

Curve Fitting & Integration by Trapezoidal Rule

Theory

It is often encountered the problem of fitting a curve to data which are subject to errors. This is contrary to the case like interpolation where it was assumed that the data are free of errors. A common strategy for such cases is to derive an approximating function that broadly fits the general trends of the data without necessarily passing through the individual points. The curve drawn is such that the discrepancy between the data points and the curve is least. The method of least square curve fitting is the most used method of curve fitting. The following equations are used to fit a non linear curve like power function –

$$ma_0 + a_1 \sum_{i=1}^m x_i + a_2 \sum_{i=1}^m x_i^2 + \dots + a_n \sum_{i=1}^m x_i^n = \sum_{i=1}^m y_i \quad \dots\dots\dots (1)$$

$$a_0 \sum_{i=1}^m x_i + a_1 \sum_{i=1}^m x_i^2 + \dots + a_n \sum_{i=1}^m x_i^{n+1} = \sum_{i=1}^m x_i y_i \quad \dots\dots\dots (2)$$

$$a_0 \sum_{i=1}^m x_i^n + a_1 \sum_{i=1}^m x_i^{n+1} + \dots + a_n \sum_{i=1}^m x_i^{2n} = \sum_{i=1}^m x_i^n y_i \quad \dots\dots\dots (3)$$

Trapezoidal rule is such a rule of determining the integration value of a function. Using this rule or equation, a limit integration can be performed. The following equation is the Trapezoidal rule–

$$\int_x^{x_n} y \, dx = \frac{h}{2} [y_0 + 2(y_1 + y_2 + \dots + y_{n-1}) + y_n] \quad \dots\dots\dots (4)$$

Code (Curve Fitting)

```
#include<bits/stdc++.h>
using namespace std;
double sumx,sumlnty,sumx2,sumxy;
int main()
{
    int m = 6;
    double x[6],y[6],lnty[6],x2[6],xy[6],a0,a1;
    for(int i=0; i<6 ; i++)
    {
        cout<<"Enter the value of x"<<i<<" : ";
        cin>>x[i];
        sumx = sumx + x[i];
        x2[i] = x[i] * x[i];
        sumx2 = sumx2 + x2[i];
        cout<<"Enter the value of y"<<i<<" : ";
        cin>>y[i];
```

```

    lny[i] = log(y[i]);
    sumlny = sumlny + lny[i];
    xy[i] = x[i] * lny[i];
    sumxy = sumxy + xy[i];
}

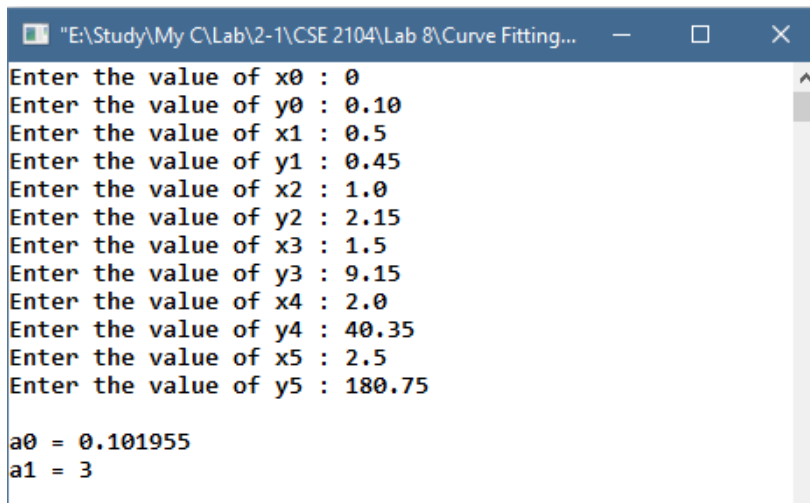
a0 = (- (sumx * sumxy) + (sumlny * sumx2)) / ((m * sumx2) - (sumx * sumx));
a1 = (- (sumlny * sumx) + (m * sumxy)) / ((m * sumx2) - (sumx * sumx));

cout<<endl;
cout<<"a0 = "<<exp(a0)<<endl;
cout<<"a1 = "<<ceil(a1)<<endl;

return 0;
}

