval)
n=length(xd);
syms p;
bdt=zeros(n,n+1);
bdt(:,1)=xd';
bdt(:,2)=yd';
for j=3:n+1
    for i=n:-1:j-1
        bdt(i,j)=bdt(i,j-1)-bdt(i-1,j-1);
    end
end
bdt;
temp=[1,cumprod(p+(0:n-2))];
a=diff(temp);
```

```
b=diff(a);
h=xd(2)-xd(1);
p=(xval-xd(n))/h;
ap=eval(a);
bp=eval(b);
c=bdt(n,2:n+1)./factorial(0:n-1);
d1val=sum(ap.*c)/h;
d2val=sum(bp.*c)/(h*h);
fprintf('\n the approximated value of first derivative of y at given x value %f is %f ',xval,d1val)
fprintf('\n the approximated value of second derivative of y at given x value %f is %f ', xval,d2val)
```

Numerical Integration :

Given set of (n + 1) data points (x_i, y_i), i = 0,1,2,.......,n of the function y = f(x), where f(x) is not known explicitly, it is required to evaluate $\int_{x_0}^{x_n} f(x)\,dx$.

Newton-Cote's Quadrature Formula (General Quadrature Formula):

This is the most popular and widely used numerical integration formula. It forms the basis for a number of numerical integration methods known as Newton-Cote's methods.

Derivation of Newton-Cotes formula:

Let the interval [a, b] be divided into n equal sub-intervals such that $a = x_0 < x_1 < x_2 < x_3 \ldots\ldots\ldots < x_n = b$. Then $x_n = x_0 + nh$.

Newton forward difference formula is

$$y(x) = y(x_0 + ph) = P_n(x) = y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{3!}\Delta^3 y_0 + \ldots\ldots\ \ldots\ldots \quad (1)$$

Where $p = \frac{x - x_0}{h}$ .Now, instead of f(x) we will replace it by this interpolating polynomial.

$\therefore \int_{x_0}^{x_n} f(x)\,dx = \int_{x_0}^{x_n} P_n(x)\,dx$, where $P_n(x)$ is an interpolating polynomial of degree n

$$= \int_{x_0}^{x_0 + nh} P_n(x)\,dx = \int_{x_0}^{x_0 + nh} \left[y_0 + p\Delta y_0 + \frac{p(p-1)}{2!}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{3!}\Delta^3 y_0 + \ldots\ldots \right] dx$$

Since $x = x_0 + ph$, $dx = h.dp$ and hence the above integral becomes

$$\int_{x_0}^{x_n} f(x)\,dx = \mathrm{h}\int_0^n \left[y_0 + \mathrm{p}\Delta \mathrm{y}_0 + \frac{\mathrm{p(p-1)}}{2!}\Delta^2 \mathrm{y}_0 + \frac{\mathrm{p(p-1)(p-2)}}{3!}\Delta^3 \mathrm{y}_0 + \ldots\ldots \right] dp$$

$$= \mathrm{h}\left[y_0(p) + \frac{\mathrm{p}^2 \Delta \mathrm{y}_0}{2} + \frac{1}{2}\left(\frac{p^3}{3} - \frac{p^2}{2}\right)\Delta^2 \mathrm{y}_0 + \frac{1}{6}\left(\frac{p^4}{4} - 3\frac{p^3}{3} + 2\frac{p^2}{2}\right)\Delta^3 \mathrm{y}_0 + \ldots\ldots \right]$$

$$= \mathrm{h}\left[ny_0 + \frac{\mathrm{n}^2 \Delta \mathrm{y}_0}{2} + \frac{1}{2}\left(\frac{n^3}{3} - \frac{n^2}{2}\right)\Delta^2 \mathrm{y}_0 + \frac{1}{6}\left(\frac{n^4}{4} - 3\frac{n^3}{3} + 2\frac{n^2}{2}\right)\Delta^3 \mathrm{y}_0 + \ldots\ldots \right]$$

$$= \mathrm{nh}\left[y_0 + \frac{\mathrm{n}\Delta \mathrm{y}_0}{2} + \frac{1}{2}\left(\frac{n^2}{3} - \frac{n}{2}\right)\Delta^2 \mathrm{y}_0 + \frac{1}{6}\left(\frac{n^3}{4} - 3\frac{n^2}{3} + 2\frac{n}{2}\right)\Delta^3 \mathrm{y}_0 + \ldots\ldots \right]$$

This is called **Newton-Cote's Quadrature for)mula.** ....(2)

Trapezoidal Rule:

Putting n = 1 in the above general formula, all differences higher than the first will become zero (since other differences do not exist if n = 1) and we get

$$\int_{x_0}^{x_1} f(x)\,dx = \int_{x_0}^{x_0+h} f(x)\,dx = \mathrm{h}\left[y_0 + \frac{1}{2}\Delta y_0 \right] = \mathrm{h}\left[y_0 + \frac{1}{2}(y_1 - y_0) \right] = \frac{h}{2}(y_0 + y_1)$$

and $\displaystyle\int_{x_1}^{x_2} f(x)\,dx = \int_{x_0+h}^{x_0+2h} f(x)\,dx = \mathrm{h}\left[y_1 + \frac{1}{2}\Delta y_1 \right] = \mathrm{h}\left[y_1 + \frac{1}{2}(y_2 - y_1) \right] = \frac{h}{2}(y_1 + y_2)$

$$\int_{x_2}^{x_3} f(x)\,dx = \int_{x_0+2h}^{x_0+3h} f(x)\,dx = \mathrm{h}\left[y_2 + \frac{1}{2}\Delta y_2 \right] = \mathrm{h}\left[y_2 + \frac{1}{2}(y_3 - y_2) \right] = \frac{h}{2}(y_2 + y_3)$$

....

Finally,

$$\int_{x_0+(n-1)h}^{x_0+nh} f(x)\,dx = \frac{h}{2}(y_{n-1} + y_n)$$

Hence,

$$\int_{x_0}^{x_n} f(x)\,dx = \int_{x_0}^{x_0+nh} f(x)\,dx = \int_{x_0}^{x_0+h} f(x)\,dx + \int_{x_0+h}^{x_0+2h} f(x)\,dx + \int_{x_0+2h}^{x_0+3h} f(x)\,dx + \ldots.. + \int_{x_0+(n-1)h}^{x_0+nh} f(x)\,dx$$

$$= \frac{h}{2}[y_0 + y_1] + \frac{h}{2}[y_1 + y_2] + \ldots\ldots\ldots + \frac{h}{2}(y_{n-1} + y_n)$$

$$= \frac{h}{2}[(y_0 + y_1) - 2(y_1 + y_2 + y_3 + y_4 + \ldots\ldots + y_{n-2} + y_{n-1}]$$

.... (3)

Simpson's 1/3rd Rule:

Putting n = 2 in Newton-Cotes Quadrature formula i.e., by replacing the curve y = f(x) by n/2 parabolas, we have

$$\int_{x_0}^{x_2} f(x)\,dx = 2h\left[y_0 + \frac{2}{2}\Delta y_0 + \frac{2(4-3)}{12}\Delta^2 y_0\right] = 2h\left[y_0 + \Delta y_0 + \frac{1}{6}\Delta^2 y_0\right]$$

$$= 2h\left[y_0 + (y_1 - y_0) + \frac{1}{6}(y_2 - 2y_1 + y_0)\right] \quad = 2h\left[\frac{1}{6}y_0 + \frac{2}{3}y_1 + \frac{1}{6}y_2\right]$$

$$= \frac{2h}{6}[y_0 + 4y_1 + y_2] = \frac{h}{3}[y_0 + 4y_1 + y_2]$$

Similarly, $\int_{x_2}^{x_4} f(x)\,dx = \frac{h}{3}[y_2 + 4y_3 + y_4]$

....

$\int_{x_{n-2}}^{x_n} f(x)\,dx = \frac{h}{3}[y_{n-2} + 4y_{n-1} + y_n]$ Adding all these integrals, we obtain

$$\int_{x_0}^{x_2} f(x)\,dx = \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + \ldots\ldots\ldots + \int_{x_{n-2}}^{x_n} f(x)\,dx$$

$$= \frac{h}{3}[y_0 + 4y_1 + y_2] + \frac{h}{3}[y_2 + 4y_3 + y_4] + \ldots\ldots\ldots + \frac{h}{3}[y_{n-2} + 4y_{n-1} + y_n]$$

$$= \frac{h}{3}\left[(y_0 + 4y_1 + y_2) + (y_2 + 4y_3 + y_4) + \ldots\ldots\ldots\ldots + (y_{n-2} + 4y_{n-1} + y_n)\right]$$

$$= \frac{h}{3}\left[(y_0 + y_n) + 4(y_1 + y_2 + y_3 + y_5 + y_{n-1}) + 2(y_2 + y_4 + y_6 + \ldots\ldots + y_{n-2})\right]$$

.... (4)

$$= \frac{h}{3}\left[\begin{array}{l} sum\ of\ the\ first\ and\ last\ ordnates) + 4(sum\ of\ the\ odd\ ordinates) \\ + 2(sum\ of\ the\ remaining\ even\ ordinates \end{array}\right]$$

With the convention that $y_0, y_2, y_4, \ldots, y_{2n}$ are even ordinates and $y_1, y_3, y_5, \ldots, y_{2n-1}$ are odd ordinates.

This is known as **Simpson's 1/3 rule** or simply **Simpson's rule**.

Simpson's 3/8 Rule:

n = 3 in Newton-Cote's Quadrature formula, all differences higher than the third will become zero and we obtain

$$\int_{x_0}^{x_3} f(x)\,dx = 3h\left[y_0 + \frac{3}{2}\Delta y_0 + \frac{3(6-3)}{12}\Delta^2 y_0 + \frac{3(3-2)^2}{24}\Delta^3 y_0\right]$$

$$\int_{x_0}^{x_3} f(x)\,dx = 3h\left[y_0 + \frac{3}{2}\Delta y_0 + \frac{3}{4}\Delta^2 y_0 + \frac{1}{8}\Delta^3 y_0\right]$$

$$\int_{x_0}^{x_3} f(x)\,dx = 3h\left[y_0 + \frac{3}{2}(y_1 - y_0) + \frac{3}{4}(y_2 - 2y_1 + y_0) + \frac{1}{8}(y_3 - 3y_2 + 3y_1 - y_0)\right]$$

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3}{8}h\left[y_0 + 3y_1 + 3y_2 + y_3\right]$$

Similarly,

$$\int_{x_3}^{x_6} f(x)\,dx = \frac{3}{8}\text{h}\left[y_3 + 3y_4 + 3y_5 + y_6\right] \quad \text{and so on.}$$

Adding all these integrals, from x_0 to x_n, where n is a multiple of 3, we get

$$\int_{x_0}^{x_n} f(x)\,dx = \int_{x_0}^{x_3} f(x)\,dx + \int_{x_3}^{x_6} f(x)\,dx + ... + \int_{x_{n-3}}^{x_n} f(x)\,dx$$

$$= \frac{3h}{8}\left[(y_0 + 3y_1 + 3y_2 + y_3) + (y_3 + 3y_4 + 3y_5 + y_6) + ... + (y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n)\right]$$

$$= \frac{3h}{8}\left[(y_0 + y_n) + 3(y_1 + y_2 + y_4 + y_5 + ... + y_{n-1}) + 2(y_3 + y_6 + y_9 + ... + y_n)\right]$$

.... (5)

Equation (5) is called **Simpson's 3/8 rule** which is applicable only when n is multiple of 3.

Memorise:

Trapezoidal Rule:

$$\int_a^b f(x)dx = \frac{h}{2}\left(y_0 + y_n + 2(y_1 + y_2 + y_3 + ... + y_{n-1})\right)$$

$$= \frac{h}{2}(sum\,of\;\;first\;and\;last\;ordinates + 2(sum\,of\;\;remaining\;ordinates))$$

Simpson's 1/3rd Rule:

$$\int_a^b f(x)dx = \frac{h}{3}\left(y_0 + y_n + 2(y_2 + y_4 + y_6 + ... + y_{n-2}) + 4(y_1 + y_3 + y_5 + ... + y_{n-1})\right)$$

$$= \frac{h}{2}(sum\,of\;\;first\;and\;last\;ordinates + 2(sum\,of\;\;even\;ordinates) + 4(sum\,of\;\;odd\;ordinates))$$

Note: For Simpsons 1/3rd Rule number of subintervals should be even.

Simpson's 3/8th Rule:

$$\int_a^b f(x)dx = \frac{h}{3}\left(y_0 + y_n + 2(y_3 + y_6 + y_9 + ...+ y_{n-3}) + 3(y_1 + y_2 + y_4 + y_5 + ...+ y_{n-2} + y_{n-1})\right)$$

$$= \frac{h}{2}(sum\, of\;\; first\, and\, last\, ordinates + 2(sum\, of\;\; multiples\, of\;\, 3\, ordinates) + 3(sum\, of\;\; remaining\, ordinates))$$

Note: For Simpsons 3/8th Rule number of subintervals should be multiple of 3.

Q . Evaluate $\int_0^1 \frac{dx}{1+x}$ using (i) Trapezoidal rule (ii) Simpson's one third rule (iii) Simpson's three eight rule. Take $h = \frac{1}{6}$ for all cases.

Solutions: Here $h = \frac{1}{6}$, Let $y = f(x) = \frac{1}{1+x}$. The values of f(x) for the points of subdivisions are as follows:

x	0	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{3}{6}$	$\frac{4}{6}$	$\frac{5}{6}$	1
$y = \frac{1}{1+x}$	1	0.8571	0.75	0.6667	0.6	0.5455	0.5

(i) Tapezoidal rule

$$\int_0^1 \frac{dx}{1+x} = \frac{h}{2}\left[(y_0 + y_6) + 2(y_1 + y_2 + y_3 + y_4 + y_5)\right]$$

$$; \frac{1}{12}\left[(1+0..5) + 2(0.8571 + 0.755 + 0.6667 + 0.6 + 0.5455)\right]$$

=0.6949.

(ii) Simpson's one third rule

$$\int_0^1 \frac{dx}{1+x} = \frac{h}{3}\left[(y_0 + y_6) + 2(y_2 + y_4) + 4(y_1 + y_3 + y_5)\right]$$

$$; \frac{1}{18}\left[(1+0.5) + 2(0.75 + 0.6) + 4(0.8571 + 0.6667 + 0.5455)\right]$$

= 0.6932.

(iii) Simpson's three eight rule

$$\int_0^1 \frac{dx}{1+x} = \frac{3h}{8}\left[(y_0 + y_6) + 3(y_1 + y_2 + y_4 + y_5) + 2y_3\right]$$

$$\approx \frac{1}{16}\left[(1+0.5)+3\left(0.8571+0.75+0.6+0.5455+2(0.6667)\right)\right]$$

= 0.6932.

MATLAB Code to numerical integration by using Trapezoidal Rule:

