Sofia Lima

15-686 Neural Computation

Dr. Tai-Sing Lee

~~8~~ 9 February, 2023

Problem Set 1: Neuronal Model

Part 1: Understanding membrane potential (6 pts)

Inside:
$[Na^+] = 50\,mM$
$[K^+] = 500\,mM$
$[Cl^-] = 40\,mM$
$[A^-] = 400\,mM$

Outside:
$[Na^+] = 460\,mM$
$[K^+] = 20\,mM$
$[Cl^-] = 540\,mM$
$[A^-] = 400\,mM$

a) Write down the Nernst potential equation and compute the Nernst potential for each ion (K, Na, Cl) that will balance its respective chemical gradient.

equilibrium potential; universal gas constant; absolute temperature

$$E = \frac{RT}{zF} \ln\left(\frac{[C_{out}]}{[C_{in}]}\right)$$

electrical potential required to counter the chemical gradient; valance of ion; Faraday's constant (electrical charge per gram-equivalent of ion)

$$E = 58\,mV \log_{10}\left(\frac{[C_{out}]}{[C_{in}]}\right)$$

$$E_K = (58\,mV) \log_{10} \frac{20}{500} = -81\,mV$$

$$E_{Na} = (58\,mV) \log_{10} \frac{460}{50} = 55\,mV$$

$$E_{Cl} = (58\,mV) \log_{10} \frac{40}{540} = -66\,mV$$

b) What could Hodgkin and Huxley conclude about which ion channel in the membrane has undergone changes in its permeability during spike generation based on the observation that the peak of the spike is at 50 mv? Explain why.

Based on observing the spike peak at 50 mV, Hodgkin and Huxley could conclude that the sodium (Na^+) channel changed permeability during spike generation. Given the calculations in part a from the chemical concentrations of the ions, the amount of injected current needed to overcome the chemical gradient of Na^+ is around 50mV. The peak of the spike at 50 mV suggests that some voltage-dependent channel that was specific for Na^+ opened due to the injection current, and Na^+ flooded into the cell such that the membrane potential then shifted toward that of sodium.

c) Calculating the resting potential of the membrane should take into account not only the concentrations of the different ions but also their respective permeability. Goldman (1947) derived the following equation to calculate the membrane's resting potential at room temperature (25°C)

$$V_m = \frac{RT}{F}\ln\left(\frac{P_K[K^+]_{out} + P_{Na}[Na^+]_{out} + P_{Cl}[Cl^-]_{out}}{P_K[K^+]_{in} + P_{Na}[Na^+]_{in} + P_{Cl}[Cl^-]_{in}}\right)$$

where P_K, P_{Na}, P_{Cl} are the membrane permeability of the three different types of ion channels. Absolute permeability is hard to measure, but relative permeability can be used instead, and 2 the relative permeability among the three ion channels are determined to be 1 : 0.03 : 0.1 for *K*, *Na* and *Cl* respectively. Substitute all these numbers to compute the resting potential of the membrane at which the electrical potential balance the chemical gradients. This is the electrical potential of the neuron inside relative to outside when the neuron is at rest. At this resting state, What happen to the *K*, *Na* and *Cl* ions? Do they stop moving or continue to flow in or out of the cells?

$$V_m = (58\ mV)\log_{10}\left(\frac{20 + (.03)460 + (.1)40}{500 + (.03)50 + (.1)540}\right)$$

$$= -67\ mV$$

At this resting state, the ions continue to flow in and out of the cell via active and passive transport. There is a *dynamic* equilibrium such that their concentrations and electric potentials don't change on average.

d) In the lecture, we showed that the membrane dynamics with injected current I_o is modeled by the following resistor-capacitor (RC) circuit equation:

$$C\frac{dV_m}{dt} + \frac{V_m(t) - V_{rest}}{R} = I_0(t)$$

The solution for membrane potential V_m of this circuit in response to a step current I_o turned on at t = 0 is given by

