EXERCISES

Introduction to MATLAB: Interface and Basics

I. Class Materials

1. Download Interface_Basics.tar

From a web browser:

Open your browser and go to http://web.mit.edu/acmath/matlab/IntroMATLAB.
Download the file **Interface_Basics.tar** to a local work directory.

Alternatively, on Athena:

Copy the file from the locker **acmath** to a local work directory.
add acmath
cp /afs/athena.mit.edu/astaff/project/acmath/matlab/IntroMATLAB/Interface_Basics.tar **.**

2. Extract this session's sub-directories and files

(Alternatively, you can download, or copy from the locker, the files one by one.)

On Athena *(or the UNIX shell on Mac OS X)****:***
tar –xvf Interface_Basics.tar

On laptops:
Use your computer's utilities, such as double click or WinZip on Windows and StuffIt on Mac.

Your local work directory should now contain the following directories and files:

Interface_Basics

Exercise_One
example1.m
MA_county_data.txt
State25.gif

Exercise_Two
example2.m

Exercise_Three
example3.m
XYZ_point_coordinates.txt

You may place and rename directories and files any way you wish. For consistency, we shall refer to the directory **Interface_Basics** as the work directory for these exercises.

II. Start MATLAB

On Athena:

Go to the work directory **Interface_Basics** using the cd command:
athena% cd Interface_Basics

Then launch MATLAB from that directory by typing at the Athena prompt:
athena% add matlab
athena% matlab &

Start the MATLAB desktop interface by typing at the MATLAB prompt:
>> desktop

On laptops:

Launch MATLAB the same way you launch any software on your laptop. Then navigate to the work directory **Interface_Basics** either from the Current Directory Window, or by using the cd command in the Command Window.

Note that typically MATLAB starts from an Applications or Programs folder, and you need to navigate to a Users folder and from there to your user directory and the work directory **Interface_Basics**. The exact path will depend on the directory hierarchy on your laptop.

III. Exercise 1: Massachusetts Census Data

Purpose

To practice the following in MATLAB:

- Importing data from text files using the Import Wizard and the load command.
- Creating and referencing the elements of matrices and vectors.
- Applying operators such as +, -, *, and / to vectors.
- Using built-in functions such as sum, mean, and std for data analysis.
- Exporting results to data files using the save command.

Background

This example uses actual 2000 census data for the state of Massachusetts, courtesy of Daniel Sheehan, GIS Specialist, MIT Information Services & Technology.

1. Open Exercise_One/example1.m in the MATLAB Editor

>> edit example1.m
Lines that start with % are comments. The rest are **MATLAB commands**.

2. Try commands from example1.m in the Command Window

- Type the commands in the Command Window (or use Cut and Paste to copy them).
- Press Enter after commands and see the results in the Command Window.
- Note how matrix elements are referred to; for example:

>> counties = textdata(2:15, 1)
creates a 14x1 vector (the names of the fourteen counties in MA) from the elements in rows 2-15 in the first column of the matrix textdata.

- Note how matrices can be created from vectors; for example:
  >> genders = [Males Females]
  creates a 14x2 matrix from two 14x1 vectors.
- Note how scalar operators can be applied to matrices; for example:
  >> allpeople = Males + Females
  creates a 14x1 vector though element-wise summation of two 14x1 vectors.
  All vectors must have the same size!
- Note that element-wise multiplication and division require the . operator (matrix multiplication and division will be covered in the Linear Algebra session); for example:
  >> density = Population ./ area
  creates a 14x1 vector (population per square mile for the fourteen MA counties).
- Note how built-in data analysis functions act on matrices; for example:
  >> MAraces = sum(races)
  creates a 1x4 row vector (total number of people per race) from a 14x4 matrix.
- Note the effect of data analysis functions on vectors; for example:
  >> MApeople = sum(MAraces)
  creates a number i.e. a 1x1 matrix (total number of people in MA of all races).
- Note that all variables are considered matrices!
  >> whos

3. Execute the M-file in the Command Window

All commands in the M-file can be executed by running the file from the Command Window:

>> example1

IV. Exercise 2: Orbital Velocity

Purpose

To practice the following in MATLAB:

- Creating and referencing the elements of matrices and vectors.
- Applying element-wise operators such as +, -, .*, and ./ to vectors.
- Solving polynomial equations using the roots and polyval functions.

Background

This example uses data from the NASA educational web site:
http://exploration.grc.nasa.gov/education/rocket/rktrflght.html
The velocity V required for a rocket to move on a circular orbit at altitude h around a planet with a mean radius R_e and gravitational constant g_0 can be computed with Johannes Kepler's formula:

$$V = \sqrt{\frac{g_0 R_e^2}{R_e + h}}$$

1. Open M-file Exercise_Two/example2.m

>> edit example2.m

2. Try commands from example2.m in the Command Window

- Type the commands in the Command Window (or use Cut and Paste to copy them).
- Press Enter after commands and see the results in the Command Window.
- Note how matrices are created; for example:
  >> Re = [3962 1079 2111; 6376 1736 3396]′
  creates a 3x2 matrix by inverting a 2x3 matrix; rows are separated by semi-colons.
- Note how the elements of a matrix are referred to; for example:
  >> g0e = g0(:, 1)
  creates a 3x1 vector from the first column in a 3x2 matrix.
- Note that element-wise multiplication and division require the • operator; for example, see computations of the 3x1 vectors Ve and Vm in English and metric units, respectively.
- Note how the built-in functions roots and polyval are used, respectively, to solve a polynomial equation and to check its solution:
  >> v = roots(p)
  >> P = polyval(p, v)
  where p is a vector of coefficients of a polynomial equation for V:

$$p_1V^2 + p_2V + p_3 = (1)V^2 + (0)V + -\frac{g_0R_e^2}{\mathrm{R}_e + h} = 0$$

3. Execute the M-file in the Command Window

>> example2

V. Exercise 3: Surface Interpolation

Purpose

To practice the following in MATLAB:

- Importing data using the Import Wizard and saving image and data files.
- Interpolating a surface from points using meshgrid and griddata.
- Plotting data using plot3 and surf and annotating figures.

1. Open M-file Exercise_Three/example3.m

>> edit example3.m

2. Try commands from example3.m in the Command Window

- Type the commands in the Command Window (or use Cut and Paste to copy them).
- Press Enter after commands and see the results in the Command Window.
- Note that plotting commands such as plot3 or surf open the Figure editor.

3. Execute the M-file in the Command Window

>> example3