MA 322: Lab Assignment #6

Due on Sunday, September 13, 2015

Jiten Chandra Kalita

Silvi Pandey-130123045

Contents

PROBLEM 1

```
#include<iostream>
#include<math.h>
#include<stdlib.h>
#include<stdio.h>
using namespace std;

int low=0;
int high=1;
int interval=2000;
double correctArea=atan(1);
double h=(double)(high-low)/(interval-1);
double function(double x);
double Trapezoidal();
double SimpsonOneThird();
double SimpsonThreeEighth();
int main()
{
    cout<<"For interval : "<<interval<<" Trapezoidal Method          : "
    <<Trapezoidal()*4       <<" Error : "<<fabs(correctArea-Trapezoidal())*4<<"\n";
    cout<<"For interval : "<<interval<<" Simpson One Third Method    : "
    <<SimpsonOneThird()*4   <<" Error : "<<fabs(correctArea-SimpsonOneThird())*4<<"\n";
    cout<<"For interval : "<<interval<<" Simpson Three Eighth Method : "
    <<SimpsonThreeEighth()*4<<" Error : "<<fabs(correctArea-SimpsonThreeEighth())*4<<"\n";
}
double function(double x)
{
    return 1/(1+x*x);
}
double Trapezoidal()
{
    double area=0;
    for(int i=0;i<interval;i++)
    {
        area+=(function(low+i*h)+function(low+(i+1)*h))*0.5*h;
    }
    return area;
}
double SimpsonOneThird()
{
    double area=0;
    double y0,y1,y2;
    for(int i=0;i<interval;i++)
    {
        y0=low+i*h;
        y1=y0+h*0.5;
        y2=y0+h;
        area+=h/6*(function(y0)+4*function(y1)+function(y2));
    }
    return area;
}
double SimpsonThreeEighth()
{
```

```
    double area=0;
    double y0,y1,y2,y3;
    for(int i=0;i<interval;i++)
    {
        y0=low+i*h;
        y1=y0+h/3;
        y2=y0+(2*h)/3;
        y3=y0+h;
        area+=(h/8)*(function(y0)+3*function(y1)+3*function(y2)+function(y3));
    }
    return area;
}
```

OUTPUT

```
For interval : 200 Trapezoidal Method          : 3.15161 Error : 0.0100209
For interval : 200 Simpson One Third Method    : 3.15162 Error : 0.010025
For interval : 200 Simpson Three Eighth Method : 3.15162 Error : 0.010025

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

```
For interval : 2000 Trapezoidal Method          : 3.14259 Error : 0.00100021
For interval : 2000 Simpson One Third Method    : 3.14259 Error : 0.00100025
For interval : 2000 Simpson Three Eighth Method : 3.14259 Error : 0.00100025

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

```
For interval : 4000 Trapezoidal Method          : 3.14209 Error : 0.000500052
For interval : 4000 Simpson One Third Method    : 3.14209 Error : 0.000500063
For interval : 4000 Simpson Three Eighth Method : 3.14209 Error : 0.000500063

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

EXPLANATION/RESULT

(a)Number of intervals taken : 200

(b)Error decreases as the number of intervals increase

(c)The accuracy increases from Trapezoidal to Simpson's One-Third rule to Simpson's Three-Eighth rule.

PROBLEM 2(a)

```
#include<iostream>
#include<math.h>
#include<stdlib.h>
#include<stdio.h>

using namespace std;
double low=0;
double high=1;
double function(double x);
double SimpsonOneThird(int interval);
int main()
```

```
{
    FILE *fp;
    fp=fopen("FileQ2.txt","w");
    int interval=2;
    double area,actualArea;
    actualArea=4*atan(1);
    while(1)
    {
        area=SimpsonOneThird(interval);
        cout<<"No of Intervals : "<<interval<<"    "
        <<"Error : "<<fabs(area-actualArea)<<endl;
        if(fabs(area-actualArea)<=0.5*pow(10,-5))
            break;
        interval=interval+2;
    }
}
double function(double x)
{
    return 4/(1+x*x);
}
double SimpsonOneThird(int interval)
{
    double area=0;
    double y0,y1,y2,h;
    h=(double)(high-low)/(interval);
    for(int i=0;i<interval;i++)
    {
        y0=low+(double)(i*h);
        y1=y0+h*0.5;
        y2=y0+h;
        area+=h/6*(function(y0)+4*function(y1)+function(y2));
    }
    return area;
}
```

OUTPUT

(a)For f(x)=$\frac{1}{1+x^2}$