```
function intval=trapezoidal(f,a,b,n)
h=(b-a)/n;
x=linspace(a,b,n+1);
y=f(x)
intval=(h/2)*(y(1)+y(n+1)+2*sum(y(2:n)));
fprintf('\nthe approximate value of given integral is %f',intval)
```

MATLAB Code to numerical integration by using Simpson's $\frac{1}{3}rd$ Rule:

```
function intval=simpsons1(f,a,b,n)
if rem(n,2)==0
   h=(b-a)/n;
   x=linspace(a,b,n+1);
   y=f(x);
   intval=(h/3)*(y(1)+y(n+1)+2*sum(y(3:2:n-1))+4*(sum(y(2:2:n))));
   fprintf('\nthe approximate value of given integral is %f',intval)
else
   error('number of subintervals n must be even')
end
```

MATLAB Code to numerical integration by using Simpson's $\frac{3}{8}rd$ Rule:

```
function intval=simpsons2(f,a,b,n)
if rem(n,3)==0
   h=(b-a)/n;
   x=linspace(a,b,n+1);
   y=f(x);
   intval=(3*h/8)*(y(1)+y(n+1)+2*sum(y(4:3:n-2))+3*(sum(y)-y(1)-y(n+1)-
sum(y(4:3:n-2))));
   fprintf('\n the approximate value of given integral is %f',intval)
else
   error('number of subintervals n must be even')
end
```

Assignment-Cum-Tutorial Questions

Section-A

Objective Questions:

1. By Newton's forward interpolation formula

   $\frac{dy}{dx} =$ ________________________________

   $\frac{d^2y}{dx^2} =$ ________________________________

2. By Newton's backward interpolation formula

   $\frac{dy}{dx} =$ ________________________________

   $\frac{d^2y}{dx^2} =$ ________________________________

3. Trapezoidal rule to find definite integral is____________________
4. Simpson's $1/3^{rd}$ rule to find definite integral is____________________
5. Simpson's $3/8^{th}$ rule to find definite integral is____________________
6. If we put n = 2 in a general quadrature formula, we get [ ]

   (a) Trapezoidal rule (b) Simpson's $1/3^{rd}$ rule

   (c) Simpson's $3/8^{th}$ rule (d) Boole's rule
7. In Simpson's $1/3^{rd}$ rule the number of subintervals should be [ ]

   (a) Even (b) odd

   (c) multiples of 3's (d) more than 'n' interval
8. If the distance *d(t)* is traversed by a particle in the 't' sec and *d(0) = 0, d(2) = 8, d(4) = 20 and d(6) = 28*, then its velocity in cm after 6 secs is [ ]

   (a) 1.67 (b) 16.67 (c) 2 (d) 2.003
9. The formula $\frac{1}{h}\left[\Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0 + ...\right]$ is used only when the point x is at [ ]

   (a) end of the tabulated set (b) middle of the tabulated set

   (c) Beginning of tabulated set (d) none of these
10. To increase the accuracy in evaluating a definite integral by Trapezoidal rule, we should take ____________________
11. Values of y = f(x) are known as x = x_0, x_1 and x_2. Using Newton's forward integration formula, the approximate value of $\left(\frac{dy}{dx}\right)_{x=x_0}$ is

    ____
12. Numerical differentiation gives [ ]

    (a) exact value (b) approximate value

(c) no result (d) negative value

13. The general quadrature formula is [ ]
(a) always same (b)depends upon interpolation formula
(c) not easy to derive (d) is also given approximate result

14. For n =1in quadrature formula, $\int_{x_0}^{x_1} f(x)\,dx$ equals to [ ]

(a) $\frac{h}{2}(f_0 + f_1)$ (b) $(f_0 + f_1)$ (c) $\frac{h}{2}(f_0 - f_1)$ (d) $\frac{h}{4}(f_0 + f_1)$

15. To apply, Simpson's 1/3rd rule, always divide the given range of integration into 'n' subintervals, where n is [ ]
(a) even (b) odd (c) 1,2,3,4 (d) 1,3,5,7

16. The process of calculating derivative of a function at some particular value of the independent variable by means of a set of given values of that function is [ ]
(a) Numerical value (b) Numerical differentiation
(c) Numerical integration (d) quadrature

17. While evaluating definite integral by Trapezoidal rule, the accuracy can be increased by [ ]
(a) h =4 (b) even number of sub-intervals
(c) multiples of 3 (d) large number of sub-intervals

Section-B

Subjective Questions

1. A curve is expressed by the following values of x and y. Find the slope at the point x =1.5

x	0.0	0.5	1.0	1.5	2.0
y	0.4	0.35	0.24	0.13	0.05

2. The population of a certain town is given below. Find the rate of growth of the population in 1961:

Year	1931	1941	1951	1961	1971
Population	40.62	60.80	71.95	103.56	132.65

3. In a machine a slider moves along a fixed straight rod. Its distance x cms along the rod is given below for various values of time 't' seconds. Find the velocity and acceleration of the slider when t = 0.3

t(sec)	0	0.1	0.2	0.3	0.4	0.5	0.6
x(cms)	30.13	31.62	32.87	33.64	33.95	33.81	33.24

4. The velocity of a train which starts from rest is given by the following table being reckoned in minutes from the start and speed in miles per hour

Minutes	2	4	6	8	10	12	14	16	18
Miles per hour	10	18	25	29	32	20	11	5	2

Estimate approximately the total distance travelled in 20 minutes.

5. The distance covered by an athlete for the 50 meter is given in the following table

Time(sec)	0	1	2	3	4	5	6
Distance(meter)	0	2.5	8.5	15.5	24.5	36.5	50

Determine the speed of the athlete at t = 5 sec. correct to two decimals.

6. A curve is drawn to pass through the points given by following table:

x	1	1.5	2.0	2.5	3	3.5	4.0
y	2	2.4	2.7	2.8	3	2.6	2.1

Find the slope of the curve at x=1.25.

7. Write a MATLAB code to find derivatives at a point by using Newtons forward interpolation formula.
8. Prepare a MATLAB code to find derivatives at a point by using Newtons backward interpolation formula.
9. Evaluate $\int_0^2 e^{-x^3} dx$ using Simpson's rule taking h=0.25
10. A river is 80 meters wide. The depth 'd' in meters at a distance x from the bank is given in the following table. Calculate the cross section of the river using Trapizoidal rule.

x	10	20	30	40	50	60	70	80
d(x)	4	7	9	12	15	14	8	3

11. Compute the value of the definite integral $\int_4^{5.2} \log x dx$ or $\int_4^{5.2} \ln x dx$ using i.Trapezoidal Rule ii. Simpson's 1/3rd Rule and iii. Simpson's 3/8th Rule.
12. Create a MATLAB function file to find a definite integral by using Trapezoidal rule.
13. Design a MATLAB code to find a definite integral by using Simpson's 1/3rd rule.
14. Write a MATLAB function file to find a definite integral by using Simpson's 3/8th rule.

Section-C

GATE/IES/Placement Tests/Other competitive examinations

1. If *f(2) = 5, f(4) = 8, f(6) = 10, and f(8) = 16* then $f''(8)$ = __________

2. Using Simpson's 1/3rd rule, find the value of the integr $\int_{0.2}^{1.4} (\sin x - \log x + e^x)\, dx$

by taking 6 sub-intervals.

3. Minimum number of subintervals required to evaluate the integral $\int_{1}^{2} \frac{1}{x} dx$

by using Simpson's 1/3rd rule so that the value is corrected up to 4 decimal places.

4. The following table gives the velocity v of a particle at time 't'

t (seconds)	0	2	4	6	8	10	12
v meters per second	4	6	16	34	60	94	136

