Lab 3 and 4 Report

Introduction

In this lab electrostatic models were created using COMSOL and MatLab to analyze the recording and stimulation of neurons. Electrostatic models are a good way to understand how voltages can be distributed in space from electrodes without the need for in person experiments or human testing. They can use real resistance, size, and current values to estimate how voltage is spread. A model was created of the head and the effects of putting an electrode at the center were observed. A deep brain simulation model was then created and analyzed how it can cause neuron activation. Data from COMSOL was also exported to estimate what a neural recording would look like from one neuron and with multiple neurons containing noise and a dead zone. Finally, the minimum extracellular current necessary to stimulate an axon was determined.

Methods

In COMSOL, an Electric Currents environment was created using a Stationary study. An Electrostatic assumption was made that says the materials used are primarily resistive in the frequencies focused on. A Stationary study means that everything is simulated at steady state at a single point in time. The electrode was modeled as a point source when in actuality it would have dimensions. These were all safe assumptions that should not drastically change results.

Initially, a testing model was created with a 1m radius, electrical conductivity of 0.33 S/m, and a point source with 1 A at the center to become familiarized with the software and ensure it is working as expected. The next step was to build a model of the head similar to what was created in the Moffit and McIntyre 2005 paper. Spheres with radii of 10, 9, 8.5 and 8 cm were generated for the scalp, skull, cerebrospinal fluid, and brain respectively. The boolean difference function was used to make shells for the scalp, skull, and cerebrospinal fluid so that only the thickness necessary was created. Resistance values of 230, 16,000, 56, and 300 Ω cm were given to the scalp, skull, cerebrospinal fluid, and brain respectively. A 1A point current at the origin was created and all external surfaces of the scalp were grounded. An adaptive mesh refinement was created with a maximum number of 3 adaptations and a rebuild mesh adaptation. The data was exported and plots of voltage versus distance to stimulus were created. One final model was created in COMSOL using the same layer of concentric spheres to model the head. The point current was set to 5 mA, a 1.25mA point current sink 3mm above the center was added, and a 3.75 mA current sink 3 mm below the center was added. The simulation was run, data was exported, and the second spatial derivative going away from the active contact and perpendicular to the probe was created.

As described there were three levels of complexity in the COMSOL models. The first consisted of just one sphere and one point source and is the simplest model that can be created to demonstrate voltage distribution. The second model introduced layers of the brain and observed how different resistances surrounding the electrode may affect the voltages. The third model introduced current sinks and realistic current amplitudes which more accurately represent what real deep brain stimulation may look like.

Models of voltage traces from simulated neuron recordings were then created. In these models equation 1 was used to find the extracellular voltage. In dipoles this equation can be expanded to equation 2. However, this paper estimates that the electrodes are close to the neurons in which a monopolar source and equation 1 is a good approximation. Additionally, this equation allows the use of superposition to compound the voltage from many places and find the total voltage. The neurons in this lab are modeled as cylindrical volume conductors. Equation 3 is an even more detailed voltage equation taking into account the cylindrical components but Holt and Koch 1999 has shown that equation 1 is a suitable estimate. Equation 1 also assumes that there is a single conductivity but programs like COMSOL help account for this.

An analytical model of neural recording was first created. Currents and their positions were created in NEURON using pyramidal neurons and loaded into files called currents_big and currents_small. Using equation 1 and sigma = 0.3333 S/m voltages were recorded 50 um, 100 um, 200 um, and 300 um away from the axon hillock. The voltages for all positions were graphed for 756 ms. A more complex model was then created that contained 10 neurons. These neurons were given random positions within a 100 um cube from the recording site. The neurons were also given random fixed firing patterns between 2-50 Hz for one second. The voltage was recorded at the origin between all the neurons. 10 μV peak to peak pink noise was added to the model. A dead zone was also added within 25 um to mimic dead neurons upon the insertion of a real electrode. The graphs of the voltages over time were plotted with and without the dead zone. Another model was created to account for the variable conductivity. A K matrix was exported from COMSOL and a cell with the hillock 50 um from the origin was created. Equation 4 was used to convert the K matrix and currents from currents_big to voltages and the data was interpolated to account for the different positions of the neuron and data from comsol. The voltage over time of the neuron was created and compared to the first analytical model. In the last part the minimum current found to cause an action potential was found experimentally using a model in a previous lab. The neuron is 5 mm in length, has a diameter of 20um, and has 1000 compartments. Using equation 1 the extracellular voltage was found for each compartment given an electrode of -1 mA was 1mm from the center and the results were plotted. Equation 6 from Warman, Grill and Durand (1992) was used to find the internal current at each compartment where ga = 3 * 10^-5 S. The highest current here gave the relationship between a 1 mA source 1mm and the intracellular current. This relationship was then used to find the extracellular current 1mm when the intracellular current is the minimum current found in NEURON.

In the first model there was only one neuron accounted for and one conductivity assumed. The second model added more realistic characteristics including the effects of 10 neurons, pink noise and a dead region. The third model only accounted for one neuron but demonstrated the effects that different conductivities have on the model. The last model assesses stimulating a neuron instead of recording activity like the other simulations.

Results

Figures 1 and 2 show the results of some of the COMSOL head models.They have a similar appearance with mostly no voltage but tiny concentrated voltage at the center where the

source is. One note to make is that the scales are different indicating model 2 may have higher voltages. Overall, increasing model complexity did not have a significant effect on the distribution of the voltages but a more realistic model may have more accurate voltage values.

Figure 3 shows the plot of the distance from the stimulus vs. the voltage. A 1/r relationship is shown where the voltage decays quickly away from the stimulus and this corresponds to what is expected in equation 1.

Figure 4 shows the result of the deep brain stimulation model. With much smaller currents, the voltage is not as spread out through the head but is more concentrated at the center. A line was created from -1 to 1 m with a step size of .0001 m along the x axis and the voltage was obtained along this line. Figure 5 shows the voltages along this line while Figure 6 shows the second derivative of the voltage along the line. It can be seen that as the line gets closer to the current sources and sinks the voltage begins to spike up. The line goes through the positive current source, an anodic stimulus. Thus, it makes sense that the voltage spikes positive where there is a positive current source. In Figure 6, there is a positive spike, a negative spike, followed by a positive spike. Right near the electrode the neuron is hyperpolarized from the anodic stimulus, shown with a negative spike making the neuron less likely to fire. This is because the voltage is already high up making it more difficult to depolarize and spike. On either side of the negative spike in Figure 6 there are positive spikes that indicate depolarized regions. These results match what is expected in equation 5. The change in membrane voltage is correlated with the second derivative. When the membrane voltage is going negative (less likely to spike) the second derivative of the external voltage is also going negative.