```
No of Intervals : 2    Error : 2.40261e-005
No of Intervals : 4    Error : 1.51131e-007

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

PROBLEM 2(b)

```
#include<iostream>
#include<math.h>
#include<stdlib.h>
#include<stdio.h>

using namespace std;
double low=0;
double high=1/(double)sqrt(2);
double function(double x);
double SimpsonOneThird(int interval);
int main()
{
    FILE *fp;
    fp=fopen("FileQ2.txt","w");
    int interval=2;
    double area,actualArea;
    actualArea=0.5*atan(1);
    while(1)
    {
        area=SimpsonOneThird(interval);
        cout<<"No of Intervals : "<<interval<<"    "
        <<"Error : "<<fabs(area-actualArea)<<endl;
        if(fabs(area-actualArea)<=0.5*pow(10,-5))
            break;
        interval=interval+2;
    }
}
double function(double x)
{
    return sqrt(1-x*x)-x;
}
double SimpsonOneThird(int interval)
{
    double area=0;
    double y0,y1,y2,h;
    h=(double)(high-low)/(interval);
    for(int i=0;i<interval;i++)
    {
        y0=low+(double)(i*h);
        y1=y0+h*0.5;
        y2=y0+h;
        area+=h/6*(function(y0)+4*function(y1)+function(y2));
    }
    return area;
}
```

OUTPUT

(a)For f(x)=$\sqrt{1-x^2}-x$

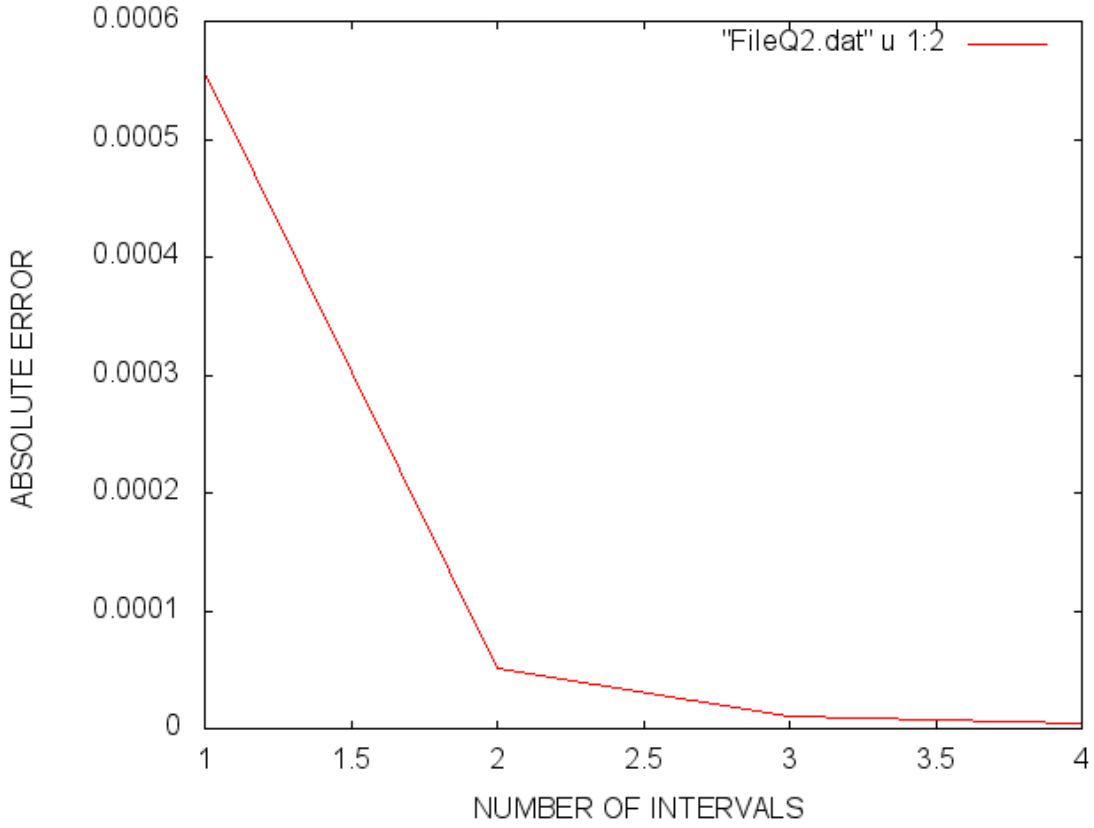
```
No of Intervals : 2    Error : 5.07167e-005
No of Intervals : 4    Error : 3.75462e-006

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

EXPLANATION/RESULT

If the initial number of intervals is taken as 1,then the required error threshold condition is met in 4 iterations

PLOT



PROBLEM 3(b)(a)