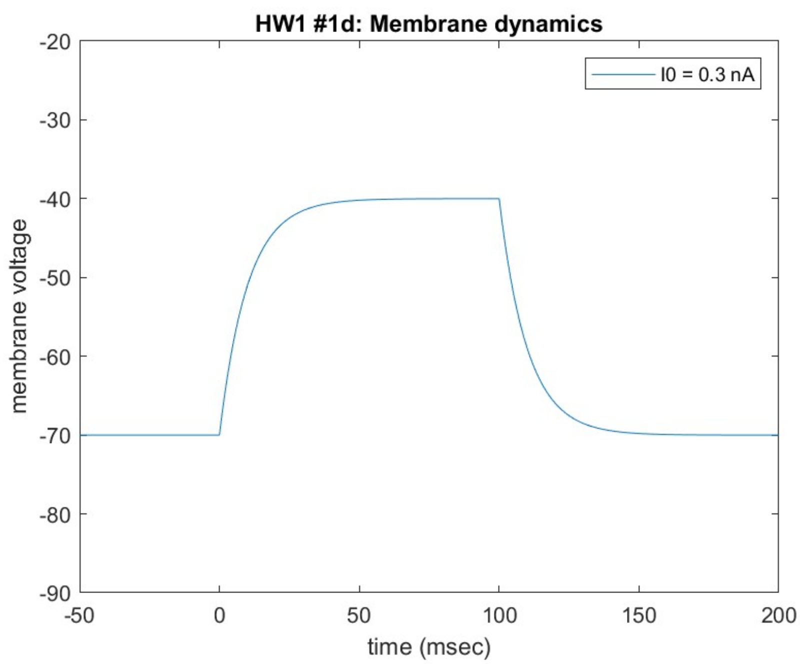
$$V_m(t) = RI_0\left(1 - e^{-\frac{t}{\tau}}\right) + V_{rest}$$

where $\tau = RC$ and R is the membrane resistance, and C is the membrane capacitance. Likewise, when the injected current is turned off at t_{off}, i.e. from I_o to zero, the membrane potential is given by,

$$V_m(t) = RI_0\left(e^{-\frac{t-t_{off}}{\tau}}\right) + V_{rest}$$

Write the Matlab code (part1d.m) to plot this solution of V_m on a graph, with y-axis showing V_m in mV, and x-axis showing time in msec (lasting from -50 ms to 200 msec), as was shown in the lecture slide. Let us say the injected current I_o level turns from 0 to 0.3 nA at $t = 0$, and stays at that level until $t = t_{off} = 100$ msec; $V_{rest} = -70$ mV, R = 100 M Ohm, C = 100 pF, $\tau = 10$ msec. See if you can reproduce the plot shown in the lecture slides. Adjust the parameter values if needed. The purpose of this question is to familiarize you with Matlab plotting functions. Submit the code with your the plot.

See code



e) While the solution to V_m can be analytical solved to obtain closed form solution as in (d), we can also obtain the solution by numerical integration. In this case, we can use the Euler method to obtain $V_m(t)$. The Euler method iteratively determines the value of V_m by determining its value at each successive time point by taking the previous time point value of V_m and adding to it the value of the derivative evaluated at the previous time multiplied by our time step.
Earlier, we were given that the model of the system is represented by the RC equation:

$$C\frac{dV_m}{dt} + \frac{V_m(t) - V_{rest}}{R} = I_0(t)$$

After some basic algebraic manipulation, we can obtain the desired form of the update equation suitable for the Euler method:

