Electric Circuits I
Laboratory 0: Introduction to MATLAB

To be completed independently during the first week of class, and submitted at the beginning of the first laboratory meeting

Objective:

- In this laboratory you will explore the basics of MATLAB, a tool for computation and visualization.
- You will do simple calculations and plot sinusoidal and exponential signals.

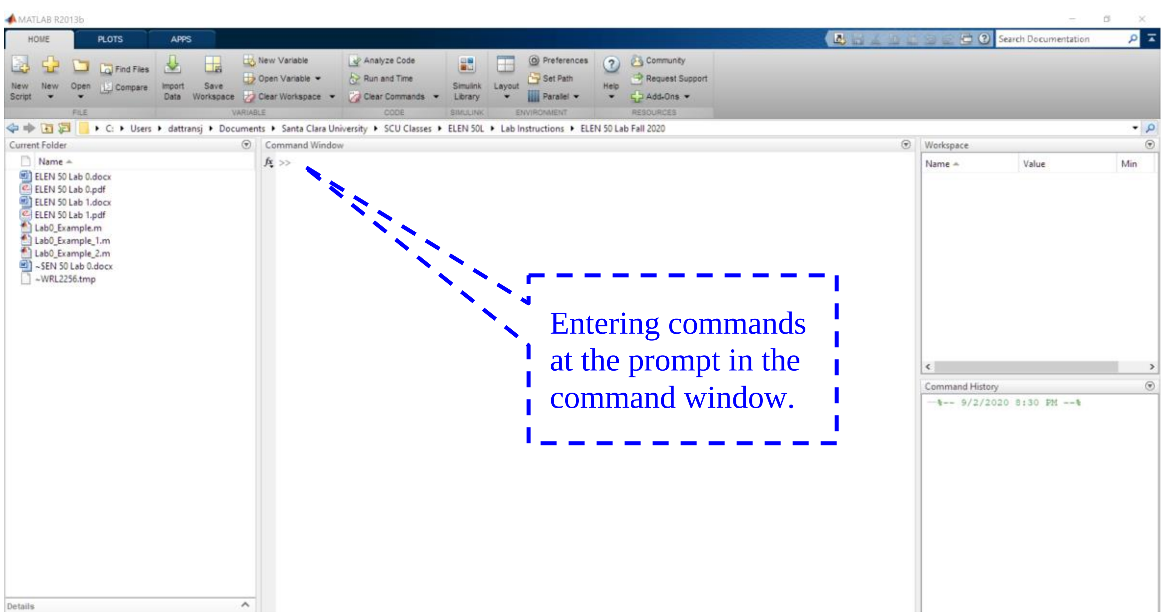
A. Background notes:

MATLAB is a versatile tool that can be used as a simple graphing calculator or a more sophisticated tool for computation and comprehensive graphical displays. In this class we will use MATLAB for homework assignments and laboratory work.

Note that MATLAB is licensed software. An alternative to MATLAB is Octave. Octave is free software that is available for various operating systems such as Linux, Windows, or MacOS. Octave uses the same commands as those in MATLAB. Check available version of Octave (https://www.gnu.org/software/octave).

B. Starting MATLAB:

In the design center log on to a system that supports MATLAB (the lab monitors should be able to direct you) and start MATLAB. You should see a screen that has a Command Window and possibly several other windows such as Command History and Workspace. In the drop-down menu under "Desktop" you can select windows and layouts.



The command window allows you to type in mathematical expressions and calculate results in a manner similar to a calculator. Under the “Help” dropdown menu, select “Using the Command window.” A new screen will provide a list of topics under Running Functions — Command Window and History.” Select “The Command Window” and read it.

Use “Help” when you have other questions about MATLAB.

C. Simple calculator operations:

In the Command Window, type the following after the >> prompt:

```
1.5*7
```

You should see the product printed as

```
ans =
    10.5000
```

and in the workspace window you should see that a variable named “ans” has been created which has a value of 10.5. Now type the following after the >> prompt:

```
53.6/7
```

You should see the result of the division on the screen and you should also see that the value of “ans” in the workspace has changed.

Parentheses can be used for more complicated computations. Try typing the following expression following the prompt.

```
(3*15+8*9)/11
```

You should get an answer of 10.6364.

