Differential Equations, Fall 2016
Computer Project 3, Due Friday, November 4

The goal of this assignment is to implement the Runge-Kutta Method and use it to solve some ODEs. You should hand in the following things:

1. A printout of `RungeKuttaMethod.m` and a printout of the figure in problem 1.
2. A printout of `exact.m` and a printout of the figure in problem 2.
3. A lower and upper bound for the singularity in problem 3, and evidence that these bounds are correct.
4. The period for E=50 and the period for E=1, along with some commands and output that show these answers are correct.

1. (3 points) The goal of this problem is to write a program to implement the Runge-Kutta Method to solve an initial value problem of the form

$$\begin{aligned} \frac{dy}{dx} &= f(x,y) \\ y(x_0) &= y_0. \end{aligned}$$

We will take $f(x,y) = x(y^2+1)$ to begin with. First, we need to create a Matlab function for $f(x,y)$. To do this, create a function called `f.m` and enter the following code.

```
function val=f(x,y)
    val=x*(y^2+1);
end
```

Test that it works by entering the command

```
> f(2,3)
```

(This should return 20.)

Now we will create the main program to implement the Runge-Kutta Method. Create a function called `RungeKuttaMethod.m`. The outline of the code will be similar to the Euler Method and Improved Euler Method programs from previous assignments, but the inner details will be different. Start with the following code in this function:

```
function [xvals yvals]=RungeKuttaMethod(xmin,xmax,steps,y0)
% calculate the step size
h=(xmax-xmin)/steps;
% now set up a list of x values
% need the list to have steps+1 numbers in it
```

```
% because the initial x is counted as 1
xvals=linspace(xmin,xmax,steps+1);
x=xmin;
% initialize y and yvals
y=y0;
yvals=[y];
% now a for loop to build the yvals
for i=1:steps
    ...
    y=...
    % add y to the list of yvals
    yvals=[yvals y];
    % update x
    x=xvals(i+1);
end
end
```

Fill in the missing parts. Now test the code by running the following commands

```
> [x y]=RungeKuttaMethod(1,1.5,5,1);
> plot(x,y)
```

Print out your program and the figure generated by the plot command.

2. (3 points) The goal of this problem is to plot the exact solution to the IVP considered in the previous problem,

$$\begin{aligned} y' &= x(y^2+1), \\ y(1) &= 1. \end{aligned}$$

First, solve this equation exactly. Then write a Matlab script called `exact.m` which contains commands for plotting the solution. The plot should be for the $1 \le x \le 1.5$. There are many ways you could go about plotting it. The exact details are left up to you. The only requirement is that the plot is reasonably accurate. Print out `exact.m` and the figure it generates. (Note: A script is just a file containing Matlab commands. It can be executed by typing in the name of the file at command line (without the `.m` extension).)

3. (3 points) Now consider the IVP

$$\begin{aligned} y' &= .5 \cdot x^2 + y^2, && (1) \\ y(0) &= 1. && (2) \end{aligned}$$

The exact solution of this equation is not defined for all x because it blows up at a certain x-value (such an x-value is called a *singularity*). The goal of this problem is to numerically find the singularity using the methods discussed in class (see also Example 2 in Section 2.6 of the book). So, run some commands to find the x-value where the solution blows up. Give your answer as a range accurate to 5 decimal places, for example "the singularity is between 2.123450 and 2.123451" (this is not the actual correct answer). (Note: the command `format long` will tell Matlab to display more decimal places.) Print out your answer and some evidence that it is correct. The evidence should show that the numerical solution is stable at the left end point of the range, but unstable and blowing up at the right end point of the range. Probably you will need to do quite a bit of experimenting to find the range. You do not need to turn in all of your work, just the final answer and evidence that it is correct. Also, on your print out, write in some hand written comments to explain what your evidence is showing.

4. (3 points) This problem is about pendulums. Suppose we have a pendulum consisting of a mass m swinging on a string or a massless rod of length L. Let $\theta(t)$ be the angle the rod makes at time t with the vertical position. Then the equation of motion of the pendulum is

$$\theta'' + \frac{g}{L}\sin\theta = 0,$$

where g is acceleration due to gravity. The conservation of energy equation for the pendulum is

$$E = \frac{1}{2}mL^2(\theta')^2 + mgL(1-\cos\theta) = \text{constant}.$$

(If you don't believe it, you can check that E really is a constant by using implicit differentiation to show that $dE/dt = 0$.) The energy equation can be solved for θ' to get the differential equation

$$\theta' = \pm\sqrt{\frac{2E - 2mgL(1-\cos\theta)}{L^2m}}.$$

This can be rewritten as

$$\theta' = \pm\sqrt{A + B\cos\theta}$$

with A, B the constants

$$A = \frac{2E}{L^2m} - \frac{2g}{L}, \quad B = \frac{2g}{L}.$$

For this problem, assume $m = 1$ kg, $g = 9.8$ m/sec, and $L = 1$ m. Then $B = 2\cdot 9.8$ and $A = 2E - 2\cdot 9.8$ and the equation is

$$\theta' = \pm\sqrt{2E - 19.6 + 19.6\cos\theta}.$$

Now, first assume E is really large, say $E = 50$ Nm. Then the pendulum will be swinging around and around in a circle, and θ' will never change sign. Assume that θ is increasing, so the equation is

$$\theta' = \sqrt{81.4 + 19.6\cos\theta}.$$

Use your Runge-Kutta program to find out how long it takes the pendulum to make one complete revolution. Your answer should be accurate to at least 3 decimal places. Note that you should get the same answer no matter what you take for the initial value of θ.

Next, assume E is small, say $E = 1$ Nm. Then the equation is

$$\theta' = \pm\sqrt{-17.6 + 19.6\cos\theta}.$$

Then the pendulum will swing back and forth. How long does it take to make one complete cycle? Your answer should be accurate to at least 3 decimal places. Hint: The sign of θ' will change at a certain time, and the program will start malfunctioning at this time because the quantity under the square root sign will become negative. The easiest way to do this problem is start with $\theta(0) = 0$, and then find the first value of t where θ' should change sign. At this instant, the pendulum will have completed 1/4 of a complete cycle.

Print out your answers along with some commands and output that show they are correct.

5. (Do not hand in: For fun only!) For small energies, the angle θ stays small, so $\sin\theta \approx \theta$, and the original ODE

$$\theta'' + \frac{g}{L}\sin\theta = 0$$

can be approximated by the ODE

$$\theta'' + \frac{g}{L}\theta = 0.$$

The smaller θ stays, the better this approximation is. The energy E satisfies $E \geq 0$, and the closer it is to 0, the smaller θ has to stay. So as E goes to 0, the approximate ODE becomes more and more accurate. Using $g = 9.8$ and $L = 1$ as in the previous problem, the approximate ODE has the exact solution

$$x(t) = C\cos\left(\sqrt{9.8}t - \alpha\right),$$

with C and α parameters. So it take $2\pi/\sqrt{9.8} = 2.00708992315449$ seconds to complete 1 cycle.

Now use your program and the equation

$$\theta' = \pm\sqrt{2E - 19.6 + 19.6\cos\theta}$$

to find out how long it takes the pendulum to complete 1 cycle, first with $E = .1$, then $E = .01$, then $E = .001$, and finally $E = .0001$. Your answers of course should be getting closer and closer to 2.00708992315449.