INSTITUTE OF INFORMATION TECHNOLOGY

JAHANGIRNAGAR UNIVERSITY

Lab Report : 05

Submission Date : 07/12/2020

Course Tittle : Numerical Analysis Lab

Course Code : ICT - 2106

Submitted To	**Submitted By**
Dr M Shamim Kaiser	MD. Shakil Hossain
Professor	Roll – 2023
IIT – JU	2^{nd} year 1^{st} Semester
	IIT – JU

1.Write a program to find out y(10) and y(1) for the following tabular data

x	x_0=3	x_1=4	x_2=5	x_3=6	x_4=7	x_5=8	x_6=9
y	y_0=2.7	y_1=6.4	y_2=12.5	y_3=21.6	y_4=34.3	y_5=51.2	y_6=72.9

Problem 1(a).

Code:

```
clc;
close all;
clear all;

x = [3 4 5 6 7 8 9];
y = [2.7 6.4 12.5 21.6 34.3 51.2 72.9];
t = length(x);
x1 = 1;

P = 0;

for j = 1:t
    P1 = 1;
    for k = 1:t
        if (k ~= j)
            P1 = P1*((x1-x(k))/(x(j)-x(k)));
        end
    end
    P = P+P1*y(j);
end
P
```

Output:

```
>> Lab5_Problem1


P =

   0.1000
```

Problem 1(b).

Code:

```
clc;
close all;
clear all;

x = [3 4 5 6 7 8 9];
y = [2.7 6.4 12.5 21.6 34.3 51.2 72.9];
t = length(x);
x1 = 10;

P = 0;

for j = 1:t
    P1 = 1;
    for k = 1:t
        if (k ~= j)
            P1 = P1*((x1-x(k))/(x(j)-x(k)));
        end
    end
    P = P+P1*y(j);
end
P
```

Output:

```
>> Lab5_Problem1

P =

   100.0000
```

2.Write a program to find out y(3) for the following tabular data

x	x_0=0	x_1=1	x_2=2	x_3=4
y	y_0=2	y_1=5	y_2=9	y_3=12

Problem 2.

Code:

```
clc;
close all;
clear all;

x = [0 1 2 4];
y = [2 5 9 12];
t = length(x);
x1 = 3;

P = 0;

for j = 1:t
    P1 = 1;
    for k = 1:t
        if (k ~= j)
            P1 = P1*((x1-x(k))/(x(j)-x(k)));
        end
    end
    P = P+P1*y(j);
end
P
```

Output:

```
>> Lab5_Problem2


P =

   12
```

3. If y_1=4, y_3=12, y_4=19 and y_x=7, then write and program to find x.

Problem 3.

Code:

```
clc;
close all;
clear all;

x = [1 3 4];
y = [4 12 19];
t = length(x);
y1 = 7;
x1=0;
for j = 1:t
    x2 = 1;
    for k = 1:t
        if (k ~= j)
          x2=x2*((y1-y(k))/(y(j)-y(k)));
        end
    end
    x1 = x1+x2*x(j);
end
x1
```

Output:

```
>> Lab5_Problem3

x1 =

   1.8571
```

5. Write the limitation of Newton's Interpolation Formulae.

Limitations of Newton interpolation formula:

(1) There is always a chance of committing some error.

(2) The calculation provides no check whether the functional values used are correct or not.

(3) We face some difficulties when we have to find the central value of the function.

6. What do you mean by interpolation, extrapolation and inverse interpolation?

Interpolation:

Interpolation is a method of constructing new data points within the range of discrete set of known data points. In engineering and science one often has a number of data points, are obtained by sampling or experimentation, and try to construct a function which closely fits those data points. This is called "Curve Fitting" or "Regression" analysis. Interpolation is a specific case of curve fitting in which the function must go exactly through the data points.

Extrapolation:

Extrapolation is used if we want to find data points outside the range of known data points. In other words, extrapolation is a method in which the data values are considered as points such as $x_1, x_2, \ldots, x_n$.

Inverse interpolation:

Using interpolation methods, we found the value of the entry y for an intermediate value of the argument x, from a given table of value of x and y. Sometimes we have to find the value of x for a given values of y not in the table. This reverse process is known as inverse interpolation. Thus inverse interpolation is defined as the process of finding the value of the argument corresponding to a given value of the function lying between two tabulated functional values.

7. Discuss on the experiment.

In this experiment, we use 3 formula.

In exercise 1, we use Newton forward and backward interpolation.

In exercise 2, we use Lagrange's interpolation.

In exercise 3, we use Lagrange's inverse interpolation.

Newton forward and backward interpolation:

Named after Sir Isaac Newton, Newton's Interpolation is a popular polynomial interpolating technique of numerical analysis and mathematics. Here, the coefficients of polynomials are calculated by using divided difference, so this method of interpolation is also known as Newton's divided difference interpolation polynomial. Newton polynomial interpolation consists of Newton's forward difference formula and Newton's backward difference formula.

Newton's forward difference formula:

If,

r=(x[i+1] – x[i])

where, i = 0, 1, ...length- 1 and x = x0 +rp where p=(r*r-i)/i!.....etc.

N(x) = [y0] + [y0, y1]sh + . . . + [y0, y1, ...yk] s(s – 1) . . . (s – k +1) h^k.

The Newton's backward can be expresses as:

N(x) = [yk] + [yk, yk-1] (x – xk) + . . . + [yk, yk-1, . . . y0] (x – xk) (x – xk-1) . . . (x – x1)

If the xk, xk-1, ... x0 are equally spaced and x = xk + sh and xi = xk– (k –i)h for i = 0, 1, . . k

Now, Newton's backward becomes:

N(x) = [yk] + [yk, yk-1]sh + . . . + [yk, yk-1, ...y0]s(s+1) . . . (s+ k – 1) hk

we use this law and find the unknown value using matlab.

Lagrange's interpolation:

Newton's interpolation Formulae is not applicable where values of x are unequally spaced. In that case Lagrange's interpolation formula is applicable, which is,

$$y(x) = \frac{(x-x_1)(x-x_2)......(x-x_n)}{(x_0-x_1)(x_0-x_2)......(x_0-x_n)} * y_0 + \frac{(x-x_0)(x-x_2).....(x-x_n)}{(x_1-x_0)(x_1-x_2).......(x_1-x_n)} * y_1 + \frac{(x-x_0)(x-x_1)......(x-x_n)}{(x_2-x_0)(x_2-x_1).....(x_2-x_n)} * y_2$$

$$+ \ ....+ \frac{(x-x_1)(x-x_2).........(x-x_{n-1})}{(x_n-x_1)(x_n-x_2).......(x_n-x_{n-1})} * y_n$$

Lagrange's inverse interpolation:

To find out value of x for a given value of y from a data table we should apply Inverse Lagrange's formula, which is

$$x(y) = \frac{(y-y_1)(y-y_2)......(y-y_n)}{(y_0-y_1)(y_0-y_2)......(y_0-y_n)} * x_0 + \frac{(y-y_0)(y-y_2).....(y-y_n)}{(y_1-y_0)(y_1-y_2).......(y_1-y_n)} * x_1 + \frac{(y-y_0)(y-y_1)......(y-y_n)}{(y_2-y_0)(y_2-y_1).....(y_2-y_n)} * x_2$$

$$+ \ ....+ \frac{(y-y_1)(y-y_2).........(y-y_{n-1})}{(y_n-y_1)(y_n-y_2).......(y_n-y_{n-1})} * x_n$$

THE END