EE202 Lab#7 Input and Output

Input and Output

Although not covered in class, as part of the lab we will be covering Chapter 13 "Input and Output" from your textbook. This is a short Chapter 13 of only 5 pages.

1. [13.2] Make a new M-File named `helloWorld`. Add the following comment at the beginning of your script. %% Lab 7 Question 1 Hello World

(a) If you have ever taken a computer programming class, this problem should not surprise you. Write a program using the disp function to display the message `Hello, World`. In the command window type help input for a hint on the next part of the question.

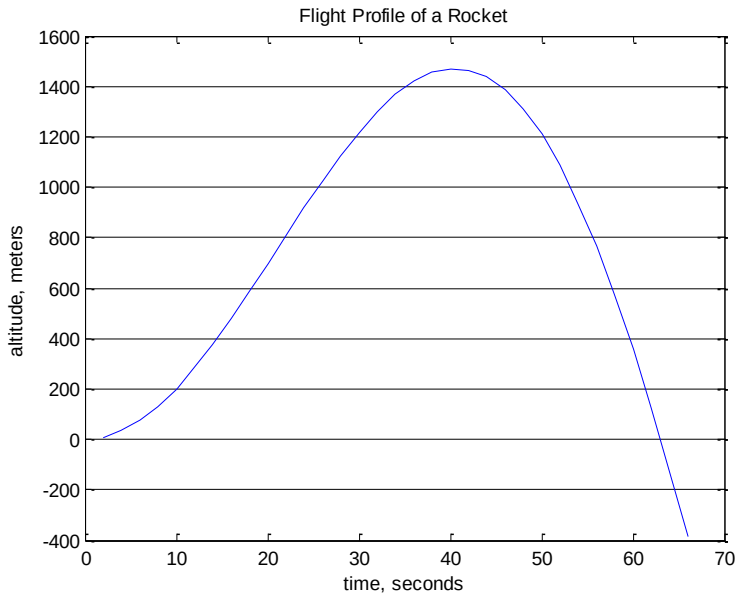
(b) Use two separate input statements to prompt a user to enter his or her first and last name without single quotes required. Use the `disp` function welcome the person. You will need to combine the names and some spaces into an array.

```
Enter your first name Jane
Enter your last name Doe
Hello Jane Doe
```

2. In lab 4 you calculated and plotted information about *a* the altitude of a rocket (in meters) as a function of time from launch. Open the M-file containing your `rocket` script. Save it under the new name `rocket7`. Add the following comment at the beginning of your script. %% Lab 7 Question 2 Rocket

(a) Applying what you learned in labs 5 and 6 update your M-file to format your output using the `fprintf` function and to graph the rocket's trajectory as shown here. The letters nn should be replaced by numbers output by your program. Please display integers and floating point numbers to the precision shown.

```
The rocket hits the ground after   nn and before   nn seconds
The maximum height is nnnn.nn and occurs at   nn seconds
```



(b) MATLAB offers a technique for entering ordered pairs of x- and y-values graphically. Using the `ginput` function have the user find the point where the rocket hits the ground. Display the answer as shown here. Units should be centimeters and seconds.

```
To within an accuracy of nnnn.nn centimeters the rocket hits the ground
after nnn.nn seconds
```

On Windows 7 running MATLAB R2007a I received the error: `java.lang.NullPointerException` You may ignore this error message.

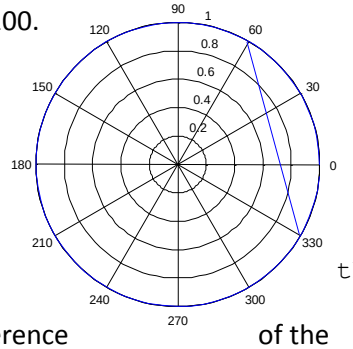
(c) Use the MATLAB `fzero` function to find the actual time, in seconds, it takes for the rocket to hit the ground. Display the answer as shown here.

```
The rocket hits the ground after 63.01 seconds.
```

3. As shown in the last problem, the `ginput` function is useful for picking distances off a graph. Let us have some more fun with this function and a little geometry by drawing the chord of a circle. Make a new M-File named `circle`. Add the following comment at the beginning of your script. %% Lab 7 Question 3 Draw the Chord of a Circle

