Important Instructions

1. Do the following using Octave or Matlab.
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 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.

1) Linear system of equations: (15 Marks)

a) Solve the following system of equations using \. Compute and display the error vector

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

b) Solve the following system of equations using **inv()** command. Compute and show values for a,b,c and d.

$$2a + 5b - c + 4d = 0$$
$$a + b + c + d = 0$$
$$4a - 3b + 6c + d = 0$$
$$2a - 5b - 3c - d = 7$$

2) **Plot a circle. (25 Marks)** It's not immediately obvious how to plot a circle in Octave/Matlab. Write the function [x,y]=getCircle(center,r) to get the x and y coordinates of a circle. The circle should be centered at center (2-element vector containing the x and y values of the center) and have radius r. Return x and y such that plot(x,y) will plot the circle.

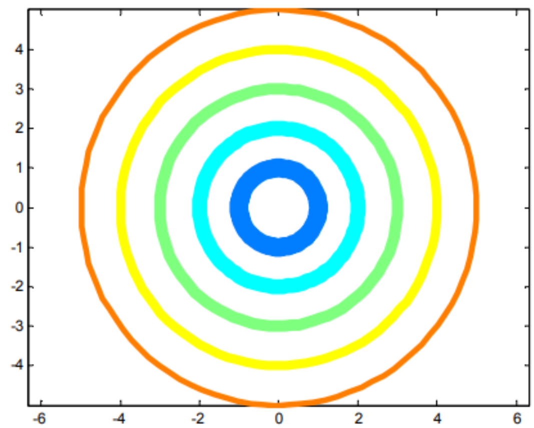
a) Recall that for a circle at the origin (0,0), the following is true:

$x(t) = \cos(t)$

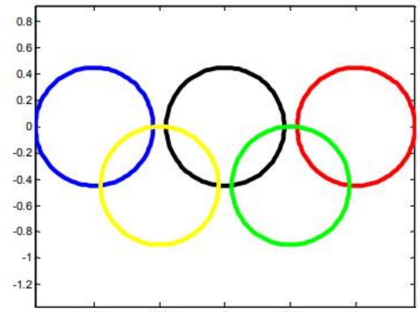
$y(t) = \sin(t)$ for t on the range $[0, 2\pi]$.

Now, you just have to figure out how to scale and translate it.

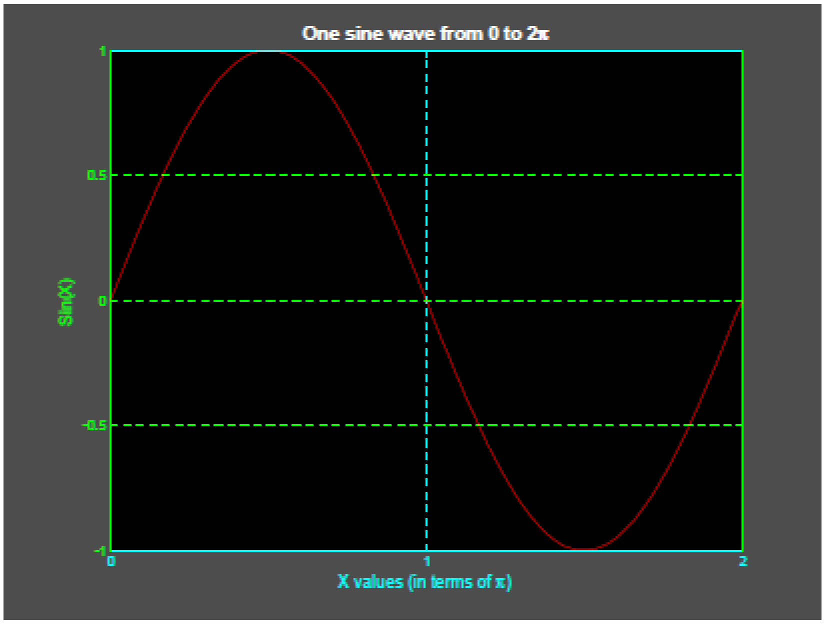
b) Write a script called `concentric.m`. In this script, open a new figure and plot five circles, all centered at the origin and with increasing radii. Set the line width for each circle to something thick (at least 2 points), and use the colors from a 5-color jet colormap (jet). The property names for line width and color are 'LineWidth' and 'Color', respectively. Other useful function calls are hold on and axis equal. It should look something like this.



c) Make a script called olympic.m. This script should use your getCircle function to draw the Olympic logo, as shown below. Don't worry about making the circles overlap in the same way as the official logo (it's possible but is too complicated for now). Also, when specifying colors, you can use the same color codes as when plotting lines ('b' for blue, 'k' for black, etc.) instead of providing an RGB vector.



