ECE 3723 - Electric Circuits II - Spring 2022
Project 1: Introduction to MATLAB

Due: 2/21/22 on Canvas

1 Introduction

MATLAB®is an extremely powerful programming environment used by engineers and scientists to perform a wide range of numerical computation, visualization, and programming. This tool is particularly valuable for electrical engineers as it is very commonly used for signal, image, and video processing, communications, and electromagnetics. The purpose of this project is introduce some of the basic functionality of MATLAB®. The purpose of this project is to look at some programming techniques that will be beneficial for future projects. This project will be focused on fundamental programming syntax, such as conditional statements and loops. These techniques, along with matrix manipulation, are vital for most programs that will be written in MATLAB®.

A suggestion for those who are not familiar/comfortable with loops: Try to think about how you would do the calculations by hand. Maybe do a test with a few points, then think about how to implement your hand-calculations with some sort of automation. Writing code is a way of thinking, not memorizing syntax. Many concepts span multiple programming languages, so it is good practice to develop this kind of thinking.

2 MATLAB®Basics

MATLAB®can be used as an advanced calculator where computations can be written as scripts and functions. So instead of just writing in single line calculations, entire design processes can be coded and optimized to realize complex components and systems. MATLAB®is a scripting environment that uses almost identical syntax as used in C and Java except without the need to create and manipulate specific handles and headers. In other words, MATLAB®can be used to program without having to deal with resource managing.

1. Create and run an m-file with a simple math calculation:
    - Open MATLAB®
    - Go to File->New->Script (or hit Ctrl+N, or click the New Script icon in the toolbar)
    - In the script window start by writing the commands ”clear all”, ”close all”, and ”clc”. Including this in scripts is a good practice to clean up the workspace before writing a script.
      Question: What do these functions do?
    - Save the file as task1.m (File->Save (Ctrl+S) or File->Save as).
    - Write ”y = 2*3 ^ 2*exp(2)”. Then run the script by selecting Run. The shortcut for running a script is F5.
      Question: What is the output in the command window?
    - Now add a semicolon behind the equation in the script.
      Question: What does the semicolon do to the output of the script?
    - Define a new variable x = 2 in the line above the previous equation. And then change the definition of y to y = x*3^x*exp(x). Now the variable x can be changed and y will change automatically when the script is run. This removes the need for manually changing the input to the equation.

2. Create an vector of numbers and plot an output of a function:

   - Use the .m file from above and create a vector called v1 with the command v1 = [1 2 3 4].
     Question: Is this a row or a column vector?
   - Now create v2 = [1; 2; 3; 4].
     Question: What do the semicolons do in this case?
   - Multiply v1 and v2 together.
     Question: Does the order matter? i.e. v1*v2 vs. v2*v1, explain why?
   - Other useful ways of creating these vectors would be v1 = 1:1:4 or v1 = linspace(1,4,4). These functions enable the creation of large vectors without having to manually populate them.
     Questions: What are the input parameters for these functions? How do they differ between the two methods?
   - Define a voltage v as v = linspace(0,12,121).
     Question: What is the purpose of choosing 121 in this equation (as opposed to any other number)?
   - Now define a 4.7 kΩ resistance with R =4.7e3 (note: the powers of ten are represented by e, not exp).
   - To find the current we need to solve Ohm's law: $i = v/R$. Create i and run the function.
     Question: What happens when you run the script?
   - MATLAB®does by default use matrix operations. To suppress those, a period is used in front of the operator. So redefine i = v./R and run it. To access specific element n in a vector in MATLAB®, the command is i(n).
     Question: What is the voltage and current when n = 89?
   - To plot a function the command "plot" can be used. Create a new line in the .m file with the command "plot(v,i)".
   - Type "help plot" to see more options for the plot function. Then add labels on the x and y axis. Add a legend with the equation. Save the plot and include in the report.

3 Conditional Statements

As with any other programming language, conditional statements are very useful when writing MATLAB®code. Most commonly used are if/elseif/else. For if statements we need to set up a conditional expression which will output either a true or a false.

- Create a new .m file.
- Define the variables x = 10 and y = 5.
- Enter the following expressions:

$$\begin{aligned} u &= x > y \\ v &= x < y \\ w &= x == y \\ ww &= x >= 2 * y. \end{aligned}$$

  Question: What are the outputs? And what is the meaning of the values?

- Look up the syntax for if statements. Then create an if statement that checks whether x is greater than y and sets y = x, else it should set x = y. **Show the code segment.**

4 For Loops

Loops are generally widely used in programming, and it can be essential in MATLAB®code as well. However, the MATLAB®engine is optimized for matrix manipulations, so whenever something can be implemented using matrix operations instead of loops, then the code will run faster. The most common loops are for loops and while loops. The for loop is often more widely used when the number of elements is well known and we want to use the loop counter for indexing. While loops more used a lot when infinite loops are desired as would be the case if a program is waiting on inputs from the users or some other type of interrupt. In this project (and class) we shall focus on the for loops.

- Create a new .m file or add to the one created above.
- Define the period $T = 1$ millisecond.
- Define the voltage amplitude $V_m = 1$ volt.
- Define the time vector t = linspace(0,T,1001).
  Question: What is the incremental value of t?
- Look up the syntax for for loops. Then use a for loop to create the vector v which will be based on

$$v(t) \quad = \quad V_m - \frac{V_m}{T}t.$$

  Be careful to use the loop counter as the vector t index and also use the same index for storing the value of v. Whenever you are plotting something in MATLAB (or any software), you should have a good idea of what it should look like. In other words, you can always plot by hand to get a comparison. This is a very simple sanity check to verify whether the code is outputting the desired plot(s).
- **Plot v vs t.**
- Now redefine the time vector t = linspace(-T,2*T,3001). And create the vector v using

$$v(t) = \begin{cases} V_m - \frac{V_m}{T}t & \text{for } 0 \leq t \leq T \\ \\ 0 & \text{elsewhere} \end{cases}$$

- **Plot v vs t over the range from -T to 2T.**
- Now modify the vector v so that is represents a periodic function with a period T over this time period (hint: it should look like a sawtooth).
  Plot v vs t over the range from -T to 2T again.

5 Deliverables

This project should be written up in a neatly organized report that includes the answers to all of the **the questions** and all plots that are generated. Every plot should be shown, and all the questions from the tasks should be answered. In future assignments (not this one), derivations of all equations are to be included in the report. All the code generated for this project must be included in an appendix (in small font, less than 12pt).