- Define an array of angles from 0 to 2π, with a spacing of π/100.
- Define an array of `ones` of the corresponding length.
- Using the `polarEE202` function draw your unit circle as shown below.
- Using the `disp` function, prompt the user to enter two locations on the circle.

```
Use the cursor to enter two locations on the
circle
```

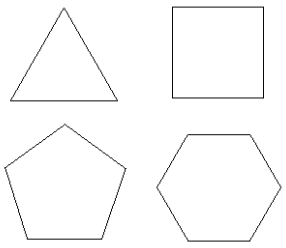


- Use the `ginput` function to pick two points on the circumference of the circle.
- Use hold on to keep the figure from refreshing, and plot a line between the two points you picked.
- Use the data from the points to calculate the length of the between them (a chord of the circle). (Hint: Use the Pythagorean theorem in your calculation.)

```
The distance between the points is 1.41 units
```

More Functions

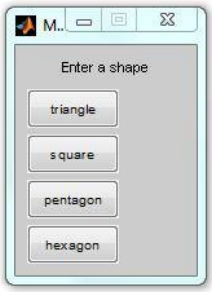
4. In order to have a closed geometric figure composed of straight lines, the angles in the figure must add up to $(n-2)(180 \text{ degrees})$, where n is the number of sides. Make a new M-File named `polygon`. Add the following comment at the beginning of your script. %% Lab 7 Question 4 Interior Angles of a Polygon



(a) Write a program using the `switch-case` statement that prompts the user to enter one of four polygons and returns the sum of the interior angles.

```
Enter either triangle, square, pentagon or hexagon:  triangle
The sum of the interior angles is:   180
```

(b) Add the following comment at this point in your script. %% (b) Sum of Interior Angles Replace your `input` and `switch-case` statements with a `menu` function. After clicking on hexagon the following output is displayed.




```
shape =
     4
The sum of the interior angles is:   720
```

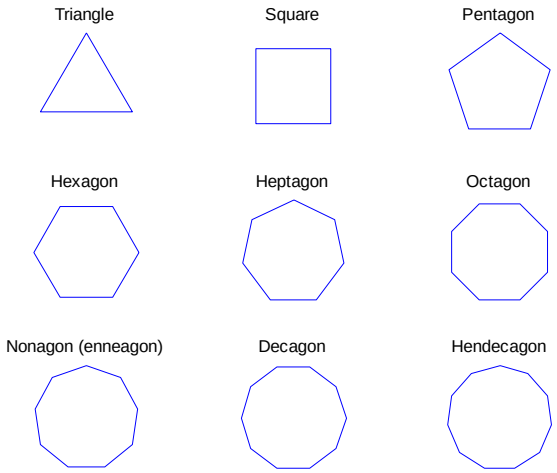
5. Now let us apply what we learned to draw a 'Triangle', 'Square', 'Pentagon', 'Hexagon', 'Heptagon', 'Octagon', 'Nonagon (enneagon), Decagon', 'Hendecagon', and a 'Six Pointed Star.' For the geeks out there; while nonagon uses the Latin *nonus* for ninth, the other polygon names are derived from the Greek. Following the Greek naming convention, a 9 sided polygon's correct name is enneagon – correction by Justin Chi.

   The easiest way to draw a polygon with n points in MATLAB is to use polar coordinates. You simply need to identify points on the circumference of a unit circle and draw lines between those points. For example, to draw a triangle you would first determine the location of the points of the triangle on the unit circle. A triangle has 3 points, which means they would be located 360 / 3 = 120 degrees or in radians $2\pi / 3 = \frac{2}{3}\pi$ apart. Let us locate the first point of the triangle at the top of the circle ($\theta = \pi/2$, r=1) and then progress counterclockwise around the circle in increments of $\frac{2}{3}\pi$ (i.e., $\frac{1}{2}\pi$, $1\frac{1}{6}\pi$, $1\frac{5}{6}\pi$,$2\ \frac{1}{2}\pi$). In other words, you want to plot your first line from $\frac{1}{2}\pi$ to$1\frac{1}{6}\pi$, your second line from $1\frac{1}{6}\pi$ to $1\frac{5}{6}\pi$, and your third line from $1\frac{5}{6}\pi$ to $2\ \frac{1}{2}\pi$. Notice that the fourth point $2\ \frac{1}{2}\pi$ is at the same coordinates as the first point $\frac{1}{2}\pi$ thus closing the shape. Here is the Matlab code to create a triangle.

