Module 1: Introduction to MATLAB

Engineering Disciplines: All

Each individual student will submit:

- 1 PDF file containing algorithms for problems 3, 4, 5, and 6
- 1 .m file containing all code for this module (Problems 1-6 plus Lab 3)

Prepare and submit these files in accordance with the ENGR 130 Style Guide and Assignment Submission Guide. Since this is an individual assignment, you can skip the group number in your file name.

Lab 1: MATLAB Vectors, Matrices, Input and Output

Materials

- Computer running MATLAB

Procedures

Complete the following questions in a single MATLAB script file, starting a new section for each question. Note that you will likely need to spend some time outside of class completing this work.

1. *Variable operators*

The total resistance (R_T) of three resistors (R_1, R_2, R_3) in parallel is given by:

$$R_T = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$$

Create variables for the three resistors, store values in each, and then calculate the combined resistance `RT` if `R1 = 80`, `R2 = 10`, and `R3 = 15`.

2. *Matrices*

a) Use the colon operator to store the integer values 1 through 5 in a vector variable called `a2`.

b) Create the following 3 x 2 matrix called `b2`:

2	3
4	9
4	11

c) Use vector `a2` to create the matrix `c2`:

$$\begin{matrix} 1 & 3 \\ 2 & 3 \\ 4 & 5 \end{matrix}$$

(You may break this into a few steps, but it can be done in one by using indices).

d) Add the matrices `c2` and `b2` together and store the result in `d2`.

e) Subtract each element in `b2` from the corresponding element in `c2` and store the result in `e2`.

f) Multiply each element in `b2` by the corresponding element in `c2` and store the result in `f2`.

g) Find the maximum of each column in `c2` and store the result in `g2`.

h) Find the maximum of each row in matrix `c2` and store the result in `h2`.

i) Find the maximum of the entire matrix `c2` and store the result in `i2`.

j) Sum of all elements of `c2` and store the result in `j2`.

3. User input and output

Before you write the MATLAB code for this question, create an algorithm that you will turn in as part of this assignment.

A power company charges $0.066 per kilowatt hour (kWh) for electricity. Prompt the user for the number of kWh (x) used in each month and tell them the cost (c) for that many kWh. Round x and c to two decimal places in your output. Use the following format, replacing `<x>` and `<c>` with the actual values:

```
Your charge for <x> kWh will be $<c>.
```

END OF LAB 1

Lab 2: Vectors and Loops

Materials

- Computer running MATLAB

Procedures

Complete the following questions in a single MATLAB script file, starting a new section for each question. Note that you will likely need to spend some time outside of class completing this work.

4. Vectors and Loops

Before you write the MATLAB code for this question, create an algorithm that you will turn in as part of this assignment.

A common statistical quantity is the standard deviation, which is a measure of how much variation there is in a data set. It can be found by:

$$s = \sqrt{\frac{\sum_{i=1}^{n} \quad (x_i - \underline{x})^2}{n-1}}$$

where n is the number of data points, x_i are the individual data points, and $\underline{x}$ is the average of the data.

Write code that does the following:

- Creates a vector with ten random integers between 1 and 100.
- Computes the average of those integers, without using any built-in functions.
- Computes the standard deviation of those integers, without using any built-in functions.
- Displays both results to one decimal place to the command window with some descriptive text.
- Displays the values obtained by the built-in functions to one decimal place to the command window with descriptive text.

5. Vector Indices and Loops

Before you write the MATLAB code for this question, create an algorithm that you will turn in as part of this assignment.

<u>Using a for loop</u>, write a MATLAB script that prompts the user to input a vector of any length and returns the vector with elements in reverse order. For example, if the user inputs [6 7 3 9], the script will return [9 3 7 6].

END OF LAB 2

Lab 3: Interfacing MATLAB with an Arduino

Materials

- Arduino Uno microcontroller
- Computer running MATLAB with Arduino add-on

Procedures

1. *Setting up your Arduino*

a. Connect the Arduino to your computer using the USB cable.

b. In the command window, type the command `arduinosetup` which will initialize a dialog box that will walk you through connecting the Arduino to MATLAB.

c. You are using an UNO type board. Select the given COM port, and keep all libraries checked to default. Click “Program” and wait for the software to upload to the Arduino. After the “Success” prompt, click “Next” to proceed.

d. Click “Test Connection” and then “Next”.

e. On the Hardware Setup Complete page, unclick “Show examples for support package”.

f. After the software is uploaded, click “Finish.” MATLAB and your Arduino will now be able to communicate with each other.

2. *LED circuit*

MATLAB can be used to control the Arduino and elements connected to it. For today, we are just going to control the Arduino’s on-board light-emitting diode (LED).

a. Start a new section in your script file with the usual header comments and “housekeeping” commands. After these, initialize your Arduino by using the command: `a = arduino();`. In the rest of your code, the variable `a` will refer to the Arduino.

b. The function `writeDigitalPin(a, <pin>, <state>)` turns one of the “pins,” or input/output ports, on the Arduino on or off. In this command, the `a` is the variable you assigned to represent your Arduino in the previous step of this procedure. The `<pin>` is the pin number that this command will manipulate; it will normally have a device connected to it, but to control the built-in LED use `‘D13’`. The `<state>` is either `0` or `1`, depending on whether you want to turn the LED off or on, respectively. In your script, after the command where you initialized the Arduino, use the `writeDigitalPin` command to turn the on-board LED on.

c. After you run the code in part b successfully, add another line of code to turn the LED off.

d. You probably noticed that the Arduino processes commands very quickly. The amount of time that the light remains on can be controlled by inserting a pause between turning the LED on and off. Use MATLAB's `help` to learn about the `pause` command and modify your code so that the LED will be on for 1 second before it turns off.

e. Modify your code so that using a loop and the `pause` command, the LED goes through 5 cycles of being on for .75 seconds and off for .5 seconds. Submit this final version of your code for this part of the module.

This is the end of the Arduino portion of this lab. You must also complete the MATLAB problem below.

6. Conditionals

Before you write the MATLAB code for this question, create an algorithm that you will turn in as part of this assignment.

The Beaufort Wind Scale is used to characterize the strength of winds. The scale uses integer values and goes from a force of 0, which is no wind, up to 12, which is a hurricane. Write code that asks the user for a number in the interval from 0 to 12, validates their input (makes sure it is in that interval) and displays the appropriate message. If the number is outside of the interval, throw an error using the `error` function. (You may assume that the input number is an integer). Display the number that was input along with the appropriate message:

0 → `There is no wind.`

1-6 → `There is a breeze.`

7-9 → `This is a gale.`

10-11 → `It is a storm.`

12 → `Hello, hurricane!`

END OF LAB 3