

COSMOS COLLEGE OF MANAGEMENT & TECHNOLOGY

Lab report on: Lagranges Interpolation
Lab report no: 04

Submitted by

Name: Aayush karki
Roll num: 200701
Group: A
Stream: BE IT
Semester: V
Subject: Numerical method

Submitted To

Department of ICT
Lab work date: 2020/8/26
Submission date: 2020/9/17

LAB NO: 04

LAB TITLE: Lagrange's Interpolation

LAB OBJECTIVE

- ↳ To understand Lagrange's Interpolation and its use in unequal interval
- ↳ To implement the solution of Lagrange's Interpolation in C and solve it using computational power.

THEORY

Lagrange's Interpolation is a method to construct polynomial that passes through a desired set of points and takes certain values at arbitrary points. Lagrange's Interpolation is used for a given data set where the interval is unequal.

Lagrange's Interpolation helps to find the equation, in way more easily than its counterpart i.e. Newton Divided difference. The general formula to find the polynomial is:

| | | | | | |
|--------|-------|-------|-------|-----|-------|
| x | x_1 | x_2 | x_3 | ... | x_n |
| $f(x)$ | y_1 | y_2 | y_3 | ... | y_n |

$$f(x) = \frac{(x-x_2)(x-x_3)\dots(x-x_n)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_n)} y_1 + \frac{(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)} y_2 + \dots + \frac{(x-x_1)(x-x_2)(x-x_3)\dots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)(x_n-x_3)\dots(x_n-x_{n-1})} y_n$$

So, there are some pros and cons of Lagrange's polynomial as:

(I) Pros

1. This formula helps to find the value of the function even when the arguments are not equally spaced.
2. This formula helps to find the value of independent variable x , corresponding to a given value of a function.

(II) Cons

1. For a polynomial of high degree, the formula involves a large number of multiplications which makes the process quite slow.
2. Can't estimate above maximum or below minimum values due to which it is not good for peak or mountainous areas.

CODR:

②

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
float xr, x[100], y[100], num, den, sum=0;
```

```
int i, j, n;
```

```
printf("Enter the value of number of terms of the table: ");  
scanf("%d", &n);
```

```
printf("Enter the values to be found");  
scanf("%f", &xr);
```

```
printf("Enter the respective value of x & y \n");  
for(int i=0; i<n; i++)
```

```
{
```

```
printf("x[%d] = ", i);
```

```
scanf("%f", &x[i]);
```

```
printf("y[%d] = ", i);
```

```
scanf("%f", &y[i]);
```

```
}
```

```
for(int i=0; i<n; i++)
```

```
{
```

```
num=1;
```

```
den=1;
```

```
for(j=0; j<n; j++)
```

```
{
```

```
if(j != i)
```

```
{
```

```
num = num * (xr - x[j]);
```

```
den = den * (x[i] - x[j]);
```

```
}
```

```
sum = sum + (num/den) * y[i];
```

```
}
```

(3)

```
printf("The required value for x= %f is %f", xr, sum);
return 0;
}
```

Test case: Enter the value of number of terms of the table: 4
 Enter the value to be found: 4
 Enter the respective value of x & y:

$$x[0] = 0$$

$$y[0] = -4$$

$$x[1] = 2$$

$$y[1] = 2$$

$$x[2] = 3$$

$$y[2] = 14$$

$$x[3] = 6$$

$$y[3] = 758$$

The required value for $x = 4.000000$ is 40.000000

DISCUSSION

In this lab work, we learned about Lagrange's interpolation, its primary use and its advantage and disadvantage. Here learn the logic behind finding the functional value for any given value of x . Also those mathematical calculation is implemented in C programming. Here the user inputs the number of terms in table, the target value for estimation, and the corresponding value of x and y . Finally the program calculates the interpolated value using Lagrange's interpolation formula.

CONCLUSION

Lagrange's interpolation is a powerful mathematical tool for approximating values within a range based on known data points. In this program, the nested loop efficiently computes the Lagrange's polynomial which is the weighted sum of given data points. The value of numerator and denominator is updated in each iteration of j and the final result is accumulated in sum variable and displayed. Thus in summary, Lagrange's interpolation is a valuable technique for filling in the gaps between data points and this example serves as an implementation in C.