```

Output (Curve Fitting)



```

E:\Study\My C\Lab\2-1\CSE 2104\Lab 8\Curve Fitting...
Enter the value of x0 : 0
Enter the value of y0 : 0.10
Enter the value of x1 : 0.5
Enter the value of y1 : 0.45
Enter the value of x2 : 1.0
Enter the value of y2 : 2.15
Enter the value of x3 : 1.5
Enter the value of y3 : 9.15
Enter the value of x4 : 2.0
Enter the value of y4 : 40.35
Enter the value of x5 : 2.5
Enter the value of y5 : 180.75

a0 = 0.101955
a1 = 3

```

Code (Trapezoidal Rule)

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int i=0;
    double h,UL = 1.0,temp,I,sumy=0.0;
    double x[10],y[10];
    x[i] = 0.0;
    cout<<"When h = 0.5 , "<<endl;

```

```

h = 0.5;
if(h == 0.5)
{
    while(x[i] <= UL)
    {
        temp = x[i];
        y[i] = 1 / (1 +(x[i]*x[i]));
        if(i>0 && x[i]!=1.00)
        {
            sumy = sumy + y[i];
        }
        i++;
        x[i] = temp + h;
    }
    I = (h /2) *(y[0]+2*sumy + y[i-1]);
    cout<<"The result is (y) : "<<I<<"\n"<<endl;
}

```

```

x[i] = 0.0;
i = 0;
cout<<"When h = 0.25 , "<<endl;
h = 0.25;
sumy = 0.0;
if(h == 0.25)
{
    while(x[i] <= UL)
    {
        temp = x[i];
        y[i] = 1 / (1 +(x[i]*x[i]));
        if(i>0 && x[i]!=1.00)
        {
            sumy = sumy + y[i];
        }
        i++;
        x[i] = temp + h;
    }
    I = (h /2) *(y[0]+2*sumy + y[i-1]);
    cout<<"The result is (y) : "<<I<<"\n"<<endl;
}

```

```

x[i] = 0.0;
i = 0;
cout<<"When h = 0.125 , "<<endl;
h = 0.125;
sumy = 0.0;
if(h == 0.125)

```

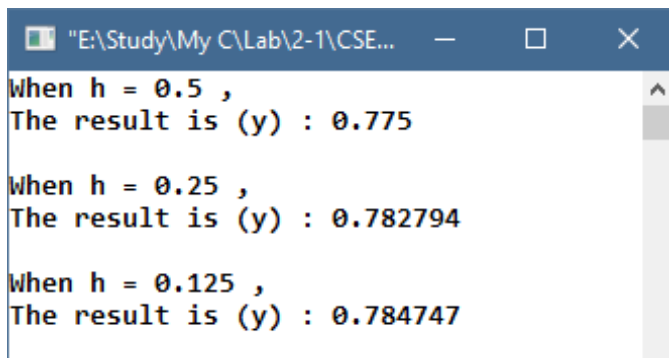
```

{
    while(x[i] <= UL)
    {
        temp = x[i];
        y[i] = 1 / (1 +(x[i]*x[i]));
        if(i>0 && x[i]!=1.00)
        {
            sumy = sumy + y[i];
        }
        i++;
        x[i] = temp + h;
    }
    I = (h /2) *(y[0]+2*sumy + y[i-1]);
    cout<<"The result is (y) : "<<I<<"\n"<<endl;
}

return 0;
}

```

Output (Trapezoidal Rule)



```

"E:\Study\My C\Lab\2-1\CSE...
When h = 0.5 ,
The result is (y) : 0.775

When h = 0.25 ,
The result is (y) : 0.782794

When h = 0.125 ,
The result is (y) : 0.784747

```

Discussion

In the above codes, one of them was the non linear curve fitting of a power function. This curve fitting method was implemented using the for loop and simple equation solving process. And then the equation no. (1),(2),(3) of theory were used to fit this non linear curve. And in the second problem, it was an integration problem which was to be solved using the rule of Trapezoidal . This was implemented using while loop for three different values of h (interval between the values of x) . Thus both the curve fitting for non linear equation and integration using Trapezoidal rule was done.