```
#include<iostream>
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

using namespace std;
int low=0;
int high=1;
double function(double x);
double integral(double a,double b);
double g(double x,double a,double b);
```

```
double GaussThreePointQuad(int interval);
int main()
{
    cout<<"For 1 interval : "<<GaussThreePointQuad(1)<<endl;
    cout<<"For 2 interval : "<<GaussThreePointQuad(2)<<endl;
    cout<<"For 10 interval : "<<GaussThreePointQuad(10)<<endl;
}
double function(double x)
{
    return pow(x,5);
}
double g(double x,double a,double b)
{
    return (b-a)*0.5*function((b-a)*0.5*x+(b+a)*0.5);
}
double integral(double a,double b)
{
    return (5*g(-pow(0.6,0.5),a,b))/9+(8*g(0,a,b))/9+(5*g(pow(0.6,0.5),a,b))/9;
}
double GaussThreePointQuad(int interval)
{
    double area=0;
    double h=(double)(high-low)/interval;
    for(int i=0;i<interval;i++)
    {
        area=area+integral(low+i*h,low+(i+1)*h);
    }
    return area;
}
```

OUTPUT

(a)For f(x) = x^5

```
For 1 interval : 0.166667
For 2 interval : 0.166667
For 10 interval : 0.166667

Process returned 0 (0x0)   execution time : 0.109 s
Press any key to continue.
```

PROBLEM 3(b)(b)

```
#include<iostream>
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

using namespace std;
int low=0;
int high=1;
double function(double x);
double integral(double a,double b);
double g(double x,double a,double b);
double GaussThreePointQuad(int interval);
```

```
int main()
{
    cout<<"For 1 interval : "<<GaussThreePointQuad(1)<<endl;
    cout<<"For 2 interval : "<<GaussThreePointQuad(2)<<endl;
    cout<<"For 3 interval : "<<GaussThreePointQuad(3)<<endl;
    cout<<"For 4 interval : "<<GaussThreePointQuad(4)<<endl;
}
double function(double x)
{
    return sin(x)/x;
}
double g(double x,double a,double b)
{
    return (b-a)*0.5*function((b-a)*0.5*x+(b+a)*0.5);
}
double integral(double a,double b)
{
    return (5*g(-pow(0.6,0.5),a,b))/9+(8*g(0,a,b))/9+(5*g(pow(0.6,0.5),a,b))/9;
}
double GaussThreePointQuad(int interval)
{
    double area=0;
    double h=(double)(high-low)/interval;
    for(int i=0;i<interval;i++)
    {
        area=area+integral(low+i*h,low+(i+1)*h);
    }
    return area;
}
```

OUTPUT

(b)For f(x) = $x^{-1}sin(x)$

```
For 1 interval : 0.946083
For 2 interval : 0.946083
For 3 interval : 0.946083
For 4 interval : 0.946083

Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```