Experiment-7
Ordinary Differential Equations

Purpose: In this experiment, you are going to approximate solution of an ordinary differential equation using different methods.

Introduction

The Runge-Kutta methods are family of implicit and explicit iterative methods for obtaining numerical approximations to solutions of ordinary differential equations.

Let an initial value problem be specified as below:

$$\frac{dy}{dx} = g(x, y), \qquad y(x_0) = y_0 \tag{1}$$

where y is an unknown function of x and $\frac{dy}{dx}$ is the rate at which y changes. The value of $y(x_{last})$ can be approximated using $g(x, y)$, x_0, y_0 and chosen step size (h).

The Euler Method

The Euler method is the simplest Runge-Kutta method with one stage.

-Pick a step-size h $>$ 0 and calculate y_{n+1} for $n = 0, 1, 2, 3...$ using

$$y_{n+1} = y_n + h * g(x_n, y_n) \tag{2}$$

First step:

$$y_1 = y_0 + h * g(x_0, y_0) \tag{3}$$

The Midpoint Method

The midpoint method is a second-order Runge-Kutta method with two stages.

-Pick a step-size h $>$ 0 and calculate y_{n+1} for $n = 0, 1, 2, 3...$ using

$$y_{n+1} = y_n + h * g(x_n + \frac{h}{2}, y_n + \frac{h * g(x_n, y_n)}{2}) \tag{4}$$

The Runge-Kutta Method

The Runge-Kutta method is also known as "classic Runge-Kutta method".

-Pick a step-size h $>$ 0 and calculate y_{n+1} for $n = 0, 1, 2, 3...$ using

$$k_1 = g(x_n, y_n) \tag{5}$$

$$k_2 = g(x_n + \frac{h}{2}, y_n + \frac{hk_1}{2}) \tag{6}$$

$$k_3 = g(x_n + \frac{h}{2}, y_n + \frac{hk_2}{2}) \tag{7}$$

$$k_4 = g(x_n + h, y_n + hk_3) \tag{8}$$

$$y_{n+1} = y_n + \frac{h * (k_1 + 2k_2 + 2k_3 + k_4)}{6} \tag{9}$$

Problem Statement

Solve the following initial value problem over the interval [-1.8,1.2] using $stepSize(h) = 0.1$ and $y(-1.8) = 1.24$.

$$\frac{dy}{dx} = y(-x^4 + x^2 - x) \tag{10}$$

Analytical solution is:

$$y(x) = e^{[-(x^5/5)+(x^3/3)-(x^2/2)]} \tag{11}$$

Lab Procedure

We highly recommend you to follow lab procedure without skipping any step and read each step thoroughly before you start.

1- Ask x_0, y_0, x_{last} and $stepSize$ values from user in main function. (10 pts) These values will be used in both Euler and midpoint methods.

2- Write two separate functions to calculate $y(x)$ and $\frac{dy}{dx} = g(x, y)$. Call these functions in main and print calculated values for $x = -0.1$ and $y = 0.5$ values. (20 pts)

y(-0.1) = 0.9947 g(-0.1,0.5) = 0.0550

```
double fY(double x)
double fYdx(double x, double y)
```

Note: You should be able to extract the y values (yEuler,yMidpoint,yRK4) from the corresponding functions to print them in the main.

3- Implement Euler method and check its operation. Do not use any loop to call Euler function more than one time in main; you should call it only once. The number of steps should be determined using x_0, x_{last} and $stepSize$ and you should allocate proper amount of memory to store obtained results considering the number of steps. (15 pts)

```
double euler(double (*fYdx1)(double, double), double xFirst, double yFirst, double xLast,
    double stepSize)
```

4- Implement midpoint method and check its operation. Do not use any loop to call midpoint function more than one time in main; you should call it only once. The number of steps should be determined using x_0, x_{last} and $stepSize$ and you should allocate proper amount of memory to store obtained results considering the number of steps. (20 pts)

```
double midpoint(double (*fYdx2)(double, double), double xFirst, double yFirst, double
    xLast, double stepSize)
```

5- Implement Runge-Kutta method and check its operation. Do not use any loop to call Runge-Kutta function more than one time in main; you should call it only once. The number of steps should be determined using x_0, x_{last} and $stepSize$ and you should allocate proper amount of memory to store obtained results considering the number of steps. (15 pts)

```
double RK4(double (*fYdx3)(double, double), double xFirst, double yFirst, double xLast,
    double stepSize)
```

