

Lecture 1 for Math 398 Section 952: A Tour of Matlab

Thomas Shores
Department of Math/Stat
University of Nebraska
Fall 2002

What is Matlab?

Matlab is

1. An interactive system for numerical computation.
2. A programmable system for numerical computation.
3. A front-end for state-of-the-art libraries, currently LAPACK and optimized BLAS.

History:

- Late 70s: Cleve Moler writes Matlab under an NSF grant as a front-end for the current state-of-the-art numerical linear algebra packages LINPACK and BLAS.
- Early 80s: Moler founds the MathWorks and introduces a commercial version of Matlab running on a PC and written in Fortran.
- Early 90s: Matlab is rewritten in C and “handle graphics”, a sophisticated graphical package is added into Matlab.
- Late 90s: Matlab evolves into an OOP supporting LAPACK and optimized BLAS.

The standard reference for this course is the superb text (recommended but not required) by Desmond Higman and Nicholas Higman, *Matlab Guide*, SIAM, Philadelphia, 2000. In addition, one can consult the Matlab documentation that will be put on reserve in the Math Library (9th floor Oldfather).

A Tutorial Introduction to Matlab

Here we go...Our procedure in all the lectures will be to work our way through the commands that are listed in the narrative that follows. Separately one should open a Matlab window for command execution. The prompt sign “>” and typewriter text indicate a command to be typed in by the instructor or reader at the Matlab command prompt. Comments will not be preceded by the prompt and will appear in normal text. I will make comments on the commands that are executed

as we read through the lecture, and students are invited to ask questions as we proceed. In the body of the Matlab command listing there will be occasionally a “%” sign. This is Matlab’s way of starting a comment, and all text until the next end of line is ignored by the Matlab interpreter. It isn’t necessary to type in these comments in order to work through this lesson in a Matlab session.

Starting and stopping.

Start up matlab and quit after a few calculations and getting some help.

```
> 1+1
> 1+1;
> pi
> % to get help on a command or built-in variable just type:
> help pi % get help on built-in variable pi
> help quit % get help on built-in command
> helpwin % or get help browser style
> quit
```

A calculator.

Restart and do the following:

```
> x=0.3
> sin(x)
> exp(x)*sin(x)/log(x)
> 2^24
> (1+sqrt(5))/2
> x
> format long
> x
> format
> z = 2 -3*i % complex numbers are no problem either
> z^2
> exp(z)
```

Matrices everywhere.

List and clear all variables and do the following:

```
> who
> clear
> a = [1 3 2; 2 1 1; 4 0 2] % create a matrix
> size(a)
> eye(3)
```

```
> a + eye(3)
> (0:5) % create a vector using the colon operator
> v = (0:5)'
```

Simple plotting.

Matlab's plotting, unlike a CAS like Maple, does not plot a function. Rather, it does a dot-to-dot on vectors of x - and y -coordinates. Most Matlab built-in scalar functions know to handle a vector argument by applying the function to each coordinate.

```
> x = 0: 0.1: 3;
> y = sin(x);
> plot(x,y);
> y = x .* sin(x);
> plot(x,y)
> % close the plot window, then
> plot(x,y)
> hold on
> plot(x, x.*sin(x))
```

Functions and scripts for experiments.

There are two kinds of text files that can be “run” under matlab, both of which have the suffix “.m” The first kind is a script file. This file contains a list of commands that one could execute at the command line. It is rather like a program in C or Fortran, but not as capable. For one thing, there is no division of program into a “main” section and various subroutines. You cannot include function definitions in a script file. Functions, that is, value returning subroutines are what function files, the second kind of executable text file, are good for.

Various files from the text A Matlab Guide are contained in the Home Directory for this course in the subdirectory MatlabGuideFiles. Open the file lorenzde.m to see an example of the definition of a function file. Notice that the prefix name of the file and the function name agree and that the first line of a function file starts with a function definition. On the other hand, a script file carries out a sequence of commands. For a good example of this, open the file collatz.m and inspect it. Matlab has a nice editor for this kind of work. Make sure these files are in the current working directory (do an “ls” or “dir”) and do the following.

```
> ls
> edit lorenzde
> edit collatz
> collatz
> help lorenzde
> lorenzde(2, [3;1])
```