**Due Date:**

**Purpose:** The goal of this project is for you to learn how to solve first order ODEs graphically using free online software, as well as solving these equations numerically, and symbolically using MATLAB, a programming language and numeric computing environment.

A compartment model is one type of mathematical model that uses first order ODEs to describe the relationship between variables. An example of a one-compartment model would be to look at the amount of money in a savings account that you make regular deposits into. Another example is the SIR Model used to model the spread of epidemics, which consists of three compartments. One for people susceptible to the disease, one for infected people, and one for recovered people.

As part of this project, you will using a one-compartment model to look at the effect a clean river flowing into a lake has on the ability to get rid of chemicals being dumped into the lake.

**Learning Goals**

The goals of this assignment are for you to learn

* how to use a computer to solve first-order ODE’s;
* create computer generated slope fields and identify equilibrium solutions;
* to numerically solve first-order ODE’s using Euler’s Method and the MATLAB **ode45** command;
* to analytically solve first order ODEs using the MATLAB **dsolve** command; and
* set up and solve a one-compartment modeling problem using MATLAB.

**Deliverable**: You are to upload a pdf of your report to the project Canvas page by the due date.

**Your report should contain the following:**

* Your first and last name.
* The solution to each problem, and a graph when requested. Be sure each graph is labeled with the differential equation the graph is representing, and that the horizontal and vertical axes are correctly labeled.
* Your audience for this report will be students in this class who are learning the material at the same time as you. Make sure you explain each section of your report as clearly as possible. Provide answers to the questions asked (please write complete and grammatically correct sentences).
* Sequences of commands used to generate your output when using MATLAB.

**Getting help**: If you have questions or difficulties, be sure to ask for help in the discussion board for Project 1.

**Part I: Graphical Methods Using a Free Online Slope Field Graphing Software (13 points)**

1. For the equation given by: 

1. You are to solve this part of the problem analytically, that means working out a solution by hand, without using a slope field.
2. If , how many equilibrium solutions are there?
3. If , how many equilibrium solutions are there?

**What to submit for 1a:** Your written or typed mathematical work showing how you found the equilibrium solutions for each of the two cases ( and ). For each case clearly state how many equilibrium solutions you found.

For problems 1b, 1c, and 1d you are to use the online program at <http://www.bluffton.edu/~nesterd/java/slopefields.html?Undamped>

to the find the slope field for each of the problems 1b, 1c and 1d.

**Note**: When using this software, you can add several initial conditions to one graph, you just need to do them one at a time.

1. For , create a slope field for the differential equation. Is the solution stable?
2. For , create a slope field for the differential equation. Is the solution stable?
3. For , create a slope field for the differential equation. Are the solutions stable?

**What to submit for 1b, 1c and 1d:** Each graph produced using the online software showing the slope fields for 1b, 1c and 1d. Be sure to include several solution curves on each graph. State in writing for each of 1b, 1c and 1d if the equilibrium solutions you can see in the corresponding slope field are stable or not. The equilibrium solutions you see in the graphs should match up with what you found analytically in part 1a.

2. For problem 2, you are to use the online program at <http://www.bluffton.edu/~nesterd/java/slopefields.html?Undamped>

to the find the slope field for the problem and add the initial condition for this problem. Graphically solve the initial value problem  for 0 ≤ t ≤ 10.

**What to submit for 2:** The graph produced using the online software showing the slope field for the differential equation on the domain from *t* = 0 to *t* = 10. The graph of the slope field should include the solution curve with initial value *x*(0)=3.14.

**Part II: Numerical Methods (19 points)**

We will be using the MATLAB programming language to numerically and analytically solve differential equations.

3. The code below is the start of a program to solve the initial value problem

for

numerically for a given number of steps, *n*.

The code to implement Euler’s Method is missing, so you need to review Euler’s Method, and determine what lines of code are needed to complete the program.

% This script solves the initial value problem using Euler's Method

n=20; % n = number of steps

t0=0; % t0 and t1 specify the time interval

t1=10;

y0=pi; % y0 is the initial condition

h=(t1-t0)/n; % h is the step size

t(1)=t0;

y(1)=y0;

for i=1:n

Add code to implement Euler’s Method here

end;

plot(t,y)

title('dx/dt = sin(xt)')

xlabel('t'), ylabel('x')

For programs that are more than a few lines long, or that you will want to modify in the future, it is best to save the program as a Script file. To do this, click on the *New Script* command in the upper toolbar to open the editor. Copy the following code into the editor. Save the program to your MATLAB folder.

At the MATLAB prompt run the script file three times changing the appropriate line of code to change the value of *n*.. For the first run use *n* = 20, then *n* = 40, and then *n* = 200.

**What to submit for 3:** Submit the code used and then the results from calling the script three times (once with *n* = 20, then with *n* = 40, and then with *n* = 200). Copy each of the three graphs and paste into your Word document along with your code.