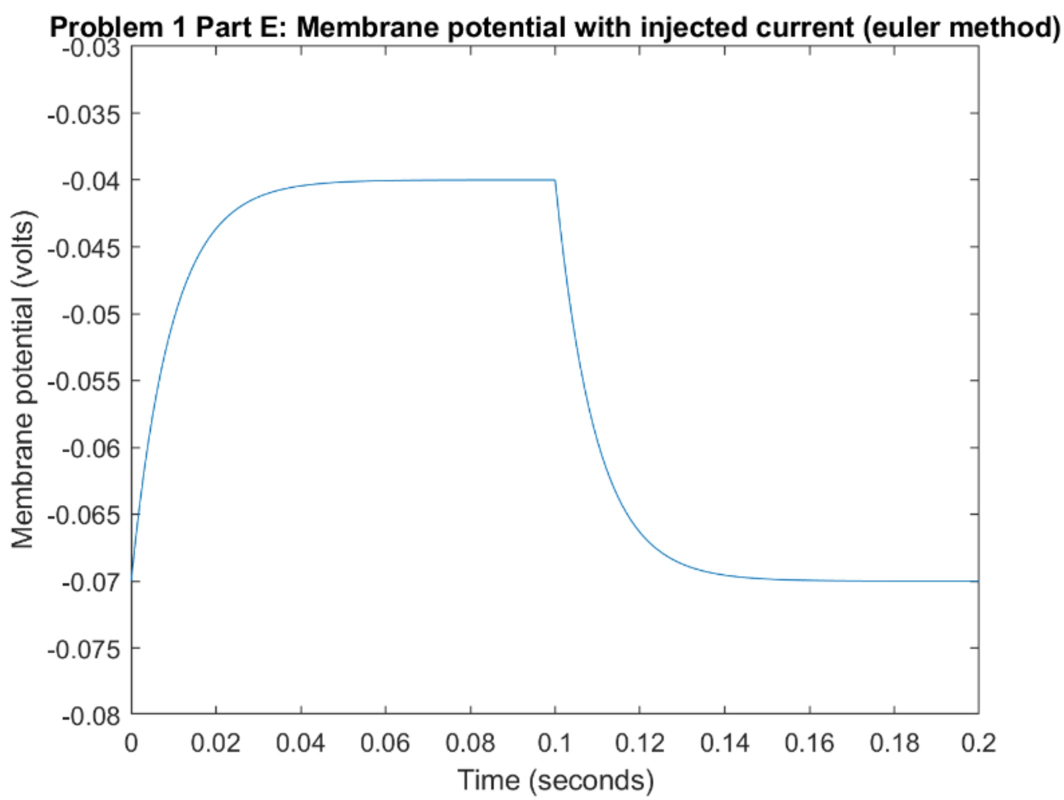
$$\frac{dV_m}{dt} = \frac{\left(I_0(t) - \frac{V_m(t) - V_{rest}}{R}\right)}{C}$$

Create a new Matlab script called `part1e.m` to implement the Euler method to simulate the evolution of V_m based on the differential equation in Part 1d, with the same parameter specifications. Plot the numerical integration result on a graph and compare it with the plot of the analytical result from 1d.

Since some of you might not have done any numerical integration, we provide you with a `part1e_provided.m` file to do this. Your task is to fill in only one line – the Euler update equation above, as commented in the code, and make the code run for 200 ms (changing one number in the constant lists). Note that dVm = Vm(next time step t) - Vm (current time step). and dt is the time step. Try several time step sizes (dt = 0.001, dt = 0.1 for example) to see what difference step size would make in this numerical simulation?

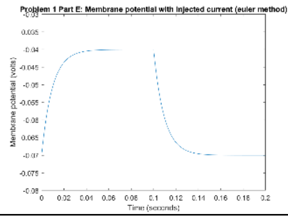
For more instructions, you can study the Matlab tutorial in Appendix E of Trappenberg, posted in Canvas which contains a section on numerical integration. The neuron simulation routines in that chapter, as used in Part 2 below, also provide an example on the simple Euler method.

The plot with the appropriate step size looks very similar to the plot generated by the analytical solution.

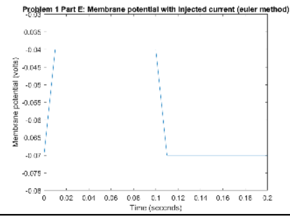


What difference step size would make in this numerical simulation? Different step sizes affect the accuracy of the solution for this first order accurate numerical integration scheme, forward Euler.

