

AMATH 410 MATLAB TECHNIQUES

This is a PARTIAL list of techniques that you'll want to be familiar with. If you're not, please experiment in MATLAB, use the help commands, and ASK the course instructors in office hours or class lab sessions. We're here to help, and all of us have questions about these things.

In addition, please make sure that you're familiar with the MATLAB functions and techniques that are listed in the sections of the Lab Manual that we have covered so far in class, and the commands used in class (please carefully look at all codes from class, which are posted online with class notes).

- Making `.m` files using the text editor. To create a new m-file, click **File--> New--> m-file** or click the bottom right underneath "File" that looks like a blank sheet of paper. Say you make a code called "myprogram.m" – you save it in a directory, make sure that this is your working directory (see below), and type `myprogram` at the command line to run this code.
- A good habit is to write "clear all; close all;" on top of your code, so that every time before it runs MATLAB clears up the workspace and existing plot windows.
- Changing the working directory in MATLAB. At the top of the matlab command window, it'll say what directory you are currently working in. Of course, this needs to be the same one where you are saving your `.m` program files and any data files to. To change the working directory, either click on the "bar" where this is displayed, or use the `cd` command at the command line.
- Use `help functionname`, where `functionname` is anything you don't know how to use yet (say, `help hist`), whenever you want more info. `doc functionname` often gives even more info, with examples – cutting and pasting these to the command line is a great way to get started. If you have no idea about the function name, say you are looking for a function that does something, you can search in the MATLAB help document using key words.
- `whos` command – type this at command line to see list of all variables and their sizes
- Or, to see values of a variable, either go to "workspace window," as described in Lab Manual, or simply type its name at the command line. e.g., type `A` then hit the enter key to see value of `A`
- Editing `.m` program files: just type `edit` and an editor will pop up; to edit the code `myfilename.m`, type `edit myfilename`
- Loading data using `load`
- Commands `size` and `length`
- Command `linspace`
- Using the colon to extract parts of vectors or matrices
- Using the semi-colon to hide the print-outs you are not interested in viewing.
- Using `for` loops.
- Using the `disp` and `num2str` commands to output data to screen in organized way – see p.15 of Lab Manual
- Using and writing MATLAB functions

- Use of anonymous functions and function handles
- Use of the `hold on` command for plotting multiple curves (or points, etc) in a single plot.
 - Syntax: `figure ; plot(xlist1,ylist1) ; hold on ; plot(xlist2, ylist2)`
 - Please compare this with what happens when you leave off the `hold on`
 - When you no longer needs the "hold on" property, type "hold off"
- Get familiar with the rich options MATLAB has to offer in plotting by checking out the help document of "plot". Simply type "help plot" in the command window and see what MATLAB likes to tell you. This is also helpful when you forget which letter stands for which color or what kind of lines in plotting.
- Use of the percent sign, `%`, to start a comment (notation that is ignored by MATLAB)
- Matrix multiplication via `*` and `^`
- Use of the 'vectorized' operations `.+`, `.*`, `.^`, etc., to do ELEMENT-WISE operations on entries of vectors or matrices. Please note that, for example, the `.*` command is therefore very much different from the matrix multiplication operation of linear algebra.
- Using `format` control to specify numbers of signif. digits (etc.)

*** And, in week 3 and later weeks (as this material is covered): ***

- Using eigenvalue / eigenvector commands via `eig`
- Use of `find` command to identify, extract, or modify target entries in a matrix or vector (e.g, sec. 8 of lab manual)
- Using `max` and `abs` – e.g. to find dominant eigenvalues / eigenvectors (sec. 8 of lab manual)
- Linear solve of matrix equations $W*c = n$, where W is matrix, n is *given* vector, and c unknown vector that are solving for. Syntax: `c=A\n`
- Using `rand` and `randn` random number generators
- Using the `mean` and `var` commands
- Plotting and properly scaling a histogram
- Simulating a Markov chain – using logical operators and repeated calls to `rand`