

## MATH.2360 Engineering Differential Equations Simple MATLAB Graphics and Some Useful Functions

Please note that in MATLAB, everything that follows the % symbol is a comment. You do not have to type % or the comments in the sample commands below.

### A. Graphing

The basic MATLAB graphing command is the *plot* command. You follow a 3-step procedure: 1) generate some  $x$  values; 2) calculate the corresponding  $y$  values; and 3) plot the points. Here is an example:

```
x = linspace(0, 2*pi); %This creates an array of numbers evenly spaced between 0 and 2π.  
y = sin(x); %This creates a vector containing the values of the sine function at each of the entries  
           %in the x vector  
plot(x, y) %This plots the (x, y) points connected with a straight line
```

To plot more than one graph on the same set of axes, you can try something like this:

```
z = cos(x);  
plot(x, y, '- ', x, z, '--') %This plots the (x, y) points connected with a solid line ('- ') and  
                           %the (x, z) points connected with a dashed line ('- -')
```

You can add a title and axis labels and you can label your curves using the following commands. **DO NOT** close the figure window containing the plot.

```
title('Graphs of sine and cosine functions') %Creates a title at the top of the graph.  
xlabel('x') %Puts a label under the horizontal axis  
ylabel('y') %Puts a label next to the vertical axis  
legend('y = sin(x)', 'y = cos(x)') %Creates a legend indicating which graph is which
```

If you don't like where MATLAB places the legend box, you can use the mouse to drag the box wherever you want it.

**In addition to solid lines and dashed lines**, you can generate dashdot graphs ('.-') or dotted graphs (':'). **You can control the color of the graph** by adding a letter after the line type (blue, green, red, cyan, magenta, yellow, or black). For example, to plot the sine graph in red using a dashdot line and the cosine graph in black using a dotted line, type

```
plot(x, y, '-r', x, z, ':k')
```

MATLAB has a trace feature. If you click on the Data Cursor toolbar icon in the figure window (the icon that has a + sign in the lower left corner) and then click on a graph, you will see the coordinates of the point you clicked on. You can use the arrow keys to move the data cursor to other points on the graph.

**OVER**

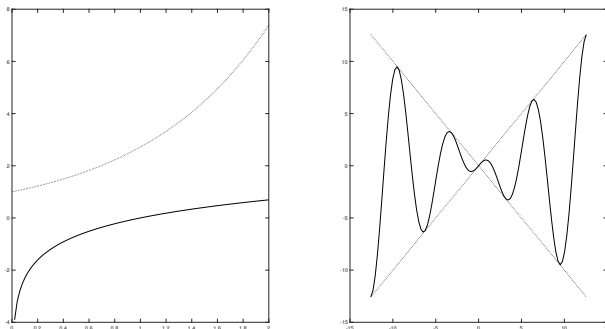
## B. Notation

MATLAB notation for operations and functions is pretty much what you would expect, with a few notable exceptions:

1. To multiply corresponding elements in two vectors, use `.*`, NOT just `*` For example, to compute the product of the two vectors `x` and `y` you generated earlier, you should type `x.*y`
2. Similarly, division is accomplished using `./` To square a number or each element of a vector, use `.^` For example, `x.^2` gives a vector containing the squares of the elements in vector `x`.
3. The exponential function is denoted `exp`. For example, to compute  $e^{1.5}$  you would type `exp(1.5)`
4. The natural log function is denoted `log`, NOT `ln`.
5. The square root function is denoted `sqrt`
6. Putting a semicolon at the end of a MATLAB statement tells MATLAB not to show the output after that statement is executed. This can be useful if you are working with a long array of numbers.

**C. Exercises** (These are just for practice. You do not have to hand them in. Your graphs should look like the graphs in the figure below.)

1. Graph  $y = e^x$  and  $y = \ln(x)$  for  $0 \leq x \leq 2$  on the same set of axes.
2. Graph  $y = x$ ,  $y = -x$ , and  $y = x \cos(x)$  for  $-4\pi \leq x \leq 4\pi$  on the same set of axes.



## D. Closing MATLAB

To close the MATLAB program, type `exit` or `quit` in the command window.