Analytical neural recordings were also done. Figure 7 shows the voltage over time recorded from 50, 100, 200 and 300um away from the axon hillock using the big currents while Figure 8 shows the same for the small currents. In each case an action potential is shown with a negative extracellular voltage indicating the start of the action potential because of the positive intracellular voltage, followed by a positive extracellular voltage because of the negative intracellular voltage. As the recording is taken farther away the amplitude of the voltage detected is smaller. With the big currents a 10 uV peak to peak is no longer detectable 300um away. For the small current a 10 uV peak to peak is no longer detectable 100 um away. This demonstrates how close recordings must be to detect neuron spikes.

This model was extended to include multiple neurons, pink noise, and a dead zone. Figure 9a shows the result of recording voltage with multiple neurons, pink noise and no dead zone. Pink noise was added because there can be thermal noise from the thermal motion of ions. Figure 9b shows the result of recording voltage with a dead zone. When electrodes are inserted in the brain to make recordings oftentimes neurons around the electrode die. This was accounted for by removing neurons within 25 um of the electrode. This makes many of the larger spikes disappear as shown from the graph and axes because the neurons that are the closest are the most detectable according to Equation 1. Figure 9 is much more crowded than figures 7 and 8 because of the multiple neurons firing and noise making it more difficult to identify the signal caused by

one neuron. Additionally, The voltages recorded in Figure 9 are greater due to having more neurons. Figure 9 is much more realistic of what neural recordings look like.

Voltages from a COMSOL head model were used to see how it affected a cell 50um from the origin. The external voltage on the cell is plotted in Figure 10. The shape of this figure looks extremely similar to Figure 7. There is an initial decrease in potential due to the intracellular voltage increase in potential and then the extracellular voltage is positive because of the negative intracellular potential. There is only a difference in magnitude from figure 7 potential because the electrode was closer there but the shape is the same. This confirms Moffit and McIntyre 2005 that different conductivity regions do not have a large effect on the extracellular voltage.

The last section looked into neural stimulation. Experimentally, it was found that the model neuron required a minimum of 76 nA to have an action potential. The extracellular voltages were found at each compartment and are shown in Figure 11. Using equation 6, Figure 12 shows the internal current of the neuron. These graphs make sense as Figure 11 shows the lowest voltage closest to the electrode and there is the highest current closest to the electrode due to the electrode having a negative current. The peak of Figure 11 was found to be -1.79 * 10^-10 A. One mA one millimeter away corresponds to -1.79 * 10^-10 A intracellular. This relationship can be applied to the internal minimum current of 76 nA. A ratio of 1mA/x = -1.79 * 10^-10 A / 76nA can be made and solved for x = 425 mA. This means it would require 425 mA minimum to generate an action potential 1 mm away from the center of the axon..

Discussion

In this report models were used to demonstrate voltage distribution, record neural traces, analyze neuron stimulation, and assess how the complexity of each model affects the results. A COMSOL model of the head with an electrode at the center gave voltage values in space and showed a 1/r relationship with voltage. A deep brain stimulation was also created that demonstrated the activating function relative to electrodes.

Neural tracings were modeled by using equation 1 to get the voltage at the position of each compartment of a cell over time and showed that the farther from the neuron it is more difficult to sense a signal. This was extended to having 10 neurons, noise, and a dead zone, making the neural traces much less clear to identify. Voltages from COMSOL were used to find the extracellular voltage at a neuron and it was shown that different conductance values did not greatly affect the model. The relationship between extracellular current, extracellular voltage, and intracellular current was used to find the minimum activation of a neuron.

The COMSOL models created are excellent to model inserting electrodes in the brain. They can be utilized to test placing the electrode on different parts of the brain instead to see how that affects what neurons can be stimulated. Additionally, a 3 dimensional electrode can be used instead of a point source to see if that has significant changes. The neural detection data helps get show how close it may be necessary to record neurons firing. The COMSOL model interpolation is an effective way of estimating stimulus in a specific region without having to manually calculate voltages. Finally, finding the minimum current to stimulate a neuron some distance away can be used to determine if a safe amount of current is able to be used in a human patient.

Appendix

Equations

$$V_{ext} = \frac{I_0}{4\pi\sigma r}$$

(Equation 1) Equation to find extracellular voltage.

$$V_{ext} = \frac{I_0}{4\pi\sigma}\frac{d\cos\theta}{r^2}$$

(Equation 2) Equation to find extracellular voltage with a dipole

$$V = \frac{I}{4\pi\sigma_e \Delta s}\log\left|\frac{\sqrt{h^2+r^2}-h}{\sqrt{l^2+r^2}-1}\right|$$

(Equation 3) Equation to find extracellular voltage in a cylindrical conductance model accounting for height and length.

$$V_{ext}(x,y,z) = I_0(x,y,z) \times K(x,y,z)$$

(equation 4)

Equation to transfer COMSOL data into voltages.

