Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

Lab Report

Topic : Numerical Integration

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 : 23-04-2018

Experiment No : 01

Experiment Name : Numerical Integration

Theory : The general problem if numerical integration may be stated as follows. Given a set of data points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ of a function $y = f(x)$, where $f(x)$ is not known explicitly, it is required to compute the value of the definite integral

$$I = \int_a^b y\,dx$$

Let, the interval [a, b] be divided into n equal subintervals such that $a = x_0 < x_1 < x_2 < \ldots\ldots < x_n = b$. Clearly, $x_n = x_0 + nh$. Hence, the integral becomes

$$\mathrm{I} = \int_{x_0}^{x_n} \mathrm{y\,dx}$$

Approximating y by Newton's forward difference formula, we obtain

$$\mathrm{I} = \int_{x_0}^{x_n} [\, y_0 + \mathrm{p}\Delta y_0 + \frac{p(p-1)}{2}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{6}\Delta^3 y_0 + \ldots\ldots\ ]\,\mathrm{dx}$$

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

$$\mathrm{I} = \mathrm{h}\int_0^n [\, y_0 + \mathrm{p}\Delta y_0 + \frac{p(p-1)}{2}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{6}\Delta^3 y_0 + \ldots\ldots\ ]\,\mathrm{dp}$$

which gives on simplification

$$\int_{x_0}^{x_n} \mathrm{y\,dx} = \mathrm{nh}\,[\, y_0 + \frac{n}{2}\Delta y_0 + \frac{n(2n-3)}{12}\Delta^2 y_0 + \frac{n(n-2)^2}{24}\Delta^3 y_0 + \ldots\ldots] \quad \ldots.(1)$$

From this general formula, we can obtain difference integration formulae by putting n = 1, 2, 3, …. Etc.

Trapezoidal Rule : Setting n = 1 in the general formula (1), all differences higher than the first will become zero and we obtain

$$\int_{x_0}^{x_1} \mathrm{y\,dx} = \mathrm{h}\,(\, y_0 + \frac{1}{2}\Delta y_0\,) = h\,[\, y_0 + \frac{1}{2}(\, y_1 - y_0) = \frac{h}{2}\,(y_0 + y_1) \quad \ldots..(2)$$

For the next interval $[x_1, x_2]$, we deduce similarly

$$\int_{x_1}^{x_2} \mathrm{y\,dx} = \frac{h}{2}\,(y_1 + y_2) \quad \ldots\ldots(3)$$

and so on. For the last [x_{n-1}, x_n], we have

$$\int_{x_{n-1}}^{x_n} y\,dx = \frac{h}{2}\,(y_{n-1}+ y_n) \quad \dots\dots(4)$$

Combining all these expressions, we obtain the rule

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{2}\,[\,y_0 + 2(\,y_1 + y_2 + \cdots + y_{n-1}\,) + y_n] \quad \dots.(5)$$

which is known as the trapezoidal rule.

Simpson's 1/3 Rule : This rule is obtained by putting n =2 in Eq. (1), i.e. by replacing the curve by n/2 arcs of second-degree polynomials or parabolas. We have then

$$\int_{x_0}^{x_2} y\,dx = 2h\,(\,y_0 + \Delta y_0 + \frac{1}{6}\Delta^2 y_0\,) = \frac{h}{3}\,(y_0+ 4y_1 + y_2\,)$$

Similarly,

$$\int_{x_2}^{x_4} y\,dx = \frac{h}{3}\,(y_2+ 4y_3 + y_4)$$

and finally

$$\int_{x_{n-2}}^{x_n} y\,dx = \frac{h}{3}\,(y_{n-2}+ 4y_{n-1} + y_n)$$

Summing up, we obtain

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}\,[\,y_0 + 4(\,y_1 + y_3 + y_5 + \cdots + y_{n-1})$$
$$+ 2\,(\,y_2 + y_4 + y_6 + \cdots + y_{n-2}\,) + y_n] \quad \dots.(6)$$

which is known as Simpson's 1/3 rule, or simply Simpson's rule.

Simpson's 3/8 Rule : Setting n = 3 in (1) we observe that all the differences higher than the third will become zero and we obtain

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

$$= \frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3)$$

Similarly

$$\int_{x_3}^{x_6} \text{y dx} = \frac{3h}{8}(y_3 + 3y_4 + 3y_5 + y_6)$$

And so on. Summing up all these, we obtain

$$\int_{x_0}^{x_n} \text{y dx} = \frac{3h}{8}[(y_0 + 3y_1 + 3y_2 + y_3) + (y_3 + 3y_4 + 3y_5 + y_6) + \cdots + (y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n)]$$
$$= \frac{3h}{8}[y_0 + 3(y_1 + y_2 + y_4 + y_5 + y_7 + \cdots + y_{n-2} + y_{n-1}) + 2(y_3 + y_6 + \cdots + y_{n-3}) + y_n]$$

This rule is called Simpson's 3/8 rule, is not so accurate as Simpson's rule.

