ENGR 105 – Introduction to Scientific Computing

Assignment 2

Due by 11:59 pm on Weds. 1/26/2022 on Gradescope

Problem 1 (10 pts): Consider an array of random numbers constructed in MATLAB using a `for` loop. Type the following code as a script, save it, and run it. Observe the form of the output `randArr`.

```
for jj = 1:20
    randArr(jj) = rand
end
```

Note that `rand` outputs a random number.

Produce an array of random numbers, `myRandArr`, with the same dimensions (but not necessarily the same exact numbers) as in the example above using a single statement. Your code should be in the form of a one-line script.

Helpful hint: check the MATLAB help document for `rand`.

Upload your script as Assignment2_problem1.m when submitting your assignment files and reference the filename in your README.txt.

Problem 2 (15 pts): Write a MATLAB function that uses the law of cosines to compute the length of a third side of triangle, *c*, given the lengths of the other two sides, *a* and *b*, and the angle opposite c, *θ*. Your function should compute *c* based on the following mathematical expression

$$c^2 = a^2 + b^2 - 2ab\cos(\theta)$$

You function should begin with the following function declaration.

```
function c = cosine_rule(a,b,theta)
```

The inputs `a`, `b`, and `theta` should be scalar (not a vector). Input `theta` should have units of degrees. Be wary of the trigonometric units used when computing cosine in MATLAB.

Using your function, compute and report *c* for the following sets of values of *a*, *b*, and *θ* by calling the function from the command line three times with each set of inputs.

Set number	***a***	***b***	***θ* (˚)**
1	5	3	45
2	1	2	60
3	3	4	90

Upload your code as cosine_rule.m when submitting your assignment files and declare the filename in your README.txt. Provide the outputs for c in your README.txt.

Problem 3 (15 pts): State, giving justifications, which of the following are <u>not</u> valid MATLAB variable names. Valid infers that the variables should (a) be able to store data, (b) should not overwrite built-in functions, and (c) should not overwrite built-in variables.

(i) `imag`
(ii) `a2`
(iii) `a.2`
(iv) `2a`
(v) `a-one`
(vi) `aone`
(vii) `_y_1`
(vii) `notValid`
(ix) `kitten mittens`
(x) `inf`
(xi) `Kitten_Mittens`
(xii) `min*2`
(xiii) `what`
(xiv) `pi`
(xv) `ENGR105`

Submit your responses in your README.txt file. You do not have to submit an .m file for this problem.

Problem 4 (10 pts): State, giving justifications, which of the following are <u>not</u> valid MATLAB statements. Valid infers that the expression on the right hand of the equal sign yields only variable `x` in the workspace memory. Assume that the workspace memory is empty when invoking any of the statements.

(i) `x = 2,15`
(ii) `x = 3.57*e2`
(iii) `x = .0`
(iv) `x = 3.57e2.1`
(v) `x = 25.82`
(vi) `x = 3.57e+2`
(vii) `x = -356231`
(viii) `x = 3,57e-2`
(ix) `x = (1 5 7)`
(x) `x = [1, 5, 7]`

Submit your responses in your README.txt file. You do not have to submit an .m file for this problem.

Problem 5 (25 pts): Hypocycloids, epicycloids, epitrochoids and hypotrochoids are all fancy names for curves generated when a circle rotates about another circle (aka roulettes). You can find interesting descriptions and .gifs for all of these roulettes on Wikipedia. Create a MATLAB function with the following function declaration that takes in scalar inputs `R`, `r`, and `d`.

```
function spirograph(R,r,d)
```

Your function should generate a curve of the following form.

$$x(\theta) = (R+r)\cos\theta + d\cos\left(\left(\frac{R+r}{r}\right)\theta\right)$$

$$y(\theta) = (R+r)\sin\theta - d\sin\left(\left(\frac{R+r}{r}\right)\theta\right)$$

You should begin by creating an array called `theta` that has values between 0 and 10π in steps of no more than 0.001 radians. You can do this using MATLAB's colon operator (aka slice notation) or the `linspace` command. Then compute values for *x* and *y* using array operations, addition, subtraction, the cosine function, etc. This should all be achieved without loops. Once you have the two arrays, `x` and `y`, your function should plot the curve using the command `plot(x,y)`. Look at the help pages for the `plot` command to determine the visualization options this command offers and how you can go about changing the color or style of the plotted line. You do not *have to* adjust the plot visualization but should feel free to do so.

Produce 3 separate plots corresponding to the `R`, `r`, and `d` values described below. Save these plots in .jpg format by selecting *File* → *Save As* from the plot window, selecting "JPEF image (*.jpg)" from the *Save as type* drop down box, and then indicating a reasonable *File name*.

Plot number	***R***	***r***	***d***
1	5	1	0.4
2	12	-1	1.5
3	7.5	-1	1

Upload your code as spirograph.m and submit each of your plots as a separate .jpg files. You are encouraged to name your image files Assignment2_problem5_plot1.jpg, Assignment2_problem5_plot2.jpg, etc. Declare all filenames (the function and images) in your README.txt.