ENGR 12800

Spring 2020

Computer Lab Practical

Instructions:

The attached practical includes six numbered problems. Prepare complete solutions for each problem to the best of your ability. Your solutions must be well organized including clear labeling of parts, and documentation of all MATLAB commands and results. Unless explicitly exempted in the problem instructions problems should include both the commands, scripts and/or functions you use and the Results in MATLAB (e.g.., the command window results, any figure windows or any pop up boxes). Do add your final answers into appropriate places in this document and upload when complete. Each problem should start on a new page. Your instructor will provide you with upload instructions and deadlines.

You are allowed to consult the internet including the lab website and the MATLAB website. You are allowed to use any previous problems you have completed yourself. You may not post anything about the practical. You should not discuss or communicate about the practical with others, except for consulting your instructor. You should type all code new and not copy and paste from previous code. You may copy and paste comments (but do edit them carefully).

If you have technical problems contact your instructor immediately.

Your Affirmation

By typing your name on the blank below (and turning in this practical) you are confirming you have followed the instructions above including:

* You have not discussed (or communicated about) this exam with anyone other than your instructor during the period from Monday, April 27th through Wednesday, April 29th.
* You have not posted any material or questions about the practical electronically or otherwise.
* The work on this document is all your own.

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



1. Function with multiple I/O and comments (23 points): The donut shaped solid shown in Figure 1 is a torus. In this figure the inner radius (a) and the outer radius (b) are labeled. Based on these dimensions the surface area and volume of a torus can be calculated using the following formula:

$$Surface Area= π^{2}\left(b^{2}-a^{2}\right)$$

$$Volume= \frac{1}{4}π^{2}\left(b-a\right)^{2}(b+a)$$

Use the steps below to develop a function that calculates and outputs the surface area and volume of a torus given the inner (a) and outer (b) radii. All input and output must happen at the command line only. Program code and the setup table (part a) must be consistent.

**Figure 1**: A Torus with inner (a) and outer (b) radii labeled. From: M.R. Spiegel, *Mathematical Handbook*, Schaum’s Outlines Series, McGraw Hill, 1968.

The function should be able to handle vector inputs.

 a. Fill in the following information: **Parts a and b should be done on this page.**

|  |  |
| --- | --- |
| **Problem Goal (brief):** |  |
| **Inputs:** | **Variable Name** | **Description** | **Units/ Dimensions** | **Input Source\*** |
|  |  |  | Command line |
|  |  |  | Command line |
| **Outputs:** | **Variable Name** | **Description** | **Units/ Dimensions** | **Output type\*** |
|  |  |  | Command line |
|  |  |  | Command line |

1. Complete a hand calculation on a test case to compare to your program results.
Use a = 4 cm and b = 5 cm as a test case.

1. Code with Comments: Copy your developed code to a word document. Use full comments on this program (including Problem ID, User introduction/help, Variable definition, Program Steps)

d. Scalar Execution: Run your developed code on the above test case. Copy the Command window call and result to the document with the code. Note: all I/O must be at the command line)

e. Vector Execution: Test the program with an input of two equal length vectors (one for the base radius and one for cone height). Add a copy the Command window call and result to your document.

1. Files & Arrays (15 points): When steel is stretched beyond its yield point it will undergo inelastic deformation and be permanently stretched. A tension test is conducted on a steel bar where the tension on the bar is slowly increased to past its yield point. The tension is then slowly decreased until there is no remaining tension on the bar. The results of this test are in the ASCII file ‘tension.csv’. In this file:
* Column 1 is the elongation of the bar in thousands of an inch,
* Column 2 is the tension corresponding to each elongation as the tension is increased,
* Column 3 is the tension corresponding to each elongation as the tension is reduced.

Both tension variables have units of pounds-force.

Prepare a script that will:

1. Interactively Input the File Name: The name of the data file must be inputted using interactive input in the script (i.e., the name of the file should not appear in the code).
2. Import the ASCII (.csv) file: Uses the input variable containing the file name to import named file into the MATLAB workspace
3. Create Vectors: create three vectors (one for each column of data) from the imputed structure
4. Displays the first five rows of data in the command Window.

Prepare a printout on a new page that includes: (1) the script,
 (2) the resulting command window call & output,

Script comments can & should be limited to: problem identification, your name, and
 identification of the program parts (i.e., parts a–d above).

