

Matlab Basics: Day 2  
*EDGE 2019*

1. **Plotting a function of one variable, (zooming in, hold on, line styles, adding a title and labeling axes).** Suppose you want to plot the function  $f(x) = 2x(1 - x)$ . The easiest way is to use ‘ezplot’:

```
ezplot('2*x*(1-x)')
```

Note the single quotes around the function, which tells Matlab that you are typing in a “string”, or a sequence of symbols, that it will then interpret. Also note the asterisks wherever there is multiplication. Matlab insists on these.

You should get a graph of the function from  $x = -6$  to  $x = 6$ , which is Matlab’s default range. To change the range, add a vector to the command:

```
ezplot('2*x*(1-x)', [0 1])
```

now plots the function between 0 and 1.

To add another graph to the same plot, say the diagonal  $y = x$ , first tell Matlab to “hold on”

```
hold on  
ezplot('x', [0 1])
```

You can find (approximately) where the two graphs intersects with the “trace” feature. Click on the icon in the menu-bar that has a plus sign on top of a piece of graph, with a bit of lined paper next to it. This is called the “Data Cursor”. Then click on the graph, and move the “Data Tip” up or down with the arrows on your keyboard. To get rid of the Data Tip, right click on it (or Control click) and then select “Delete Data Tip”.

Clear the figure for the next example:

```
clf
```

The less easy, but more flexible, way to plot a function is to first create a vector of x-values, then a vector of  $f(x)$  values, and then plot one against the other. For example:

```
x = 0:.001:1;
```

creates a vector called “ $x$ ” with the numbers between 0 and 1, at intervals of .001. Then create the values of  $f(x)$  and plot one against the other. Notice that we needed to use “component-by-component” multiplication to multiply the vectors  $x$  and  $1 - x$ :

```
f=3.2*x.*(1-x);
plot(x,f)
```

Add the diagonal in red:

```
hold on
plot(x,x,'r')
```

You can use the figure editing tools by selecting the arrow in the figure menu bar, and then selecting objects in the graph to edit. (Try changing one of the lines to a dashed line, for example). You can also insert labels and text using the “Insert” pull-down menu in the figure window. You can also plot only dots, or other symbols:

```
plot(.5, .5, 'm+')
```

adds a magenta plus sign at the point (.5, .5). You can change these using the plot edit tools as well. There is also a zoom feature in the menu bar (but there’s nothing much to zoom on in this case!)

2. **Plotting a function of two variables.** Matlab can plot in 3-d as well. Suppose we want to plot a function of two variables:  $f(x, y) = x^2 + y^2$ . Create vectors for  $x$  and  $y$  as before

```
x=-1:.01:1;
y=-2:.01:2;
```

Now we have to create GRIDS so that each  $x$ -value is paired with every possible  $y$ -value. There’s a special command for this called “meshgrid”:

```
[X,Y] = meshgrid(x,y);
```

$X$  and  $Y$  are now gigantic matrices (check them out if you like by typing, e.g. “ $X$ ” followed by “enter”). Now we calculate the function at each of the  $(x, y)$ -pairs and plot. First we clear the figure again:

```
clf
f = X.^2 + Y.^2;
surf(x,y,f)
```

Note that we need to use the capital letters (the matrices) in the first case, but the lower case (the vectors) for the axes of the plot. “surf” stands for surface, which is what you get. You can rotate the graph around using the rotate icon in the figure menu-bar (it looks like a near-circular arc with an arrow on it surrounding a little box). Check it out!

3. **Writing a function as an M-file and using it.** We often want to store routines that calculate things so that we can use them over again without retyping. These are stored in files called ‘M-files’ and they always end with the ‘.m’ extension. There are some of these in your folder, and the easiest way to figure these out is to look at one of them. Open the file called ‘logistic.m’. It should open in the “Editor” window and look like this:

```
function y=logistic(x)

global a

y=a*x.*(1-x);
```

The word “function” is in blue because it is special: it tells Matlab that the file is a function that takes an argument (its input, in this case a number or vector called ‘x’) and it produces output, in this case the variable y, which is also a vector. The name of the function is ‘logistic’, but Matlab really only cares what you name the file when you STORE it, not what you name the function here on the first line (to avoid confusion, these should probably be the same thing). The second line says that ‘a’ is a global variable, meaning that this function will use the value of a that you declared OUTSIDE of the function (in the command window). The next line tells Matlab how to calculate y. Notice that we used the ‘.\*’ for component-wise multiplication so that we can calculate the results for a vector of  $x$ -values. Go back to the command window and type

```
global a
a = 1;
logistic(.5)

ans =

    0.2500
```

These commands tell Matlab to declare  $a$  to be a global variable (so that it is accessible to the function ‘logistic’), to set the variable  $a$  equal

to 1, then to run the routine 'logistic' with  $x = .5$ . The answer should be:  $y = 1 \cdot x(1 - x) = .5(1 - .5) = .25$  which is what we got.

Now we will modify the M-file a bit by removing the global variable. Instead, we will pass the parameter  $a$  to the function as another input, or "argument". Go back to the Editor and change the file to look like this:

```
function y=logistic2(x,a)

y=a*x.*(1-x);
```

and save this new M-file as 'logist2.m' (in the same folder). Back in the command window try changing the values of  $a$ , which is now the second argument:

```
logistic2(.2,2)
```

```
ans =
```

```
0.3200
```

```
logistic2(.2,4)
```

```
ans =
```

```
0.6400
```

Now try your hand at creating a new function called 'cubic' which will return the function:  $f(x) = ax^3 + (1 - a)x$ . It should take two arguments,  $x$  and  $a$ . Plot the function in the range  $x = -1$  to  $x = 1$  when  $a = 3.5$ . Superimpose the diagonal (in red) and the axes (in black).

*Here is one way to do this:* The M-file should look like:

```
function y=cubic(x,a)

y=a*x.^3 + (1-a)*x;
```

and then in the Command Window type (don't forget to clear the figure first)

```
x=-1:.01:1;
y=cubic(x,3.5);
plot(x,y)
```

```
hold on
plot(x,x,'r')
plot([-1 1],[0 0],'k',[0 0],[-1 1],'k')
title('The cubic function  $y = 3.5 x(1-x)$ ','fontsize',20)
```

Note that you can plot several lines with one command, that 'k' stands for black, and that you can also add a title from the command window.