Mech 221: Computer Pre-Lab 4

Hand in the solutions to the two questions in the pre-lab at the *beginning* of the lab.

In the upcoming lab, you will solve systems of linear equations numerically using code built into MATLAB, as well as the Forward Euler algorithm. More specifically, this pre-lab will cover:

- Accessing elements of matrix in MATLAB: individual entries, rows and columns.
- The MATLAB command `legend` that will enhance figures with many plots on them.
- An introduction to "`ode45`", a numerical differential equation solver built into MATLAB.
- Transforming second and higher order ODEs into first-order systems of equations. First order systems can be solved with `ode45`.

MATLAB commands to know for the lab

`A(1,2)` : If `A` is an $n \times m$ matrix in MATLAB (n rows and m columns) this will return the value in the first row, second column.

`A(1,:)` : If `A` is an $n \times m$ matrix in MATLAB (n rows and m columns) this will return the first row as a vector (a row vector of length m).

`A(:,2)` : If `A` is an $n \times m$ matrix in MATLAB (n rows and m columns) this will return the second column as a vector (a column vector of length n).

`legend` : This command has many options, we will start with the simplest way to use it. Once you have finished plotting all of your plots on the same figure (remember to use `hold on`), remember the order that you plotted them in. Once you have the names of your plots, type in the following command:

```
legend('name1','name2',...);
```

where `name1, name2, ...` are the names of the plots that will appear in the legend. The ordering of their entry into the legend command corresponds to the order by which you plotted them (i.e: `name1` will correspond to the line first drawn in the figure). To take the legend away, type in `>> legend off`.

The MATLAB command `ode45` that approximates solutions of Differential Equations

Given a function f, an interval $[a, b]$, and initial condition c, MATLAB has built-in routines to solve

$$\dot{y} = f(t, y), \quad y(a) = c, \quad a \leq t \leq b \tag{1}$$

MATLAB can be used to find approximate solutions to vector DEs, but let's start with this scalar DE first. A general purpose algorithm is in `ode45`, which can be called using the following syntax where `a`, `b` and `c` have been defined with the scalar values from the problem above.

```
[t,y] = ode45('func',[a b], c);
```

The elements of this call are discussed below from left to right

- `ode45` is a MATLAB function that returns two vector outputs. The square brackets with the comma separator on the left is how you can assign these two outputs to named variables `t` and `y`.
- `t` is a column vector containing time values between a and b at which the solution was computed.
- `y` is vector of the same length as `t` containing corresponding y values. The command `plot(t,y)` would then plot the approximate solution between a and b.
- `func` is the name of an .m file function with two arguments (t and y) that returns the value $f(t, y)$.

Let us look at a vector first order DE with two components $y_1(t)$ and $y_2(t)$ also solved for t between a and b.

$$\begin{aligned} \dot{y_1} &= f_1(t, y_1, y_2), \quad y_1(a) = c_1 \\ \dot{y_2} &= f_2(t, y_1, y_2), \quad y_2(a) = c_2 \end{aligned}$$

As seen below (and in the lectures), second order scalar equations can be converted to first order systems with 2 components. Third order scalar equations can be converted to first order systems with 3 components, etc. In this way, higher order problems can be approximated with numerical methods.

To approximate the two component problem above, the intial data c_1 and c_2 must be put into a *column* vector:

```
c = zeros(2,1);
c(1) = c1;
c(2) = c2;
```

The command `c = [c1 c2]'` would have the same result. The call to `ode45` has the same syntax as the scalar case

```
[t,y] = ode45('func2',[a b], c);
```

where `t` is a column vector of times at which the solution is computed as before and

- `y` is now a matrix with 2 columns and a number of rows equal to the size of `t`. `y(:,1)` contains the approximation of y_1 in a column vector at the corresponding times, `y(:,2)` the approximation of y_2. `y(4,:)` is a row vector with the approximate values $y_1(t)$ and $y_2(t)$ at the time `t(4)`. The command `plot(t,y)` will plot two curves, $y_1(t)$ in blue and $y_2(t)$ in green.

- `func2` is the name of an .m file function with two arguments (t and y) where y will be a column vector with two components y_1 and y_2. The function must return a column vector with two entries $f_1(t, y_1, y_2)$ and $f_2(t, y_1, y_2)$.

ODE systems with more than two components follow the same pattern.

