Neuroscience Computation Final Homework

# Due by Oct. 2nd at 5 pm

The final project in Computation for Neuroscience is designed to be a collaborative programming project where the work is split up between the people in the course but in the end a single, hopefully functional program will be produced. We will go through the stages of completing this project at the next class on Tuesday, but I wanted to give you the outline now so you can being work on it if possible.

The project is being set up to run using a central version management program git under a repository called github. I have posted a description of many of the common git commands and we will go over this in detail on Tuesday. A preliminary piece of code has been created that the class will need to work on to create the final product. The idea is to have the class split up the work so that the programming can be done more efficiently. I have described below the different issues that need to be corrected in the code. Your job is to work with the provided github repository. It is probably easiest to clone it into a new repository that is created and managed by someone in the class, then submitting the final product at the end back to the repository that I created. Each person should be handed a job and should branch the main repository onto their computer for their work.

# I have created a repository on github called: [NeuroProgrammingBCM](https://github.com/NeuroProgrammingBCM)/[MysteryProg](https://github.com/NeuroProgrammingBCM/MysteryProg)

I have created a preliminary code to simulate ion channels in some manner. Download this program onto your machine or clone it using git, and try running it to determine what it does. I will suggest to you that it simulates channels in some way, but is not sufficient for our purposes since it doesn’t include voltage-dependence, and driving force and is otherwise poorly documented. Also, this program runs in a debug and non-debug mode so be sure to test both. You job is to:

1. Document the different modules in this code and their dependencies. Also, write unit tests for each of these modules.
2. Modify the code to allow the production of currents rather than what the output is at this point. Consider the parameters that might need to be adjusted to provide user flexibility in creating differ currents
3. This code was written in a module manner. Write a version that could be used to create multiple channel objects for testing the ability of a test algorithm to separate different channel events based on amplitude.
4. Use the code to simulate what happens if multiple channel objects are allowed to run at one time at different potentials. This should be done to run as a voltage-clamp type experiment.

It is important to note that what we want in the end is code that will do what this code is able to do, but it will be better documented and tested. What you have been given is like a prototype. At certain points you may find it easier to rewrite something rather than work with what you have been given. But, if you rewrite it you need to do something to show that your solution is superior to what you started with.