Code :

```
#include <bits/stdc++.h>
using namespace std;
double f(double x){
   double r=(1.0/(1+x));
   return r; }
double trapezoidal(int m,double v[],double  k)
{
   double s=v[0]+v[m];
   double sum=0;
   for(int i=1;i<m;i++)
      sum=sum+v[i];
   sum=sum*2;
   sum=sum+s;
   sum=(k/2.0)*sum;
   return sum;
}
```

```
double simpson_o_t(int n,double u[],double l)
{
   double s=u[0]+u[n];
   double sum=0;
   for(int i=1;i<n;i++)
   {
      if(i%2==0)
         sum=sum+(2*u[i]);
      else
         sum=sum+(4*u[i]);
   }
   sum=sum+s;
   sum=(l/3.0)*sum;
   return sum;
}
double simpson_t_e(int o,double w[],double p)
{
   double s=w[0]+w[o];
   double sum=0;
   for(int i=1;i<o;i++)
   {
      if(i%3==0)
         sum=sum+(2*w[i]);
      else
         sum=sum+(3*w[i]);
   }
   sum=sum+s;
   sum=(((3.0)*p*sum)/(8.0));
   return sum;
```

```
}
int main()
{
    for(;;)
    {
        double a,b,h;
        int g;
        cout<<"Enter the lower limit & upper limit : ";
        cin>>a>>b;
        cout<<"Enter the difference (h) : ";
        cin>>h;
        g=round((b-a)/h);
        double x[g+1],y[g+1];
        for(int i=0;i<=g;i++)
        {
            x[i]=a+(i*h);
            y[i]=f(x[i]);
        }
        double j=(log(1+b))-(log(1+a));
        double j1,j2,j3;
        int d;
        cout<<endl<<"1. Trapezoidal Rule."<<endl<<"2. Simpson's 1/3
Rule."<<endl<<"3. Simpson's 3/8 Rule."<<endl<<"4. Compare."<<endl<<"5.
Exit."<<endl<<ends<<"Enter the option(1-5) : ";
        cin>>d;
        if(d>5)
            cout<<"Invalid input."<<endl;
        if(d==5)
            break;
        switch(d)
    {
    case 1:
        {
            cout<<endl<<"1. Trapezoidal Rule : "<<endl;
            j1=trapezoidal(g,y,h);
            cout<<"Value : "<<j1<<endl;
            cout<<"Error : "<<fabs(j1-j)<<endl<<endl<<endl<<endl;
        }break;

    case 2:
        {
            cout<<endl<<"2. Simpson's 1/3 Rule : "<<endl;
            j2=simpson_o_t(g,y,h);
            cout<<"Value : "<<j2<<endl;
            cout<<"Error : "<<fabs(j2-j)<<endl<<endl<<endl<<endl;
        }break;
    case 3:
        {
            cout<<endl<<"3. Simpson's 3/8 Rule : "<<endl;
            j3=simpson_t_e(g,y,h);
            cout<<"Value : "<<j3<<endl;
            cout<<"Error : "<<fabs(j3-j)<<endl<<endl<<endl<<endl;
        }break;
    case 4:
        {
            cout<<endl<<"4. Compare : "<<endl<<endl;
            cout<<"Calculated Value : "<<j<<endl;
            j1=trapezoidal(g,y,h);
            cout<<"Trapezoidal Rule : "<<j1<<" : |"<<j<<"-"<<j1<<"| = 
"<<fabs(j-j1)<<endl;
```

```
                j2=simpson_o_t(g,y,h);
                cout<<"Simpson's 1/3 Rule : "<<j2<<" : |"<<j<<"-"<<j2<<"| =
"<<fabs(j-j2)<<endl;
                j3=simpson_t_e(g,y,h);
                cout<<"Simpson's 3/8 Rule : "<<j3<<" : |"<<j<<"-"<<j3<<"| =
"<<fabs(j-j3)<<endl;
                if((fabs(j1-j)<fabs(j2-j)) && (fabs(j1-j)<fabs(j3-j)))
                  cout<<endl<<"Trapezoidal is the best rule."<<endl;
                if((fabs(j2-j)<fabs(j1-j)) && (fabs(j2-j)<fabs(j3-j)))
                  cout<<endl<<"Simpson's 1/3 is the best rule."<<endl;
                if((fabs(j3-j)<fabs(j1-j)) && (fabs(j3-j)<fabs(j2-j)))
                  cout<<endl<<"Simpson's 3/8 is the best rule."<<endl;
                cout<<endl<<endl<<endl;
              }break;
      }
   }
}
```

Input :

```
Enter the lower limit & upper limit : 0 1
Enter the difference (h) : .1
1. Trapezoidal Rule.
2. Simpson's 1/3 Rule.
3. Simpson's 3/8 Rule.
4. Compare.
5. Exit.
 Enter the option(1-5) : 4
```

Output :

```
4. Compare :
Calculated Value : 0.693147
Trapezoidal Rule : 0.693771 : |0.693147-0.693771| = 0.000624223
Simpson's 1/3 Rule : 0.69315 : |0.693147-0.69315| = 3.05013e-006
Simpson's 3/8 Rule : 0.680347 : |0.693147-0.680347| = 0.0127998
Simpson's 1/3 is the best rule.
```

Discussion : In this program, lower limit and upper limit of a function was taken as input. The difference between two x of f(x) was also taken as input. Then, the option was chosen as needed. After that, the rule was algorithmize as it was described in the rule. There were five option. Three for three rules and one for comparing which rule is the most efficient and other for exit.