ECE 3723 - Electric Circuits II - Fall 2021
Project 2: Introduction to MATLAB part 2

Due: 9/21/21 on Canvas

Introduction

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. If you can code in any one programming language then you can code in any language.

1 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.**

2 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 matrices 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 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

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

  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

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

- **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.
  (hint: it should look like a sawtooth)
  (hint: think about how to extract the period. Functions related to rounding and/or modulus might be useful)
  (hint: remember the definition of a periodic function $v(t) = v(t + nT)$)
  Plot v vs t over the range from -T to 2T again.

3 Deliverables

This project should be written up in a neatly organized report that includes the results from sections 1 and 2, answers to all the questions and all plots that are generated. The format of the report needs to be formal, which includes a front page, introduction, main body, and conclusions. All the code generated for this project must be included in an appendix (in small font, less than 12pt).

4 Grading

- The report itself is worth 15 points, which will be given based on the structure of the report, grammar, clarity, and formatting.
- Answering the questions listed in the handout in the main body of the report is worth 75 points.
- Turning in the code in an appendix is worth 10 points.