EXERCISES

Introduction to MATLAB: Practice

I. Class Materials

1. Download Practice.tar OR Practice.zip

From a web browser:

Download the file **Practice.tar** or **Practice.zip** from
http://web.mit.edu/acmath/matlab/IntroMATLAB to a local directory.
On Windows, if you do not have WinZip, download **Practice.zip**.

Alternatively, on Athena:

```
athena% add acmath
athena% cp  /mit/acmath/matlab/IntroMATLAB/Practice.tar .
```

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

On Athena *(or the UNIX shell on Mac OS X)**:***

```
tar –xvf Practice.tar
```

On laptops:

Use your computer's utilities, such as double click or WinZip on Windows or StuffIt on Mac. Without WinZip on Windows, double click on **Practice.zip** and select File->Extract All. Your local work directory should now contain the following directories and files:

Practice

- **Exercise_One**
  - rocketGUI.fig
  - rocketprogram.m
  - velocitycallback.m
  - orbitalvelocity.m

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

There are additional materials for this class. You may download or copy them from the acmath locker (optional) or watch the demos in class and look at the files later.

- 1_010_statistics.tar
- 1_010_statistics.zip
- MATLAB_File_Exchange_Downloads.tar
- MATLAB_File_Exchange_Downloads.zip
- 6_013_electromagnetics.tar
- 6_013_electromagnetics.zip

II. Start MATLAB

On Athena:

athena% cd Practice
athena% add matlab
athena% matlab &
>> desktop

On laptops:

Launch MATLAB and navigate to the work directory **Practice**.

III. Exercise 1: Rocket Velocity – A program with a Graphical user Interface

Purpose

To practice the following in MATLAB:

- Creating the graphical layout of a **Graphical User Interface (GUI)** with GUIDE.
- Writing **function M-files** for the GUI, including **callback functions**.
- Debugging a program using the MATLAB Editor in Debug Mode.

Background

This example, which we also used in Exercise Three in Session 4: Programming, is based on NASA's educational site: http://exploration.grc.nasa.gov/education/rocket/rktrflght.html. We shall create a MATLAB program with a Graphical User Interface, similar to the Circular Orbit Calculator (a Java applet), which runs on that web site.

1. Open rocketGUI.fig in the GUIDE Editor

>> cd Exercise_One
>> guide rocketGUI.fig

2. Understand how the GUI was created in GUIDE

- The file rocketGUI.fig was created in GUIDE. See the options to the left of the graphical layout, listing widgets that can be used to create a GUI.
- Note what **widgets** were used to create the GUI: two **popup menus**, one **edit text** field, one **push button**, and many **static text** fields.
- Open the Property Inspector of GUIDE by double clicking on a widget, or by selecting View->Property Inspector. For example, open it for the popup menu next to Planet.
- Most of the attributes have default values. Only the following were edited in GUIDE:
  - Attribute Tag was set to planet_popup.
  - Attribute String was set to an empty string.
  - Attribute Callback was set to nothing here. (The Callback is written in an M-file.)
  - Attributes for various Color and Font properties were set as desired.

- Note that the GUI does not do anything here: rocketGUI.fig is a graphic file that only tells MATLAB what graphical layout we want for the GUI. The code behind the GUI, i.e. functions that invoke actions when a user interacts with the widgets, is written in M-Files.

3. Close rocketGUI.fig

- Save the rocketGUI.fig file by clicking the Save icon in GUIDE (or File->Save As).
- Close the graphical file rocketGUI.fig before proceeding to the corresponding **M-Files**. Use GUIDE only to create the graphical layout of a program.

