Mech 221: Computer Lab 5

Hand in the solutions to the three questions in the lab at the *end* of the lab.

Question 1: Using "if" statements

In this question, we will program some basic piecewise functions using "if" statements. Some functions that we know could make use of the "if" statement are the absolute value function, the maximum function (i.e: the maximum of two numbers), and the Heaviside function introduced in the pre-lab. We will be programming all of these.

First, let us consider the maximum function, $\max(a, b)$ operating on two numbers a and b. This function will return the value of a if $a > b$, or it will return b otherwise. How do we tell a computer to do this? Follow the simple steps below.

First make a new .m file function, and entitle it '`myMax.m`'. This function will accept two numbers as input and return one number of output. Now consider what we want this function .m file to do if we pass it two numbers. Considering how we defined the max function above, we should try to code the 'myMax' to do the same thing.

- Write the code to complete the '`myMax`' function using ONE "if" statement. Naturally you will want to test it to make sure it works. (Note: When we say "if" statement, this includes "if-else" statements).
- Write similar .m file functions for the absolute value function, and the Heaviside function. Name them '`myAbs.m`', and '`myHeaviside.m`' respectively.
- *Submit the code for your .m file functions:* `myMax.m`*,* `myAbs.m`*, and* `myHeaviside.m`

Question 2: Solving a Oscillator with Heaviside Forcing

Recall from the pre-lab that you were asked to solve an oscillator with discontinuous forcing analytically. Now we will get you to solve it numerically as well. Recall the initial-value problem:

$$\ddot{y} + y = H(t-2), \quad y(0) = 0, \dot{y}(0) = 1, \quad 0 \le t \le 10 \tag{1}$$

- Recall that you rewrote this as a first order system in the pre-lab. Write an .m file function that will calculate $\vec{f}(t, \vec{y})$ for our problem $\dot{\vec{y}} = \vec{f}(t, \vec{y})$, $\vec{y} = (y, \dot{y})$. You will need to use `myHeaviside` in your .m file function.

- Use `ode45` to get a numerical solution for the above problem. Plot the approximate solution for $y(t)$. On the same plot add the exact solution to this problem you found in the pre-lab.

- *Hand in the appropriately labelled plot generated above.*

Question 3: The Oscillator with Surface Friction

The scaled equation introduced in the pre-lab that governs an oscillator subject to friction is given by:

$$\ddot{y} + y = F(y, \dot{y}) \quad y(0) = 1, \dot{y}(0) = 0 \tag{2}$$

The frictional force is scaled to $\beta = 0.3$. If $\dot{y} > 0$ then $F = -\beta$; if $\dot{y} < 0$ then $F = \beta$ (the friction force opposes the motion if the object is in motion). If $\dot{y} = 0$ (the mass is not moving) there are three cases to consider:

1. if $|y| \leq \beta$ then $F = y$ (the spring force $-y$ cannot overcome the frictional force and the mass stays at rest).

2. if $y > \beta$ then $F = \beta$.

3. if $y < -\beta$ then $F = -\beta$.

- Write the problem (2) as a first order system.

- Write a MATLAB function that implements the RHS vector of this system *exactly*. Print out your function when you have it working.

- Use `ode45` to get a numerical solution for the above problem. Plot the approximate solution for $y(t)$. On the same plot add the exact solution to this problem you found in the pre-lab. Print out this plot.

- Use `ode45` to get a solution to the problem when $\beta = 0.07$. Plot the approximate solution for $y(t)$. Make sure you use a long enough time interval for the computation to see the mass come to rest on the plot. Print out this plot.

- Notice that these computations use a lot more time steps than previous computations you have done. Consider Figure 3 in the pre-lab and the discussion around it. Describe why this problem is difficult for MATLAB's `ode45` solver.

- *Hand in the two appropriately labelled plots generated above and your code for the RHS function of the system corresponding to the oscillator with firction. Also hand in your discussion of the reason* `ode45` *uses so many time steps for this problem.*