MatLAB Tutorial II: Plotting Graphs and Importing and Plotting Data

2-D graphing:

1. Graph a function.
2. Animate function graph
3. Import and graph data
4. Graph data and a function (two plots on one graph).

1. Make and plot a function. Here is a sample script:

```
clear;
clf();

%define constants
c = 3; %speed of light in 1e-8 nm/ns
A = 1; %amplitude
lambda = 400; %wavelength is 500 nm

%dependent variables
k =2*pi/lambda; %wave number k = 2?/lambda
w = 2*pi*c/lambda; % angular frequency omega = 2?f = 2?c/lambda

%variables for plotting
x = 0:1:2*lambda; %position in nm
t = 1;  %time in nanoseconds

%wave equation
E = A*sin(k*x-w*t); %wave propagating to the right

% Plot EM wave
plot(x, E) % plot wave

%Label graph
title('EM wave propagating along the X axis.')
xlabel('Distance in nanometers (X axis)')
ylabel('E Field (Y axis)')
```

2. Graph an animated function. Here is a sample script,
note the 'r' to color the graph.

To animate a graph, you essentially create an array of plots that are displayed using M(t)=getframe within a for … end loop.

```
clear;
clf();

%define constants
c = 3; %speed of light in 1e-8 nm/ns
```

```
A = 1; %amplitude
lambda = 650; %wavelength is 500 nm

%dependent variables
k =2*pi/lambda; %wave number k = 2 π /lambda
w = 2*pi*c/lambda; % angular frequency omega = 2 π f = 2 π c/lambda

%variables for plotting
x = 0:1:2*lambda; %position in nm

for t = 1:300 %This for loop will animate the graph, there are 300 frames. Step size must be an integer.

clf;
hold on; %hold allows multiple plots to be plotted on same graph.

%wave equation
E = A*sin(k*x-w*t); %wave propagating to the right

% Plot EM wave
plot(x, E, 'r') % plot wave, 'r' sets the color to red.

%Label graph
title('EM ray propagating along the X axis.')
xlabel('Distance in nanometers (X axis)')
ylabel('E Field (Y axis)')

M(t) = getframe; %This is what is doing the animation
end
```

3. Import and plot data.
Create some x,y data in Excel, export as a .csv function. Check with preview that it is appropriate data, and then import into MatLAB

For example, create 1 to 180 in col A and use Excel function in col B with the noise size in cell D1:

```
=COS(A1*3.141/90)+$D$1*RANDBETWEEN(-1,1)
```

Note: you need to export as .csv data.

Next, import data into matrix a using:

```
>> a = csvread('testdata.csv');
```

This can be any .csv data such as from Pasco DataStudio or LabVIEW.

You can select a single column of a matrix by a(:,n) where a is the matrix and n is the column number. So you could create a vector of x data and a vector of y data using:

```
>> x = a(:,1);
```

>> y = a(:,2);

As an aside, you can combine two row vectors into a 2xn matrix by:
>> e=0:1:10;
>> f=0:2:20;
>> g=[e;f];

(this makes a matrix of two rows of 11 elements)

However, if you want two columns, try transposing:
>> h=g';

or you can do the transposing first with a comma:
>> h=[e', f'];

but you probably DO NOT want to do:
h=[e'; f'];
or
h=[e, f];
because these make single column or row vectors.

Plot the data:
>>plot(x, y)
or
>>plot(a(:,1), a(:,2))

4. Load data, plot and compare to theory. Here is the script to read in a data file and compare it to a cosine curve:

```
clear;
clf;
%Import Data
a = csvread('testdata.csv');
x = a(:,1);
y = a(:,2);
%Theoretical curve
b = 0:0.01:180;
c = cos(2*pi/180*b);

plot(x,y,'ro')% ro plots red o for each data point

hold on; % this saves the graph for multiple plots, you could also plot both at the same time using plot(x,y1,'-ro',x,y2,'-.b')

plot(b,c, '-.b') % -.b is a dash dotted blue line

legend('Experiment', 'Theory')%this provides a graph legend
```