Suppose you made an error typing the expression and the 9 should have been 19. It is not necessary to retype everything to get the correct result. In the command window, use the up arrow to reprint the previous line, which is the incorrect expression. You can now use the left and right arrows to move the cursor position and you can insert into or delete parts of the expression you typed previously. Use the left arrow to move the cursor to where the “1” should have been. Enter the “1” in the expression and hit “enter”. Now you should see an answer of 17.9091.

Work to be submitted:

- Using MATLAB, calculate the term grade point average of a student who took three courses and received an A- (worth 3.7) in a 4 unit course, a B (worth 3.0) in a 5 unit course, and a C+ (worth 2.3) in a 3 unit course.

- Copy the expression on the screen and the results, and paste that into the document for your lab submission.

D. Some MATLAB functions.

In this class we will be using sinusoidal and exponential functions.

1. **Exponential function**: For the exponential function e^t use the MATLAB function exp.
   After the prompt type:

   ```
   exp(0)
   ```

   You should see the response

   ```
   ans =
   ```

1

Type exp(-1) and then exp(-5). You should see answers of 0.3679 and 0.0067.

It is possible to compute several values of the exponential function at the same time by using a list of values for the function argument. The list, or vector of values, must be typed inside a set of square brackets. Type the following after the prompt:

```
exp([0, -1, -2, -3, -4, -5])
```

You should see the following result.

```
ans =
   1.0000  0.3679   0.1353   0.0498   0.0183  0.0067
```

In the workspace window you should see that "ans' is now the list of the 5 values you have just computed. Alternatively, you can create a vector of values and assign a variable name to it so that a variable name can be used instead of the explicit list of vector values. Type the following and note the result in the command window and the workspace window. The computed result should be the same as the previously computed list.

```
>> t=[0, -1, -2, -3, -4, -5];
>> exp(t)
```

2. **Cosine function**: Type the following to create a vector of angle values:

```
a = pi*[0:5]
```

You should see

```
a =
    0   3.1416  6.2832   9.4248   12.5664   15.7080
```

Note that MATLAB knows that π (pi) is 3.14159… , so you can just use the name pi instead of explicitly entering the value of π . Also note that the expression 0:5 creates a list in integers from 0 to 5, so the vector **a** has elements with values $[0\ \pi\ 2\pi\ 3\pi\ 4\pi\ 5\pi]$. In addition, note that when you define a vector without ending the statement with a semicolon, MATLAB lists all the values in the vector for you. The cosine function is defined for argument values in radians. Type the following:

```
>> cos(a)
```

Now type the following:

```
>> a=pi*0.5*[0:5];
>> cos(a)
```

The result should be six values of the cosine function evaluated from a = 0 to a = 2.5pi at intervals of pi/2.

Work to be submitted:

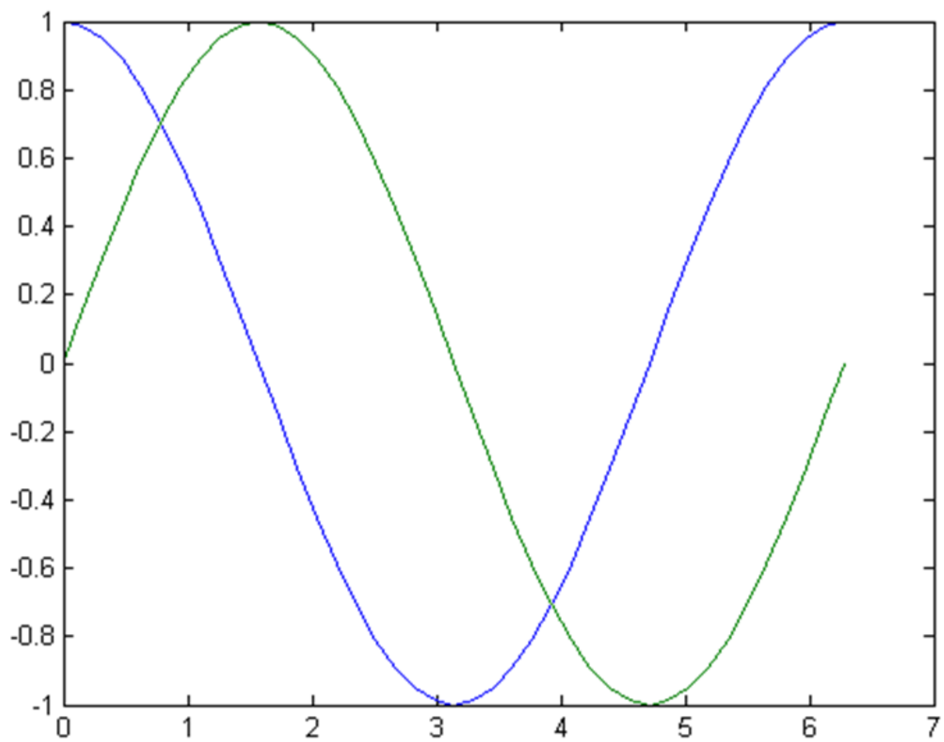
- Using MATLAB, calculate 20 values of the cosine function evenly distributed over one cycle or period of the function.