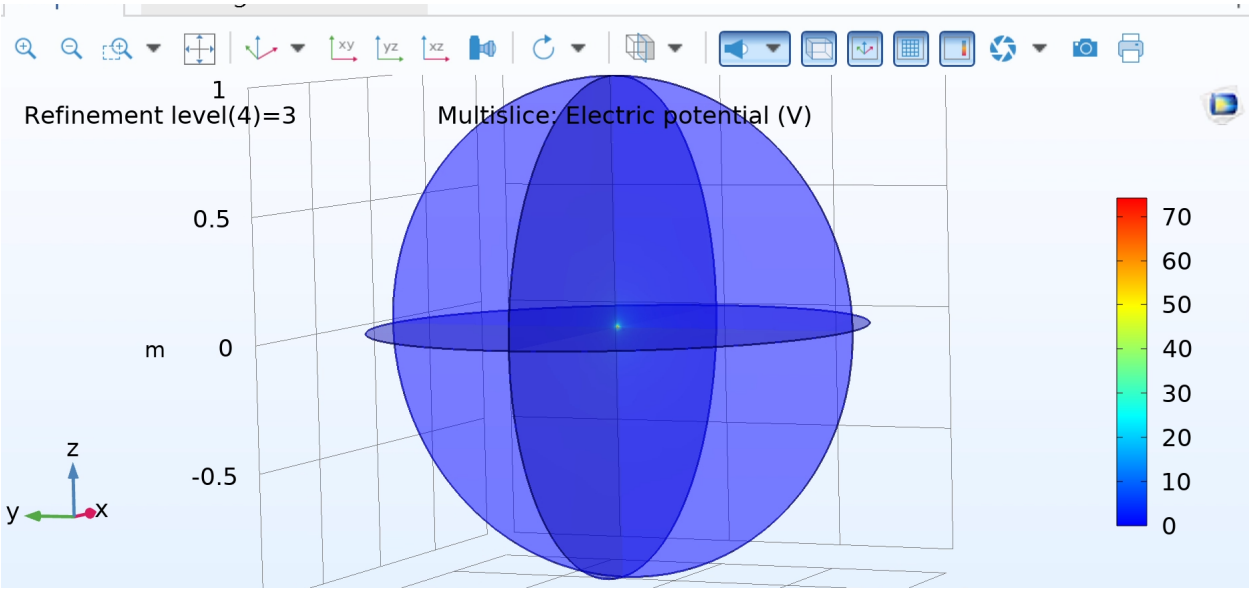
$$r_i c_m \frac{\partial V_m}{\partial t} = \frac{\partial^2 V_e}{\partial x^2}$$

(Equation 5) Equation describing the activating function.

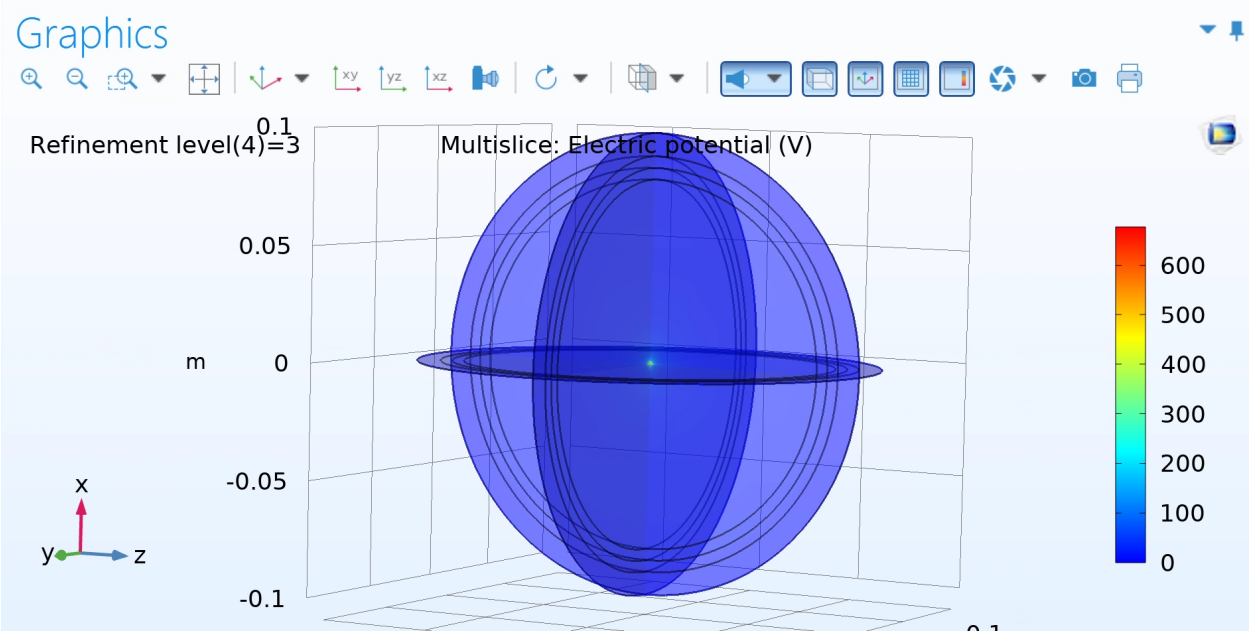
$$I_{\text{int}}(n) = g_a \Delta^2 V_e = g_a (V_e(n-1) - 2V_e(n) + V(n+1))$$

(equation 6) Equation to find intracellular voltage with spatial second derivative.