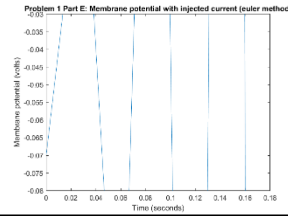
dt=0.001



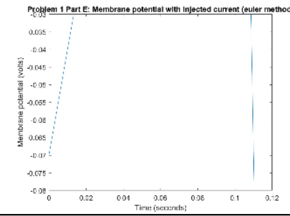
dt=0.01



dt=0.03



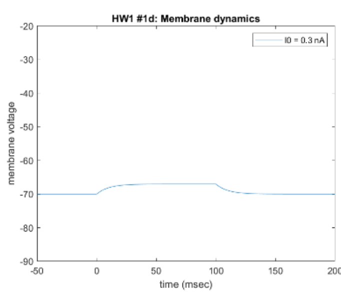
dt=0.1



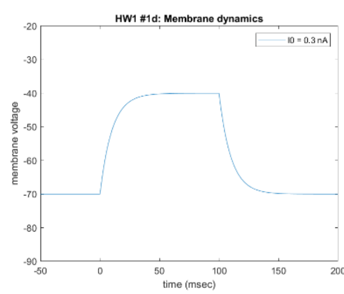
f) **Individual Exploration:** Your task is to explore how parameters resistor R and capacitor C of the model influence the behavior of the model to different kinds of input. The goal here is for you to gain better understanding of how resistors and capacitors work. You are encouraged to read up on these elements. Discuss your experimental observations and understanding of these circuit elements and provide some graphs to support your observation. Best 5 answers (most creative or thorough exploration) will be awarded up to 1 additional bonus point. Interesting run-up might get some partial bonus points.

The analytic solution of the system is not dependent on C. Therefore, when using this method, I found no difference when changing the value of C using this method.

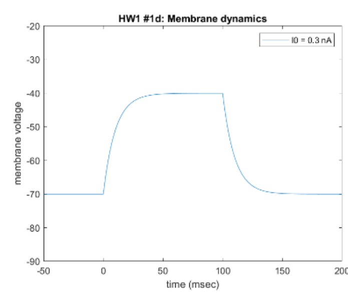
R=10
C=100



R=100
C=10

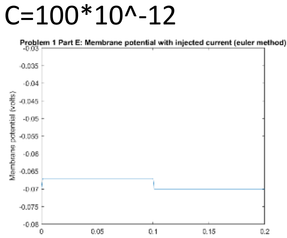


R=100
C=1

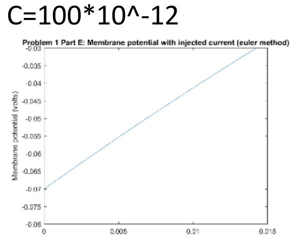


The forward Euler method was affected by changing either R or C. R seems to affect the magnitude of the response in the membrane potential.

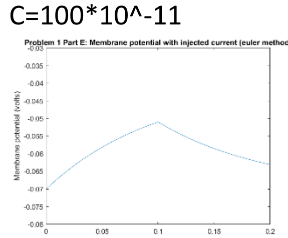
R=100*10^5
C=100*10^-12



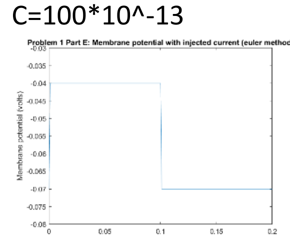
R=100*10^7
C=100*10^-12



R=100*10^6
C=100*10^-11



R=100*10^6
C=100*10^-13



Part 2: Numerical Integration of Spiking Neuron Models (4 pts)

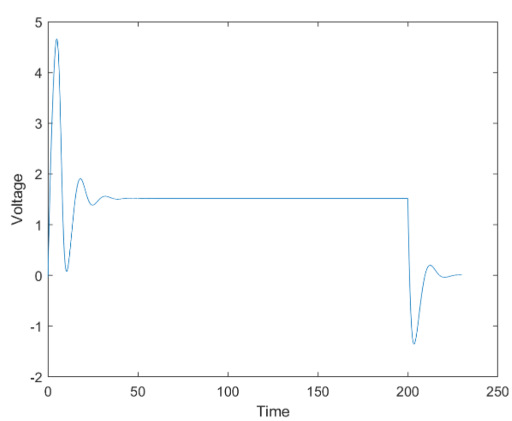
In the course assignment 1 folder, you should find a number of Matlab routines that simulate different models of spiking neurons. Here, we will consider the Hodgkin-Huxley's model (`hh.m`). Please refer to the description of these programs in Chapters 2 and 3, as well as Appendix E of Trappenberg's Fundamentals of Computational Neuroscience that we are providing in the Course Materials folder in Blackboard.

