Numerical Analysis Semester 081 Problem Set 15 Reference Sections 8.04

1. For the following system of first order differential equations with initial conditions,

$$\frac{dx}{dt} = x - 4y$$

$$\frac{dy}{dt} = x + y, \quad x(0) = 2; y(0) = 3$$

a) complete the first **two** steps by hand using Euler's Method with a step size of h = 0.05.

b) complete the first step (just one step!) by hand using Runge-Kutta order 4 Method with a step size of h = 0.05.

2. Rewrite the following 2^{nd} order differential equation with initial conditions as a set of first order differential equations with initial conditions and write in standard (Normal) form.

$$\frac{d^2y}{dt^2} + 3t\frac{dy}{dt} + (y(t))^2 = \sin(t), y(0) = 1, y'(0) = 5$$

3. (Computer Exercise)

a) Rewrite the following 2^{nd} order differential equation with initial conditions as a set of first order differential equations with initial conditions and write in standard (Normal) form.

$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 5y = e^{-t}, y(0) = 3, y'(0) = -5$$

b) Apply MATLAB's ode45 method to solve the system over the interval [0,5]
Write down the commands that you used (command with appropriate arguments)

c) Compare the solution to the true solution at the point t = 5.

$y(t) = 3e^{-2t}\cos(t) + e^{-2t}\sin(t)$ (write down both values and compute the error)

d) Plot the true solution (part c) and your approximate solution from part b) using MATLAB commands like those in the last lab. Print the graph.

For part b) You will need to set up a function m-file fode.m representing the "vectorized" right hand side of the system you developed in part a)

Example of how this is done: For the system in problem 1 you could use:

```
function dy = fode(t,y)
% y is now a vector representing the right hand side of d.e.
% y(1) is first function y(2) is second
dy = zeros(2,1);    % a column vector representing right hand sides
dy(1) = y(1) - 4*y(2)   % adapt these for your problem
dy(2) = y(1) + y(2)
```

The call for this system on the interval [0,1] would look like:

>> [t,y]=ode45(@fode,[0,1],[2,3]);

Plot the two approximating functions for the solution to the system:

>> clf
>> plot(t,y(:,1) 'r-')
>> hold on
>> plot(t,y(:,2),'g-')