Spring 2018

Lab 2 - Vectors and 2-d Plotting

Objectives: To learn how to use MATLAB to plot data in a vector array, in particular:

- How to enter data into a vector
- How to define vectors
- How to plot the data
- How to add additional attributes to your plot

1. **Boot It Up**

   (a) Start up MATLAB

   - Before you begin all lab assignments, start by typing *diary lab*02 to begin recording your session. This is critical since you will be required to show this file to get credit for your lab.

2. **Vector Basics**

   (a) MATLAB is a numerical package so all variables must be assigned a number (or numbers) to avoid errors. You may also want to use capital letters to indicate a vector (this may be handy later on when you have a long script and you need to follow what's going on). Enter these vectors into the command window:

```
X = [2 9 0 7 5 3]
Y = [24 65 -84 7 -62 33]
Z = [b r o n y]
Z = [3 9 3 9]'
check your session variables by typing whos
```

   (b) Operations on vectors are simple, but can also be a source of frustration if the inner dimensions do not match. Evaluate these in the command window (note: some of these will error):

```
X+X
X+Y
X*X
X*X'
X*Y
X*transpose(Y)
X+Z
Z^2
```

   (c) Element-wise operations do not necessarily depend on vector dimensions to work properly. Evaluate these in the command window:

```
X.+X
X.+Y
X.*X
X.*X'
X.*Y
X.*transpose(Y)
X.+Z
Z.^2
X1 = cos(X)
Z1 = exp(Z)
```

   (d) Vectorization is a process in MATLAB that allows you to build long vectors very quickly. There are also built-in functions that fill your array with prescribed values. The *linspace* command divides your array into equally spaced values (default is 100). Try these:

```
W = [0:20]
V = 0:5:500
U = [-35:-2:12]
T = 0:.001:1
X2 = linspace(0,2*pi)
Z2 = linspace(0,2*pi,1000)
A = zeros(1,30)
B = ones(30,1)
C = rand(1,7)
D = nan(7,1)
check your session variables by typing whos
```

(e) Inspecting elements in your vector can be done by using parentheses. You can also add elements to your array if necessary. Enter these into the command window:

```
X(4)
X(8)
X(7)
X(8) = 23
X(7)
X
T(2:9)
T(2:9)+T(10:17)
X(1:3)+Y(3:6)
```

(f) There are several helpful functions built-in to MATLAB that you can use to get information about your data. These include: max(Y), min(T), length(T), sort(W), sum(X), prod(Y), median(S), mean(Y), std(S), size(T). Evaluate the following:

```
max(T)
min(X)
length(T)
sort(X)
sum(Z)
prod(Y)
median(U)
mean(V)
std(X)
size(T)
```

(g) Plotting your data in MATLAB is simple. There are many types of plots and options (the command 'clf' clears the current figure. *hold on* does the opposite by keeping data in the same plot). Try the following:

```
X = linspace(-2*pi,2*pi);
Y1 = sin(X);
Y2 = cos(X);
Z = linspace(0,3,1000);
Z1 = exp(Z);
plot(X)
clf,plot(X,Y1)
clf,plot(X,Y2)
clf,plot(Z,Z1)
clf,plot(X,Y1,'g:',X,Y2,'m-')
clf,polar(X,Y1)
P = [2 6 3 12 9 3 2];
clf,pie(P)
B = [34 67 10 45 82 50 38 39 43 32 78 29];
clf,bar(B,'r')
```

(h) One last feature that could be helpful is the use of *subplot*. A subplot provides several plots in one figure window, and the syntax is as follows: subplot(row,column,position). Once you execute a subplot command be sure to specify all attributes for that plot (xlabel, title, etc) before calling the next subplot command. Here are a few examples to try:

```
subplot(2,2,1),plot(X,Y1)
subplot(2,2,2),plot(Z,Z1)
subplot(2,2,3),plot(X,Y2)
subplot(2,2,4),plot(X,tan(X))
clf
subplot(2,1,1),plot(X,Y1)
subplot(2,1,2),plot(Z,Z1)
clear all
```

3. **Putting it Together**

   (a) From the File menu in MATLAB, select New: M-File. You should get a new blank window. In the window, type the following:

```
% Author: Justice Beaver (or your name)
% Plot cosine and sine from a to b with c data points

% reset all session variables to avoid
clear all

% Enter 3 numbers
a = input('Enter a starting point for your plot: ');
b = input('Enter an ending point for your plot: ');
c = input('How many points do you want to use? ');

% create domain array
X = linspace(a,b,c);
% calculate cosine and sine for all values of X
Y = cos(X);
Z = sin(X);

% plot the results
clf % clear any figures that may be open

% add both plots to the same figure
% you could also use:
% hold on
% plot(X,Y,'b.'),plot(X,Z,'y.')
plot(X,Y,'b.',X,Z,'y.')

% add a plot title
title('Cosine and Sine')

% change axes color to almost black
set(gca,'Color',[0.1 0.1 0.1]);

% label the x-axis
xlabel('x')
% label the y-axis
ylabel('y=sin(x) and cos(x)')

% add a legend
legend('cos(x)','sin(x)')
```

   (b) Save the script and give it the name $lab0201.m$.

4. **Time to Get Serious**

   (a) Create a script called $lab0202.m$ that solves the following problems. Put your name in the script. Label each evaluation with a comment and use appropriate variables.

- Create a column vector that has 28 elements with all values equal to 56.87. Use both the *ones* and *zeros* functions to accomplish this task.
- Create a vector of random numbers with 49 values (you may use the *randi* function if you want integers). Find the sum and product of all values in your vector; sort your vector; find the mean, median, and standard deviation of your data set; determine the length and size of your vector; find the maximum and minimum values; find the value in positions 5,9,12,34 and add them together; plot your data using the *bar* function.

(b) Write a script, *lab*0203.*m*, that plots $r = cos(2\theta)$ from 0 to 2π using both the 'plot' command and 'polar' command. Have both plots appear in the same figure by using *subplot*. (Recall $x = r\cos\theta$ and $y = r\sin\theta$ when plotting in cartesian coordinates.) If the axes are compressed, use *axis equal*. Once you have run your script and have it working, modify your code to plot $r = \cos n\theta$ where n is any integer entered by the user and the range of θ is also defined by the user (use the command *input* to prompt the user). Label your axes and include a title. Be sure to comment your code!

(c) Write a script entitled *lab*0204.*m* that draws a scaled-model of our Solar System. Use 0.005 Astronomical Units (AU) for the radius of the Sun, and the following values for the radius of the planets (if you feel bad for Pluto for being downgraded you can add it at 39.8 AU):

Planet	AU
Mercury	0.4
Venus	0.7
Earth	1.0
Mars	1.5
Jupiter	5.2
Saturn	9.5
Uranus	19.2
Neptune	30.1

Use different colors for each planet, make the axes background black, and be sure to add a title and legend to your plot. After your script is complete, use the zoom tool to interact with your Solar System. Again, be sure to comment.

(d) You should submit 5 files to your TA for review: your diary file, *lab*0201.*m*, *lab*0202.*m*, *lab*0203.*m* and *lab*0204.*m*. Do not forget to upload all of these files to Canvas!

(e) Once the TA has checked your files you are free to go. Log off of your machine before you leave the lab.