Study the code in `hh.m` and modify the code appropriately to perform the following experiment to obtain the so-called frequency-current (f-I) curve, which specifies the input-output relationship of the neuron.

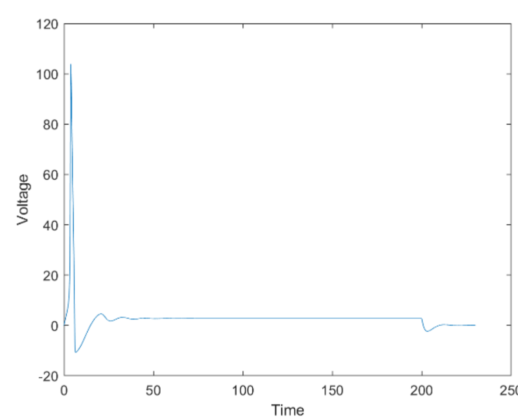
a) First, we will run a few simulations on the HH model with 200 ms stimulation of different external input current I_{ext} = 2, 4, 7, and 14. The provided hh.m set the input current to 2 at 0 ms and set it to 0 at 200 ms. In your report, provide 200-ms duration plots of spiking activities of neurons with stimulation current I_{ext} = 2, 4, 7, and 14. Note that the resting membrane voltage in this routine is not set to -70 mV, but to -10 mV, with the Nernst potentials adjusted accordingly. Don't worry about this. Your task is to substitute in different external stimulation current and observe what happen, and include the membrane potential plots under the stimulation of the four currents. What do you observe, noting anything you find interesting? Comment and explain your observations.

As the strength of the stimulus increases, the spike frequency in the response increases. With a very low injection current, there is no spike at all, but rather a mini EPSP. Interestingly, when the current turns off, it appears that the membrane potential approaches the Ernst potential for potassium. This could be due to the dynamics of the gates for each ion, as defined by *g*. Another observation from those responses that formed a spike train is that the height of the first spike is usually larger than the ones that follow. This could also follow the rules of the ion-specific dynamics such that the "slower" ion channels prevent such a peak membrane potential.