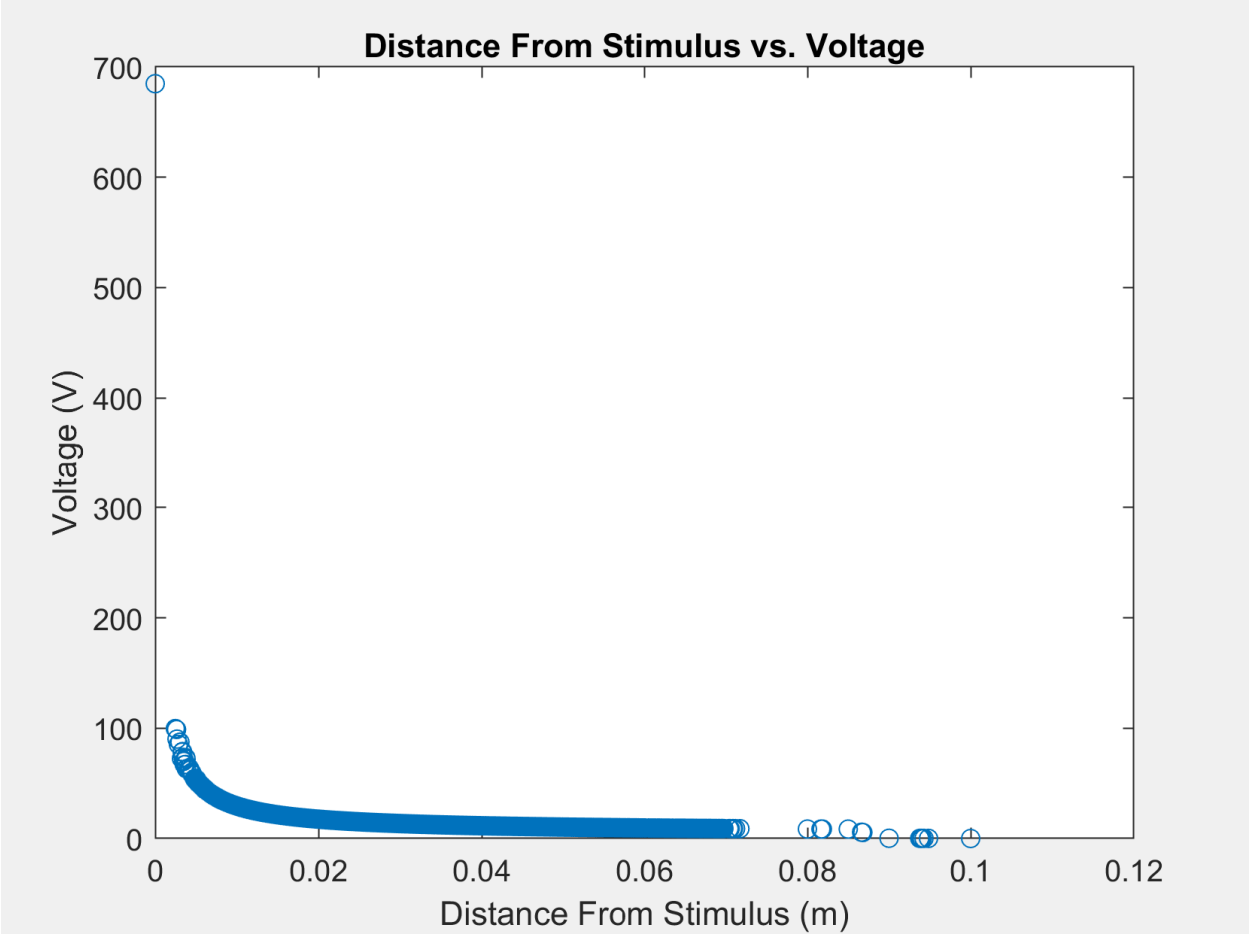
Figures



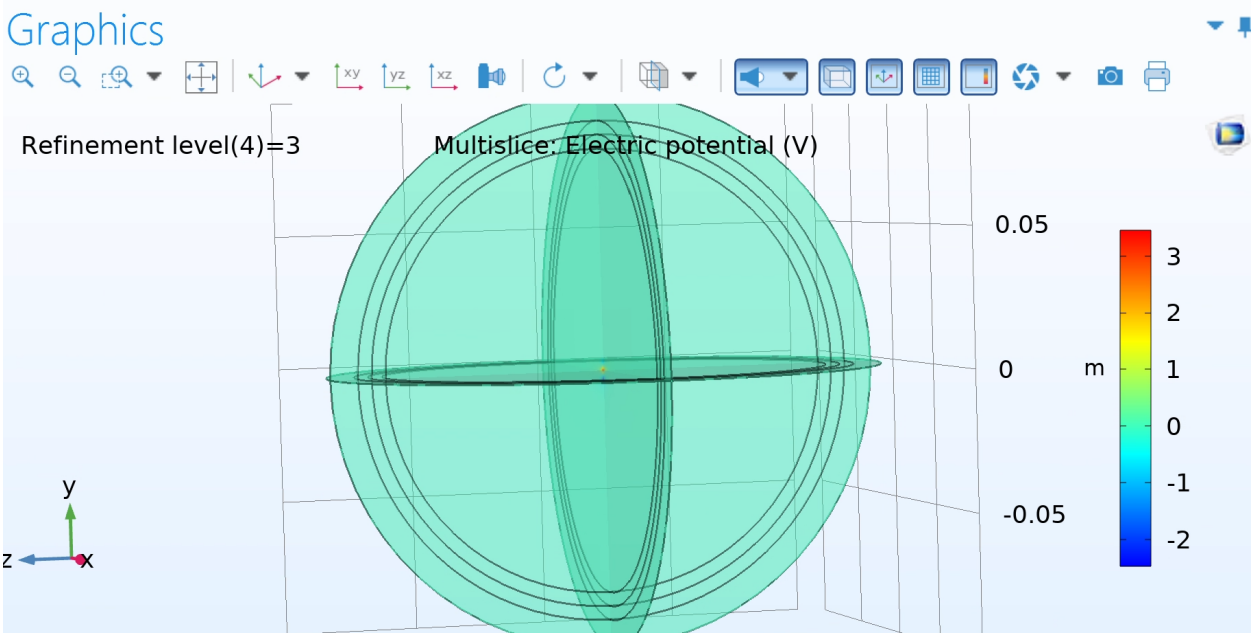
(Figure 1) Simple model of voltages with one sphere and point source



