

Matlab Commands Math 2250 Fall 2011

Created by Victor Camacho

This file discusses several commands and functions in Matlab that will be very useful in Math 2250/2280 courses. If you have any additional suggestions for what could be added to this file feel free to let me know.

If you are not already familiar with Matlab, I would suggest first reading the Introduction to Matlab¹ file to familiarize yourself with the basic properties. The instructions in this file assume you are already familiar with how to login to your account, open Matlab, enter basic commands into Matlab, create an m-file, save an m-file, and execute an m-file.

I would highly suggest trying out all of the commands in this document to familiarize yourself with them. Many of these commands will be helpful for doing the assigned projects as well as checking your work on homework and other problems.

Help:

Any time you need help in Matlab you can go to the "Help" menu and select "Product Help". This will open a new window with, among other things, a search box where you can enter the subject you need help with, or the specific command.

If you already know the name of the function you wish to use, such as the "sqrt" function, you would type the following text onto the command line:

```
>> help sqrt
```

The output of this command will be essentially everything you need to know about the square root function, it will also provide links to related functions and documentation.

Comments:

In many circumstances you may find it useful to insert comments into your Matlab code. In an m-file you may have many lines of code that do different things. It will be useful to remind yourself what each parts of the code does. Matlab will automatically ignore all of the text you have entered on a line after a percentage symbol (%). Everything entered on the line after this symbol will be colored green, to indicate that it is a comment and will not be read by Matlab. Consider the following commands that have been entered into an m-file:

¹ http://www.math.utah.edu/~camacho/teaching/fall2011/files/Matlab_Intro.pdf

```

-----
% In this file I calculate the hypotenuse of a triangle.
% 1. Define the length of the two legs of the triangle
a = 3;
b = 4;
% 2. Use the Pythagorean theorem to calculate the length
% of the hypotenuse.
c = sqrt(a^2 + b^2);
% 3. Output the final answer.
c
-----

```

Notice how the use of comments helps you understand exactly what the program is doing at every step. Even if you're not entirely familiar with all of these commands yet. Every line that starts with a % symbol is interpreted as a comment and colored green. Also notice the use of white space here. Matlab does not care how many lines separate each command or comment, so you can separate your code into distinctive chunks if you need to in order to help your understanding. Finally, notice that all but the last command ends in a semi-colon. If you want the result of a specific command to be displayed, you can remove the semi-colon, otherwise Matlab will suppress the output. This is generally a good idea for m-files since there will be many commands but you will typically only care about the end result.

Variables and Arithmetic

In Matlab it is possible to assign a value to a given variable. Suppose you wanted to set the letter a to the value of 2. You would simply enter

```
>> a = 2
```

Now, anytime you use the letter a (by itself) it will be interpreted as the number 2. For example, Matlab will correctly compute all of the following arithmetic operations with a:

```

>> a + 1      % This outputs a plus 1, "ans=3"
>> a^3        % This outputs a cubed, "ans=8"
>> sqrt(a)    % This outputs square root of a, "ans=1.4142"

```

```
>> abs(a)      % This outputs the absolute value of a, "ans=2"
```

You can subsequently change the value of `a` by reassigning it to a different value. After typing the following command, `a` will now have a value of 3 anytime it is used (until it is reassigned):

```
>> a = 3
```

You can also reassign the value of `a` using an expression in terms of `a`. Consider the following commands:

```
>> a = a + 1    % This sets a+1 as the new value of a. So if
                % previously a had a value of 3, it now has a
                % value of 4.
```

```
>> a = 2*a     % This will double the current value of a and set
                % that to the new value of a.
```

```
>> a = sqrt(a) % This will take the square root of the current
                % value of a and set that to the new value of a.
```

Matlab is fairly flexible with the names of variables. You are basically allowed to use any string of letters (both upper and lower case), numbers, and underscores. The only restriction to this is that the string **MUST** start with either an upper or lower case letter. All of the following would be valid variable names in Matlab: `a`, `a1`, `a_12`, `applepie`, `apple_pie`, `axq4350dkdfjks`, `q1123443342`, `a__c4`. Matlab has a few of its own internally defined variables and functions that you will not be able to assign in your code. These include things like `pi`, `i`, `sin`, `cos`, `log`, `exp`, `abs`, `sqrt`, and so on.

Matlab Functions

Matlab has many internally defined functions for you to use when doing various computations. The functions may be prepared to receive one or more arguments. The general format for calling a function is as follows:

```
>> function_name(argument1)
>> function_name(argument1, argument2)
>> function_name(argument1, argument2, argument3)
```

You first write the function name, then open parenthesis, then the list or arguments that the function will receive, where each argument is separated by a comma, and then close the parenthesis. Let's look at a few examples:

```
>> sin(0)      % The sin function only has one argument. This
                % command will compute the sine of 0.
```