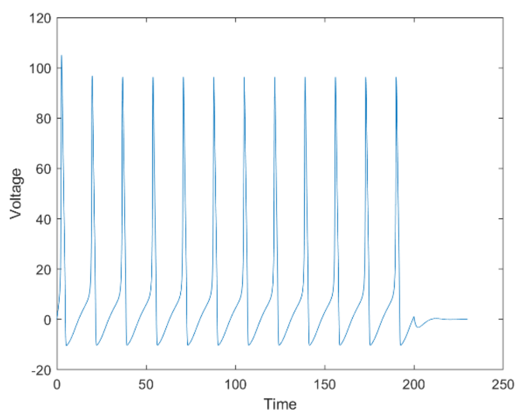
I=2



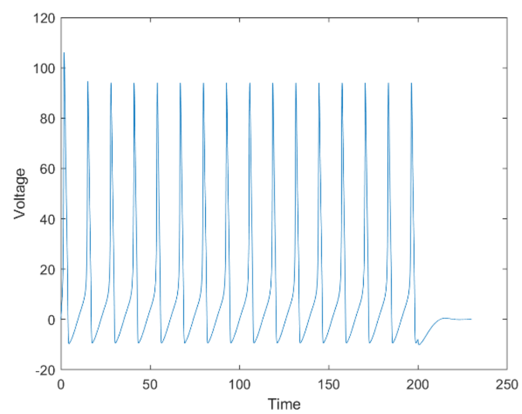
I=4



I=7

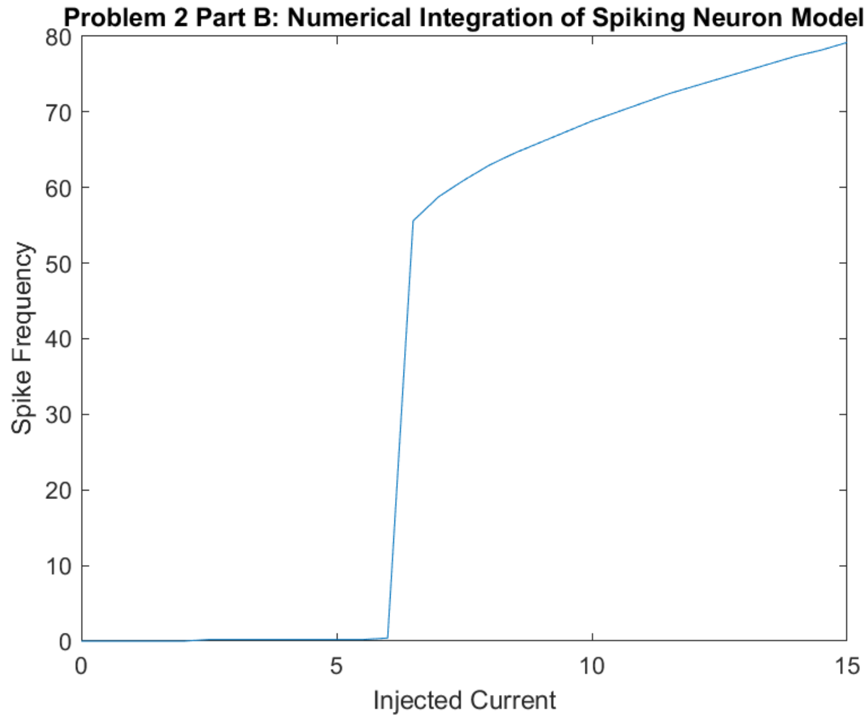


I=14



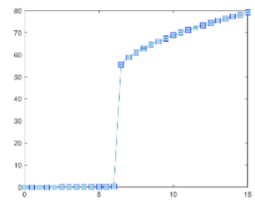
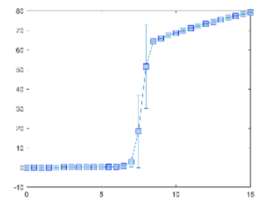
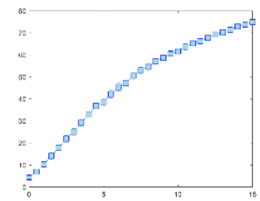
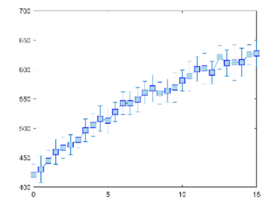
b) We provide you with a ‘window discriminator’ in spikeFrequency.m which is a Matlab function that takes a time series of membrane potential, a defined threshold, and the duration of the time series as input, and returns the spiking rate (# of spikes/second or Hz) computed from the time series. Set your threshold based on the observation on the spike height so that the window discriminator will count the number of spikes in the input time series. You should generate a long duration of membrane potential such as 5000 ms as input so that you can have a more accurate estimate of the spike rate. Test the neuron with I_{ext} value ranging from 0 to 15, at 0.5 increment (or finer if you wish), and plot the frequency-current (f-I) curve, with spiking rate on the y-axis and input current strength in the x-axis.
Include in your report this input-output or f-i curve. Describe at least two features of this f-I curve of the basic HH model. What could have contributed to these features? Are these features desirable or not desirable from a computational perspective? What are the implications on the computation that a neuron can perform? You are welcome to speculate, and you don’t have to answer all the questions, but should think hard about it, or read about it and provide some meaningful ideas. Submit your graphs, observations and discussions in your report.

Two features of this f-I curve are: (1) the curve serves as a threshold function such that if the injected current is above 6.5 then the neuron is “ON” and (2) the maximum possible firing rate seems to saturate around 80 spikes per second, however there is some range in the “ON” state. The implications on the computation that a neuron can perform are that they have a global signal that is more binary and may be able to fine-tune this signal, and potentially these neurons may be able to make subtle distinctions within the “ON” state.

