Assignment 4

To compile mod files (`*.mod`), run `nrnivmodl` in the directory where the mod files are. Also run your simulations in this directory.

1. Simplifying the network simulation

Run `demo9.py` and `demo9_1.py` and compare the results.

1. Which one ran faster?
2. Read the two programs and explain what are main differences between them.
3. In which conditions can you use simplifications in `demo9_1.py`? Discuss in which situations the simplication scheme cannot be used.

2. Membrane potential fluctuation by random synaptic inputs

`hw4_2.py` simulates a single passive neuron that is bombarded by random excitatory synaptic inputs. A current clamp is also embedded to probe the membrane property. The simulation will repeat 100 times and save the result in `voltages_hw4_2.csv`. The simulation result will show an initial effect up to t<100 ms, and please use the data beyond this range.

1. By using the current clamp, measure the *effective* membrane conductance g_L' from the averaged membrane potential. Try different number of synaptic inputs N_{exc} and plot how g_L' depends on N_{exc}. Also, try different firing rate f_{exc} as well. How does g_L' compare with $g_L + g_{exc} f_{exc} N_{exc} \tau_{exc}$? Here g_L = 0.5 nS is the original membrane conductance. τ_{exc} = 1.25 ms is a time scale for synapse deactivation, and g_{exc} = 1 nS is the synaptic conductance.
2. Plot how the *measured* membrane reversal potential E_L changes with N_{exc} and f_{exc}. How does it compare with $g_L E_L/g_L'$?
3. Plot how the *variance* of the membrane potential (σ_V^2) scales with N_{exc}, f_{exc}, and g_{exc}. In particular, find m, n, and k in the scaling relation $\sigma_V^2 \sim N_{exc}^m f_{exc}^n g_{exc}^k$.
4. Is the membrane potential fluctuation sufficiently Gaussian? If so, assuming that our neuron fires when the membrane potential reaches a fixed threshold V=V_{th} (e.g. V_{th} = -50 mV), discuss how the firing rate of the neuron will be affected by N_{exc}, f_{exc}, g_{exc}, etc.?

3. Vogels-Abbott network model

`hw4_3.py` is a simulation of 1000 randomly connected neurons. 20% of the neurons are inhibitory while the rest is excitatory. Each neuron gets inputs from 10% of the neurons in each of the excitatory and inhibitory pool. The network is driven by external stimuli for 7 ms at the simulation start, but, after that, the simulation runs without any external stimulus. The data will be save to `spikes_hw4_3.csv`.

The default value for the excitatory conductance (`gexc`) is g_{exc} = 1.5 nS and the inhibitory conductnace (`ginh`) is $g_{inh} = r\, g_{exc}$ where r = 6.2.

1. Run the simulation twice and compare the results. Is firing of neurons deterministic or random, given the same initial condition? Make a very small change in one of the parameters (g_{exc}, r) and observe how the firing pattern changes.
2. By trying different r values, find a maximal r that can generate stable and asynchronous firing in the network for 200 ms. With the maximal r, what is the average firing rate?
3. Try several different g_{exc} from 2 nS up to 30 nS, and find the maximal r for asynchronous firing and average firing rate. What pattern do you see? Explain your results.
4. Compare the firing pattern between the small g_{exc} (~ a few nS) and large g_{exc} (> 10 nS) case. Discuss why the differences arise.
5. (Optional) Discuss how your results compare with [Ostojic, S. (2013) Nat Neurosci *17*, 594-600] (Figure 1).