Learning objectives and outcomes:

The problems sets are designed to learn basic aspects of neural computation, where the emphasis is on articulating these models and analyses into working MATLAB code. These problems are the perfect place to assimilate and build upon what you have learned in lecture and recitation.

To achieve these goals, problems will combine: building numerical simulations of neuronal models and processes, analyses and visualization of data coming from either experimental data or as output of simulations, and a bit of analytical and conceptual work.

More specifically, PSET 1 covers the material presented in lectures 1-3 and recitations 1&2. The problems will touch on diffusion in 1D, equivalent circuit model and Nernst potential.

The expected learning outcomes for this PSET are:

- Be acquainted and be comfortable using the MATLAB Integrated Development Environment (IDE) for scientific computing.
- Be able to load .mat files
- Be able to produce simple plots, and data visualizations.
- Be able to extract relevant parts of a numerical array.
- Be able to initialize variables and vectors.
- Be able to compute the mean and variance of an array, both row and column wise.
- Be able to implement for-loops for basic numerical integration of differential ordinary equations.
- Be able to write basic functions.
- Be able to present your results as a cohesive and structured report.

MATLAB functions you will need:

Loading files: `load.`

Plotting: `figure, plot, subplot, xlim, ylim, title, xlabel, ylabel, legend, hold on, hold off.`

Summary statistics: `var, mean.`

For more information on MATLAB functions and commands, check the provided cheat sheet or the more extensive MATLAB documentation.

Problem 1: Diffusion in 1D

Diffusion in 1D can be seen as a random walk. A random walk is the process by which randomly-moving objects wander away from where they started. The simplest random walk to understand is a 1-dimensional walk. Suppose that a particle is sitting at the center of a integer number line. Let's call this position, x = 0. Then, at each time point the particle takes a step of length 1 either to the left or to the right, with equal probability (1/2 in each direction). During the diffusion process, the particle keeps taking steps either left or right each time independently of what has happened at previous time points.

The file *randomwalk.mat* contains a matrix *X* that summarizes a random walk experiment for multiple particles. The code used to generate this data is in file *RandomWalk.m*. This code is for you to take a look at, you don't need to run it or modify it to answer the questions below.

The matrix *X* contains 500 rows and 1001 columns. In this matrix, each row represents the position of a given particle over time. These positions are in units of microns (1 micron = 10^{-6} m). Conversely, each column represents a time point in our experiment. Thus, the first column is the position of all particles at t = 0. We have tracked the particles positions in time increments of 1 millisecond. So, column 2 are the positions of all particles a t = 1 msec, column 3 at t = 2 msec and so on.

With this information answer the following questions:

1. **Look at matrix *X* and determine the initial positon for all particles.**
2. **What is the value of t for the last column of matrix *X*?**
3. **Plot the trajectory of particle 14 as a function of time.**
4. **In a single figure panel, plot the trajectory of all particles as a function of time. What do you see? Why each particle follows a different trajectory?**
5. **Plot the mean displacement of the particles as a function of time. Interpret this plot and discuss what is happening.**
6. **Plot the mean square displacement of the particles as a function of time. Is there any trend in this plot? Explain.**
7. **Use the plot in question 6 to estimate the diffusion coefficient of these particles. Briefly explain the methodology implemented to estimate this quantity.**

Problem 2: Equivalent circuit model

The equivalent circuit model of a neuron that we built in class is essentially a simple RC circuit with the addition of battery to capture the non-zero membrane resting potential. The lipids on the cell membrane provides electrical insulation between the intracellular fluid and the extracellular fluid. This creates a capacitor, with capacitance proportional to the cell's surface area (*A*). The cell membrane is not a perfect insulator. Pores in the membrane allow ions to move across it. A cell's total membrane conductance (G_m) represents how easily ions flow across the membrane. Note that the total conductance is also proportional to the cell's area and recall that conductance is the reciprocal of resistance.

For simplicity let's assume that our modeled neuron is a perfect sphere of radius 0.06 mm and resting membrane potential V_{rest} = -70 mV. The membrane of this neuron has a specific membrane capacitance (c_m) of 10 nF/mm² and specific membrane conductance (g_m) of 1 μS/mm².

This modeled neuron obeys the following differential equation:

$$V(t) + \tau_m \frac{dV(t)}{dt} = V_{rest} + R_m I_e, \qquad (2.1)$$

where $V(t)$ is the membrane potential at time t, V_{rest} is the resting membrane potential, R_m is the membrane resistance, C_m is the membrane capacitance, I_e is the current injected, and $\tau_m = R_m C_m$ is the membrane time constant.

The objective of this exercise is to understand how this cell responds to injected current. To achieve this objective, we need to solve equation 2.1 to find $V(t)$ for a given current waveform. We can do this either analytically or numerically. Luckily, the above differential equation has analytical solution for constant applied current:

$$V(t) = V_{rest} + R_m I_e + (V_0 - V_{rest} - R_m I_e)\exp\left(-\frac{t - t_0}{\tau_m}\right) \qquad (2.2)$$

where t_0 is a reference time when the initial condition is set to V_0.