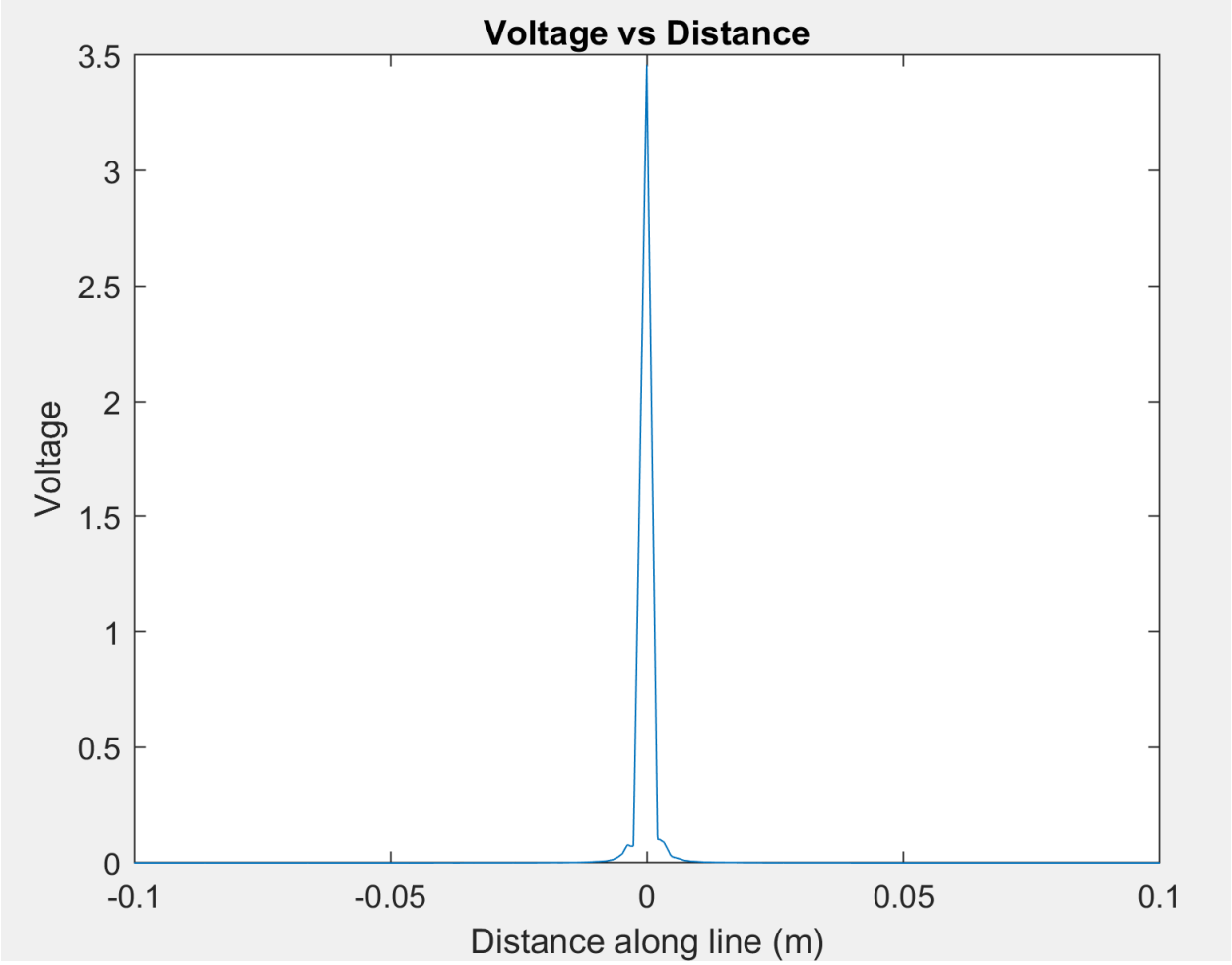
(Figure 2) Voltage distribution of model of head with concentric sphere and point source electrode.



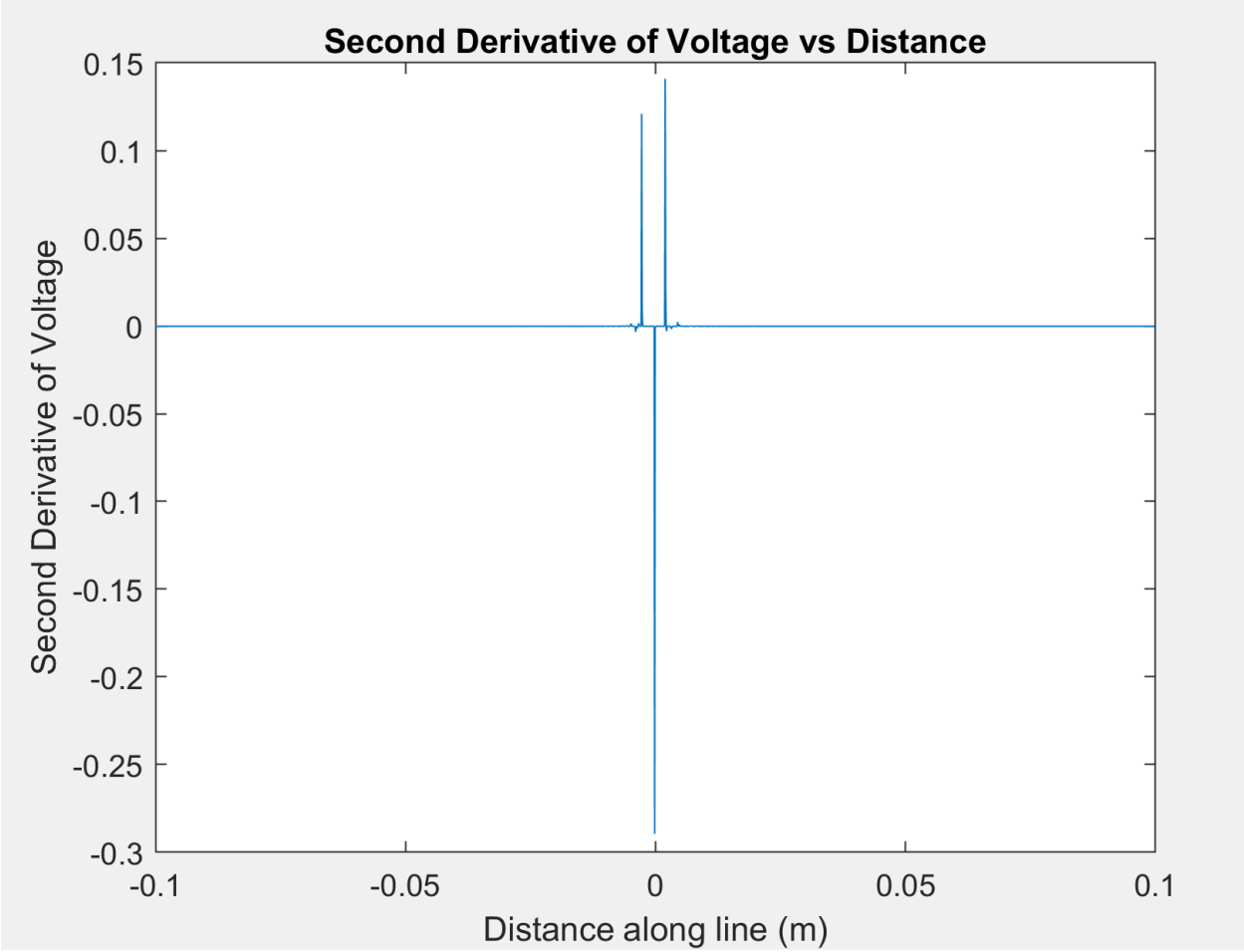
(Figure 3) Voltage from COMSOL over the distance from stimulus.



