Due Date : 25th November 2016

Important Instructions

1. Do the following using GNU **Octave**.
2. Hard copy is not required. Submit via the LMS link.
3. Do all the following questions and **copy the commands you used and the important output (such as graphs, plots, answers and error messages) to a report (doc, pdf, etc) and submit.**
4. **For each question, also save the command line inputs and outputs and upload as a separate text file. (e.g. For question 1, save as "Q1.txt").**
   Hint: Use the **diary** command.

A diary of a session is initiated with the diary command followed by the file name that you want to keep the text file. You then type in all of the necessary commands. When you are done, enter the diary command alone, and it will write all of the output to the file and close the file. In the example below a file called "save.txt" is created that will contain a copy of the session.

```
>>diary save.txt
        ... enter commands here...
>>diary
```

This will create a file called "save.txt" which will hold an exact copy of the output from your session. (See end of the assignment for an additional note on advanced file I/O)

1) **Linear system of equations.**
   Solve the following system of equations using ****. Compute and display the answer/error vector.

   i.
   $$x + 4y = 34$$
   $$-3x + y = 2$$

   ii.
   $$2x - 2y = 4$$
   $$-x + y = 3$$
   $$3x + 4y = 2$$

   iii.
   $$3a + 6b + 4c = 1$$
   $$a + 5b = 2$$
   $$7b + 7c = 3$$

   (10 marks)

2) **Polynomial fitting.**
   i. Evaluate $y = x^2$ for $x = -4:0.1:4$.
   ii. Add random noise to these samples. (Use **randn)**. Plot the noisy signal with **.** markers.
   iii. Now fit a 2nd degree polynomial to the noisy data.
   iv. Plot the fitted polynomial on the same plot, using the same x values and a red line. (Use the **holdon** command).

   (10 marks)

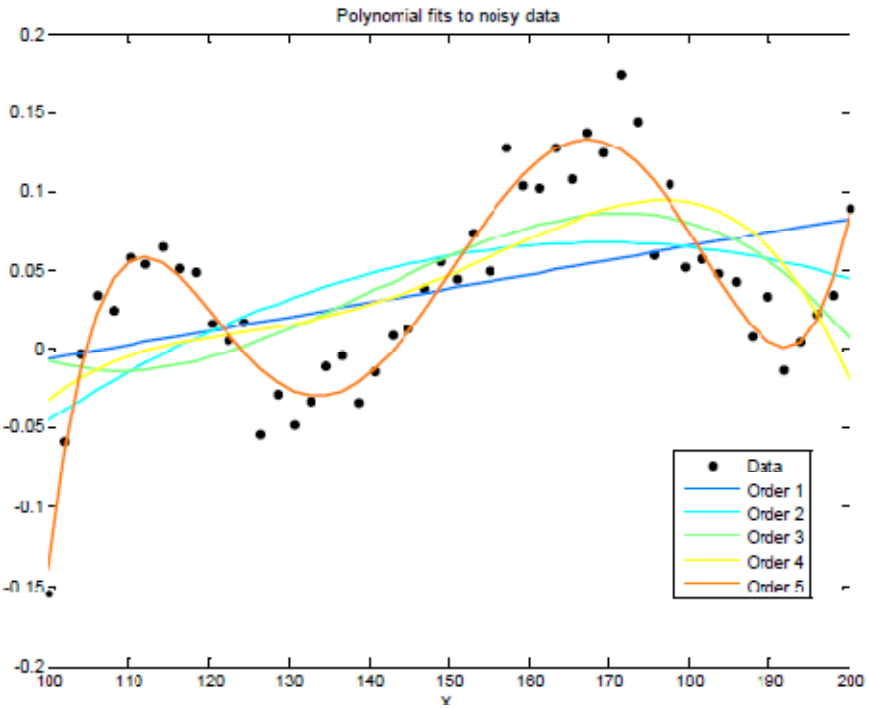
3) Polynomial fitting.

Write a script to load the data file randomData.mat (which contains variables `x` and `y`) and fit first, second, third, fourth, and fifth degree polynomials to it.

Plot the data as blue dots on a figure, and plot all five polynomial fits using lines of different colors on the same axes. Label the figure appropriately.

To get good fits, you'll have to use the centering and scaling version of **polyfit** (the one that returns three arguments, see **help**) and its counterpart in **polyval** (the one that accepts the centering and scaling parameters).

It should look like this:



(20 marks)

4) **Semilog plot**.

