Homework Assignment 2

*** Due Thursday 10/2/14 by noon on Sakai under HW2. Upload your submission as a Word document. ***

Please reaffirm the Duke Community Standard at the top of your assignment:
"I have adhered to the Duke Community Standard in completing this assignment." [Electronic Student Signature]

Please explain your methods, show all work, explain your results, include units, and label your axes. Upload your .hoc files with clear filenames. Also provide code snippets in your Word document as indicated in the instructions.

This homework explores various factors that affect threshold, and it will serve as an introduction to the NEURON simulation environment (neuron.yale.edu).

Part 1 – Binary search algorithm in NEURON

Let's implement a binary search algorithm that can be used later to find thresholds. To find threshold, you'd stimulate your neuron using an amplitude Istim1 and check if there's an action potential: if there is, you are at or above threshold, but if there isn't, you are below threshold. You'd then adjust your stimulus amplitude accordingly, and try again. Rinse and repeat.

For now, let's just "search" for the value of some float A, rather than threshold. Let's say A=4.3, and you want to write an algorithm that determines the value of A within a resolution of 0.1, knowing only that A ϵ [0, 10]. You could do this with a brute force approach…:

1. myguess = 0
2. myguess >= A?
   a. Yes? Then, myguess ~= A.
   b. No? Then, myguess = myguess+0.1 and repeat step 2.

But instead, let's use a smarter approach: binary search algorithm.

1.1 Google how a basic binary search algorithm works. Briefly explain the algorithm by writing pseudo-code. Don't forget to include an exit condition!

1.2 Implement your binary search algorithm in NEURON:

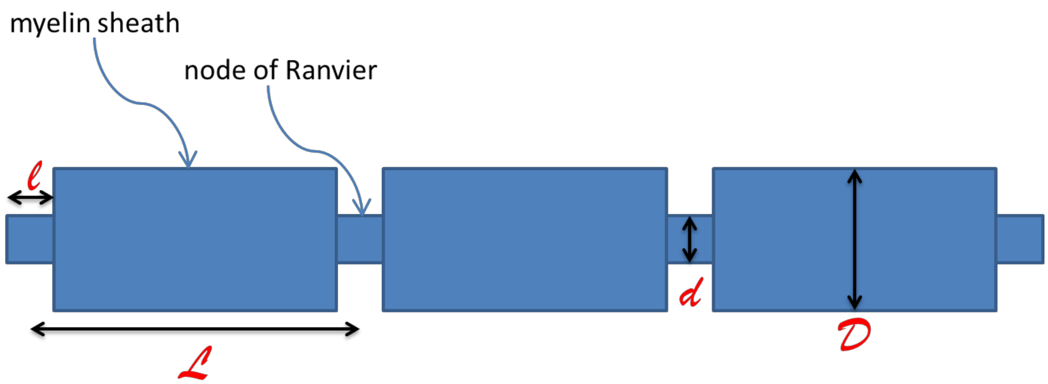
- Define your parameters.
- Implement your binary search algorithm. Some useful NEURON functions…
  - while(1)
  - if
  - else if
  - break
- Print your result to the neuron shell using NEURON's "print".

Provide your code in your Word report and a screenshot of the neuron.sh window showing your result.

Part 2 – Model axon in NEURON

Let's model a simple myelinated axon in NEURON. Our axon is an 8um fiber (i.e. D=8um). As per typical geometrical relationships in mammalian axons, d=0.7D, L=100D, and l=1um. The fiber is immersed in an infinite homogeneous conducting medium with ρ_e=500Ω–cm. Assume that the myelin is perfectly insulating. The axoplastic resistivity is 200Ω–cm and the specific membrane capacitance is 2uF/cm^2.

- Show your calculations and results for d and L.