Find (i) the distance moved by the particle in 12 seconds and also (ii) the acceleration at t = 2 sec.

5. A rocket is launched from the ground. Its acceleration is registered during the first 80 seconds and is given in the table below. Using Simpson's 1/3rd rule, find the velocity of the rocket at t= 80 seconds.

t sec	0	10	20	30	40	50	60	70	80
f(cm/sec^2)	30	31.63	33.34	35.47	37.75	40.33	43.25	46.69	50.67

-----------------@-@-@-@-@----------------

UNIT–V: Numerical Solutions of Ordinary Differential Equations

Pre-requisite : Commands of MATLab

Objectives:

• To the numerical solutions of a first ordered Ordinary Differential Equation together with initial condition.

Syllabus:

Taylor's Series Method - Euler Method - Modified Euler Method - Runge – Kutta Fourth order Method.

Subject Outcomes:

At the end of the unit, Students will be able to

• solve Ordinary Differential equations using Numerical methods.

• implement the numerical methods in Matlab in order to solve Ordinary Differential equations.

The important methods of solving ordinary differential equations of first order numerically are as follows

- Taylors series method
- Euler's method
- Modified Euler's method of successive approximations
- Runge- kutta method

To describe various numerical methods for the solution of ordinary differential equations we consider the general 1st order differential equation

$dy/dx = f(x,y)$ ------- (1)

with the initial condition $y(x_0) = y_0$

The methods will yield the solution in one of the two forms:

i) A series for y in terms of powers of x, ,from which the value of y can be obtained by direct substitution.

ii) A set of tabulated values of y corresponding to different values of x.

TAYLOR'S SERIES METHOD

To find the numerical solution of the differential equation

$$\frac{dy}{dx} = f(x, y) \rightarrow (1)$$

With the initial condition $y(x_0) = y_0 \rightarrow$ (2)

$y(x)$ can be expanded about the point x_0 in a Taylor's series in powers of $(x - x_0)$ as

$$y(x) = y(x_0) + \frac{(x - x_0)}{1} y'(x_0) + \frac{(x - x_0)^2}{2!} y''(x_0) + \ldots\ldots\ldots\ldots + \frac{(x - x_0)^n}{n!} y^n(x_0) \rightarrow (3)$$

In equation3, $y(x_0)$ is known from Initial Condition . The remaining coefficients $y'(x_0), y''(x_0), \ldots\ldots\ldots y^n(x_0)$ etc are obtained by successively differentiating equation1 and evaluating at x_0. Substituting these values in equation3, $y(x)$ at any point can be calculated from equation3. Provided $h = x - x_0$ is small.

When $x_0 = 0$, then Taylor's series equ3 can be written as

$$y(x) = y(0) + x.y'(0) + \frac{x^2}{2!} y''(0) + \ldots\ldots + \frac{x^n}{n!} y^n(0) + \ldots\ldots\ldots \rightarrow (4)$$

<u>Note</u>: We know that the Taylor's expansion of y(x) about the point x_0 in a power of $(x - x_0)$is.

$$y(x) = y(x_0) + \frac{(x - x_0)}{1!} y^{I}(x_0) + \frac{(x - x_0)^2}{2!} y^{II}(x_0) + \frac{(x - x_0)^3}{3!} y^{III}(x_0) + \ldots \rightarrow (1)$$

Or

$$y(x) = y_0 + \frac{(x - x_0)}{1!} y_0^I + \frac{(x - x_0)^2}{2!} y_0^{II} + \frac{(x - x_0)^3}{3!} y_0^{III} + \ldots\ldots$$

If we let $x - x_0 = h$. (i.e. $x = x_0 + h = x_1$) we can write the Taylor's series as

$$y(x) = y(x_1) = y_0 + \frac{h}{1!} y_0^I + \frac{h^2}{2!} y_0^{II} + \frac{h^3}{3!} y_0^{III} + \frac{h^4}{4!} y_0^{IV} + \ldots.$$

$$\text{i.e. } y_1 = y_0 + \frac{h}{1!} y_0^I + \frac{h^2}{2!} y_0^{II} + \frac{h^3}{3!} y_0^{III} + \frac{h^{IV}}{4!} y_0^{IV} + \ldots\ldots \qquad \rightarrow (2)$$

Similarly expanding y(x) in a Taylor's series about x = x_1. We will get.

$$y_2 = y1 + \frac{h}{1!} y_1^I + \frac{h^2}{2!} y_1^{II} + \frac{h^3}{3!} y_1^{III} + \frac{h^4}{4!} y_1^{IV} + \ldots\ldots \quad \rightarrow (3)$$

Similarly expanding y(x) in a Taylor's series about x = x_2 We will get.

$$y_3 = y_2 + \frac{h}{1!} y_2^I + \frac{h^2}{2!} y_2^{II} + \frac{h^3}{3!} y_2^{III} + \frac{h^4}{4!} y_2^{IV} + \ldots\ldots \quad \rightarrow (4)$$

In general, Taylor's expansion of y(x) at a point x= x_n is

$$y_{n+1} = y_n + \frac{h}{1!} y_n^I + \frac{h^2}{2!} y_n^{II} + \frac{h^3}{3!} y_n^{III} + \frac{h^4}{4!} y_n^{IV} + \ldots.. \quad \rightarrow (5)$$

Example 1. Using Taylor's expansion evaluate the integral of $y' - 2y = 3e^x, y(0) = 0$ at $x = 0.2$. Hence compare the numerical solution obtained with exact solution .

Sol: Given equation can be written as $2y + 3e^x = y', y(0) = 0$

Differentiating repeatedly w.r.t to 'x' and evaluating at $x = 0$

$$y'(x) = 2y + 3e^x, y'(0) = 2y(0) + 3e^0 = 2(0) + 3(1) = 3$$
$$y''(x) = 2y' + 3e^x, y''(0) = 2y'(0) + 3e^0 = 2(3) + 3 = 9$$
$$y'''(x) = 2.y''(x) + 3e^x, y'''(0) = 2y''(0) + 3e^0 = 2(9) + 3 = 21$$
$$y^{iv}(x) = 2.y'''(x) + 3e^x, y^{iv}(0) = 2(21) + 3e^0 = 45$$
$$y^{v}(x) = 2.y^{iv} + 3e^x, y^{v}(0) = 2(45) + 3e^0 = 90 + 3 = 93$$

In general, $y^{(n+1)}(x) = 2.y^{(n)}(x) + 3e^x$ or $y^{(n+1)}(0) = 2.y^{(n)}(0) + 3e^0$

The Taylor's series expansion of $y(x)$ about $x_0 = 0$ is

$$y(x) = y(0) + xy'(0) + \frac{x^2}{2!} y''(0) + \frac{x^3}{3!} y'''(0) + \frac{x^4}{4!} y''''(0) + \frac{x^5}{5!} y'''''(0) + \ldots.$$

Substituting the values of $y(0), y'(0), y''(0), y'''(0), \ldots\ldots\ldots.$

$$y(x) = 0 + 3x + \frac{9}{2} x^2 + \frac{21}{6} x^3 + \frac{45}{24} x^4 + \frac{93}{120} x^5 + \ldots\ldots..$$

$$y(x) = 3x + \frac{9}{2} x^2 + \frac{7}{2} x^3 + \frac{15}{8} x^4 + \frac{31}{40} x^5 + \ldots\ldots.. \rightarrow \text{equ1}$$

Now put $x = 0.1$ in equ1

$$y(0.1) = 3(0.1) + \frac{9}{2}(0.1)^2 + \frac{7}{2}(0.1)^3 + \frac{15}{8}(0.1)^4 + \frac{31}{40}(0.1)^5 = 0.34869$$

Now put $x = 0.2$ in equ1

$$y(0.2) = 3(0.2) + \frac{9}{2}(0.2)^2 + \frac{7}{2}(0.2)^3 + \frac{15}{8}(0.2)^4 + \frac{31}{40}(0.2)^5 = 0.811244$$

$$y(0.3) = 3(0.3) + \frac{9}{2}(0.3)^2 + \frac{7}{2}(0.3)^3 + \frac{15}{8}(0.3)^4 + \frac{31}{40}(0.3)^5 = 1.41657075$$

Analytical Solution:

The exact solution of the equation $\frac{dy}{dx} = 2y + 3e^x$ with $y(0) = 0$ can be found as follows

$\frac{dy}{dx} - 2y = 3e^x$ Which is a linear in y.

Here $P = -2, Q = 3e^x$

I.F = $\int_e^{pdx} = \int_e^{-2dx} = e^{-24}$

General solution is $y.e^{-2x} = \int 3e^x.e^{-2x}dx + c = -3e^{-x} + c$

$\therefore y = -3e^x + ce^{2x}$ where $x = 0, y = 0$ $\quad 0 = -3 + c \Rightarrow c = 3$

The particular solution is $y = 3e^{2x} - 3e^x$ or $y(x) = 3e^{2x} - 3e^x$

Put $x = 0.1$ in the above particular solution,

$y = 3.e^{0.2} - 3e^{0.1} = 0.34869$

Similarly put $x = 0.2$, $y = 3e^{0.4} - 3e^{0.2} = 0.811265$

put $x = 0.3$, $y = 3e^{0.6} - 3e^{0.3} = 1.416577$

MATLAB CODE FOR THE IMPLEMENTATION OF TAYLOR SERIES METHOD:-

To find the numerical solution of the differential equation

$$\frac{dy}{dx} = f(x, y) \text{ with the initial condition } y(x_0) = y_0$$

