Objectives: To learn how to use arrays in MATLAB for plotting, in particular:

- How to enter and work with arrays
- How to make and label graphs

Outcome:

- Write a script that graphs some curves and labels the graph.

1. Back to MATLAB

1. In Windows, go to the Start menu and find MATLAB. Select it and wait for MATLAB to start.
2. Arrange the windows so that you can both see this document and the MATLAB window.
3. Remember you’ll have to save any files you create in My Documents and tell MATLAB to look in My Documents to find your files.
4. To get help on any command in MATLAB, like help with the plot command, type help plot. You can also use the resources on the help menu or from the Class Resource page on Online@UT.

2. Arrays and Array Arithmetic

1. MATLAB works as easily with arrays (or lists of numbers or vectors) as it does with single numbers (called scalars). To enter an array you can:
   - Enter the numbers individually, starting and ending with square brackets [ and ]. For example v = [-1 0 2 3 6 9 10]. You can use spaces or commas to separate the numbers. If it takes more than one line, put three dots (...) at the end of each line that is continued on the next.
   - For evenly spaced sequences of numbers you can use colon notation. By specifying the beginning, end and (optionally) the increment you can have MATLAB make your array. For example [1:10] produces an array with the numbers from 1 to 10. [7:9] produces 7, 8, 9. If you put a third number in the middle, you can change the spacing from 1, [1:2:10] produces 1, 3, 5, 7, 9. You can also use fractional and negative increments. Try [0:0.2:1] and [5:-1:2].
   - For evenly spaced sequences you can also use the built-in function linspace (try help linspace to find out how to use it).

2. Arithmetic for arrays comes in two forms:
   - For arithmetic with scalars, use regular notation:
     - \[3\times[1:4] - 2\text{ produces } 1, 4, 7, 10\]
     - \[[1:4]/4\text{ produces } 0.25, 0.50, 0.75, 1.0\]
     - \[t = -1 + 3\times[0:400]/400\text{ produces } 401 \text{ evenly spaced numbers from } -1 \text{ to } 2 \text{ (linspace can also do this)}\]
   - For two arrays of the same length (same number of values), use regular notation for addition and subtraction + and −, but the ‘dot’-form for multiplication, division and exponentiation .*, ./ and .\^.
     - Each of these performs the action term by term:
       - \[x = [1 \ 2 \ 3] \text{ and } y = [4 \ 5 \ 6]\]
       - \[x + y \text{ and } y - x\text{ produce } 5, 7, 9 \text{ and } 3, 3, 3\]
       - \[x.*y \text{ produces } 4, 10, 18\]
       - \[x./y \text{ produces } 0.25, 0.40, 0.50\]
       - \[y.^2 \text{ produces } 16, 25, 36\]
       - If you forget the ‘dot’ MATLAB will give you an error message about matrix dimensions not agreeing. We’ll find out later why it does this.
   - Most of the functions work exactly the same on arrays as they do on scalars. For example, if \[x = \pi*[0:8]/4\] (0 to 2\(\pi\) by \(\pi/2\)) then sin(x) produces sin of each value.

3. Plotting

1. To produce a plot in MATLAB you need two arrays, one for the \(x\)-values and one for the \(y\)-values. These can come from a calculation or you can enter them by hand. For example, if \[x = [1 \ 2 \ 3 \ 4] \text{ and } y = [2 \ 3 \ 2 \ 5] \text{ then you can plot these by } \text{plot}(x,y)\]. This produces a simple plot in a new window with the points connected by a solid line.
2. To plot a function and make it look smooth, we need lots of points. 400 seems to be a good number, you can adjust as you see fit. If we want to plot the graph of the function \(f(x) = x^2 - 3x + 2\) for \(x\) between -2 and 4, we set \(x = -2 + 6*[0:400]/400\), then compute \(y = x.*2 - 3*x + 2\) (Note the use of the dot-form). Then \text{plot}(x,y) plots it.
3. Once we have a graph, we can add some annotations: \text{xlabel('x label')} puts x label on the x-axis, \text{ylabel('y label')} puts y label on the y-axis and \text{title('Title')} puts Title as the title of the graph.
1. We are going to graph two curves on the same set of axes. The commands that create the graph will be in a script `proj3.m`. Make sure the file has your name and is well commented. The curves:

- The first curve will be the graph of \( y = f(x) \) for \( x \) in the range from -4 to 4, where \( f \) is the cubic function,
  \[
  f(x) = \frac{1}{4}x^3 - \frac{1}{24}x^2 - \frac{61}{24}x + \frac{28}{12}
  \]

- The second curve is a parametric curve
  \[
  x(t) = 2t \cos(\pi t), \quad y(t) = 4 \sin(\pi t), \quad \text{for } t \text{ between 0 and 2}.
  \]

For this curve, you create an array of \( t \) values and then use the formulas to compute the corresponding \( x \) and \( y \) values.

2. In your script, create the two curves, and plot them on the same graph. Label the graph and add a legend. Include your name in the title of the graph. Style counts so be creative with your colors and line-styles. When your graph is done, choose Export under the File menu of your graph, and export it as a jpeg-file with the name ‘proj3.jpg’.

3. For this project, submit the two files: ‘proj3.m’ and ‘proj3.jpg’.

4. As usual, please fill out the survey when you are done with the lab.