1. Graphing (6 pts): Using data imported in Problem 2, Graph the two series for tension vs. elongation
	1. One series should be for increasing tension and the other for decreasing tension.
	2. Put both on one plot (i.e., one set of x-y axes).
	3. Plot the elongation on the x-axis (the two tensions on the y).
	4. Plot the entire data set (not simply the five rows displayed previously).
	5. Completely label the resulting graph and its two curves (including axis labels and legend).
	6. Make sure graph and fonts are easily read and follow standard graph format conventions.

Copy the commands used (either command window code or as a script) and the requested graph.

**If stuck** on successfully reading in the csv file: You can use the data file tension.mat (with a loss of the 10 points for reading the csv and addressing the file). At the command line type >> ***load tension.mat*** to load the three variables ***elongation***, ***TensionUp***, and ***TensionDown***. The file tension.mat must be in your current working directory.

1. Structured Programming – loop (21 points): Figure 2 shows a MATLAB function that converts a negative number to -1, a positive number to +1, and zero to zero. Enter this code, and save it to your MATLAB test folder. You do not have to copy the comments.

function y=sgn(x)

% Determines the sign of

% an input variable.

% y = sgn(x)

% input: x = any real number

% output: y = -1 for x < 0

% 1 for x > 0

% 0 for x = 0.

if x<0

 y=-1;

elseif x>0

 y=1;

else

 y=0;

end

1. Validate this function as is and copy the validation runs to a document. On the document with the validation runs, briefly explain your validation logic (i.e., how the test cases you choose test that the function is working correctly in all aspects.). You do not need to include a copy of the code at this stage.
2. Modify this function so that it will handle vector inputs and evaluate them element-by-element. ***Comments may be limited to your name and problem Identification***. Add a copy the modified code to the document with the part a results.

**Figure 2:** MATLAB function for Problem 3 that flags the sign of a number.

1. Execute the modified function for the following two test cases:
2. x1= [-3.5, 4, 0, -2.1, 3.6];
3. x2= [-1, -4, 3];

Add the results of these test cases the material from parts (a) and (b). Print out.

1. Structured Programming – Conditionals (21 points): A Microchip model MCP3204-CI/P integrated circuit (IC) is used to convert analog signals to digital signals. The price of these IC chips varies with the number purchased and is shown in the table below.

|  |  |
| --- | --- |
| Quantity Purchased | Price/IC |
| 1 – 34 | $ 4.01 |
| 35 – 99 | $ 3.09 |
| 100 – 499 | $ 2.40 |
| > 500 | Call for quote |

For example if the order is 50 chips select the $3.09 price and multiply by the number of chips: 3.09\*50 = $154.50. So the program should return $159.50.
N.B. the command >> ***format bank*** will result in numbers formatted with two decimal places.

If the number of chips is more than 499 the program should return NaN as the command line output and display a text message “Call for Quote” in the command window.

Table 1 outlines the basic setup for this problem. The I/O in your program must match this setup.

**Table 1:** Goal and I/O structure for the Chip Price Problem

|  |  |
| --- | --- |
| **Goal**  | This program is to return the cost of a requested number of circuits, including price variation for quantity.  |
| **Inputs** | **Variable Name** | **Description** | **Units/ Dimensions** | **Input Source\*** |
| qty | The number of items to be purchased | Count | Command line |
| **Outputs** | **Variable Name** | **Description** | **Units/ Dimensions** | **Output type\*** |
| Cost | Total cost of the purchase or NaN | $  | Command line |
|  | Display ‘Call for quote’ for large qty. | text | Display  |

a. In the space below show hand calculations on an appropriate number of test cases to compare to your program results. Plan enough cases to provide for adequate program validation.

1. Develop your code. Comments may be limited to ID., student name and date.
2. Execute your code for the test cases calculated in part a.

Copy the function code developed and the execution for the test cases to a document and print out.

1. Empirical Models (14 points) Each of the two problems below shows a different set of data plotted three ways. From the plots deduce the likely model that will fit this data.

Your answer for each data set should include

* 1. The name of the likely model linear, exponential or power (not the name of the plot).
	2. The corresponding equation (using m and b for the model parameters) for that model.
	You are not expected to determine the values of m or b.
	3. A brief explanation of why you choose that type of model.

**Write answers on this page in the space provided after each data set’s graphs.**

* 1. First Data Set



* 1. Second Data Set