```
function [ soltable ] = odetaylor( f,x0,y0,xn,h,not )

% here f is a symbolic function function f(x,y) such that y=y(x)

syms x y(x);

xd=x0:h:xn;

n=length(xd);

yd=zeros(1,n);

yd(1)=y0;

for i=1:n-1

   fold=f;

   d=ones(1,not-1);

      for j=1:not-1

      d(j)=subs(fold,{x,y},{xd(i),yd(i)});

      fold=subs(diff(fold,x),diff(y(x),x),fold);

      end

   yd(i+1)=yd(i)+sum(d.*(h.^(1:not-1)./factorial(1:not-1)));

end

soltable=[xd' yd'];

end
```

EULER'S METHOD

It is the simplest one-step method and it is less accurate. Hence it has a limited application.

Consider the differential equation $\frac{dy}{dx}$ = f(x,y) →(1)

With $y(x_0) = y_0$ →(2)

Consider the first two terms of the Taylor's expansion of y(x) at $x = x_0$

$y(x) = y(x_0) + (x - x_0)\, y^1(x_0)$ →(3)

from equation (1) $y^1(x_0) = f(x_0,y(x_0)) = f(x_0,y_0)$

Substituting in equation (3)

$\therefore y(x) = y(x_0) + (x - x_0)\, f(x_0,y_0)$

At $x = x_1$, $y(x_1) = y(x_0) + (x_1 - x_0)\, f(x_0,y_0)$

$\therefore y_1 = y_0 + h\, f(x_0,y_0)$ where $h = x_1 - x_0$

Similarly at $x = x_2$, $y_2 = y_1 + h\, f(x_1,y_1)$,

Proceeding as above, $y_{n+1} = y_n + h\, f(x_n,y_n)$

This is known as Euler's Method

Example 1. Using Euler's method solve for x = 2 from $\frac{dy}{dx} = 3x^2 + 1$, y(1) = 2, taking step size

(I) h = 0.5 and (II) h=0.25

Sol: Here $f(x,y) = 3x^2 + 1$, $x_0 = 1, y_0 = 2$

Euler's algorithm is $y_{n+1} = y_n + h\, f(x_n,y_n)$, n = 0,1,2,3,….. →(1)

h = 0.5 $\therefore x_1 = x_0 + h = 1+0.5 = 1.5$

Taking n = 0 in (1) , we have $x_2 = x_1 + h = 1.5 + 0.5 = 2$

$y_1 = y_0 + h\, f(x_0,y_0)$

i.e. $y_1 = y(0.5) = 2 + (0.5)\, f(1,2) = 2 + (0.5)(3 + 1) = 2 + (0.5)(4)$

Here $x_1 = x_0 + h = 1 + 0.5 = 1.5$

$\therefore y(1.5) = 4 = y_1$

Taking n = 1 in (1),we have

$y_2 = y_1 + h\, f(x_1,y_1)$

i.e. $y(x_2) = y_2 = 4 + (0.5)\, f(1.5,4) = 4 + (0.5)[3(1.5)^2 + 1] = 7.875$

Here $x_2 = x_4 + h = 1.5 + 0.5 = 2$

$\therefore y(2) = 7.875$

h = 0.25 $\therefore x_1 = 1.25, x_2 = 1.50, x_3 = 1.75, x_4 = 2$

Taking n = 0 in (1), we have

$y_1 = y_0 + h\, f(x_0,y_0)$

i.e. $y(x_1) = y_1 = 2 + (0.25)\, f(1,2) = 2 + (0.25)\,(3 + 1) = 3$

$y(x_2) = y_2 = y_1 + h\, f(x_1,y_1)$

i.e. $y(x_2) = y_2 = 3 + (0.25)\, f(1.25,3)$

$= 3 + (0.25)[3(1.25)^2 + 1]$

$= 4.42188$

Here $x_2 = x_1 + h = 1.25 + 0.25 = 1.5$

$\therefore y(1.5) = 5.42188$

Taking n = 2 in (1), we have

i.e. $y(x_3) = y_3 = h\, f(x_2,y_2)$

$= 5.42188 + (0.25)\, f(1.5,2)$

$= 5.42188 + (0.25)\, [3(1.5)^2 + 1]$

$= 6.35938$

Here $x_3 = x_2 + h = 1.5 + 0.25 = 1.75$

$\therefore y(1.75) = 7.\ 35938$

Taking n = 4 in (1),we have

$y(x_4) = y_4 = y_3 + h\, f(x_3,y_3)$

i.e. $y(x_4) = y_4 = 7.35938 + (0.25)\, f(1.75,2)$

$= 7.35938 + (0.25)[3(1.75)^2 + 1]$

$= 8.90626$

Note that the difference in values of y(2) in both cases (i.e. when h = 0.5 and when h = 0.25).The accuracy is improved significantly when h is reduced to 0.25 (Exact solution of the equation is $y = x^3 + x$ and with this $y(2) = y_2 = 10$.

MATLAB CODE FOR THE IMPLEMENTATION OF EULER'S METHOD:-

To find the numerical solution of the differential equation

$\frac{dy}{dx} = f(x, y)$ with the initial condition $y(x_0) = y_0$