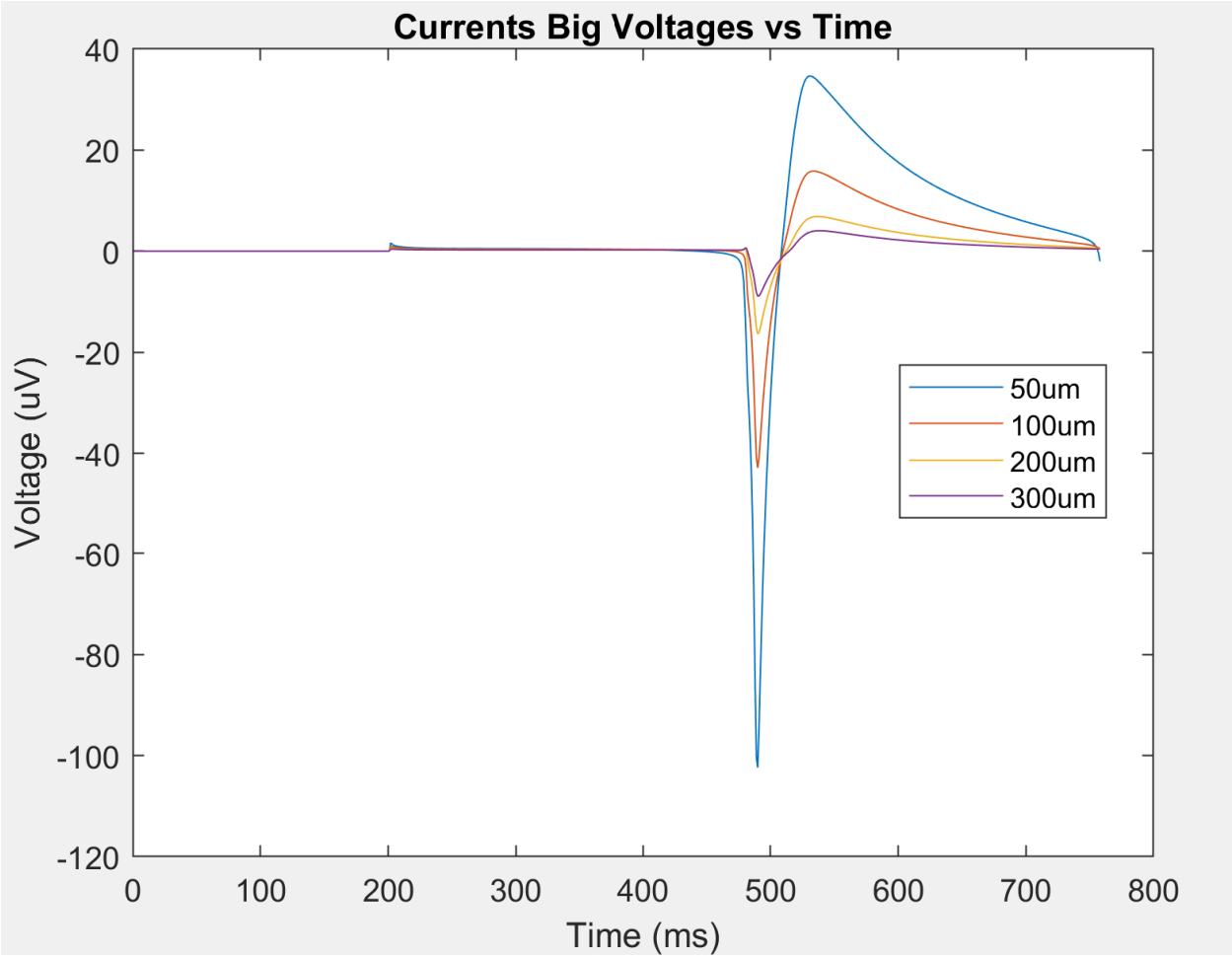
(Figure 4) Deep brain stimulation model with 2 current sinks and a current source.



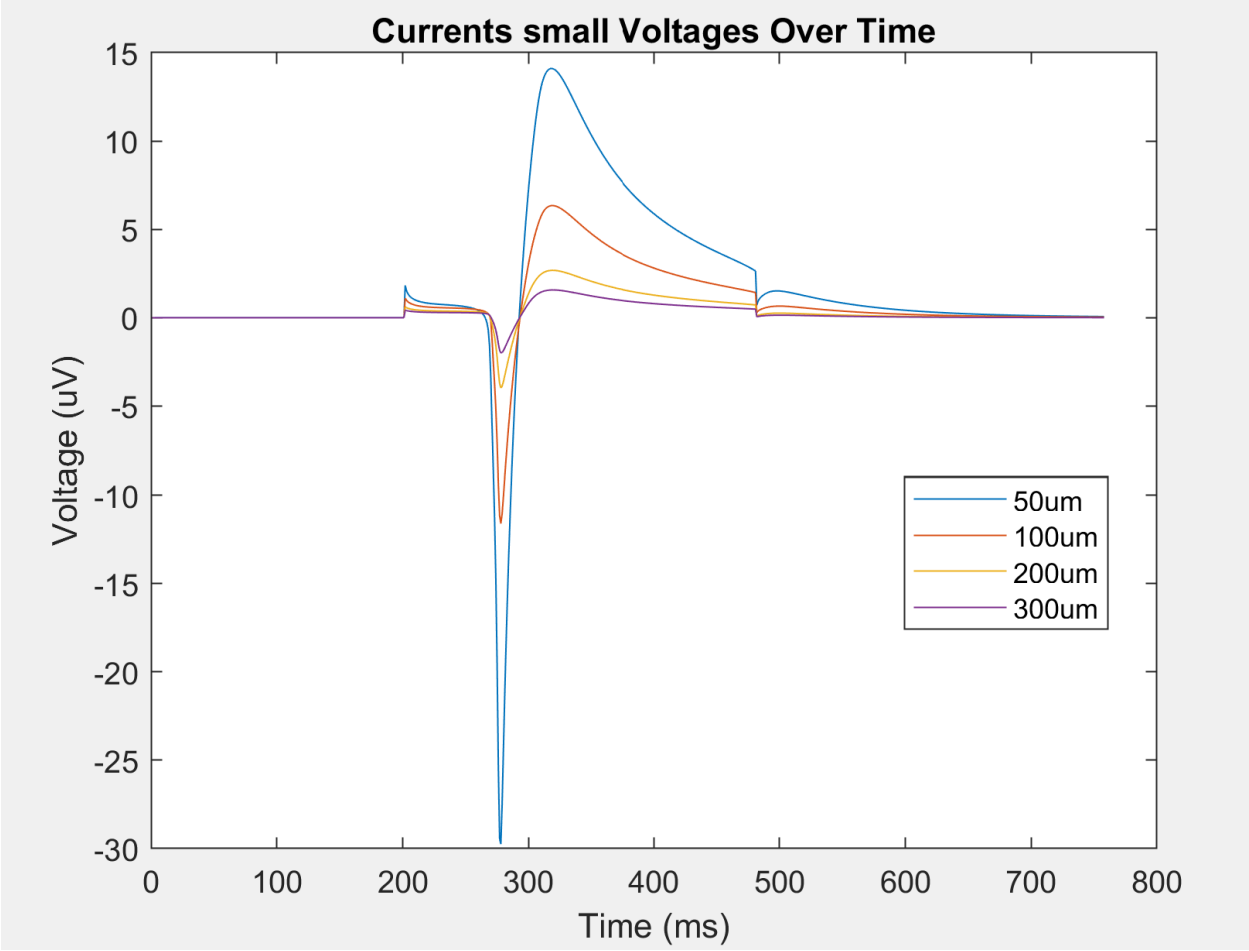
(Figure 5) Voltage over distance of line through deep brain stimulation model.



