EXERCISES

Introduction to MATLAB: Graphics

I. Class Materials

1. Download Graphics.tar

From a web browser:

Open your browser and go to http://web.mit.edu/acmath/matlab/IntroMATLAB.
Download the file **Graphics.tar** to a local work directory.
On Windows, if you do not have WinZip, download **Graphics.zip** instead.

Alternatively, on Athena:

Copy the file from the locker **acmath** to a local work directory.

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

On laptops:

Use your computer's utilities, such as double click or WinZip on Windows or StuffIt on Mac.
If you downloaded **Graphics.zip** on Windows, double click on it and select File->Extract All.

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

Graphics

Exercise_One
example1.m
grid_x.dat
grid_y.dat
interp_spline_z.dat
XYZ_point_coordinates.txt

Exercise_Two
example2.m
x_coord.txt
y_line.txt
y_quadratic.txt
y_cubic.txt

Exercise_Three
example3.m
cubic.tif

There is an additional file for Exercise Three, called cubic.avi, in the acmath locker. Download or copy cubic.avi to your Exercise_Three directory only if you have enough space for it: about 72 MB. If not, watch the demo in class and then try the file later.

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

II. Start MATLAB

On Athena:

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

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 **Graphics** either from the Current Directory window or by using the cd command in the Command Window.

III. Exercise 1: 3D Surface and Point Graphs

Purpose

To practice the following in MATLAB:

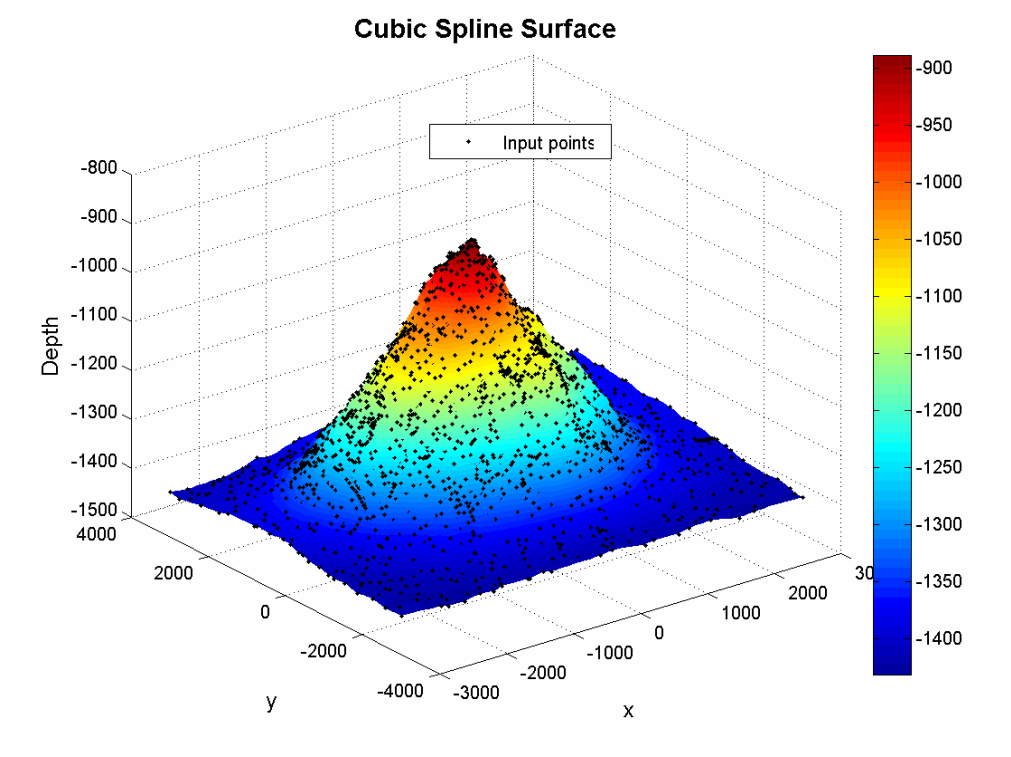
- Creating 3d plots using commands such as plot3 and surf.
- Adding multiple graphs to the same plot.
- Creating and calling figure and axes handles using functions gcf and gca.
- Annotating figures using commands such as title, xlabel, and legend.
- Using **Plot Edit mode** and the Property Editor to change graphics attributes.

To prepare for the Programming session by:

- Naming plots and axes, so that you can refer to them in a program.
- Representing everything you can do in the Figure editor and the Property Editor with MATLAB command lines.

Background

We shall use the point coordinates and the interpolated cubic spline surface from Example Three of Session1: Interface and Basics, and create an annotated 3d graph (see figure).