```
function [ soltable ] = odeeuler( f,x0,y0,xn,h )

x=x0:h:xn;
n=length(x);
y=zeros(1,n);
y(1)=y0;
for i=1:n-1
y(i+1)=y(i)+h*f(x(i),y(i));
end
soltable=[x' y'];
end
```

Modified Euler's method

It is given by $y^{(i)}{}_{k+1} = y_k + h/2 f\left[\left(x_k, y_k\right) + f\left(x_{k+1}, 1\right)_{k+1}{}^{(i-1)}\right], i = 1, 2....., ki = 0, 1.....$

Working rule :

i) Modified Euler's method

$$y^{(i)}_{k+1} = y_k + h/2 f\left[\left(x_k, y_k\right) + f\left(x_{k+1}, 1\right)_{k+1}^{(i-1)}\right], i = 1, 2....., ki = 0, 1.....$$

ii) When $i=1$ y^{0}_{k+1} can be calculated from Euler's method

iii) k=0, 1……… gives number of iteration. $i = 1, 2...$

gives number of times, a particular iteration k is repeated

Suppose consider dy/dx=f(x, y) -------- (1) with $y(x_0) = y_0$----------- (2)2

To find $y(x_1) = y_1$ at $x = x_1 = x_0 + h$

Now take k=0 in modified Euler's method

We get $y_1^{(1)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(i-1)}\right)\right]$……………………… (3)

Taking i=1, 2, 3...k+1 in equation (3), we get

$y_1^{(0)} = y_0 + h/2\left[f\left(x_0, y_0\right)\right]$ (By Euler's method)

$$y_1^{(1)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(0)}\right)\right]$$

$$y_1^{(2)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(1)}\right)\right]$$

$$y_1^{(k+1)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(k)}\right)\right]$$

If two successive values of $y_1^{(k)}, y_1^{(k+1)}$ are sufficiently close to one another, we will take the common value as $y_2 = y\left(x_2\right) = y\left(x_1 + h\right)$

We use the above procedure again

Example1. Using modified Euler's method, find the approximate value of x when $x = 0.3$ given that $dy/dx = x + y$ and $y(0) = 1$

Sol: Given $dy/dx = x + y$ and $y(0) = 1$

Here $f(x, y) = x + y, x_0 = 0,$ and $y_0 = 1$

Take h = 0.1 which is sufficiently small

Here $x_0 = 0, x_1 = x_0 + h = 0.1, x_2 = x_1 + h = 0.2, x_3 = x_2 + h = 0.3$

The formula for modified Euler's method is given by

$$y_{k+1}^{(i)} = y_k + h/2\left[f\left(x_k + y_k\right) + f\left(x_{k+1}, y_{k+1}^{(i-1)}\right)\right] \rightarrow (1)$$

<u>Step1:</u> To find $y_1 = y(x_1) = y(0.1)$

Taking k = 0 in eqn(1)

$$y_{k+1}^{(i)} = y_0 + h/2\left[f\left(x_0 + y_0\right) + f\left(x_1, y_1^{(i-1)}\right)\right] \rightarrow (2)$$

when i = 1 in eqn (2)

$$y_1^{(i)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(0)}\right)\right]$$

First apply Euler's method to calculate $y_1^{(0)}$ = y_1

$$\therefore \; y_1^{(0)} = y_0 + h\, f\left(x_0, y_0\right)$$

= 1+(0.1)f(0.1)

= 1+(0.1)

= 1.10

$$now\left[x_0 = 0, y_0 = 1, x_1 = 0.1, y_1(0) = 1.10\right]$$

$$\therefore \; y_1^{(1)} = y_0 + 0.1/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(0)}\right)\right]$$

= 1+0.1/2[f(0,1) + f(0.1,1.10)

= 1+0.1/2[(0+1)+(0.1+1.10)]

= 1.11

When i=2 in equation (2)

$$y_1^{(2)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(1)}\right)\right]$$

= 1+0.1/2[f(0.1)+f(0.1,1.11)]

= 1 + 0.1/2[(0+1)+(0.1+1.11)]

= 1.1105

$$y_1^{(3)} = y_0 + h/2\left[f\left(x_0, y_0\right) + f\left(x_1, y_1^{(2)}\right)\right]$$

= 1+0.1/2[f(0,1)+f(0.1 , 1.1105)]

= 1+0.1/2[(0+1)+(0.1+1.1105)]

= 1.1105

Since $y_1^{(2)} = y_1^{(3)}$

$\therefore$ y_1 = 1.1105

<u>Step:2</u> To find $y_2 = y(x_2) = y(0.2)$

Taking k = 1 in equation (1) , we get

$$y_2^{(i)} = y_1 + h/2\left[f\left(x_1, y_1\right) + f\left(x_2, y_2^{(i-1)}\right)\right] \rightarrow (3)$$

I = 1,2,3,4,.....

For i = 1

$$y_2^{(1)} = y_1 + h/2\left[f\left(x_1, y_1\right) + f\left(x_2, y_2^{(0)}\right)\right]$$

$y_2^{(0)}$ is to be calculate from Euler's method

$$y_2^{(0)} = y_1 + h \; f\left(x_1, y_1\right)$$

= 1.1105 + (0.1) f(0.1 , 1.1105)

= 1.1105+(0.1)[0.1+1.1105]

= 1.2316

$$\therefore \; y_2^{(1)} = 1.1105 + 0.1/2\left[f\left(0.1, 1.1105\right) + f\left(0.2, 1.2316\right)\right]$$

= 1.1105 +0.1/2[0.1+1.1105+0.2+1.2316]

= 1.2426

$$y_2^{(2)} = y_1 + h/2\left[f\left(x_1, y_1\right) + f\left(x_2 y_2^{(1)}\right)\right]$$

= 1.1105 + 0.1/2[f(0.1 , 1.1105) , f(0.2 . 1.2426)]

= 1.1105 + 0.1/2[1.2105 + 1.4426]

= 1.1105 + 0.1(1.3266)

= 1.2432

$$y_2^{(3)} = y_1 + h/2\left[f\left(x_1, y_1\right) + f\left(x_2 y_2^{(2)}\right)\right]$$

= 1.1105+0.1/2[f(0.1,1.1105)+f(0.2 , 1.2432)]

= 1.1105+0.1/2[1.2105+1.4432)]

= 1.1105 + 0.1(1.3268)

= 1.2432

Since $y_2^{(3)} = y_2^{(3)}$

Hence y_2 = 1.2432

Step:3

To find $y_3 = y(x_3)$ = y y(0.3)

Taking k =2 in equation (1) we get

$$y_3^{(1)} = y_2 + h/2\left[f\left(x_2, y_2\right) + f\left(x_3, y_3^{(i-1)}\right)\right] \rightarrow (4)$$

For i = 1 ,

$$y_3^{(1)} = y_2 + h/2\left[f\left(x_2, y_2\right) + f\left(x_3, y_3^{(0)}\right)\right]$$

$y_3^{(0)}$ is to be evaluated from Euler's method .

$$y_3^{(0)} = y_2 + h\, f\left(x_2, y_2\right)$$

= 1.2432 +(0.1) f(0.2 , 1.2432)

= 1.2432+(0.1)(1.4432)

= 1.3875

$\therefore\ y_3^{(1)}$ = 1.2432+0.1/2[f(0.2 , 1.2432)+f(0.3, 1.3875)]

= 1.2432 + 0.1/2[1.4432+1.6875]

= 1.2432+0.1(1.5654)

= 1.3997

$$y_3^{(2)} = y_2 + h/2\left[\ f\left(x_2, y_2\right) + f\left(x_3, y_3^{(1)}\right)\right]$$

= 1.2432+0.1/2[1.4432+(0.3+1.3997)]

= 1.2432+ (0.1) (1.575)

= 1.4003

$$y_3^{(3)} = y_2 + h/2\left[\ f\left(x_2, y_2\right) + f\left(x_3, y_3^{(2)}\right)\right]$$

= 1.2432+0.1/2[f(0.2 , 1.2432)+f(0.3 , 1.4003)]

= 1.2432 + 0.1(1.5718)

= 1.4004

$$y_3^{(4)} = y_2 + h/2\left[\ f\left(x_2, y_2\right) + f\left(x_3, y_3^{(3)}\right)\right]$$

= 1.2432 + 0.1/2[1.4432+1.7004]

= 1.2432+(0.1)(1.5718)

= 1.4004

Since $y_3^{(3)} = y_3^{(4)}$

$\therefore$ The value of y at x = 0.3 is 1.4004

MATLAB CODE FOR THE IMPLEMENTATION OF Modified Euler's method METHOD:–

To find the numerical solution of the differential equation

$$\frac{dy}{dx} = f(x, y) \text{ with the initial condition } y(x_0) = y_0$$