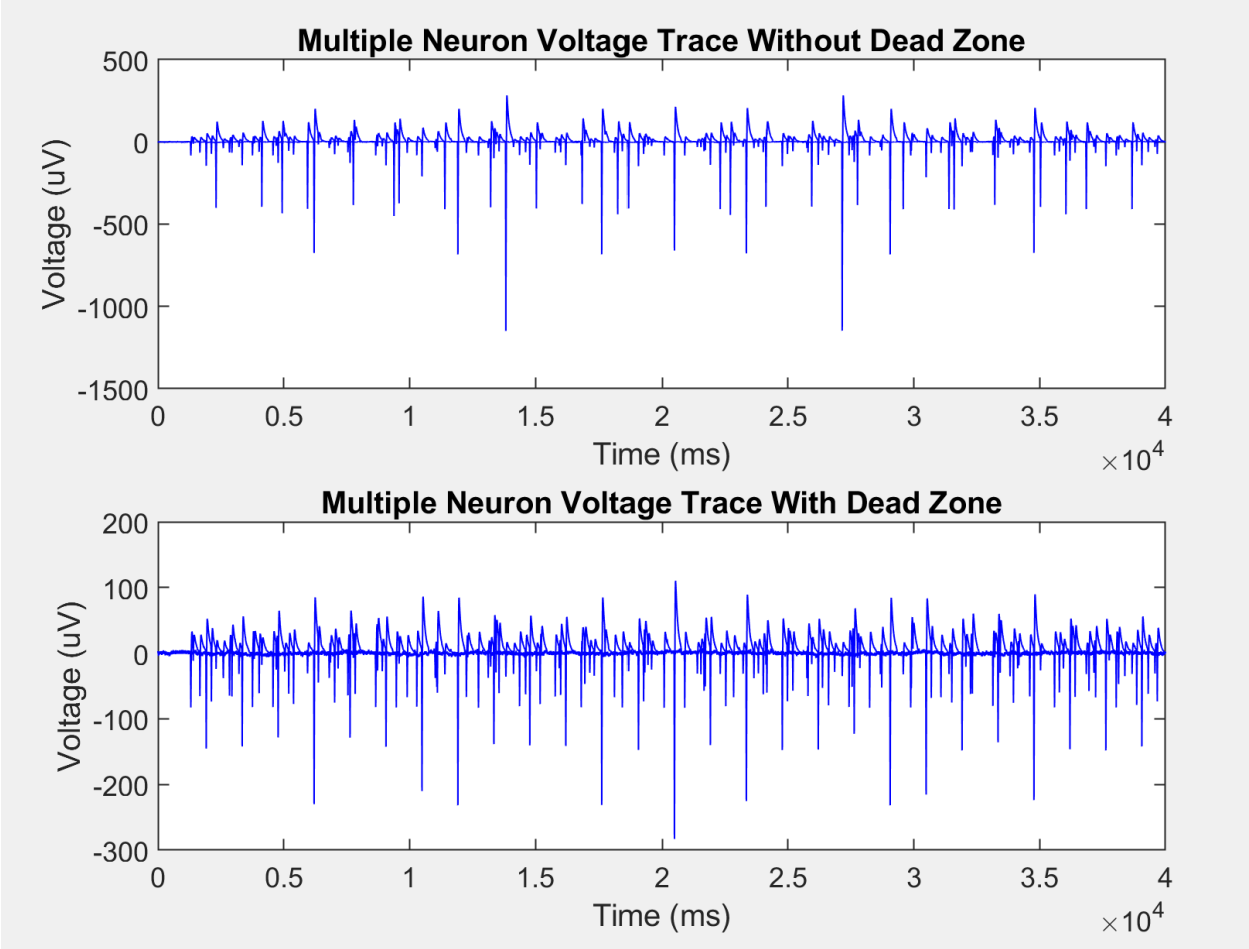
(Figure 6) Activating function of line going through deep brain stimulation model.