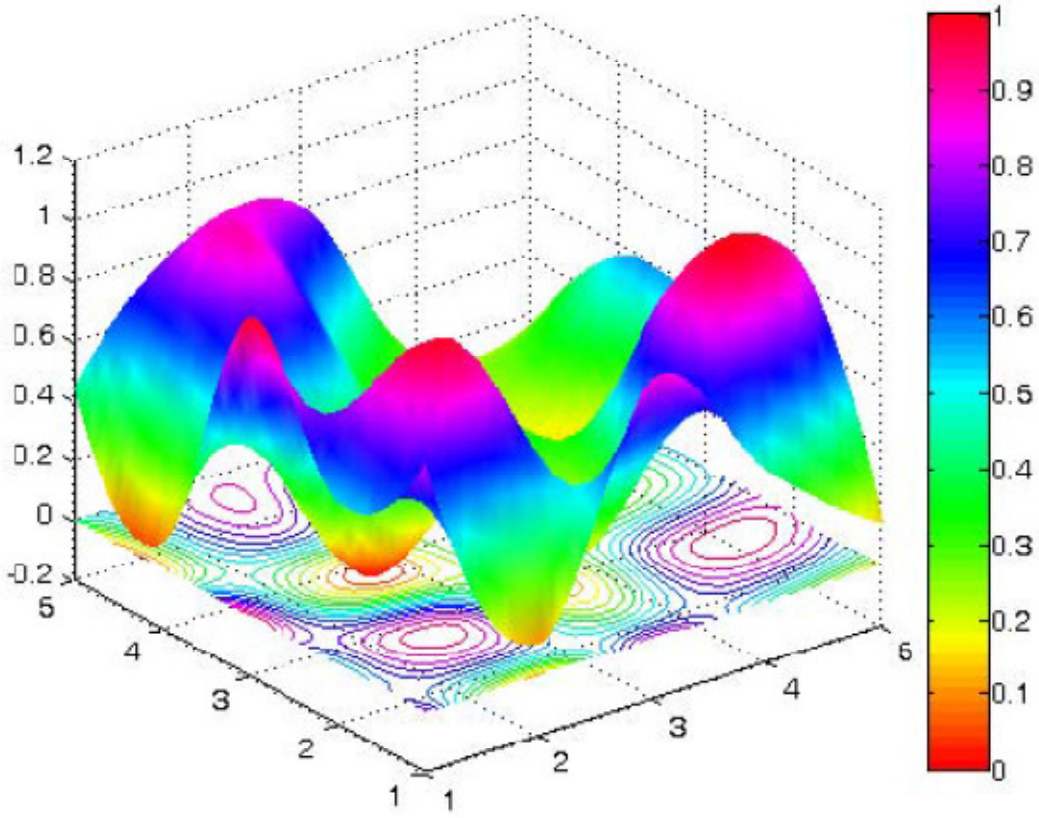
Over the past 5 years, the number of students in a class has been 15, 25, 55, 115, 144. Class size seems like it's growing exponentially. To verify this, plot these values on a plot with a log y scale and label it (**semilogy**, **xlabel**, **ylabel**, **title**).

Use magenta square symbols of marker size 10 and line width 4, and no line connecting them. You may have to change the x limits to see all 5 symbols (**xlim**).

If the relationship really is exponential, it will look linear on a log plot.

(10 marks)

5) **Interpolation and surface plots**. Write a script called `randomSurface.m` to do the following.
   a. To make a random surface, make Z0 a 5x5 matrix of random values on the range [0,1] (use **rand**).
   b. Make an X0 and Y0 using **meshgrid** and the vector 1:5 (use the same vector for both inputs into meshgrid). Now, X0, Y0 and Z0 define 25 points on a surface.
   c. We are going to interpolate intermediate values to make the surface seem smooth. Make X1 and Y1 using **meshgrid** and the vector 1:0.1:5 (again use the same vector for both inputs into meshgrid).
   d. Make Z1 by interpolating X0, Y0 and Z0 at the positions in X1 and Y1 using cubic interpolation (use **interp2**, specify **cubic** as the interpolation method).
   e. Plot a surface plot of Z1. Set the colormap to **hsv** and the shading property to **interp** (use **surf, colormap, shading).**
   f. Hold on to the axes and plot a 15-line contour on the same axis. (check the LevelStep, LevelList properties for setting the spacing between contour lines). This will make a 2-dimensional plot.
   g. Add a colorbar (use **colorbar**).
   h. Set the color axis to be from 0 to 1 (use **caxis**). The final figure should look something like this. (If the figure is not copy/pasting into your document properly, try changing the figure copy options to use a bitmap).



(25 marks)

6) **Loops and flow control**.
Make function called loopTest(N) that loops through the values 1 through N and for each number n it should display 'n is divisible by 2', 'n is divisible by 3', 'n is divisible by 2 AND 3' or 'n is NOT divisible by 2 or 3'.

Use a **for** loop, the function **mod** or **rem** to figure out if a number is divisible by 2 or 3, and **num2str** to convert each number to a string for displaying. You can use any combination of **if**, **else**, and **elseif**.

(25 marks)

Additional Note

C Style Read/Write

In addition to the high level read/write commands you already know, Matlab allows C style file access. This is extremely helpful since the output generated by many home grown programs is in binary format due to disk space considerations. This is an advanced subject, and we do not go into great detail here. Instead we look at the basic commands. After looking at this overview we highly recommend that you look through the relevant help files. This will help fill in the missing blanks.

The basic idea is that you open a file, execute the relevant reads and writes on a file, and then close a file. One other common task is to move the file pointer to point to a particular place in the file, and there are two commands, **fseek** and **ftell** to help.

Here we give a very simple example. In the example, a file called "laser.dat" is opened. The file identifier is kept track of using a variable called *fp*. Once the file is opened the file position is moved to a particular place in the file, denoted pos , and two double precision numbers are read. Once that is done the position within the file is stored, and the file is closed.

```
fp = fopen('laser.dat','r');
fseek(fp,pos,'bof');
tmp = fread(fp,2,'double');
pos = ftell(fp);
fclose(fp);
```