4. Open all M-files in the Exercise_One directory in the MATLAB Editor

```
>> edit rocketprogram.m
>> edit velocitycallback.m
>> edit orbitalvelocity.m
```

5. Read and understand rocketprogram.m and veloctiycallback.m

- rocketprogram.m is a **function M-File**, which is meant to run as a program.
- velocitycallback.m is a function M-File, which defines a **Callback function**, i.e. a function that is executed when a user interacts with widgets of the GUI that was created in the graphical file rocketGUI.fig.
- Read and try to understand the lines in both files, before running the program rocketvelocity from the Command Window.
- Note the use of built-in function openfig in both files to open the graphical layout in file rocketGUI.fig whenever the function is called (or to reuse it if it is already open):
  fig1 = openfig('rocketGUI.fig', 'reuse')
- Note the use of built-in function findobj in both files to find a widget in the GUI and return a handle to its Tag attribute (the Tag was set in GUIDE); for example:
  planet_popup = findobj(fig1, 'Tag', 'planet_popup')
  altitude_input_text = findobj(fig1, 'Tag', 'altitude_input_text')
  velocity_result_text = findobj(fig1, 'Tag', 'velocity_result_text')
- Note the use of built-in function set in rocketprogram.m to set the String attribute of popup menus (multiple options) and other widgets (one string); for example:
  set (planet_popup, 'String', {'Earth', 'Mars'}) ;
  set (radius_value_text, 'String', '6376');
- Note the use of built-in function set in rocketprogram.m to set the Callback attribute of popup menus, edit text fields, and push buttons; for example:
  set (planet_popup, 'Callback', 'velocitycallback') ;
- velocitycallback is a function defined in the **function M-file** velocitycallback.m.
- Note the use of built-in function get in velocitycallback.m to get the String attribute of an edit text field; for example:
  H_text = get (altitude_input_text, 'String')
  and that the string has to be converted to a number before computations:
  H = str2num (H_text);
- Note the use of get to get the Value attribute of a popup menu, i.e. the number of the string that is currently selected among several possible strings in the popup; for example:
  planet = get (planet_popup, 'Value')

- Note the use of built-in function set in velocitycallback.m to get the String attribute of a static text field and that all numbers have to be converted to strings first; for example:
  set (velocity_result_text, 'String', num2str(V));
- Add a third option to select the Moon in the popup menu for planets. Which file(s) do you have to modify? The mean radius R_e and gravitational constant g_0 of the Moon are:
  In metric units: R_e = 1736 km; g_0 = 1.615 m/sec^2.
  In English units: R_e = 1079 miles; g_0 = 5.3 ft/sec^2.

6. Execute the M-File velocityprogram from the Command Window

- Run the program to open the GUI, then use the widgets to select or enter arguments, and explain what happens in terms of specific command lines in the **function M-Files**:
  >> rocketprogram

7. Run the function orbitalvelocity in Debug Mode

- orbitalvelocity is a function defined in the M-file orbitalvelocity.m, and can be used independently of the program rocketprogram or the callback velocitycallback.
- Set stops in the file orbitalvelocity.m by clicking next to the line numbers 20, 23, 31 (if, elseif, else statements), 35, 42, (case statements), 36, and 43 (computations of V).
- Run orbitalvelocity from the Command Window:
  orbitalvelocity(1079, 5.3, 200, 'English'); for example:
  and see how the Command Window changes to the debug mode:
  K>>
  See how the program stops at the marked lines.
- Press the Step icon in the Editor's menu bar to execute the program line by line. Identify the program's flow control by if, elseif, else statements and by switch, case statements.
- Hold the cursor over any variable in the file to see when a value is gets assigned to that variable while the steps of the program are executed.
- Press the Run icon to finish the program run, or the Exit Debug Mode icon to switch back to non-debugging mode.

VI. MIT MATLAB Demos

- Statistical simulation programs for **1.010 Uncertainty in Engineering**.
  CC Andreas Langousis and Daniele Veneziano.
  MIT Department of Civil and Environmental Engineering.
- Interactive demo programs with Graphical User Interface for **6.013 Electromagnetics**.
  © 2004 Xiaowei He and Markus Zahn.
  MIT Department of Electrical Engineering and Computer Science.

VII. MATLAB File Exchange Demos

http://www.mathworks.com/matlabcentral/fileexchange/