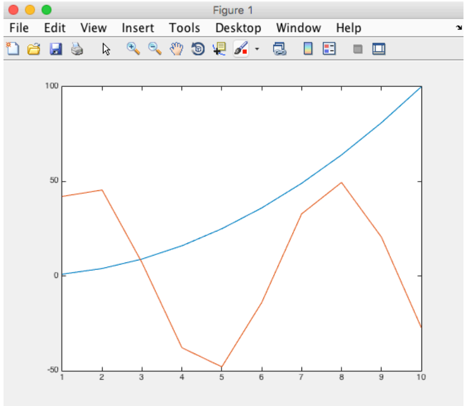
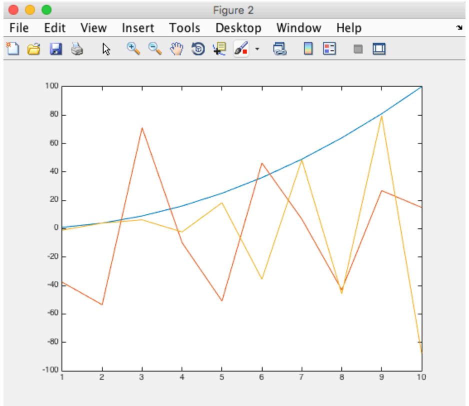
Instant MATLAB tutorial. Jason D. Kahn, Dept. of Chem. & Biochem., U Maryland College Park 4/2016

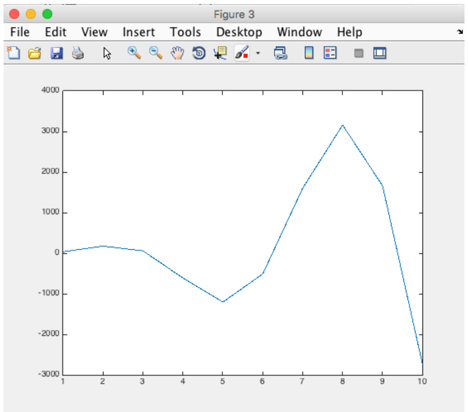
This brief tutorial is intended to allow you to use Matlab programs provided to you and to collect and graph some of the results. There is a vast literature on Matlab. This just hits a few very common issues. Retype or copy and paste the commands after the >> (line by line, not including the >>) into the Matlab command line and see what happens in the command line and the figure windows. The % sign indicates a comment, no need to retype the % sign or text after it on the same line. There are a lot of other tutorials out there, for example http://www.mathworks.com/help/matlab/getting-started-with-matlab.html, but most of them focus on creating programs rather than just using them.

```
>> 2+2
ans =
     4
>> z = 2+5  % assigns the result to a variable z that can be re-used
z =
     7
>> z
z =
     7
>> q = z^2;   % The semicolon suppresses output. q is calculated but not displayed
>> q
q =
    49
>> x = 1:10   % Defines a vector x – Matlab is built to handle arrays of all dimensions easily
x =
     1     2     3     4     5     6     7     8     9    10
>> y1 = x.^2   % The . in .^ means element by element squaring, not trying to dot vectors or square a matrix
y1 =
     1     4     9    16    25    36    49    64    81   100
>> y2 = 50*sin(x);
>> y3 = 100*exp(-0.1*x)*cos(2*x)   % Throws an error b/c it’s trying to doing matrix mult rather than elt by elt
Error using  *
Inner matrix dimensions must agree.

>> y3 = 100*exp(-0.1*x).*cos(2*x)
y3 =
  -37.6545  -53.5158   71.1312   -9.7532  -50.8923   46.3117    6.7902  -43.0304   26.8465   15.0125
>> plot(x,y1)  % Opens up a new figure window and plots y1 vs. x
>> hold on     % Tells Matlab to plot new plots into the same figure window rather than making a new one
>> plot(x,y2)
>> f2 = figure  % The f2 = figure command returns a handle f2 that allows you to access the figure again.
f2 =
  Figure (2) with properties:
      Number: 2
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [680 678 560 420]
       Units: 'pixels'
  Show all properties
>> plot(x,y1)  % The plot goes into the new plot window
>> hold on
>> plot(x,y3)
>> figure   % Makes a new figure
>> plot(x,y1.*y2)
>> figure(f2)  % Get back to figure 2
>> hold on
>> plot(x,y1.*sin(y2))
```

The results should look like the screen shots below.

