Differential Equations, Fall 2016
Computer Project 1, Due Friday, September 9

This assignment will walk you through some basic Matlab commands. You should hand in the following things:

a) a printout of the programs “slopefield.m” and “graphslopefield.m”

b) a printout of the output generated by running the commands from problems 1)-3) below

c) a printout of the slopefield graph generated in problem 5)

1. (3 points) Open Matlab. Entering commands into the command window will tell Matlab to do things immediately. Using a semicolon will suppress the output. Enter the following commands:

```
> 4+4
> 4+4;
> x=5
> y=6;
> x+y+x*y-3*x^2
```

2. (3 points) The basic data type in Matlab is a matrix. A matrix is a 2d array of numbers. Individual entries can be accessed by their coordinates. The first coordinate is the vertical position (row number) and the second coordinate is the horizontal position (column number).

```
> A=[1 2;3 4]
> A(1,1)
> A(1,2)
> A(2,1)
> A(2,2)
> A(1,1)=-10
> B=[1 2 3 4]
> C=[1; 2 ;3; 4]
```

The transpose command ' turns rows into columns and vice versa. Input the following commands to see how this works.

```
> A'
> B'
> C'
```

A colon can be used to access a range of entries. The following commands give some examples.

$>$ `A(:,1)`

$>$ `A(:,2)`

$>$ `A(1,:)`

$>$ `A(2,:)`

$>$ `B(1,2:4)`

$>$ `C(3:4,1)`

$>$ `1:10`

$>$ `2:5`

Matrices can easily be appended together to make bigger matrices.

$>$ `X=[A A]`

$>$ `Y=[A;A]`

$>$ `Z=[X;B]`

$>$ `W=[Y C]`

3. (3 points) Each user created function in Matlab resides in its own file of the same name (with a .m extension). We will walk through creating a new function in this problem.

   First, some basic file commands. The command pwd will print working directory, the command cd will change directory, the command ls will list files, and the command mkdir will make a directory.

   Create a directory for yourself. In this directory, create a file called "test.m" (by selecting New → Function from the menu at the top). (Alternatively, open up any text editor and create a new file called test.m in your directory. Make sure the text editor does not put a .txt extension onto the file.)

   Enter the following text into the file test.m:

```
function A=test(B,C)
    A=B+C;
end
```

   The first line means that a function called test is being defined. The function takes in two inputs (B and C), and returns the value A. Run the function using the following command from the command window:

   $>$ X=[1 2 3]

   $>$ Y=[4 5 6]

> test(X,Y)

4. (3 points) Now we are going to create a slope field. In this problem we will create the slope field function. The slope field function will be defined piecewise, so we will need to use an if statement to program it. In the next problem we will graph the slope field.

   The slope field we will consider is the one for the equation

$$\frac{dy}{dx} = f(x, y)$$

   with

$$f(x, y) = \begin{cases} y(y-2) & y \geq 0 \\ .5y^2 \cos x - y & y < 0. \end{cases}$$

   Create the function $f(x, y)$ in a Matlab file called "slopefield.m". (So the function will be called from Matlab as slopefield(x,y)). Here is some code to get you started (you need to fill in the ...).

```
function val=slopefield(x,y)
    if y >=0 then
        val=...
    elseif y<0 then
        val=...
    end
end
```

5. (3 points) Now we want to call this function for some values of (x, y) and plot a little line segment of the correct slope for each of these points. This will require several commands, so we will create a new file to hold these commands. Matlab calls such a file a script, select New → Script from the menu and create a file called "graphslopefield.m". Here is some code to get you started. You need to fill in the .... Some of the commands are explained below.

```
xvals=linspace(-4,4,41);
yvals=linspace(-4,4,41);
hold on;
for x=xvals
    for y=yvals
        slope=slopefield(x,y);
        ...
        xnew=...
        ynew=...
        plot([x xnew],[y ynew])
    end
end
```

The first two lines create the (x, y) values to use. The command "linspace(a,b,n)" returns a list of n evenly spaced numbers starting at a and ending at b. The command "hold on" tells Matlab to plot everything on the same graph. Then we have two "for ... end" loops, the first one loops over all entries in xvals, and the next one loops over all entries in yvals. In general, a loop of the form

```
for i=data
   ...
end
```

will tell Matlab to run the code ... in the interior of the loop several times. The first time it runs, i will be set to the first entry in data, the second time it runs i will be set to the second entry in data, and so on. In the graphslopfield.m program, there are actually two nested for loops, which will cause the code to be run for all combinations of (x,y) values with x in xvals and y in yvals. In the interior of the for loops is a plot command which will plot a line segment from the point (x,y) to (xnew,ynew). What you need to do is write in the code to calculate xnew and ynew so that the line segment has the correct slope. Also, from an aesthetic point of view, it would be nice if you could find a way to make it so that the line segment always has the same length.

Now run your program by calling it from the command prompt:

> `graphslopefield`

This should cause a graph of the slope field to pop up (it might take a little while). You might want to adjust the size of the graph so that the x and y axes have the same scale (just drag the corner of the window to adjust the size). You can draw in some solution curves on the printout of the slope field if you want to.