An Introduction to Matlab: Part 1

The general idea in this lecture is that we start walking the user through some basic Matlab commands. This introduction starts very easily, explicitly giving the user Matlab commands to try, and progresses to the point where the user should know the commands to use without being given them. This first lecture covers

- Getting and running Matlab
- Basic Matlab commands
- Doing calculator type work with Matlab

How to get and run Matlab

Matlab is installed on most lab machines on campus, and it is definitely installed at the ILC. Since I'm not sure of the configuration on any other computer, you'll want to ask the lab staff how to load Matlab (there should be just a link to Matlab someone in the program files).

Matlab is free for download if you are enrolled at the University of Arizona. You can download the license for Matlab ONLY if you are physically at the university or if you are logged in using VPN. Once you have the license number, you should be able to download Matlab from mathworks anywhere outside of campus (please let me know if this is not the case)

- Follow the instructions here: https://sitelicense.arizona.edu/matlab/matlab_instructions.php
- Note that you absolutely HAVE TO use your email.arizona.edu address when you register with Mathworks.
- Once installed, you should have a nice Matlab icon on your desktop. Double click to load Matlab.

Using Matlab for the first time

Odds are, when you opened Matlab, you have a whole bunch of windows that appeared. Most likely you have a *Current Directory* window, a *Command History* window, and a *Command Window*. You may have more, you may have less. The really important window here in the *Command Window*. This is where all of your Matlab commands will go for now.

1. *Matlab help/doc commands:* Here we learn how to use *help* and *doc*.

   The two most important commands to remember in Matlab are *help* and *doc*. If you know what function you want to use, but you do not know how to use it, then you can type *help FunctionName*. If you want even more information about this function, you can type *doc FunctionName* to bring up a new window (called the *Help* window). Just typing *doc* at anytime will bring up the *Help* window. Once in the *Help* window, you can use the *Search* feature on the upper left to aid you in finding whatever you may need.

   (a) Type *help sin* and press RETURN.
   (b) Type *doc sin* and press RETURN.
   (c) In the *Help* window, search for *natural log* and see what happens. What is the command in Matlab for the natural log?
   (d) Close the *Help* window and go back to the *Command Window*.

2. *Assigning/Unassigning Variables:* Here we learn how to use *=, ; , who,* and *clear*

   (a) To assign a variable in Matlab, we use the = . Assign a to have the value 3 by typing the following
   $a = 3$
   Matlab should provide you confirmation that a is 3 by printing the result out. If you do not want Matlab to re-print your result, end your command with a semicolon ; Try:
   $b = pi;$
   You can double check that b is in fact the approximation of π by simply typing b, without a semicolon (since it will re-print the result)
   b

(b) You have now defined the variables a and b. Sometimes, you would like to know exactly what variables you have already defined. You can do this by using the *who* command. Type

who

on the command line now. Matlab should tell you that your variables are a and b.

(c) Let us say that we do not want a to be 3 anymore. We can assign it another value (by typing, for example, *a=4*) or we can simply unassign it so it has no value. To do this, type

clear a

at the command line. Now type *who* again. You should only have one variable, b, now. To clear EVERYTHING that you've assigned, you use *clear all.*

3. *Other useful commands:* Here we learn how to use the *format* command, the *clc* command, and the *up/down* keys.

(a) Matlab may display your answer in too few or too many digits. To control this, you can use the *format* command. Try *help format* for more information. Type

format long

then ask Matlab to display the value of *pi* by typing

pi

Next, type *format short* and do the same. Unless we change something, all results will now be shown in *format short* form now that you have typed *format short.*

(b) One more useful command is *clc.* Running this command clears the *Command Window.* It does not erase any of your variables, but instead just cleans up you window.

(c) Lastly, anyone familiar with Unix will know that the *up* and *down* keys allow you to cycle through your previous commands. Try pressing *up* a few times, then press *down* a few times. You'll notice your previous commands will be displayed. This is extremely useful when you have made a mistake.

4. *Calculator Work:* Here we learn to use basic calculator commands, such as *+* , *-* , */* , *** , *^* , sin*, e,* etc.

(a) At this point, you can use Matlab just like a calculator. Try adding, subtracting, multiplying and dividing some numbers using the usual commands.

(b) Matlab can also use complex numbers. Try setting $x = 1 + i$. Find $x * x$. This should be what you would expect from complex arithmetic.

(c) To raise a scalar to a power in Matlab, we use the usual carrot key *^* . So, in the last part, we could have calculated *x^2* instead of *x*x.*

(d) The carrot key will work with fractional powers, but there is a *sqrt* function also. Note that this is a FUNCTION. Thus you need to pass in an argument. So to find $\sqrt{2}$ using the *sqrt* function, we type

sqrt(2)

(e) The same goes for the exponential function. To find e^2, we use the function *exp. exp* does NOT have any value, it is a function. So

exp(2)

gives us the value of e^2, while

exp^2

gives us an error, since exp is NOT a number.

(f) Lastly, we use Matlab to compute the value of inverse trig functions. Note they are not defined as *arcsin* or *arccos*, but rather as *asin, acos,* and so on. Try to compute arctan(1) using Matlab.