

Notes on MATLAB Assignments

These notes are written to introduce the MATLAB part of this course. For additional MATLAB tutorial materials, go to the course web site math.rutgers.edu/courses/550A and follow the links there.

Demonstration of MATLAB: In this course you will use MATLAB to create and operate on matrices and vectors. The commands needed to do this are short and easy to remember, because MATLAB specializes in matrix computations and uses standard linear algebra notation.

The basic mode of MATLAB is interactive. After you start the MATLAB program and obtain the prompt `>>`, you type commands that MATLAB then executes when you press the **Enter** key. If you are running MATLAB in a Windows environment you can type `demo` at the MATLAB prompt and get a choice of animated demonstrations. Click on **Desktop Environment** and run the playback files (you may want to return to this demo later as you work through the lab and prepare your writeup). In the demos **Matrices** run **Basic matrix operations**. This gives a slide show that demonstrates how matrices are entered and displayed. Also run the demo **Matrix manipulation**.

Suppressing Displays: When you place a semicolon at the end of a command, the command will be executed but the result will not be displayed on the screen. This is very useful when you are creating big matrices. Try typing `z = [3:0.2:5];` to create a long row vector with equally spaced entries (note the semicolon at the end). Then type `z` to display the vector `z`.

If a command statement does not fit on one line, use three periods `...` followed by the **Enter** key to indicate that the statement continues on the next line.

Format Commands: You can control how numbers are displayed on the screen (this does not affect the internal arithmetic). Type

```
y = [4/3 1.2345e-6];
```

Then observe what display you get when you type

```
format short, y
```

at the prompt. Repeat this using `format short e, y` `format long, y` `format long e, y` `format rat, y`

Script Files: For more complicated MATLAB calculations you should use *scripts*. A script contains one or several MATLAB commands and is stored as a text file with a descriptive name such as `mymatrix.m`, for example (the extension `.m` is required). When you type the name of a script (without the extension `.m`) at the MATLAB command prompt the commands within the script are executed, affecting the variables in the global workspace. Such scripts are called *m-files*. The advantage of having scripts is that you can execute the commands in the script at any time by typing the *name* of the script instead of the *contents* of the script.

Writing Scripts: When you need to write a script file, use the following procedure: Start MATLAB and click on *File*, then *New*. Move the pointer to the right and click on *m-Files*. This will open the MATLAB Editor/Debugger Window, and you can type the script commands in this window. You can take any m-file, edit it (just as you would edit any text file), and then save it under a different name to obtain a new m-file.

Running Scripts: After you have created an m-file and saved it to your directory, you must set the *Path* so that MATLAB can find this file. Click on *File*, then *Set Path* and follow the directions to add your directory to the list of path names.

Diary File: You will need to record the results of your MATLAB sessions to generate your lab reports. To do this, create a folder (subdirectory) to save your MATLAB work in. Then use the **Current Directory** field in the desktop toolbar to change the directory to this folder. Now type

```
diary lab1.txt
```

followed by the **Enter** key. Now each computation you make in MATLAB will be saved in your directory in a text file named `lab1.txt`. You can then edit this file using your favorite text editor. When you have finished your MATLAB session you can turn off the recording by typing `diary off` at the MATLAB prompt.

If you want to stop your MATLAB session before completing a lab assignment, you can reopen the diary file the next time you start MATLAB. If you use the same file name, the results of your new MATLAB session will be written at the end of the old diary file. You may prefer to use different names (such as `lab1a.txt`, `lab1b.txt`) for each session on an assignment, and then merge the files when you prepare your lab write-up with a text editor. Of course, for the other labs you will change the filename to `lab2.txt`, and so forth.

Lab Write-up: After opening your diary file, type the comment line

```
% Math 550 MATLAB Lab Assignment #1
```

at the MATLAB prompt. Type `format compact` so that your diary file will not have unnecessary spaces. Put labels to mark the beginning of your work on each part of each question, so that your edited lab write-up has the format

```
% Question 1 (a) ...
:
% Question 1 (b) ...
```

and so on. Be sure to answer all the questions in the lab assignment. Insert comments in your diary file as you work through the assignment.

Saving Workspace: It is a good idea to save the workspace every few minutes, in case your session is accidentally terminated or if you want to close MATLAB and resume work later. To do this, first choose **Set Path** from the **File** menu to set the path to your subdirectory (if you are using the latest version of MATLAB, you can change paths using **Current Directory** window). Then type

```
save lab1
```

This will create a binary file in your folder with the name `lab1.mat` that contains the data in your current workspace. You can restore the workspace by typing

```
load lab1
```

Random Seed: When you start your MATLAB session, initialize the random number generator by typing

```
rand('seed', abcd)
```

where *abcd* are the last four digits of your Student ID number. This will ensure that you generate your own particular random vectors and matrices.

BE SURE TO INCLUDE THIS LINE IN YOUR LAB WRITE-UP

Editing of Lab Write-up: The standard mode for MATLAB is as a command-line program (you can recycle commands using the up-arrow key ↑). This means that you must edit your diary file to remove unnecessary or incorrect lines that were generated during the MATLAB session. Remove all errors and other material that is not directly related to the questions. Your write-up should only contain the commands and answers for the MATLAB calculations and your answers to the questions. Remove unnecessary page breaks and blank spaces. Put your name and student ID number on each page. (If you have difficulty doing this using your text editor, you can write this information by hand after printing the report.)