```
function [ soltable ] = odemodifiedeuler( f,x0,y0,xn,h,tol )
x=x0:h:xn;
n=length(x);
y=zeros(1,n);
y(1)=y0;
for i=1:n-1
    yp=y(i)+h*f(x(i),y(i));
    y(i+1)=y(i)+h*(f(x(i),y(i))+f(x(i),yp))/2;
    while abs(yp-y(i+1)>tol)
        yp=y(i+1);
        y(i+1)=y(i)+h*(f(x(i),y(i))+f(x(i),yp))/2;
    end

end
soltable=[x' y'];
end
```

Runge – Kutta Methods

I. Second order R-K Formula

$y_{i+1} = y_i+1/2\ (K_1+K_2)$,

Where $K_1 = h\ (x_i, y_i)$

$K_2 = h\ (x_i+h, y_i+k_1)$ for i= 0,1,2-------

II. Third order R-K Formula

$y_{i+1} = y_i+1/6\ (K_1+4K_2+ K_3)$,

Where $K_1 = h\ (x_i, y_i)$

$K_2 = h\ (x_i+h/2,\ y_0+k_1/2)$

$K_3 = h\ (x_i+h,\ y_i+2k_2-k_1)$

For i= 0,1,2-------

III. Fourth order R-K Formula

$y_{i+1} = y_i+1/6\ (K_1+2K_2+ 2K_3 +K_4)$,

Where $K_1 = h\ (x_i, y_i)$

$K_2 = h\ (x_i+h/2,\ y_i+k_1/2)$

$K_3 = h\ (x_i+h/2,\ y_i+k_2/2)$

$K_4 = h\ (x_i+h,\ y_i+k_3)$

For i= 0,1,2-------

Example 1. Apply the 4^{th} order R-K method to find an approximate value of y when x=1.2 in stepsof 0.1, given that $y^1 = x^2+y^2$, y (1)=1.5

sol. Given $y^1= x^2+y^2$,and y(1)=1.5

Here $f(x,y)= x^2+y^{2,}$ y_0 =1.5 and x_0=1,h=0.1

So that x_1=1.1 and x_2=1.2

Step1:

To find y_1 :

By 4^{th} order R-K method we have

$y_1=y_0+1/6\ (k_1+2k_2+2k_3+k_4)$

$k_1= h\ f(x_0,y_0) = (0.1)\ f(1,1.5) = (0.1)\ [1^2+(1.5)^2] = 0.325$

$k_2= hf\ (x_0+h/2,y_0+k_1/2) = (0.1)\ f(1+0.05,1.5+0.325) = 0.3866$

and

$k_3= h\ f((x_0+h/2,y_0+k_2/2) = (0.1)\ f(1.05,1.5+0.\ 3866/2) = (0.1)[(1.05)^2+(1.6933)^2]$

= 0.39698

$k_4= hf(x_0+h,y_0+k_3) = (0.1)f(1.0,1.89698) = 0.48085$

Hence

$$y_1 = 1.5 + \frac{1}{6}\left[0.325 + 2(0.3866) + 2(0.39698) + 0.48085\right]$$
$$= 1.8955$$

Step2:

To find y_2, i.e., $y(x_2) = y(1.2)$

Here x_1=0.1,y_1=1.8955 and h=0.1

by 4th order R-K method we have

$y_2 = y_1+(1/6)\ (k_1+2k_2+2k_3+k_4)$

k_1= hf(x_1,y_1) = (0.1)f(0.1,1.8955) = (0.1) [1^2+$(1.8955)^2$] = 0.48029

k_2 = hf (x_1+h/2,y_1+k_1/2) = (0.1)f(1.1+0.1,1.8937+0.4796) = 0.58834

k_3 = hf((x_1+h/2,y_1+k_2/2) = (0.1)f(1.5,1.8937+0.58743) = (0.1)[$(1.05)^2$+$(1.6933)^2$]

= 0.611715

k_4 = hf(x_1+h,y_1+k_3) = (0.1)f(1.2,1.8937+0.610728) = 0.77261

Hence

y_2 = 1.8937+(1/6) (0.4796+2(0.58834)+2(0.611715)+0.7726) = 2.5043

$\therefore$ y =2.5043 where $x = 0.2$.

MATLAB CODE FOR THE IMPLEMENTATION OF 4th order R-K METHOD: –

To find the numerical solution of the differential equation

$\frac{dy}{dx} = f(x, y)$ with the initial condition $y(x_0) = y_0$

```
function [ soltable ] = odeRK( f,x0,y0,xn,h )

n=length(x);

y=zeros(1,n);

y(1)=y0;

