Heaven's Light is Our Guide
Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

Lab Report

Topic : Numerical Solution of Ordinary Differential Equation

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 : 08-05-2018

Experiment No : 01

Experiment Name : Solution By Taylor's Series

Theory : Considering the differential equation

$$y' = f(x, y) \qquad \text{.........(1a)}$$

with the initial condition

$$y(x_0) = x_0 \qquad \text{..........(1b)}$$

If y(x) is the exact solution of (1), then the Taylor's series for y(x) around x = x_0 is given by

$$y(x) = y_0 + (x - x_0)y_0' + \frac{(x-x_0)^2}{2!} + y_0'' + \cdots \qquad \text{......(2)}$$

If the values of y_0', y_0'', … are known, then (2) gives a power series for y. Using the formula for total derivatives, we can write

$$y'' = f' = f_x + y' f_y = f_x + f f_y$$

Where the suffixes denoted partial derivative with respect to the variable concerned. Similarly, we obtain

$$y''' = f'' = f_{xx} + f_{xy} f + f\big(f_{yx} + f_{yy} f\big) + f_y(f_x + f_y f)$$
$$= f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_x f_y + f {f_y}^2$$

and other higher derivatives of y. The method can easily be extended to simultaneous and higher-order differential equations.

Code :

```
#include <bits/stdc++.h>
using namespace std;
int fact(int a)
{
   if(a==0)
      return 1;
   else
      return (a*fact(a-1));
}
```

```
double df1(double f,double g)
{
   double r=(f-(g*g));
   return r;
}
double df2(double f,double g)
{
   double r=(1-(2*g*df1(f,g)));
   return r;
}
double df3(double f,double g)
{
   double r=-(2*g*df2(f,g))-(2*df1(f,g)*df1(f,g));
   return r;
}
double df4(double f,double g)
{
   double r=-(2*g*df3(f,g))-(6*df1(f,g)*df2(f,g));
   return r;
}
double df5(double f,double g)
{
   double r=-(2*g*df4(f,g))-(6*df1(f,g)*df3(f,g))-(6*df2(f,g)*df2(f,g));
   return r;
}
int main()
{
   double x0,xd,y[10];
   cout<<"Enter the x0 : ";
```

```
    cin>>x0;
    cout<<"Enter the y0 : ";
    cin>>y[0];
    cout<<"Enter the desire x : ";
    cin>>xd;
    y[1]=df1(x0,y[0]);
    y[2]=df2(x0,y[0]);
    y[3]=df3(x0,y[0]);
    y[4]=df4(x0,y[0]);
    y[5]=df5(x0,y[0]);
    double sum=1;
    for(int i=1;i<=5;i++)
        sum=sum+((pow(xd,i)*y[i])/fact(i));
    cout<<endl<<"Value : "<<sum<<endl;
}
```

Input :

Enter the x0 : 0

Enter the y0 : 1

Enter the desire x : 0.1

Output :

Value : 0.913794

Discussion : In this program, initial value of x and y were taken as input. Then, for which x the output was to be shown, that desired x was taken as input also. After that, the solution of Taylor's series was algorithmize. In the algorithm, there was determined third differential equation for the calculation. But in the program, there was determined to fifth differential equation for more accurate and efficient answer.

Experiment No : 02

Experiment Name : Solution By Euler's Method

Theory : Considering the differential equation

$$y' = f(x, y) \qquad \text{.........(1a)}$$

with the initial condition

$$y(x_0) = x_0 \qquad \text{..........(1b)}$$

Suppose that we wish to solve the equation (1) for values of y at $x = x_r = x_0 + rh$ ($r = 1, 2, \ldots..$). Integrating Eq. (1), we obtain

$$y_1 = y_0 + \int_{x_0}^{x_1} f(x, y)dx \qquad \text{...........(2)}$$

Assuming that $f(x, y) = f(x_0, y_0)$ in $x_0 \leq x \leq x_1$, this gives Euler's formula

$$y_1 = y_0 + hf(x_0, y_0) \qquad \text{...........(3a)}$$

Similarly for the range $x_1 \leq x \leq x_2$, we have

$$y_2 = y_1 + \int_{x_0}^{x_1} f(x, y)dx$$

Substituting $f(x_1, y_1)$ for $f(x, y)$ in $x_1 \leq x \leq x_2$ we obtain

$$y_2 = y_1 + hf(x_1, y_1) \qquad \text{...........(3b)}$$

Proceeding in this, we obtain the general formula

$$y_{n+1} = y_n + hf(x_n, y_n) \qquad \text{...........(4)}$$

The process is very slow and to obtain reasonable accuracy with Euler's method, we need to take a smaller value for h.

Code :

```
#include <bits/stdc++.h>
using namespace std;
double df(double f,double g)
{
   double r=(f-(g*g));
   return r;
}
```

```
int main()
{
    double x[200],y[200],xd;
    cout<<"Enter the x0 : ";
    cin>>x[0];
    cout<<"Enter the y0 : ";
    cin>>y[0];
    cout<<"Enter the desire x : ";
    cin>>xd;
    for(;;) {
        double h;
        cout<<endl<<endl<<"Enter the value of h (Enter 0 to break ) : ";
        cin>>h;
        if(h==0)
            break;
        int g=(xd-x[0])/h;
        for(int i=0;i<=g;i++)  {
            x[i+1]=x[i]+h;
            y[i+1]=(y[i]+(h*df(x[i],y[i])));
        }
        cout<<endl<<"Value : "<<y[g]<<endl;
    }
}
```

Input :

Enter the x0 : 0

Enter the y0 : 1

Enter the desire x : 0.1

Enter the value of h (Enter 0 to break) : 0.01

Output :

Value : 0.912576

Discussion : In this program, initial value of x and y were taken as input. Then, for which x the output was to be shown, that desired x was taken as input also. With x , h was also taken in input as it is the difference between each x. After that, the solution of Euler’s Method was algorithmize. The process is very slow and to obtain reasonable accuracy with Euler’s method, we need to take a smaller value for h.