Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

Lab Report

Topic : Numerical Differenciation

Course Title : Sessional Based on CSE 2103

Course Code : CSE 2104

Submitted By :

Md. Tanzid Hasan

Section : A

Roll No : 1603054

Submitted To :

Shyla Afroge

Assistant Professor

Dept. of Computer Science & Engineering

Date : 16-04-2018

Experiment No : 01

Experiment Name : Fitting a straight line

Theory : It is known that the maximum and minimum values of a function can be founded by equating the first derivative to zero and solving for the variable. The same procedure can be applied to determine the maxima and minima of a tabulated function.

Consider Newton's forward difference formula

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

Differentiating this with respect to p, we obtain

$$\frac{dy}{dp} = \Delta y_0 + \frac{2p-1}{2!}\Delta^2 y_0 + \frac{3p^2-3p+2}{3!}\Delta^3 y_0 + \ldots\ldots\ldots\ldots \qquad \ldots..(2)$$

For maxima or minima $\frac{dy}{dp}$=0. Hence, terminating the right-handed side, for simplicity, after the third difference and equating it to zero, we obtain the quadratic for p

$$c_0 + c_1 p + c_2 p^2 = 0 \qquad \ldots\ldots(3)$$

where

$$c_0 = \Delta y_0 - \frac{1}{2}\Delta^2 y_0 + \frac{1}{3}\Delta^3 y_0$$

$$c_1 = \Delta^2 y_0 + \Delta^3 y_0$$

and

$$c_2 = \frac{1}{2}\Delta^3 y_0$$

values of x can be found from the relation of $x = x_0 + ph$.
After that the y(x) can be determined by from Newton's interpolation forward or backward method.

Code :

```
#include <bits/stdc++.h>

using namespace std;

int fact(int n)

{

   int multi=1;
```

```
    for(int i=1;i<=n;i++)
        multi=multi*i;
    return multi;
}
int main()
{
    int n;
    double m,h;
    cout<<"How many elements you want to input : ";
    cin>>n;
    double a[n];
    double b[n][n];
    double q[n];
    cout<<"Enter x and corresponding y : "<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<"x"<<i<<" : ";
        cin>>a[i];
        cout<<"y"<<i<<" : ";
        cin>>b[0][i];
    }
    for(int i=0;i<n;i++)
        for(int j=0;j<n-i;j++)
            b[i+1][j]=b[i][j+1]-b[i][j];
    h=a[1]-a[0];
    double p=.5-((b[1][0])/b[2][0]);
    double x=a[0]+(p*h);
    if(x<=a[n/2])
        {
```

```
        double sum1=0;
double h1=a[1]-a[0];
double p1=((x-a[0])*1.0)/h1;
for(int i=0;i<n;i++)
    for(int j=0;j<n-i;j++)
        b[i+1][j]=b[i][j+1]-b[i][j];
double q1[n];
q1[0]=1;q1[1]=p1;
double g1=p1;
for(int i=2;i<n;i++)
{
    g1=g1*(p1-i+1);
    q1[i]=(g1*1.0)/fact(i);
}
for(int i=0;i<n;i++)
{
    sum1=sum1+(q1[i]*b[i][0]);
}
cout<<"Answer : "<<sum1<<endl;
    }
if(x>a[n/2])
    {
        double sum2=0;
double h2=a[1]-a[0];
double p2=((x-a[n-1])*1.0)/h2;
for(int i=0;i<n;i++)
    for(int j=i;j<n;j++)
        b[i+1][j+1]=b[i][j+1]-b[i][j];
double q2[n];
```

```
    q2[0]=1;q2[1]=p2;
    double g2=p2;
    for(int i=2;i<n;i++)
    {
        g2=g2*(p+i-1);
        q2[i]=(g2*1.0)/fact(i);
    }
    for(int i=0;i<n;i++)
        sum2=sum2+(q2[i]*b[i][n-1]);
    cout<<"Answer : "<<sum2<<endl;
        }
    return 0;
}
```

Input :

How many elements you want to input : 5

Enter x and corresponding y :

x0 : 1.2

y0 : 0.9320

x1 : 1.3

y1 : 0.9636

x2 : 1.4

y2 : 0.9855

x3 : 1.5

y3 : 0.9975

x4 : 1.6

y4 : 0.9996

Output : Answer : 1.00442

Discussion : In this program, some values of y was taken for some value of x. Then from these values, p was determined by differentiating the series. After determining p, x was calculated from p and h. Once x is calculated, y(x) can be determined by from Newton's interpolation forward or backward method.