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 Copy and Paste to copy them).
- Press Enter after commands and see the results in the Command Window.
- Note the use of functions gcf and gca to get the handle to the current figure and the current axes, respectively; for example:
  >> fig1 = gcf
  >> axes1 = gca
  >> axes1 = get(gcf, 'CurrentAxes')
- Note the use of the function set to change attributes of figures; for example:
  >> set(fig1, 'Color', [0.95 0.87 0.73])
- Start **Plot Edit mode** for figures; for example:
  >> plotedit(fig1)
- Note the use of functions for annotating figures; for example:
  >> xlabel('x')
  >> legend(p1, 'Input Points')

- Save files using the File->Save As menu or using the function saveas; for example:
  >> saveas(fig1, 'surface.tif')

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: 2D Curve and Point Subplots

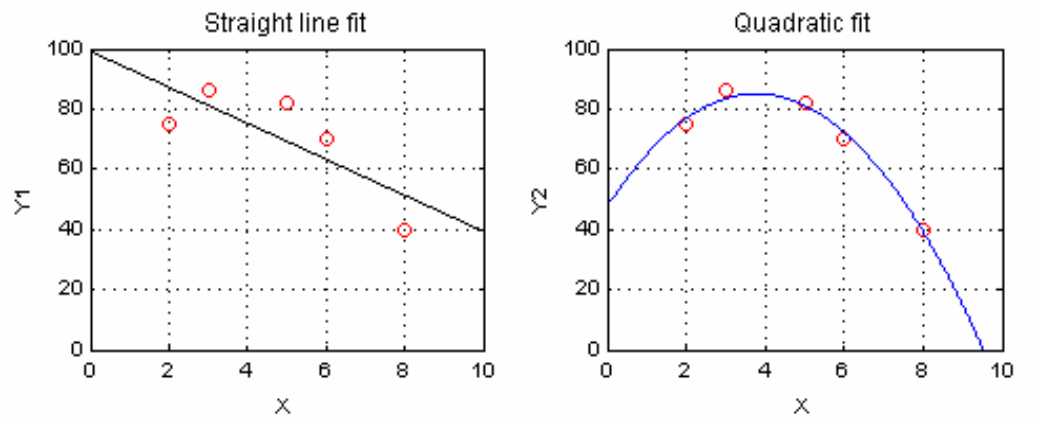
Purpose

To practice the following in MATLAB:

- Creating and customizing figures with subplots using the subplot command.
- Naming figures, subplots and axes, so that you can refer to them in a program.

Background

We shall use the polynomial curves we created to fit a set of data points in Example Two of Session 2: Linear Algebra and Calculus, and create a 2d graph with subplots (see figure).



1. Open Exercise_Two/example2.m in the MATLAB Editor

```
>> edit example2.m
```

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

- Type the commands in the Command Window (or use Copy and Paste to copy them).
- Press Enter after commands and see the results in the Command Window and in the Figure window. The Figure window is created when you call plotting commands.
- Note the use of the function subplot to create subplots in a figure; for example:
  >> s1 = subplot(2, 2, 1)
  >> p1 = plot(X, Y, 'ro')
- Note the use of the function gca to get the handle to the current axes; for example:
  >> axes1 = gca
- Note the use of functions axes and axis for axes' customization; for example:
  >> axes(axes1); axis([0 10 0 100]);

- Note the use of functions such as legend for annotating figures; for example:
  >> legend(P2, 'Y2=aX^2+bX+c')
- Save files using the File->Save As menu or using the function saveas; for example:
  >> saveas(1, 'subplots.tif')

3. Execute the M-file in the Command Window

```
>> example2
```

V. Exercise 3: Helix Animation

Purpose

To practice the following in MATLAB:

- Creating animations using the drawnow command.
- Saving and exporting animations using functions such as getframe and movie2avi.

To prepare for the Programming session by:

- Writing and using for loops.

Background

A Helix is a 3d curve described by the coordinates:

$$x(t) = a\cos(t)$$
$$y(t) = a\sin(t)$$
$$z(t) = bt$$

1. Open Exercise_Three/example3.m in the MATLAB Editor

```
>> edit example3.m
```

2. Read the file example3.m

- This file is a program. Before running it, read and try to understand all command lines.
- Note the use of function set to set the DoubleBuffer Attribute of the current figure to "on", which ensures that the figure will not flicker during redrawing:
  set(gcf, 'DoubleBuffer', 'on')
- Note the use of functions length and for to create a loop through the elements of a vector:
  ```
  for i = 1 : length(x)
      command lines
  end
  ```
- Note the use of function drawnow to force plotting at each step of the loop:
  drawnow
- Note the use of function set and gca to set the axes of the plot to be the same at every step of the animation loop; for example:
  set(gca, 'XLim', [-2 2], 'YLim', [-2 2], 'ZLim', [0 20])
- Note how the plots at every step are saved into a matrix using function getframe:
  M(i) = getframe(gcf)
- Note how the animation gets exported (outside of MATLAB) using movie2avi:
  movie2avi (M, 'helix.avi', 'fps', 30, 'quality', 100)

3. Execute the M-file in the Command Window

After understanding the code, run the file from the Command Window to see the animation:

```
>> example3
```

4. Watch the AVI file outside of MATLAB

If you created the file helix.avi, you may watch it outside of MATLAB. AVI is a Windows format, so it should play fine in whatever player you use on Windows.

You may need special software to see AVI files on other platforms. For example, the Apple QuickTime Player may not run AVI files. A free player that can show AVI movies on Mac OS X is the VLC Media Player (download from http://www.videolan.org/).

On Athena, you can watch AVI files using the program xanim.

```
athena% add graphics
athena% xanim helix.avi
```

In general, a good place to find out what extra programs are available on Athena is the web site **What Runs Where on Athena**: http://web.mit.edu/acs/www/whereruns.html.

VI. Create graphics with your own data

By now, you should have learned how to import data, organize them in matrices, manipulate them using operators and built-in functions, and create 2d and 3d annotated plots in MATLAB. Try all this using your own data.