This equation constitutes the core a numerical integration method to solve first order linear ordinary differential equations known as the exponential Euler scheme. This scheme is further discussed in the accompanying document *NumericalIntegration.pdf*.
By implementing this numerical scheme in MATLAB, we will be able to simulate the response of our modeled neuron to square pulses of injected current and answer the following questions:

1. **Draw the equivalent circuit model for this neuron, labeling intracellular space, extracellular space, membrane capacitance, and membrane resistance.**
2. **Calculate the total membrane conductance G_m and the total membrane resistance R_m for this modeled neuron.**
3. **Calculate the total membrane capacitance C_m for this modeled neuron.**
4. **What is the analytical value of the time constant for this modeled neuron?**
5. **Complete the provided MATLAB code (file *RCpassive.m*, check for addtional instructions on code comments) to implement the exponential Euler scheme. Make sure that:**
    a. **You set the cell's capacitance, resistance, and resting potential (using the values calculated and defined above).**
    b. **You set the initial condition V_0 (i.e. $V(t=0)$) to $V_0 = V_{rest}$ and the current injected I_e to 100 pA starting at 100ms and finishing at 200 ms.**

      c. **Your code updates *V(t)* at every time step using the exponential Euler method.**

6. **Simulate the cell for 1 second using a time-step of 0.1 ms. Plot the voltage as a function of time in one panel and the square current pulse in another.**
7. **From the previous plot estimate the time constant of the cell (Recall that as the voltage changes from V_0 to steady state, only 1/e of the total voltage change is left at time t = τ_m).**
8. **How does the value obtained in question 7 compares to the value obtained in question 4?**
9. **Show analytically (manipulate the equations, no MATLAB simulations) that equation 2.2 is indeed a solution to the RC differential equation 2.1.**
10. **Show analytically that equation 2.2 satisfies $V(t_0)= V_0$.**
11. **Show analytically that for equation 2.2, as $t \to \infty$, $V(t) \to V_{rest} + R_m I_e = V_\infty$ also known as the steady state of the system.**
12. **Suppose the cell is at V_{rest} at *t=0*, when a current step of I_e = 500 pA is turned on. How long will it take for the cell to reach -60 mV? What is the steady state voltage for this value of I_e? Compute these answers analytically.**
13. **Let's add an extra channel to our neuron. The extra channel has a resistance R_{extra} and is added in parallel to the other components. Does this new channel make the membrane time constant slower or faster? Show calculations to support your conclusion.**

Problem 3: Nernst potential

Ions will drift across the membrane down their concentration gradient from areas of high concentration to areas of low concentration. However, this creates a charge gradient in the opposite direction, since ions are charged. These two opposing forces balance at the Nernst potential where the net flow across the membrane for a given ion species is zero.
The Nernst potential is given by the following formula:

$$E = \frac{kT}{q} \ln\left(\frac{C_e}{C_i}\right)$$

where k is the Boltzmann constant with value 1.38×10^{-23} (J/K), T is the temperature in Kelvins (T(K) = T(C) + 273.15), and q is the charge of a monovalent ion with value 1.6×10^{-19} (C).
For this problem let's consider a neuron with the following intra- and extra-cellular ionic concentrations: $[K^+]_{in}$=186mM, $[K^+]_{out}$=4.8mM, $[Ca^{++}]_{in}$=50nM, $[Ca^{++}]_{out}$=1.5mM.

With this information answer the following questions:

1. **Complete the starter code provided in function *NersntPotential.m* so that it calculates the Nernst potential in mV, for ions of any arbitrary valence at a specified temperature in degrees Celsius.**

2. **Calculate the Nernst potentials of K^+ and of Ca^{++} for this cell at 37C, using the function you wrote in question 1.**

Problem 4: Ionic fluxes and Nernst potential

A spherical cell of radius 0.06 mm cell with a specific membrane capacitance of 10 nF/mm^2 is sitting in a bath a 37C. The only channel present in this cell is a Potassium conductance and the concentration of this is: 186 mM intracellularly Potassium and 4.8 mM extracellularly. (Note that these numbers are the same as in problem 2 and 3).

This cell starts at 0 mV with all the potassium channels closed. At time t_0 the channels are suddenly opened.

Given this information, answer the following questions

1. **Calculate the initial (before the channels open) number of potassium ions inside the cell.**
2. **What is the steady state membrane potential a long time after the channels open?**
3. **Calculate how much net charge must accumulate for the cell to reach steady state. Remember that the charge accumulated by a capacitor is $q=CV$.**
4. **Calculate the net change in number of potassium ions. In which direction is this net flux?**
5. **How much does this flux of K^+ ions change the cell's potassium concentration? Would this change affect the physiological properties of the neuron?**

9.40 Introduction to Neural Computation
Spring 2018