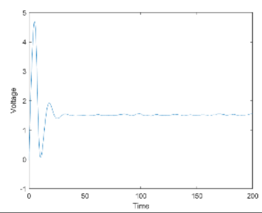
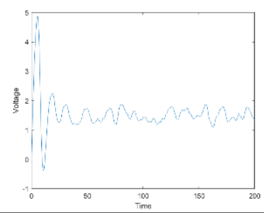
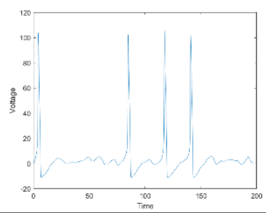
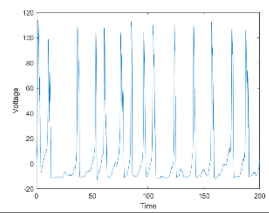
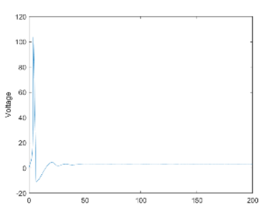
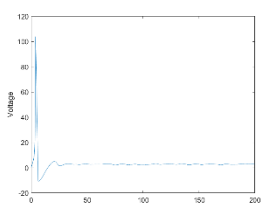
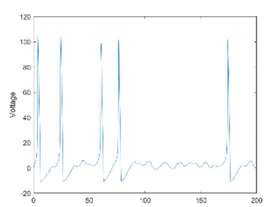
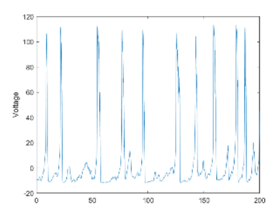
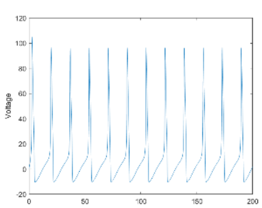
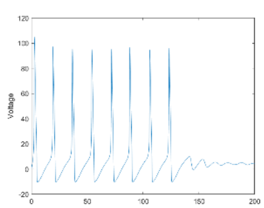
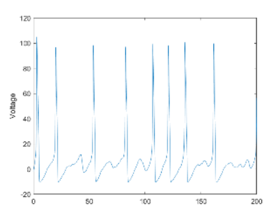


c) While the current we inject to the neuron is deterministic, in an animal, the neuron's input can come from many different sources, such as top-down feedback, cognitive state of the animals, other neurons' interaction, stochasticity in synaptic transmission. We call those unknown input noises and can model them using a Gaussian noise. There are at least two possible approaches you can add Gaussian white noise to the HH model neuron: (1) adding $\sigma * randn$ to the input or (2) directly to variable x in the integration step in the program. Here σ is a scalar specifying the standard deviation of the Gaussian white noise provided by *randn*. Is there a difference between these two approaches? Study the effect of noises on the spiking of the neurons and the f-I curve for a range of σ (the appropriate range might be different depending on how you add the noises, e.g. you might need to use smaller value if added to x than if you add to the input, or HH model might crash). In generating the f-I curve, for each particular fixed input current, you may want to repeat the simulations 20 times and then compute the average so that your f-I curve would be more smooth. Plot the new spike train for 200 msec as well as the new f-I curve with a number of σ you choose. How does the noise change the f-I curve? What is the significance or implications of such change? Is this desirable or not desirable? How does the change vary with the noise level? Treat this as a scientific experiment to discover of the impact of noises on the "effective activation function" of the neuron. Back up your claims and conclusions with graphs and observations.

The plots represent the f-I curve for a range of sigma (i.e. the amount of noise from the Gaussian random variable added directly to x). The lines represent the average across 20 runs, and the errorbars represent the standard deviations.

sigma=0.00001 	sigma=0.0001 	sigma=0.001 	sigma=0.01 

The plots below represent spike trains from the last round of simulation at three inputs. The noise has a drastic effect on the responses. As the noise increases (left to right columns), the firing rate becomes more sporadic such that the frequency of spikes is less regular. In the cases of low injection current, some noise levels will even causes spikes when there are none in the noiseless case. The ability to tolerate noise is an important trait for neurons to control the firing rate.

I=2 sigma=0.00001 	I=2 sigma=0.0001 	I=2 sigma=0.001 	I=2 sigma=0.01 
I=4 sigma=0.00001 	I=4 sigma=0.0001 	I=4 sigma=0.001 	I=4 sigma=0.01 
I=7 sigma=0.00001 	I=7 sigma=0.0001 	I=7 sigma=0.001 	I=7 sigma=0.01