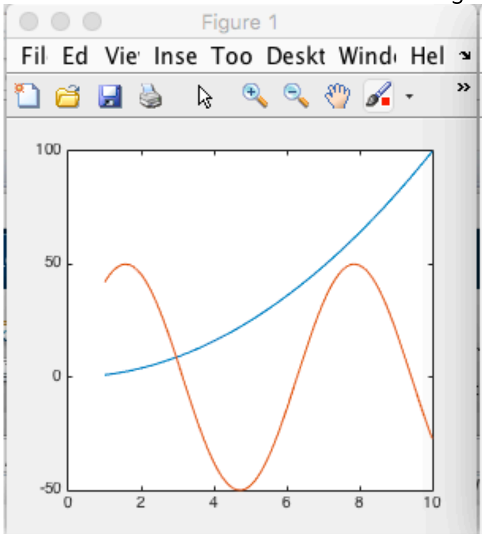


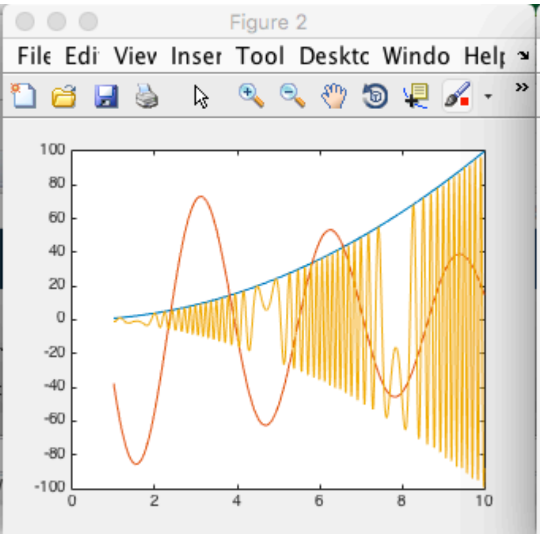


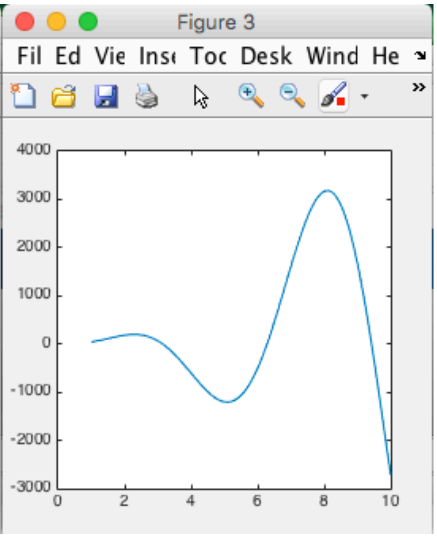


If you are curious, cut and paste the block of lines below into the command window to generate higher-quality figures that show the same functions -- see what we miss by sampling too infrequently:

```
% Higher res plots. Cut and paste the whole block.
close all
x = 1:.01:10;  % creates a vector with spacing of 0.01, i.e. 1, 1.01, 1.02...9.99 10.00
y1 = x.^2;
y2 = 50*sin(x);
y3 = 100*exp(-0.1*x).*cos(2*x);
plot(x,y1)
hold on
plot(x,y2)
f2 = figure
plot(x,y1)
hold on
plot(x,y3)
figure
plot(x,y1.*y2)
figure(f2)
hold on
plot(x,y1.*sin(y2))
%% Down to here. Now look at the figures
```