6- Print calculated values obtained with all methods and function you wrote for $y(x)$ in the second step for each iteration with values $x_0 = -1.8$, $y_0 = 1.24$, $x_{last} = 1.2$ and $stepSize = 0.1$. Comment on results. Check Figures: 1 and 2 (20 pts)

```
Please enter x0 and y0
-1.8 1.24
Please enter xLast and stepSize
1.2 0.1
                        Euler      Midpoint   RK4        Exact
Step 0-y(-1.800000)--- 1.240000---1.240000---1.240000---1.240031
Step 1-y(-1.700000)--- 0.563258---0.828280---0.784419---0.784309
Step 2-y(-1.600000)--- 0.351354---0.623880---0.578086---0.578000
Step 3-y(-1.500000)--- 0.267254---0.523950---0.481377---0.481307
Step 4-y(-1.400000)--- 0.232177---0.481456---0.440928---0.440864
Step 5-y(-1.300000)--- 0.220995---0.474454---0.434046---0.433984
Step 6-y(-1.200000)--- 0.223955---0.492179---0.450142---0.450077
Step 7-y(-1.100000)--- 0.236639---0.528796---0.483636---0.483567
Step 8-y(-1.000000)--- 0.256657---0.580440---0.530895---0.530819
Step 9-y(-0.900000)--- 0.282322---0.643702---0.588746---0.588663
Step 10-y(-0.800000)--- 0.312076---0.714885---0.653777---0.653684
Step 11-y(-0.700000)--- 0.344233---0.789763---0.722110---0.722008
Step 12-y(-0.600000)--- 0.376931---0.863721---0.789539---0.789427
Step 13-y(-0.500000)--- 0.408232---0.932179---0.851909---0.851789
Step 14-y(-0.400000)--- 0.436298---0.991140---0.905612---0.905484
Step 15-y(-0.300000)--- 0.459613---1.037696---0.948027---0.947893
Step 16-y(-0.200000)--- 0.477166---1.070330---0.977789---0.977651
Step 17-y(-0.100000)--- 0.488542---1.088958---0.994823---0.994683
Step 18-y(0.000000)--- 0.493911---1.094706---1.000141---1.000000
Step 19-y(0.100000)--- 0.493911---1.089506---0.995483---0.995342
Step 20-y(0.200000)--- 0.489461---1.075622---0.982892---0.982753
Step 21-y(0.300000)--- 0.481551---1.055200---0.964307---0.964172
Step 22-y(0.400000)--- 0.471048---1.029890---0.941224---0.941092
Step 23-y(0.500000)--- 0.458537---1.000572---0.914441---0.914312
Step 24-y(0.600000)--- 0.444208---0.967182---0.883900---0.883776
Step 25-y(0.700000)--- 0.427790---0.928639---0.848623---0.848504
Step 26-y(0.800000)--- 0.408535---0.882898---0.806760---0.806647
Step 27-y(0.900000)--- 0.385265---0.827186---0.755816---0.755710
Step 28-y(1.000000)--- 0.356520---0.758542---0.693138---0.693041
Step 29-y(1.100000)--- 0.320868---0.674734---0.616748---0.616661
Step 30-y(1.200000)--- 0.277420---0.575564---0.526490---0.526416
```

Figure 1: $y(x)$ values obtained with stepSize $= 0.1$

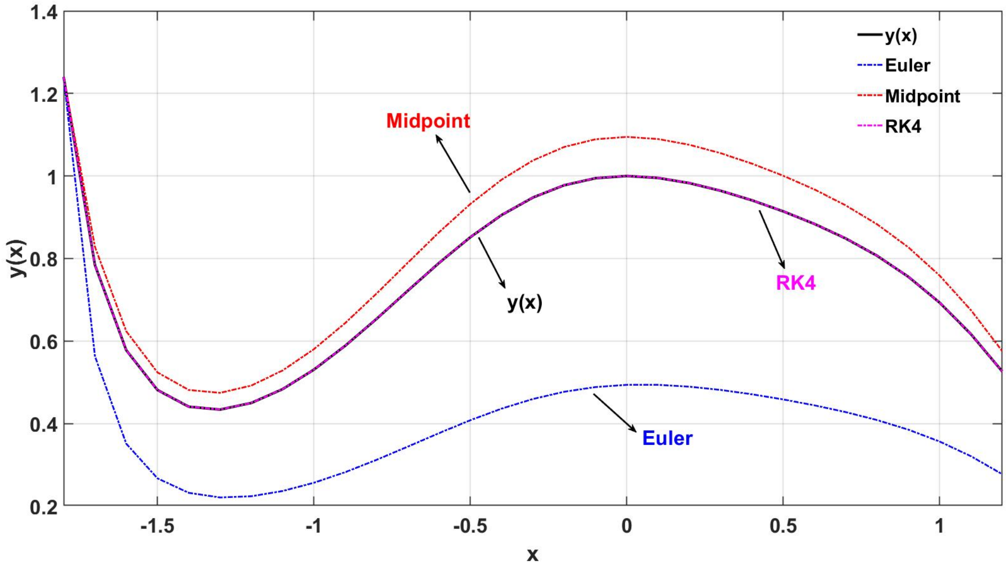

Figure 2: $y(x)$ values obtained with stepSize $= 0.1$