- Copy the expression on the screen and the results, and paste that into the document for your lab submission.

3. **Simple plots.**
MATLAB can also be used to create plots. When you ask for a plot, a figure window is created showing your plot. Type the following instructions. Don't forget the semicolon.

```
>> a1=pi*0.05*[0:40];
>> plot( a1,cos(a1), a1,sin(a1))
```

In the figure window you should see a plot like the one below of the cosine in blue and the sine in green for angle values from 0 to 2pi. The functions are evaluated at intervals of 0.05pi.



Under "file" in the figure window, select "save as" and then, for "save as type", select .png. Choose a name for the plot and save it. You can then insert this .png image file into a document such as a lab report. In the command window, type "help plot" after the >> prompt. You will see a description of many other plot options.

Work to be submitted:

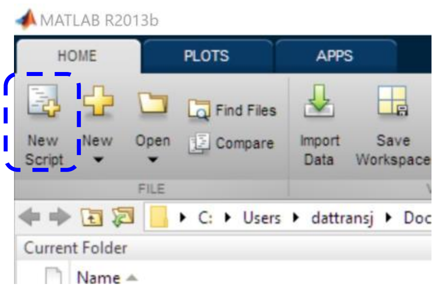
- Using MATLAB, plot the value of exp(-t) vs t over the range from 0 to 5 with the function evaluated at intervals of 0.1. Save the plot and insert the screen instruction and plot into the document for your lab submission.

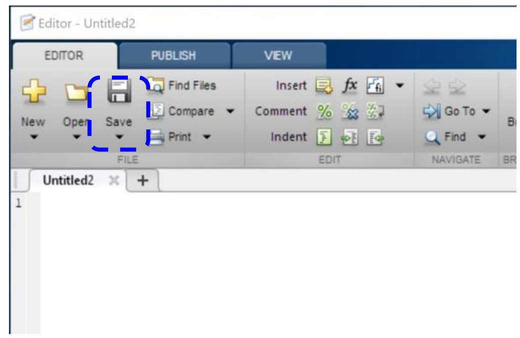
- Using MATLAB, plot the value of cos(a) vs a over the range from 0 to 6π with the function evaluated at intervals of 0.05π. Save the plot and insert the screen instruction and plot into the document for your lab submission.

E. MATLAB Command Scripts

Instead of entering command each time at the prompt >>, MATLAB (or Octave) offers a command script, in which commands are saved into a command script file (.m), to run many commands are sequentially run.

- Select "New Script" from Home tab and a new script editor will pop up:

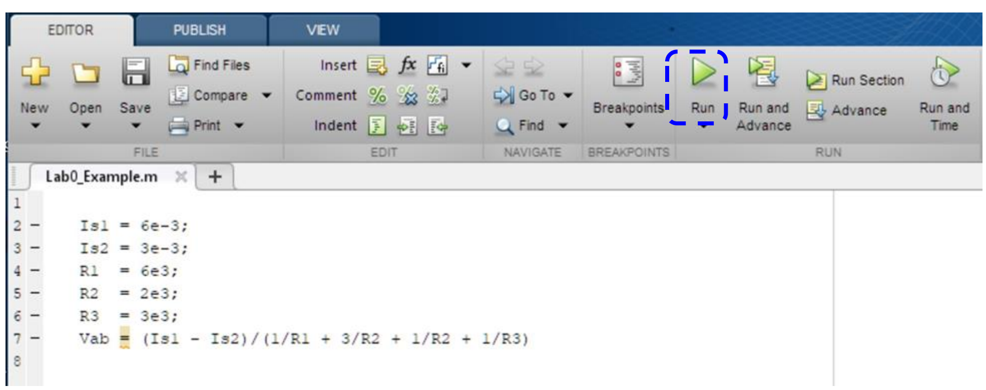




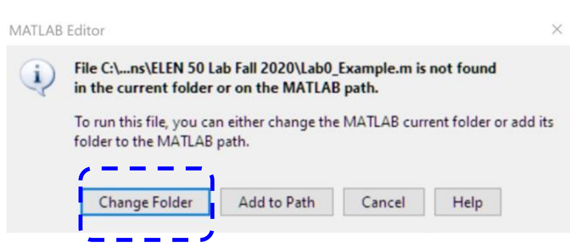
- Click on "Save" and save the command script file (e.g. Lab0_Example.m). Once the script file is saved, the title of the editor window will change to the name of the script file.
- Entering the following commands into the script file and save it.

```
Is1 = 6e-3;
Is2 = 3e-3;
R1  = 6e3;
R2  = 2e3;
R3  = 3e3;
Vab = (Is1 - Is2)/(1/R1 + 3/R2 + 1/R2 + 1/R3)
```

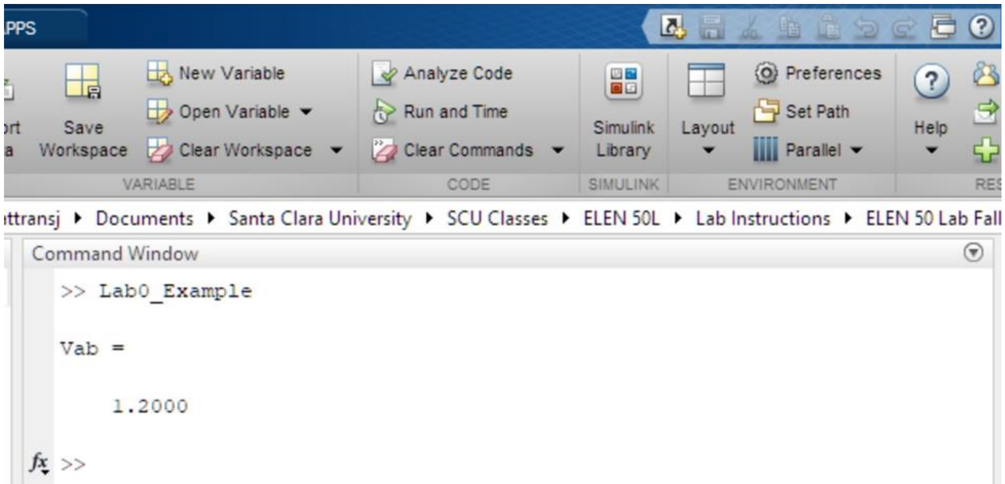
- To run the script file, click on "Run":



Note: *if the script is run the first time, a small window will pop up and select "Change Folder)*



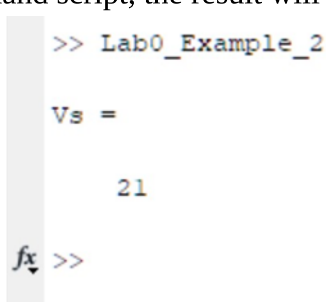
- If there is no error, the result will be displayed on the command window:



- Create another script file and enter the following commands:

```
Vs1 = 6;
Is  = 2e-3;
Vo  = 4;
R1  = 3e3;
R2  = 2e3;
R3  = 3e3;
R4  = 12e3;
R5  = 1e3;
 % ***************************************************************
% system of equations
% ***************************************************************
IR5 = Vo/R5;
IR1 = -Is + IR5;
Vad = -Vs1 + R1*IR1 + R2*IR5 + Vo;
IR3 = -Vad/R4 + Is - IR5;
Vs  = -R3*IR3 + Vad
```

- If there is no error in the command script, the result will be:



Work to be submitted:

- Save both command script files and submit them along with Lab report.

F. Laboratory Report:

For your laboratory report submit the document with the requested computations and plots. At the top of the document be sure to include your name, the date, and the laboratory section day and time.