Homework exercise: MATLAB introduction & numerical integration

Due: February 1st, by 10:00 PM

Objectives:

1. Practice writing MATLAB scripts and creating functions.
2. Investigate the dynamics of the integrate-and-fire model using Euler's method.

Deliverables:

- The code (and written answers) should be contained in a single MATLAB script file. Each exercise should be separated into a new code section (use '*%%*' to create a new code section). Written responses can be as comments ('*%*') or in comment blocks (see below for usage). Please indicate the exercise number and sub-letter before code/responses so it is easy to follow.
- The MATLAB function (the .m file) that you will write for exercise #3.
- These two MATLAB files need to be uploaded to Canvas.

Exercises:

1. In a MATLAB script, create two matrices, **A** and **B** (you pick the size) and fill them with your favorite numbers.
    a. Use the '*size*' command to determine the size of each matrix.
    b. Perform the following mathematical operations using these matrices: addition, subtraction, and element-wise multiplication. Use the '*disp*' command to display the result in the command window for each mathematical operation.

2. Lookup the MATLAB help page for the '*linspace*' command. In a MATLAB script, use the '*linspace*' command to create two vectors (you can pick the input arguments).
    a. For the first vector, only use two input arguments, *x1* and *x2*.
    b. For the second vector, use three input arguments, *x1*, *x2*, and *n*.
    c. In a block comment (use '*%{*' to open the comment and '*%}*' to close), briefly describe the similarities and differences between using the '*linspace*' command and the colon (':') operator to create vectors.

3. Create a MATLAB function named 'exponential_decay' to implement the following equation:

$$y(t) = e^{-kt}$$

The function should take two input arguments, '*time*', and '*kappa*', and return '*y_t*'. Implement the equation using element-wise operations so that a vector/matrix of values could be used as an input.

   a. In a MATLAB script, use either a '*for loop*' or vectorization to evaluate the function, where $\kappa = 2$ and $t = 0{:}\,0.1{:}\,3$.
   b. Plot the output and label the x-axis and y-axis.

4. Open '*Integrate_and_fire_code.m*' in MATLAB. First, read through the code to gain a general understanding of what the code is doing. With '*Current_source*' set to '*step*' and '*refractory_period*' set to '*false*', run the code and look at the plot.
   a. In a block comment, indicate how many spikes were observed and briefly describe what is being shown in the two subplots.
   b. Set '*refractory_period*' to '*true*' and run the code again. How many spikes were observed?
   c. Change '*Current_source*' to '*sine*' and re-run the code (tip: you can also use the F5 key to run your script). Indicate how many spikes were observed during each phase of the sine function.
   d. Change '*Current_source*' back to '*step*' ('*refractory_period*' should still be '*true*') and calculate the time interval (T) between spikes and the rate (units of spikes/second, calculated as 1/T). The variable '*spike_times*' might be helpful for calculating these values.
   e. Calculate the spiking rate (as calculated in **d**) for different input current values (1, 2, 4, 6, 8, & 10 nA) by editing line 45 (the multiplication on the '*ones*' function).
      i. NOTE: It may be easier to turn the '*Integrate_and_fire_code.m*' script into a function so you can easily pass in the input current values.
   f. Write code to plot spiking rate (on the y-axis) versus input current (on the x-axis) for the current values calculated in **e**.