for i=1:n-1
```

```
    k1=h*f(x(i),y(i));
    k2=h*f(x(i)+h/2,y(i)+k1/2);
    k3=h*f(x(i)+h/2,y(i)+k2/2);
    k4=h*f(x(i)+h,y(i)+k3);
    y(i+1)=y(i)+(k1+2*(k2+k3)+k4)/6;
end
soltable=[x' y'];
end
```

Assignment-cum-Tutorial Questions

A. Questions testing the remembering / understanding level of students

I Objective Questions

1. If $\frac{dy}{dx}$ = f(x,y),y(x_0) = y_0,the formula for fourth order Runge – Kutta method is _______

2. In which of the following methods, successive approximations are used?

a) Picard's method b) Taylor series method

c) Adams-Bashforth method d) None of these

3. Which among the following is the self-starting method?

a) Adams-Bashforth method b) Milne's method

c) Runge-Kutta method d) Predictor method

4. Among the following, which is the best for solving initial value problem?

a) Modified Euler's method b) Picard's method

c) Runge-Kutta method of fourth order d) Taylor series method

5. Which of the following is a step-by-step method?

a) Picard's method b) Taylor series method

c) Adams-Bashforth method d) None of these

B. Questions testing the ability of students in applying the concepts

I) Multiple choice Questions

1. If $\frac{dy}{dx}$ = -y, y(0) = 1, h = 0.01 then by Euler's method, the value of y_1 = __________

a) 0.099 b) 0.0981 c) 0.99 d) none

2. If $\frac{dy}{dx} = \frac{y-x}{y+x}$, y(0) = 1 and h = 0.02, using Euler's method the value of y_1 = _________

a) 1.02 b) 1.04 c) 1.03 d) none

3. If $\frac{dy}{dx} = x^2+y^2$, y(0) = 0 using Taylor's series method, the value of y(0.4)=___________

a) 0.2133 b) 0.02133 c) 0.002133 d) None

4. The value of y at x= 0.1 using Runge – Kutta method of fourth order for the differential equation $\frac{dy}{dx}$ = x – 2y, y(0) = 1 taking h = 0.1 is __________

a) 0.825 b) 0.0825 c) 0.813 d) None

5. The value of y at x = 0.1 using modified Euler's method up to second approximation for dy/dx = x – y, y(0) = 1 is _________

a) 0.909 b) 0.0909 c) 0.809 d) None

6 If $\frac{dy}{dx} = 1 + y^2$, $f(x_0,y_0)$ =1,h – 0.2,K_1 = 0.2,K_2 = 0.202,K_3 = 0.20204,K_4 = 0.20216, then the value of y_1 by fourth order Runge – Kutta method is ________

a) 0.0202 b) 0.202 c) 0.102 d) None

7. Using Runge – Kutta method, the approximate value of y(0.1) if $\frac{dy}{dx} = x + y^2$,y = 1

where x =0 and $f(x_0,y_0) = 1$ $K_1 = 0.1$, $K_2 = 0.115$, $K_3 = 0.116$, $K_4 = 0.134$ is

a) 1.116 b) 1.001 c) 1.211 d) None

8. Using Runge-kutta method, to solve the differential equation $\frac{dy}{dx} = x + y$, h=0.1 and y(0) = 1.

(i) The values of k_1, k_2, k_3 and k_4 respectively are

a) 0.11, 0.121, 0.1, 0.005

b) 0.1, 0.11, 0.1105, 0.12105

c) 0.111, 0.11105, 0.121005, 0.121

d) None of these

(ii) For the above problem y(0.1) = ________

a) 1.11034 b) 1.33011 c) 1.43001 d) None of these

II Problems

1. Solve $y^1 = x - y^2$, y(0) = 1 using Taylor's series method and evaluate y(0.1), y(0.2).
2. Given $y^1 = x + \sin y$, y(0)=1 compute y(0.2) and y(0.4) with h=0.2 using modified Euler's method
3. Using R-K method, find y(0.2) for the equation dy/dx=y-x , y(0)=1,take h=0.22.

   given that y=1 when x=0 and $\frac{dy}{dx} = x - y$
4. Using Taylor's series method, solve the equation $\frac{dy}{dx} = x^2 + y^2$ for $x = 0.4$ given that $y = 0$ when $x = 0 4$.
5. Find the solution of $\frac{dy}{dx}$ = x-y , y(0)=1 at x =0.1 , 0.2 ,0.3 , 0.4 and 0.5 using modified Euler's method.
6. Write the R-K method of 4th order formula for the solution y_1 of $\frac{dy}{dx} = f(x, y), y(x_0) = y_0$.
7. Using R-K method, estimate y(0.2) and y(0.4) for the equation $dy/dx = y^2 - x^2 / y^2 + x^2$, y(0)=1, h=0.2.
8. Explain the Modified Euler's method.

9. Find y(0.1) & y(0.2) using Euler's modified formula given that $\frac{dy}{dx} = x^2 - y$, y(0) = 1
10. Find y(0.1) & y(0.2) using Runge - Kutta 4th order formula given that $y^1 = x^2 - y$ & y(0) =1.
11. Write a code in MATLAB to find the numerical solution of first order ordinary differential equation using R-K Method of fourth order.
12. Write a code in MATLAB to find the numerical solution of first order ordinary differential equation using Euler's Method.

C. Questions testing the analyzing / evaluating ability of students

1. Solve by Taylor's series method, the equation $\frac{dy}{dx} = \log xy$ for y(1.1) and y(1.2), given y(1) = 2.
2. Using R – K method, solve for y at x = 1.2, 1.4 from $\frac{dy}{dx} = \frac{2xy + e^x}{x^2 + xe^x}$ given y(1) = 0.

GATE QUESTIONS

1. Consider an ordinary differential equation dx/dt=4t+4 if x =0 at t = 0, the increment in x calculated using Runge-Kutta fourth order multi-step method with a step size of $\Delta t = 0.2$ is ___________ (GATE2014)

   (A) 0.22 (B) 0.44 (C) 0.66 (D) 0.88

UNIT-VI

CURVE FITTING

Objectives:

- To understand the application of '**Least Square Method**'

Pre-requisite : Commands of MATLab

Syllabus: Fitting a straight line - Parabolic curve - exponential curve - power curve by the method of least squares.

Learning Outcomes:

At the end of the unit, Students will be able to

- Fit a straight line to the given data.
- Fit a Parabola to the given data.
- Fit an exponential curve and a power curve the given data.

Least Square Method:

The principle of least squares is one of the popular methods for finding a curve fitting a given data. Say $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be ***n*** observations from an experiment. We are interested in finding a curve

$$y = f(x) \qquad (1)$$

Closely fitting the given data of size '***n***'. Now at $x = x_1$ while the observed value of y is y_1, the expected value of y from the curve (1) is $f(x_1)$. Let us define the residual by

$$e_1 = y_1 - f(x_1) \quad \ldots\ldots..(2)$$

Likewise, the residuals at all other points x_2, ..., x_n are given by

$$e_2 = y_2 - f(x_2)$$

....................(3)

....

$$e_n = y_n - f(x_n)$$

Some of the residuals $e_i's$ may be positive and some may be negative. We would like to find the curve fitting the given data such that the residual at any x_i is as small as possible. Now since some of the residuals are positive and others are negative and as we would like to give equal importance to all the residuals it is desirable to consider sum of the squares of these residuals, say E and thereby find the curve that minimizes E. Thus, we consider

$$E = \sum_{i=1}^{n} e_i^2 \qquad (4)$$

and find the best representative curve (1) that minimizes (4).

2.4.2 **Least Square Fit of a Straight Line**

Suppose that we are given a data set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ of 'n' observations from an experiment. Say that we are interested in fitting a straight line

$$y = ax + b$$

to the given data. Find the 'n' residuals e_i's by:

$$e_i = y_i - (ax_i + b), \qquad i = 1, 2.....n \qquad (2)$$

Now consider the sum of the squares of e_i's i.e

$$E = \sum_{i=1}^{n} e_i^2$$

$$= \sum_{i=1}^{n} [y_i - (ax_i + b)]^2 \qquad (3)$$

Note that E is a function of parameters a and b. We need to find a,b such that E is minimum. The necessary condition for E to be minimum is given by:

$$\frac{\partial E}{\partial a} = \frac{\partial E}{\partial b} = 0 \qquad (4)$$

The condition $\frac{\partial E}{\partial a} = 0$ yields:

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{n} \quad 2x_i[y_i - (ax_i + b)] = 0$$

$$a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \qquad (5)$$

i.e

Similarly the condition $\frac{\partial E}{\partial b} = 0$ yields

$$a\sum_{i=1}^{n} x_i + nb = \sum_{i=1}^{n} y_i \qquad (6)$$

Equations (5) and (6) are called as normal equations,which are to be solved to get desired values for a and b.

The expression for E i.e (3) can be re-written in a convenient way as follows:

$$E = \left(\sum_{i=1}^{n} y_i^2 - a\sum_{i=1}^{n} x_i y_i - b\sum_{i=1}^{n} y_i\right) \qquad (7)$$

Example: Using the method of least squares, find an equation of the form

$y = ax + b$ that fits the following data:

x	0	1	2	3	4
y	1	5	10	22	38

Solution: Consider the normal equations of least square fit of a straight line i.e

$$a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \qquad (1)$$

$$a\sum_{i=1}^{n} x_i + nb = \sum_{i=1}^{n} y_i \qquad (2)$$

Here n=5.

From the given data, we have,

x	y	xy	x^2
0	1	0	0
1	5	5	1
2	10	20	4
3	22	66	9
4	38	152	16
$\sum_i x_i = 10$	$\sum_i y_i = 76$	$\sum_i x_i y_i = 243$	$\sum_i x_i^2 = 30$

Therefore the normal equations are given by:

30a +10b =243 ....(3)

10a+5b=76.......................(4)

On solving (3) and (4) we get

a = 9.1 , b= - 3 ....(5)

Hence the required fit for the given data is

y=9.1x - 3 ....(6)

Remarks:

(1) Experimental data may not be always linear. One may be interested in fitting either a curve of the form$(a) y = ax^b \;\; or \; (b) \; y = ae^{bx}$ However, both of these forms can be linearized by taking logarithms on both the sides. Let us look at the details:

$$Case \quad a) \qquad y = ax^b \qquad (1)$$

On taking logarithms on both the sides we get:

$$\log_{10} y = \log_{10} a + b \log_{10} x \quad (2)$$

$$Y = \log_{10} y \ , \quad A = \log_{10} a \ , \quad X = \log_{10} x \quad (3)$$

Say

Using (3) in (2) we get

$$Y = A + bX \qquad (4)$$

which is linear in X, Y.

$$Case \quad b) \quad y = ae^{bx} \qquad (5)$$

On taking logarithms we get

$$\log_{10} y = \log_{10} a + bx \log_{10} e \quad (6)$$

$$\log_{10} y = Y, \quad A = \log_{10} a \ , \quad b \log_{10} e = B \qquad (7)$$

$\therefore$ we get

$$Y = A + Bx \quad (8)$$

which is linear in Y, x.

Example: By the method of least square fit a curve of the form $y = ax^b$ to the following data:

x	2	3	4	5
y	27.8	62.1	110	161

Solution.

Consider $y = ax^b$ **----(1)**

On taking logarithm on both the sides we get

$$log_{10}\, y = log_{10}\, a + b\, log_{10} x \qquad (2)$$

$$Y = log_{10}\, y \ , \ A = log_{10}\, a \ , \ X = log_{10}\, x \qquad (3)$$

Using (3) in (2) we get

$$Y = A + bX \qquad (4)$$

Data in modified variables X,Y

X	0.3010	0.4771	0.6021	0.6990
Y	1.4440	1.7931	2.0414	2.2068

Normal equations corresponding to the straight line fit (4) are:

$$b\sum_{i=1}^{4} X_i^2 + A\sum_{i=1}^{4} X_i = \sum_{i=1}^{4} X_i Y_i \qquad (5)$$

$$b\sum_{i=1}^{4} X_i + nA = \sum_{i=1}^{4} Y_i \qquad (6)$$

From the modified data we get

X	Y	XY	X^2
0.3010	1.4440	0.4346	0.0906
0.4771	1.7931	0.8555	0.2276
0.6021	2.0414	1.2291	0.3625
0.6990	2.2068	1.5426	0.4886
$\sum_i X_i = 2.0792$	$\sum_i Y_i = 7.4853$	$\sum_i X_i Y_i = 4.0618$	$\sum_i X_i^2 = 1.1693$

$\therefore$ normal equations take the form:

$$1.1693b + 2.0792A = 4.0618$$

$$2.0792b + 4A = 7.4853$$

On solving for **b & A,** we obtain,

$$b = 1.9311, \quad A = 0.8678.$$

$$a = antilog\,\mathrm{A} = 7.375$$

.

∴ The desired curve is $y = 7.375x^{1.9311}$

Matlab code to fit exponential curve of the form y=a*e^(bx)