3. Handles. We'll use handles to set various properties of a figure in order to make it look like this: **(20 Marks)**



1) Do all the following in a script named handlesPractice.m
2) First, make a variable **x** that goes from 0 to 2π , and then make **y=sin(x)**.
3) Make a new **figure** and do **plot(x,y,'r')**
4) Set the x limit to go from 0 to 2π (**xlim**)
5) Set the **xtick** property of the axis to be just the values [0 pi 2*pi], and set ***xticklabel*** to be {'0','1','2'}. Use **set** and **gca**
6) Set the **ytick** property of the axis to be just the values -1:.5:1. Use **set** and **gca**
7) Turn on the grid by doing **grid on**.
8) Set the **ycolor** property of the axis to green, the **xcolor** property to cyan, and the **color** property to black (use **set** and **gca**)
9) Set the colorproperty of the figure to a dark gray (I used [.3 .3 .3]). Use **set** and **gcf**
10) Add a title that says '**One sine wave from 0 to 2π**' with fontsize **14**, fontweight **bold**, and color **white**. Hint: to get the π to display properly, use \pi in your string. Matlab uses a Tex or Latex interpreter in xlabel, ylabel, and title. You can do all this just by using title, no need for handles.
11) Add the appropriate x and y labels (make sure the π shows up that way in the x label) using a **fontsize** of **12** and **color cyan** for x and green for y. Use **xlabel** and **ylabel**
12) Before you copy the figure to paste it into word, look at copy options (in the figure's Edit menu) and under '**figure background color'** select 'use figure color'.

4) Throwing a ball: (25 Marks)

Below are all the steps you need to follow, but you should also add your own meaningful comments to the code as you write it.

a) Start a new file in the Octave Editor and save it as `throwBall.m`

b) At the top of the file, define some constants (you can pick your own variable names).

I. Initial height of ball at release = 1.5m
II. Gravitational acceleration = 9.8 m/s^2
III. Velocity of ball at release = 4 m/s
IV. Angle of the velocity vector at tie of release = 45 degrees

c) Next, make a time vector that has 1000 linearly spaces values between 0 and 1, inclusive.

d) If *x* is distance and *y* is height, the equations below describe their dependence on time and all the other parameters (initial height *h*, gravitational acceleration *g*, initial ball velocity *v*, angle of velocity vector in degrees θ). Solve for *x* and *y*.

i. $x(t) = v\cos\left(\theta\frac{\pi}{180}\right)t$. We multiply θ by $\frac{\pi}{180}$ to convert degrees to radians.

ii. $y(t) = h + v\sin\left(\theta\frac{\pi}{180}\right)t - \frac{1}{2}gt^2$

e) Approximate when the ball hits the ground

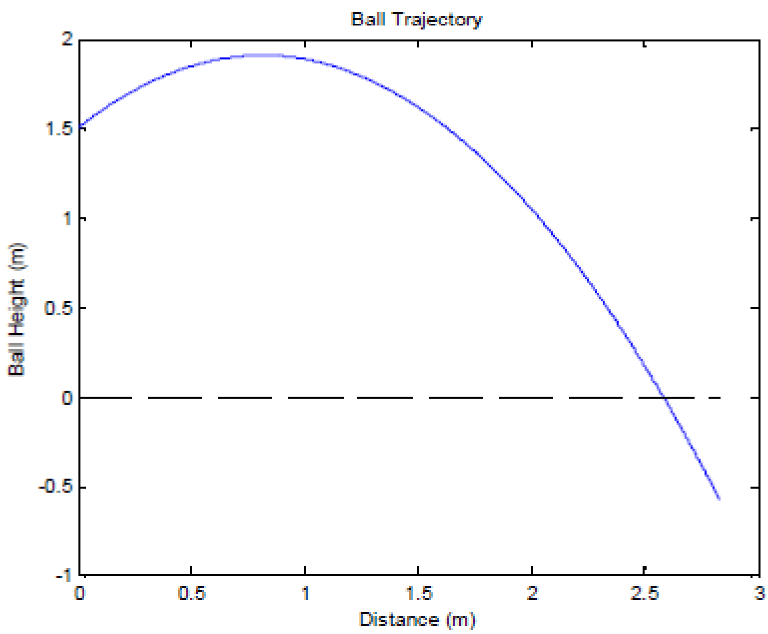
I. Find the index when the height first becomes negative (use **find**)
II. The distance at which the ball it the ground is value of x at that index
III. Display the words: *The ball hits the ground at a distance of x meters.* (Where x is the distance you found in part II above).

f) Plot the ball's trajectory

I. Open a new figure (use **figure**)
II. Plot the ball's height on the y axis and the distance on the x axis.
III. Label the axes meaningfully and give the figure a tittle. (use **xlabel**, **ylebel** and **title**)
IV. Hold on to the figure (use **hold on**)
V. Plot the ground as a dashed black line. This should be a horizontal line going from 0 to the maximum value of x. (use **max**). The height of this line should be 0. (see **help plot** for line colors and styles).

g) Run the script from the command window and verify that the ball indeed hits the ground around the distance you estimated in e, ii. You should get something like this:

```
>> throwBall
The ball hits the ground at a distance of 2.5821 meters
```



5. (a) Suppose that x=2 and y=5. Use Octave to compute the following. **(5 Marks)**

(i) $\frac{yx^3}{x-y}$ (ii) $\frac{x^5}{x^5-1}$

(b) Suppose x=-7-5i and y = 4 + 3i. Use Octave to compute **(5 Marks)**

(i) x + y (ii) xy (iii) x/y

(c) Use Octave to find the roots of $\mathbf{13x^3 + 182x^2 - 184x + 2503 = 0}$ **(5 Marks)**