Matlab for 12.006J/18.353J Problem Set 3

Note: This document is available for download in the assignments section.

Another Note: PLEASE use the matlab command `subplot` to put more than one plot on a page and save paper and trees. There are more details below in Section 3. Thank you.

1 Solving ODEs

You can use the matlab function `ode45` to solve sets of ODEs. ode45 integrates a system of ordinary differential equations using 4th and 5th order Runge-Kutta formulas. To get started, we will use a simple example, the undamped pendulum:

$$\frac{d^2\theta}{dt^2} + \omega^2 \sin\theta = 0 \tag{1}$$

Let $y_1 = \theta$ and $y_2 = \dot{\theta}$, we can rewrite this ODE as two 1st order ODEs:

$$\frac{dy_1}{dt} = y_2 \tag{2}$$

$$\frac{dy_2}{dt} = -\omega^2 \sin y_1 \tag{3}$$

Copy the file `vanderpol.m` to your working directory (note that the equations in the code are for an undamped pendulum, though you will be changing them to analyze the Van der Pol equation). Read the comment carefully.

The comments are those lines which begin with a % in the file. This file specifies the system of ODE for ode45 to integrate. To get help on ode45, type

```
>> help ode45
```

To integrate the above ODEs, for Matlab versions 5.2 and higher, type

```
>> [t,y] = ode45('vanderpol', [0 10], [1.0 1.0]);
```

The first argument 'vanderpol' is the name of the function defined in the file `vanderpol.m`. The name of the function must be delimited by single quote. The 2nd argument [0 10] specifies the time interval of integration, ie. t=0 to 10. The 3rd argument [1.0 1.0] are the initial conditions, ie. y(1) = 1.0 and y(2) = 1.0 at t=0.

[t,y] is the set of data returned by ode45, t is a vector that contains the time, y is a matrix of 2 columns. y(:,1) is the set of y(1) and y(:,2) is the set of y(2). Try typing

```
>> y

y =

    1.0000    1.0000
    1.0492    0.9571
    1.0961    0.9129
    1.1409    0.8678
    1.1833    0.8217
    1.3591    0.5829
    1.4741    0.3356
    1.5268    0.0857
    1.5170   -0.1640
    1.4450   -0.4129
    1.3108   -0.6584
    1.1164   -0.8927
    0.8665   -1.1020
    0.5690   -1.2677
    0.2386   -1.3661
   -0.1065   -1.3818
   -0.4455   -1.3132
   -0.7126   -1.1971
   -0.9506   -1.0403
   -1.1522   -0.8552
   -1.3125   -0.6540
   -1.4313   -0.4415
--more--
```

The first column is the θ and second column is the $\dot{\theta}$, which are defined as y(1) and y(2) in `vanderpol.m`

To do this problem set, modify the vanderpol.m code to implement the appropriate system of ODEs and type

```
>> [t,y] = ode45('vanderpol', ..., ...);
```

2 Plotting Your Results

To put more than one plot on a figure, use the subplot command. For example, the following command tells matlab that you are going to put three columns and

two rows of plots on your figure, and that you currently want to work in the first of those subregions.

```
>> subplot(321)
```

If you finish giving matlab commands for the first plot, then typing the following moves you to working in the next subsection of the figure.

```
>> subplot(322)
```

For more explanation, type

```
>> help subplot
```

To obtain a plot of phase space trajectory, type

```
>> plot(y(:,1),y(:,2),'-');
```

To obtain time series plots of y(1) and y(2), type

```
>> subplot(2,1,1);
>> plot(t,y(:,1));
>> subplot(2,1,2);
>> plot(t,y(:,2));
```

To print 2 curves in one plot, type

```
>> plot(t,y(:,1));
>> hold on
>> plot(t,y(:,2));
```

This should print the time series for y_1 and y_2 together in a single plot. To disable hold, just type

```
>> hold off
```

To save the current figure as a postscript file called figure.ps for later printing, type

```
>> print -dps figure.ps
```

3 Finding Attractors

In this and most future problem sets you will be asked to simulate a dynamical system. Realize that the transient dynamics of the system going from the initial condition to an attractor are not very important. What is important is the dynamics of the system on the attractor, since this will be the behavior of the system at all future time. The charachteristics of the system are revealed from the attractor not the transient process. RUN YOUR SIMULATIONS LONG ENOUGH to reach it. These attractors may be a fixed point, limit cycle, or a strange one. It is easy to realize when the system has reached the attractor. Most of the time you will notice the system has settled into some sort of cycle, as opposed to a transient growth or decay process.

4 Saving Your Work

Rather than typing all your matlab commands at the command prompt, it is possible to type them into a matlab script file, which can be edited using the built in matlab editor or any other text editor. The script should have a descriptive name which ends in .m (for example: my_ps3_script.m). The advantage of doing this is that it allows you to run several commands at once, simply by typing the name of the file at the matlab command prompt. It makes it easy to change things without redoing everything, and it gives you a record of the commands you used. An simple example script would be:

```
%%%%%%%%%%%%
% my_ps3_script.m
%
% A script I'm writing to illustrate the concept of scripting.
% The % symbols comment out anything that comes after them.

close all; clear all;  % This clears old figures and variables
x=[0:0.01:10];
y=sin(x);
figure;
plot(x,y,'b--')
title('A test plot of the sine of x');
xlabel('x'); ylabel('sin(x)');
print -dpsc sinx.ps

%%%%%%%%%%%%%%%
```

And this script would be run by typing

```
>> my_ps3_script
```

Finally, to save all your data in case you want to look at it again without calculating everything again, type

```
>> save pset3
```

Your data will be stored in a file named pset3.mat. The next time you start matlab and want to reload the data in pset3, type

```
>> load pset3
```