Transforming a higher-order DE to a first order system

The solver can only deal with first-order equations, but it compensates for this by working with *vector-valued* unknown functions as descriobed above. A scalar ODE with lots of derivatives can be transformed into a first-order vector ODE with lots of components. Here's how.

Consider an n^{th} order ODE:

$$\frac{d^n y}{dt^n} = f\left(t, y, \frac{dy}{dt}, ..., \frac{d^{n-1}y}{dt^{n-1}}\right) \tag{2}$$

Now, consider a new set of variables $\{x_1, x_2, \cdots x_n\}$, and define them the following way.

$$x_1 = y, \quad x_2 = \frac{dy}{dt}, ... \quad , x_n = \frac{d^{n-1}y}{dt^{n-1}} \tag{3}$$

Now consider what happens when we differentiate these variables $\{x_n\}$ with respect to t:

$$\frac{dx_1}{dt} = x_2, \quad \frac{dx_2}{dt} = x_3, ... \quad , \frac{dx_{n-1}}{dt} = x_n \tag{4}$$

$$\frac{dx_n}{dt} = f\left(t, x_1, x_2, ..., x_n\right) \tag{5}$$

Notice that we now have a vector-valued, first-order ODE which we can solve with `ode45`.

Example: Consider the following ODE:

$$\ddot{y} + y = 0 \quad y(0) = 1 \quad \dot{y}(0) = 0 \tag{6}$$

The solution for this problem is $y(t) = \cos t$. If we want to test `ode45` on this known solution we must first transform this problem into a linear system. Following the notes above, we introduce $x_1 = y$ and $x_2 = \dot{y}$. For these variables, initial conditions are given: $x_1(0) = 1$ and $x_2(0) = 0$. Remember to put these values into a column vector before using them as the last argument of `ode45`. Now by this choice of variables, $\dot{x_1} = x_2$. Since $x_2 = \dot{y}$ when we consider $\dot{x_2}$ we see that it is $\ddot{y}$ which by the DE is equal to $-y$ which is $-x_1$. Summarizing

$$\dot{x_1} = f_1(t, x_1, x_2) = x_2 \tag{7}$$

$$\dot{x_2} = f_2(t, x_1, x_2) = -x_1 \tag{8}$$

When you write your MATLAB function for the RHS of this system, don't forget to put the vector $\vec{f} = (x_2, -x_1)$ in a column.

Question 1: Transforming a third-order ODE

Transform the third order equation for $y(t)$ given by

$$y''' + e^t y'' + \cos(y) = 0$$

into a first order system with three components. *Hand in the resulting system and your work.* Note that in lab #4 you will be solving such a third order equation numerically (but not this one).

Question 2: RLC Circuit

An RLC circuit is an electric circuit in which a resistor, capacitor and inductor are hooked up in series. This circuit is (or will be) the object of study in your Electronics lectures due to it's oscillatory properties. For an RLC circuit, we have the following equations:

$$V_L + V_C + V_R = 0,\ V_L = L\dot{I},\ V_R = IR,\ \dot{V}_C = I/C \tag{9}$$

which, after setting $V = V_C$, leads to the system of equations

$$\begin{array}{rcl} \dot{V} & = & \frac{1}{C} I \\ \dot{I} & = & -\frac{1}{L} V - \frac{R}{L} I. \end{array} \tag{10}$$

Suppose that $R = 4$, $L = 2$, $C = 0.5$, and that the initial conditions are given by $V(0) = 2$, $I(0) = 1$.

- Combine the system (10) to get a second order equation for I.
- Rewrite the system in the form $\dot{\vec{X}} = A\vec{X}$ for a 2×2 matrix A (Make sure to define $\vec{X}$ appropriately).
- Show that the exact solution is given by

$$\begin{pmatrix} V(t) \\ I(t) \end{pmatrix} = \begin{pmatrix} 2e^{-t} + 4te^{-t} \\ e^{-t} - 2te^{-t} \end{pmatrix} \tag{11}$$

- *Hand in the expression you obtain for the second order equation for I and the matrix form of the system you derived above. Show all your work. Show carefully that the functions for V and I above satisfy the DE system.*

Coming up in Lab #4

Completing the pre-lab will prepare you for the lab, in which the following will be covered:

- Learning how to use `ode45`.
- Using the Forward Euler method to solve a *system* of ODEs.
- Using `ode45` to solve a system of ODEs.
- Converting a new third order DE to a system with three components and finding its solution.