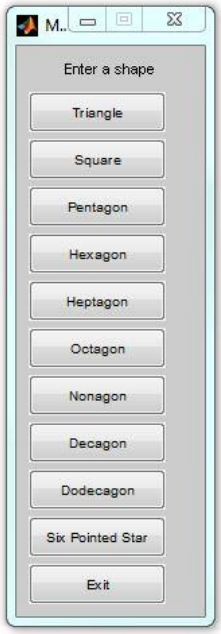
```
theta=pi/2:2*pi/3:2*pi + pi/2;
r = ones(1,length(theta));
polarEE202(theta,r)
```

   As seen in the figure below polygons with an odd number of points, like the triangle, all start at $\frac{1}{2}\pi$ (triangle, pentagon, heptagon, nonagon (enneagon), and hendecagon). For a square with four points the angle between each point would be $2\pi/4 = \frac{1}{2}\pi$ and we would want to locate our first point at $\frac{1}{4}\pi$. As seen in the figure below polygons with an even number of points, like the square, all will start from a different angle, such that the top of the polygon is flat. For the hexagon and decagon, you can place the first point at ($\theta = 0$, r=1).

   Make a new M-File named `polygons_drawing`. Add the following comment at the beginning of your script. %% Lab 7 Question 5 Nine Polygons Using a `for loop` and a `switch-case` statement create the following figure.



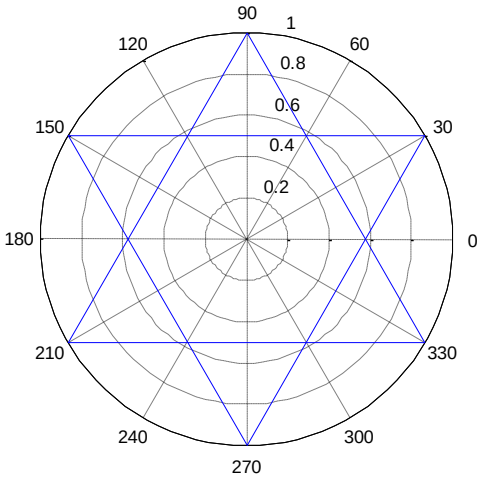
To remove the grid lines and remove the labels I would recommend using Google. The best solution I could find was the trick mentioned at compgroups; hopefully, you can find a more "correct" answer. You also may want to consider a modified version of polarEE202. If you like programming challenges, my solution only contained 3 cases (square, hexagon and decagon, octagon) with the `otherwise` default condition handling all the other polygons (*the best solution I have seen requires no switch-case statement*). If you do not like programming challenges then I would recommend a `case` statement for each polygon. Please let me know if you accepted the programming challenge (*possible bonus point opportunity*).



6. Open the M-file containing your `polygons_drawing` script. Save it under the new name `polygons_menu`. Change the comment at the beginning of your script to %% Lab 7 Question 6 Polygon and Star Menu.

   (a) Applying what you learned in question 4 and 5 replace the `for` statement with a `while` loop and a menu allowing the user to select the shape to be drawn. The `while` loop should check for the condition where the user presses the exit button. For this problem, please keep the polar grid on (default).

   (b) As the second to last menu item add a six-pointed star. The six pointed star can be constructed from two triangles, with the second offset from the first by 60 degrees ($\pi/3$). You will need to use the `hold on` function to keep the two plots in the same figure.



Please obtain my signature for question 5. Be ready to explain your thought process and any code optimization.

Appendix A Creating Your Lab Report

How to Organize and Clean up Your Work

To clean up past work, place the following line at the beginning of your M-File(s).

```
format compact
clear,clc, close all  %  The close all command closes all figure windows
```

To make more than one figure in an M file, use the function `figure(n)`, where **n** is the next figure to be drawn.

How to Publish your Lab

Matlab can format your M-file and the resulting outputs for publishing (i.e., a format you can turn in with your lab) by selecting File → Publish To →Word Document. If the conversion fails then try File → Publish To HTML, open in IE or Firefox and convert to PDF or simply print.