```
function [a,b]=exp1fit(x,y) % to fit y=a*e^(bx)

% on applying logarithm ln, we have lny=lna+bx

n=length(x);

A=[sum(x) n;sum(x.*x) sum(x)];

B=[sum(ln(y));sum(ln(y).*x)];

coef=A\B;

a=exp(coef(2)); % since coef(2)=lna

b=coef(1);

end
```

Matlab code to fit exponential curve of the form y=a*b^x

```
function [a,b]=exp2fit(x,y) % to fit y=a*(b^x)

% on applying log10 we have logy=loga+xlogb

n=length(x);
```

```
A=[sum(x) n;sum(x.*x) sum(x)];

B=[sum(log10(y));sum(log10(y).*x)];

coef=A\B;

a=10^coef(2);

b=10^coef(1);

end
```

Matlab code to fit power curve of the form y=a*x^b

```
function [a,b]=powerfit(x,y) % to fit y=a*(x^b)

% on applying logarithm ln, we have lny=lna+blnx

n=length(x);

A=[sum(ln(x)) n;sum(ln(x).*ln(x)) sum(ln(x))];

B=[sum(ln(y));sum(ln(y).*ln(x))];

coef=A\B;

a=exp(coef(2)); % since coef(2)=lna

b=coef(1);

end
```

Least Square fit of a parabola

Given a data set of n observations $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ of an experiment .Now we try to fit a best possible parabola

$$y = ax^2 + bx + c \qquad (1)$$

following the principle of least square. Finding the appropriate parabola amounts to determining the constants **a,b,c** that minimize the sum of the squares of the residuals $e_i's, i = 1,2, \ldots n$ given by

$$E = \sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n} [y_i - (ax_i^2 + bx_i + c)]^2 \qquad (2)$$

The necessary condition for *E* to be minimum is

$$\frac{\partial E}{\partial a} = \frac{\partial E}{\partial b} = \frac{\partial E}{\partial c} = 0 \qquad (3)$$

Now the condition $\frac{\partial E}{\partial a} = 0$ yields

$$\frac{\partial E}{\partial a} = -\sum_{i=1}^{n} 2[y_i - (ax_i^2 + bx_i + c)](x_i^2) = 0$$

$$a\sum_{i=1}^{n} x_i^4 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i^2 y_i \qquad (4)$$

i.e

Similarly $\frac{\partial E}{\partial b} = 0$ yields

$$\frac{\partial E}{\partial b} = -\sum_{i=1}^{n} 2[y_i - (ax_i^2 + bx_i + c)]x_i = 0$$

$$a\sum_{i=1}^{n} x_i^3 + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \qquad (5)$$

i.e

Finally $\frac{\partial E}{\partial c} = 0$ yields

$$a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i + nc = \sum_{i=1}^{n} y_i \qquad (6)$$

Equations (4), (5) and (6) are called as normal equations whose solution yields the values of the constants a, b and c and thus the desired parabola.

Example: Given the following data from an experimental observation

y:	9.4	11.8	14.7	18.0	23.0
x:	1.0	1.6	2.5	4.0	6.0

fit a parabola in the form $y = ax^2 + bx + c$ following the principle of least square.

Solution) Here **n=5**

The normal equations for finding a parabolic fit are:

$$a\sum_i x_i^4 + b\sum_i x_i^3 + c\sum_i x_i^2 = \sum_i x_i^2 y_i$$

$$a\sum_i x_i^3 + b\sum_i x_i^2 + c\sum_i x_i = \sum_i x_i y_i \quad (1)$$

$$a\sum_i x_i^2 + b\sum_i x_i + nc = \sum_i y_i$$

x	y	x^2	x^3	x^4	xy	x^2y
1.0	9.4	1.0	1.000	1.0000	9.40	9.400
1.6	11.8	2.56	4.096	6.5536	18.88	30.208
2.5	14.7	6.25	15.625	39.0625	36.75	91.875
4.0	18.0	16.00	64.00	256.0000	72.00	288.000
6.0	23.0	36.00	216.00	1296.0000	138.00	828.000
$\sum_i$ =15.1	76.9	61.81	110.721	1598.6161	275.03	1247.483

$\therefore$ The normal equations are:

$$1598.6161a + 110.721b + 61.81c = 1247.483$$
$$110.721a + 61.81b + 15.1c = 275.03 \qquad (2)$$
$$61.81a + 15.1b + 5c = 76.9$$

On Solving (2) for **a,b,c** we get

$$a = 1.0390, b=7.5088, c=-20.1411$$

Matlab code to fit quadratic curve i.e., parabola y=ax^2+bx+c :-

```
function [coef]=parabolafit(x,y)

% this is the function file to fit parabola whose

% equation is y=ax^2+bx+c

n=length(x);

% Coefficient matrix of the normal equations to fit y=ax^2+bx+c
```

```
A=[sum(x.^2) sum(x) n;
   sum(x.^3) sum(x.^2) sum(x);
   sum(x.^4) sum(x.^3) sum(x.^2)];
B=[sum(y);sum(y.*x);sum(y.*(x.^2))];
disp([A,B]);
coef=A\B;
end
```

Assignment-Cum-Tutorial Questions

Section-A

Objective Questions:

1. Write the normal equation in method of least squares to fit a straight line.
2. Write the normal equation in method of least squares to fit a parabola.
3. Is it necessary that the curve due to method of least squares agree at the points in the given data?
4. In method of least squares to fit a straight line, is the following statement is ture?
   "The mean of the data points is always a point on the straight line."

Section-B

Subjective Questions

1. Derive the normal equations in fitting a straight line in Method of least squares.

2. Derive the normal equations in fitting a parabola in Method of least squares.

3. If p is the pull required to lift the weight by means of a ulley block, find a linear law of the form p=a+bw, connecting p and w, using the following data:

w (lb):	50	70	100	120
p (lb):	12	15	21	25

Compute p, when w=150lb.

4. The results of measurement of electric resistance R of a copper bar at various temperatures t^oC are listed below:

t:	19	25	30	36	40	45	50
R:	76	77	79	80	82	83	85

Find a relation R=a+bt when a and b are constants to be determined by you.

5. Write a code in MATLAB to implement Least Squares Method to fit a straight line.

6. Derive the normal equations in fitting a parabola in Method of least squares.

7. Write a code in MATLAB to implement Least Squares Method to fit a parabola curve.

8. Fit a parabola $y= a+bx+cx^2$ to the following data:

x:	2	4	6	8	10
y:	3.07	12.85	31.47	57.38	91.29

9. The following table gives the results of the measurements of train resistances; V is the velocity in miles per hour. R is the resistance in pounds per ton:

V:	20	40	60	80	100	120
R:	5.5	9.1	14.9	22.8	33.3	46.0

If R is related to V by the relation $R=a+bV+cV^2$, find a, b and c.

10. Fit the exponential curve $y = ae^{bx}$ to the following data:

x:	2	4	6	8
y:	25	38	56	84

11. Predict y at x=3.75 , by fitting a power curve to the given data:

x:	1	2	3	4	5	6
y:	298	4.26	5.21	6.10	6.80	7.50

12. Write a code in MATLAB to implement Least Squares Method to fit a power curve.