4. **Example**: The function **ode45** is a program built into MATLAB that can be used to solve first order differential equations. The following code solves the IVP  on the interval 

% First order ODE solution using ode45 with Anonymous Function Method

% script name: ode45solver.m

clc; clear all; clf

% use anonymous function

ode1=@(x,y) (-2\*x.\*y);

% call solver with anonymous function name

% independent variable range and initial value of dependent variable

[x,y]=ode45(ode1,[0,5],1);

% plotting segment

plot(x,y, 'Linewidth',2)

xlabel('x'), ylabel('y'), grid on,

title('Solution to ODE dy/dx = (-2xy')

**Task:** Change the above code to solve the IVP for  with  Graph the solution to the IVP.

**What to submit for 4:** Paste into your Word document a copy of the modified code and a graph showing the solution to the IVP.

**Part III: Analytical Methods (23 points)**

MATLAB’s differential equations solver function called **dsolve** provides analytical solutions to first-order differential equations, with or without initial conditions.

5. **Example**: The general solution to the differential equation, can be obtained using the code

syms y(t) a

dsolve(diff(y,t) == a\*y)

**Task**: Find the general solution to the differential equation .

**What to submit for 5:** Submit the code used in 5 and copy and paste the MATLAB output from the code given in 5 into your Word document.

6. **Example**: To find the solution to the initial value problem with use the code

syms y(t)

eqn = diff(y,t) == 2\*y;

cond = y(0) == 5;

ySol = dsolve(eqn, cond)

ySol is a symbolic expression. That means you cannot evaluate ySol at  by typing ySol(2). Instead, use the **subs** command to display the values of ySol at certain times *t*. For instance, the value of ySol at is given by typing the command:

subs(ySol,'t', 0.5) % to get exact answer

double(subs(ySol,'t', 0.5)) % to get an approximate answer

In practice, the goal is to study a family of solutions at various initial conditions. To do this, the initial conditions need to be placed in a vector. As an example, to evaluate ySol at the three values of *t*: use the command:

double(subs(ySol, 't', [0.5,0.6,0.7]))

To plot the solution ySol on the interval [0, 2] use the commands:

fplot(ySol, [0, 2])

xlabel('t'); ylabel('y');

title('Solution to IVP dy/dt = 2y, y(0)=5');

**Task**: Solve the initial value problem  Graph the solution to the IVP, and evaluate the solution at the three values of *t*: .

**What to submit for 6** Submit the lines of code used along with the output solutions and the graph from MATLAB that were outputted from each of the commands.

7. **Example**: This example solves an IVP and plots solution curves for different initial conditions.

IVP: 

The initial conditions are  on the interval 

syms y(t) c;

eqn = diff(y,t) == 2\*y;

ic = y(0) == c;

Sc = dsolve(eqn, ic)

% Since we wish to plot the solutions curves with initial values y(0) = 4, 5, 6, 7 on the

% interval [0, 2], also type the following commands into the MATLAB prompt:

for cs = 4:1:7; % this create a vector starting at 4 and ending at 7, with a

Scs = subs(Sc, 'c', cs); % step size of 1

fplot(Scs, [0, 2])

hold on

end

hold off

axis([0, 2, 0, 25]);

title('Solution to IVP dy/dt = 2y, y(0)=c');

lgd = legend ('4', '5', '6', '7', 'Location','southeast');

title(lgd,'Initial Conditions');

xlabel('t'); ylabel('y'); grid on;

**Task**: Solve the following IVP and plot solution curves for the different initial conditions.

IVP: 

**What to submit for 7:** Submit the lines of code used along with the output solutions to each command, and the graph from MATLAB that was produced from your code.

**Part IV: Application Problem (17 points)**

8. Suppose a lake has a river running through it and a factory built next to the lake begins releasing a chemical into the lake. The water running into the lake is free of the chemical, and initially the lake is free of the chemical. Assume that the factory dumps chemicals into the lake at a rate of *T* = 100 kg/day and that the volume of the lake is *V* = 10,000 m3.

1. If the river is flowing at a rate of *R* = 1,000 m3/day, how does the mass of chemical in the lake evolve over a 60 day time period? State the initial value problem you are solving. Show a graph of your solution and discuss any observations. Make sure each axis of your graph is properly labeled and there is a title for the graph. What is the concentration (in kg/m3) of the chemical in the lake after 60 days?
2. Now assume that the river is flowing at a rate of *R* = 2,000 m3/day. How does the mass of chemical in the lake evolve over a 60 day time period? State the initial value problem you are solving. Show a graph of your solution and discuss any observations. Make sure each axis of your graph is properly labeled and there is a title for the graph. What is the concentration (in kg/m3) of the chemical in the lake after 60 days?

**What to submit for 8:** For each of 8a and 8b:

* Submit your model differential equation and your initial conditions resulting from the problem description given in 8a and 8b.
* Include a graph (you can use code from previous problems for these two problems) in order to produce a graphical solution of the initial value problems.
* Write at least one sentence describing what each graph tells you about the amount of the chemical in the lake over time.
* Submit the number (based on your graph or analytic work) to the question asking what the concentration of the chemical is in the lake after 60 days. In both cases this can be calculated from information in the appropriate graph.