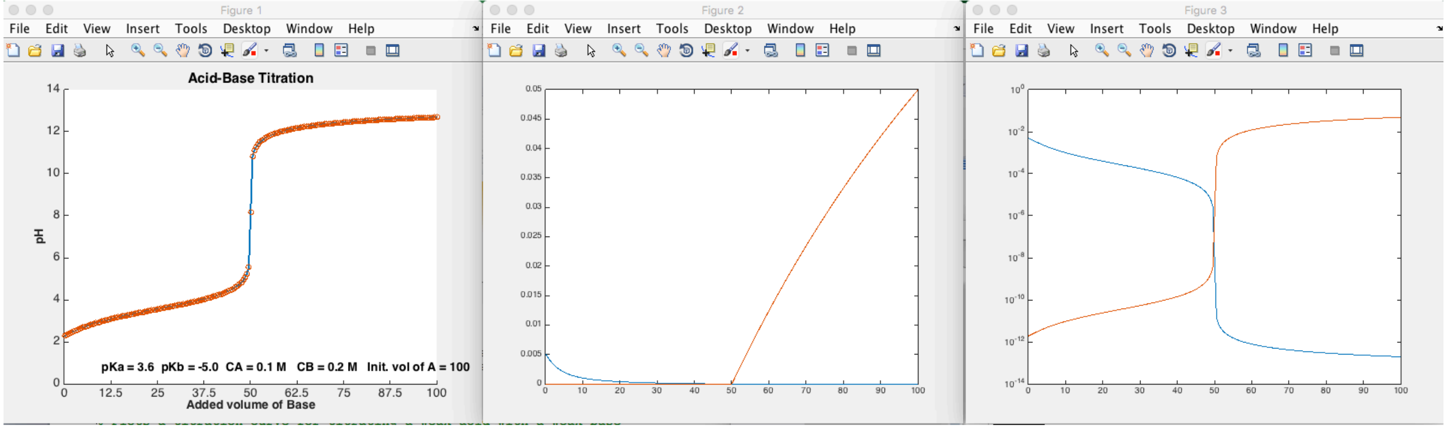




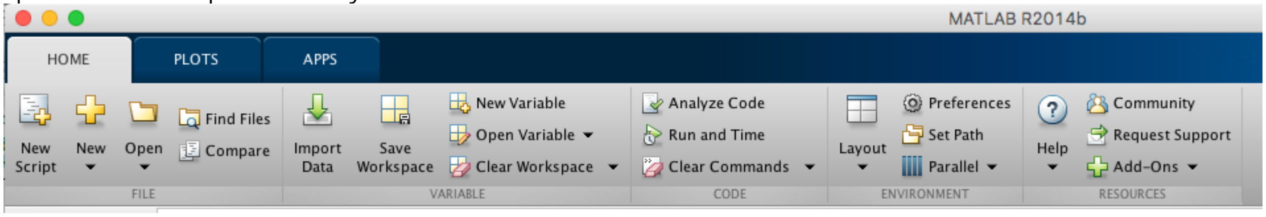


Many Matlab functions generate output that can be captured and re-used. For example, here I use the “weakweak3” program to generate a titration curve. The function makes plots by itself, but if you want to combine outputs, rescale, etc. you can capture the output and work with it yourself. The beginning of the weakweak3 function tells you what it does:

