1 Instructions

Please submit your programming solutions to the lab exercises as scripts attached to an e-mail addressed to `suadusum@ucsd.edu`. In MATLAB, a script is also referred to as an **m-file**. For novice users, there is some further discussion below about saving your working code in this format.

Ideally, the subject line of your e-mail should start with `SIO135/SIO236` and contain the lab exercise number (e.g., “SIO135/SIO236 Lab 1”). Give your m-file attachment a filename containing your last name, first name, and the lab number as follows: `lastname_firstname_lab#.m`. For example, my e-mail (with the subject line “SIO135/SIO236 Lab 1” would have an attachment named `Adusumilli_Susheel_lab1.m`.

If you are not yet familiar with working in MATLAB, please proceed to Section 2 and complete the exercises in Section 3. On the other hand, students who only need to review their previous knowledge of MATLAB should just submit their solution for the exercise in Section 4.

2 About MATLAB

MATLAB (or its open-source equivalent like Octave or Python) is a great research tool because it has lots of handy built-in functions for manipulating vectors and matrices. There are two basic things you need to know about MATLAB.

2.1 Finding help

If you are not familiar with MATLAB, look through the “Getting Started”, “Examples”, and “Demos” section of the Help window, or read through an online tutorial. At any time, you can use the Help Index to look up a specific function. For example if you wanted to know more about the function `ifftshift`, you can type `help ifftshift` at the command prompt. You can also use the Help Index to look for a topic if you do not know the name of the function you need (e.g., “inverse Fourier transform”). There are also numerous online resources for quick tutorials or function reference:

`http://www.math.utah.edu/lab/ms/matlab/matlab.html`

`http://www.mathworks.com/help/matlab/functionlist.html`

2.2 Making a MATLAB script (m-file)

A script is a text file containing lines of commands (e.g., `yourscript.m`). When you run the script by typing `yourscript` at the MATLAB command prompt, MATLAB then executes each line one at a time exactly as if you had typed it into the command line. Lines beginning with `%` are treated by MATLAB as comments and are ignored. A script can be written with any text editor, but MATLAB has a built-in m-file editor. You should use an m-file for all your assignments. It allows you to save your work, and you can easily keep track of what you have done and make changes. Here’s an example m-file:

Listing 1: Sample contents of an m-file

```
% SIO 135/236 Lab 1

% Example Exercise # 1
x = [0:1:2048]
% (other commands follow)

pause

% Example

pause
```

Notice the `pause` commands separating the lines of MATLAB code pertaining to different exercises for the same lab assignment. This allows for the execution of one section of your m-file at a time. It is recommended that you follow this format when you save your work as an m-file for these lab exercises.

Again, for labs in this class please name your m-files using the following convention: `lastname_firstname_lab#.m`, where `lastname` is your family name, `firstname` is your given name and `lab#` is the number of the lab.

3 Exercises for New Users

1) Use your own computer or login to a computer in the lab using your regular UCSD username and password. Open MATLAB from the start menu. If you've never used MATLAB, look at a demo or a tutorial website (see above). Use an m-file to save your work, and make sure you save it in a place you can find it again (e.g., your Desktop or WorkArea).

2) Write a short program to generate a sine (or cosine) function 2048 points long. Generate exactly 32 or 64 full cycles over this interval. Plot the results and add labels to the axes. Always label your axes. *Always.*

3) Create two 6x6 matrices, A and B, of any numbers or function you like. Multiply A and B using matrix multiplication. Multiply corresponding elements of A and B. Plot both results using `imagesc` and add a colorbar. Change one plot to grayscale using colormap.

4 MATLAB Review Exercise

4) Exercise for advanced MATLAB users (skip problems 1–3). The class notes on platforms and orbits includes the following formulas for calculating a ground track for a satellite in a circular orbit about a spherical earth (don't worry about converting from geocentric to geodetic coordinates). Use this formula to calculate and plot the ground track of any remote sensing satellite. The following parameters will produce an exact 10-day repeat track for the Topex satellite.

$$\omega_e = 2\pi/86164.1 \tag{4.1}$$

$$\omega_s = 2\pi/6745.72 \tag{4.2}$$

$$\omega_n = \omega_e - \omega_s \frac{10}{127} \tag{4.3}$$

$$i = 66.01 \frac{\pi}{180} \tag{4.4}$$

$$\theta(t) = \sin^{-1}\left[\sin \omega_s t \sin i\right] \tag{4.5}$$

$$t(\theta) = \omega_s^{-1} \sin^{-1}\left[\frac{\sin\theta}{\sin i}\right] \tag{4.6}$$

$$\phi(t) = \tan^{-1}\left[\frac{-\sin\omega_e' t \cos\omega_s t + \cos\omega_e' t \sin\omega_s t \cos i}{\cos\omega_e' t \cos\omega_s t + \sin\omega_e' t \sin\omega_s t \cos i}\right] + \phi_0 \tag{4.7}$$