- Start with CodingExample.hoc provided in the NEURON tutorial folder on Sakai.
  - Geometrical parameters
    - Change the node_diam, node_length and myelin_length appropriately.
  - Electrical parameters
    - Change the node_cm and rhoa appropriately.
    - Based upon Vm(t) with Istim=0, what is Vrest (i.e. v_init)?
    - Add a variable called "ap_thresh".
  - Other parameters
    - Change dt to 1us so that you have multiple time points even for short stimuli.
    - Navigate to where you installed NEURON. Go to nrn73\src\hh.mod. In this file, look at the comment block at the top. Change "celsius" in your hoc code accordingly
  - Axon
    - Briefly explain the equation for Ra.
    - Replace the passive mechanism ("pas") with NEURON's built in Hodgkin-Huxley mechanism ("hh"). Note that the hh mechanism already has all its electrical parameters defined.
  - Stimulus
    - Change the stimulus location so it's at the center of the axon. (Hint: Try NEURON's "int" function.)
    - Add a recording vector to record stim.i.
  - Instrumentation
    - Add an APCount object tied to node #20. Note that to do this more thoroughly, you'd want to check for action potentials at all nodes. Why? Well, what if you check for an action potential at the end of your axon, and it comes back zero simply because the action

potential wasn't given enough time to propagate. But let's just ensure that we select a sufficiently long tmax.

  - Plots
    - Modify the Vm(t) graph to plot the time course for the 45th node; remember NEURON starts counting from zero. Adjust the axis limits as needed.
    - Add a second graph to plot your stimulus vector.

Ok, we're ready to run some simulations!

- For PW=mydur=0.1ms, select myamp such that it's superthreshold.
- Provide labelled screenshots of Vm(t) and Istim(t). Make sure you can see the full action potential.
- Provide your code in your Word report.

Part 3 – Intracellular threshold with PW=0.1ms

Let's combine Parts 1 and 2.

- Start with your Part 2 code.
- Add your threshold parameters to proc params().
- Add a new procedure containing your binary search algorithm. Also make the two following additions to your algorithm:
  - Before starting your search, make sure you check that your chosen upper bound is sufficiently large to elicit an action potential. Hint: Check myapc.n.
  - You also need to do one other check. When you're done searching (i.e. within your chosen resolution), you need to make sure that your final choice is superthreshold. If not, perhaps use your last superthreshold guess.

In your Word report, provide your threshold procedure and a screenshot of neuron.sh showing your threshold for the axon using PW=mydur=0.1ms.

Part 4 – Extracellular threshold with PW=0.1ms

Modify your code from Part 3 to use extracellular stimulation (ρ_e=1000Ω-cm) instead of intracellular stimulation. See the two slides on extracellular stimulation in the NEURON tutorial slides on Sakai for help. Note that in NEURON, 3.1415… is "PI" (uppercase). Also note that we now want to use a cathodic stimulus.

Find the threshold for a 0.1ms pulse with the electrode placed 1mm from the fiber (perpendicular distance) over the middle node of Ranvier. Provide a screenshot of neuron.sh showing your threshold.

Part 5 – Threshold-fiber diameter relationship

Ok, let's use our code to get some data! Plot extracellular threshold with PW=0.1ms and electrode-fiber distance=1mm as a function of fiber diameter for D=1 to 15um with Δ2um. Recall that d=0.7*D and L=100*D, as shown for Part 2. You can maintain the node length at 1um.

NOTE: If you choose to automate looping through the diameters in NEURON rather than doing it manually, make sure you reset your e_extracellular values to zero for all nodes and reset your upper and lower bounds (for your binary search algorithm) at the start of each simulation. But feel free to obtain your values manually (one diameter at a time) rather than coding a batch run.

Provide your results in a table and a labelled graph.

Part 6 – Threshold-distance relationship

Now let's examine the effect of electrode-fiber distance. Go back to your 8um axon and PW=0.1ms. Obtain threshold for the following 7 electrode-fiber distance: {0.1, 0.2, 0.5, 1, 2, 5, 10}mm. Once again, provide your results in a table and a labelled graph. Use a logarithmic scale on your x axis.

Hint: Whether you're finding thresholds manually (one electrode-fiber distance at a time) or with a batch function, consider that you should lower the initial upper bound for your binary search algorithm when your electrode is closer. I suggest finding threshold for the furthest position, then using that as your upper bound for your second further position, and so on.

Part 7 – Application

Find a paper where they're examining the threshold-diameter relationship or the threshold-distance relationship for a specific application. Briefly and clearly summarize this paper (max of 250 words) by providing the following information:

- Full citation
- Threshold-distance or threshold-diameter?
- Application and reason for application