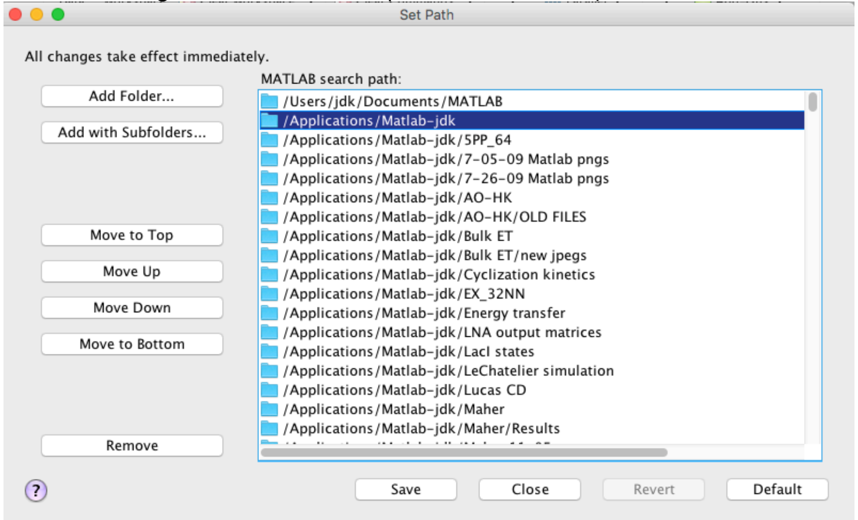
```
function titr = weakweak3(CA,CB,volA,volB,pKA,pKB)
% Plots a titration curve for titrating a weak acid with a weak base
% Jason Kahn U Maryland 2/25/2016  version 1.01
% [titr = ]weakweak3(CA,CB,volA,volB,pKA,pKB);  for example:
% titr = weakweak3(.1,.05,100,225,4.75,3.5); % or just
% weakweak3(.75,1.0,300,250,6.15,-5); % for titration with a strong base
% Titration of a weak acid A with concentration CA and volume volA with a
% weak base B of concentration (CB), from a volume of zero up to volB.
% KAA = 10^(-pKA) is the acid dissociation constant of A and KBB = 10^(-pKB)
% is the base dissociation constant of B. Volume units do not matter (as
% long as they are the same!), typically they would be ml.
% Returns a matrix with columns volB, pH, [A], and [B] in the "titr" variable.

>> close all    % Closes figure windows. Make sure you don’t use “delete” instead as it goes after files!
>> titr1 = weakweak3(0.100,0.2,100, 100,3.56,-5);  % Captures output as the matrix titr1
>> figure
%The colon : indicates all of the rows or columns, so titr1(:,1) means the entire first column of the matrix.
%The plot function plots sets of x,y vectors to give multiple lines:
>> plot(titr1(:,1),10.^(-titr1(:,2)),titr1(:,1),10^-14./10.^(-titr1(:,2)))  % Plots [H+] and [HO-] instead of pH
>> figure
>> semilogy(titr1(:,1),10.^(-titr1(:,2)),titr1(:,1),10^-14./10.^(-titr1(:,2)))  % Semilog plot brings back the
appearance of the pH plot but the y axis is concentrations
```



Finally, a word about the Matlab Path. The Path is a list of places that Matlab looks for programs. If a program fails to run or complete, frequently it is because the function or some sub-function that it calls is not in the Path. When you set up Matlab for the first time, you should create a folder/directory on your disk somewhere and add it to the Path. Then whenever you download Matlab programs, put them into that directory. You may need to update the Path periodically to include the contents of subfolders. Click on Set Path on the Home tab.



In the dialog box that comes up as shown below, choose “Add with Subfolders” to add your folder, Save, Close



All changes take effect immediately.

MATLAB search path:

```
/Users/jdk/Documents/MATLAB
/Applications/Matlab-jdk
/Applications/Matlab-jdk/5PP_64
/Applications/Matlab-jdk/7-05-09 Matlab pngs
/Applications/Matlab-jdk/7-26-09 Matlab pngs
/Applications/Matlab-jdk/AO-HK
/Applications/Matlab-jdk/AO-HK/OLD FILES
/Applications/Matlab-jdk/Bulk ET
/Applications/Matlab-jdk/Bulk ET/new jpegs
/Applications/Matlab-jdk/Cyclization kinetics
/Applications/Matlab-jdk/EX_32NN
/Applications/Matlab-jdk/Energy transfer
/Applications/Matlab-jdk/LNA output matrices
/Applications/Matlab-jdk/LacI states
/Applications/Matlab-jdk/LeChatelier simulation
/Applications/Matlab-jdk/Lucas CD
/Applications/Matlab-jdk/Maher
/Applications/Matlab-jdk/Maher/Results
```

When you get back to the command line, you may need to have Matlab look at its path again:

```
>> rehash
```

When in doubt about whether a function is available or which version is being used, use the “which” command:

```
>> which weakweak3
/Applications/Matlab-jdk/weakweak3.m
>> which weakweak4
'weakweak4' not found.
```