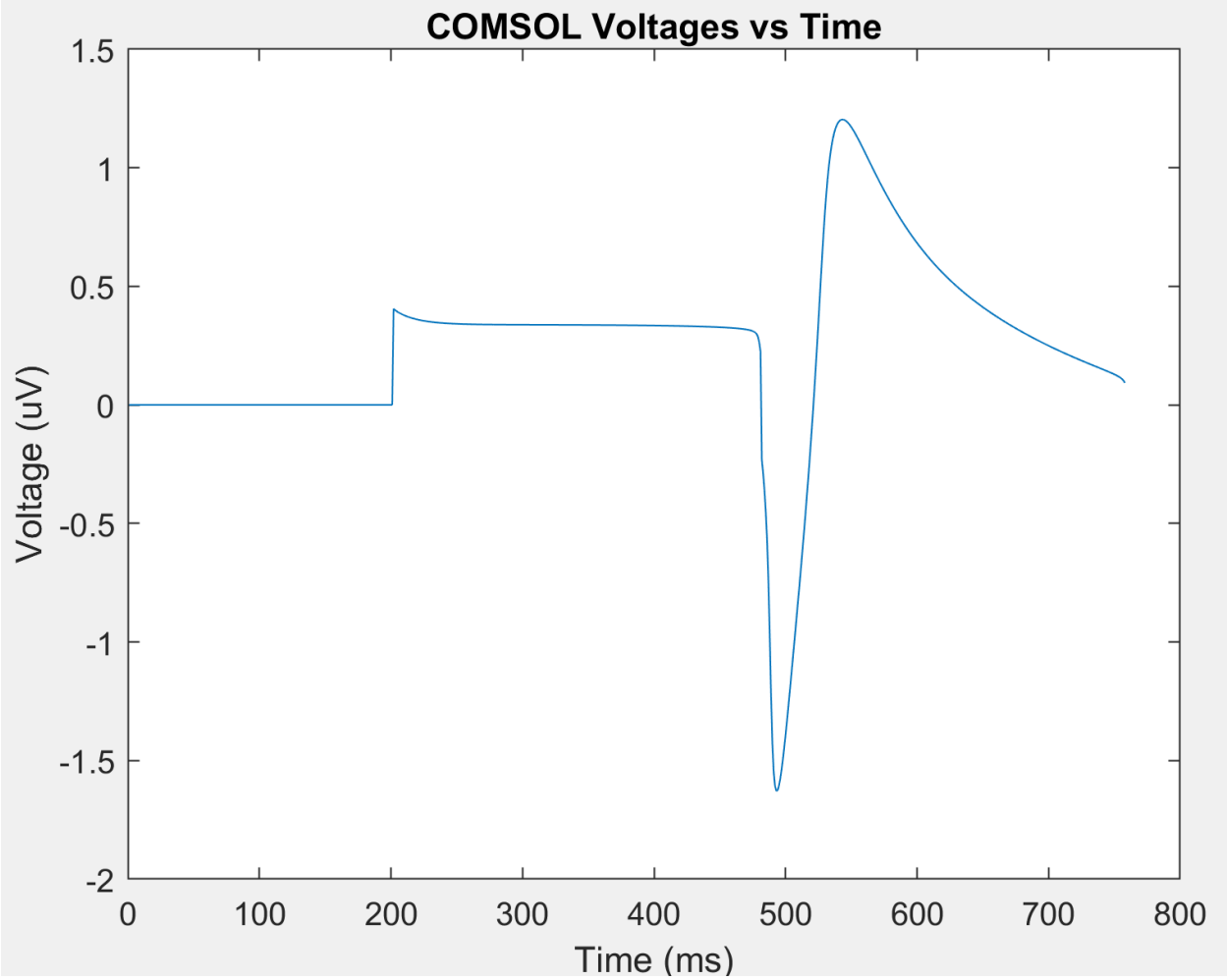
(Figure 7) Voltage recordings from one neuron different distances away for currents_big.



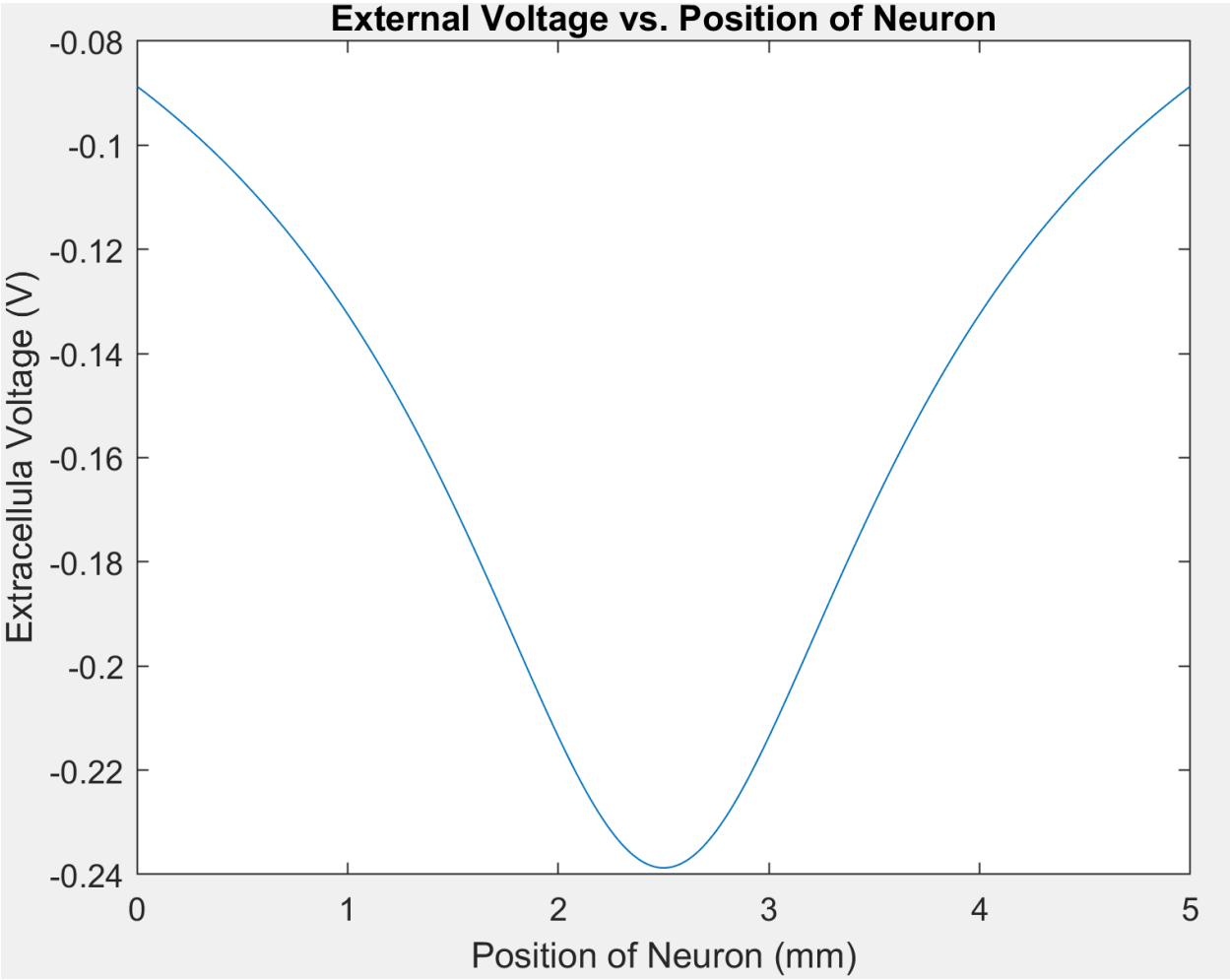
(Figure 8) Voltage recordings from one neuron different distances away for currents_small.



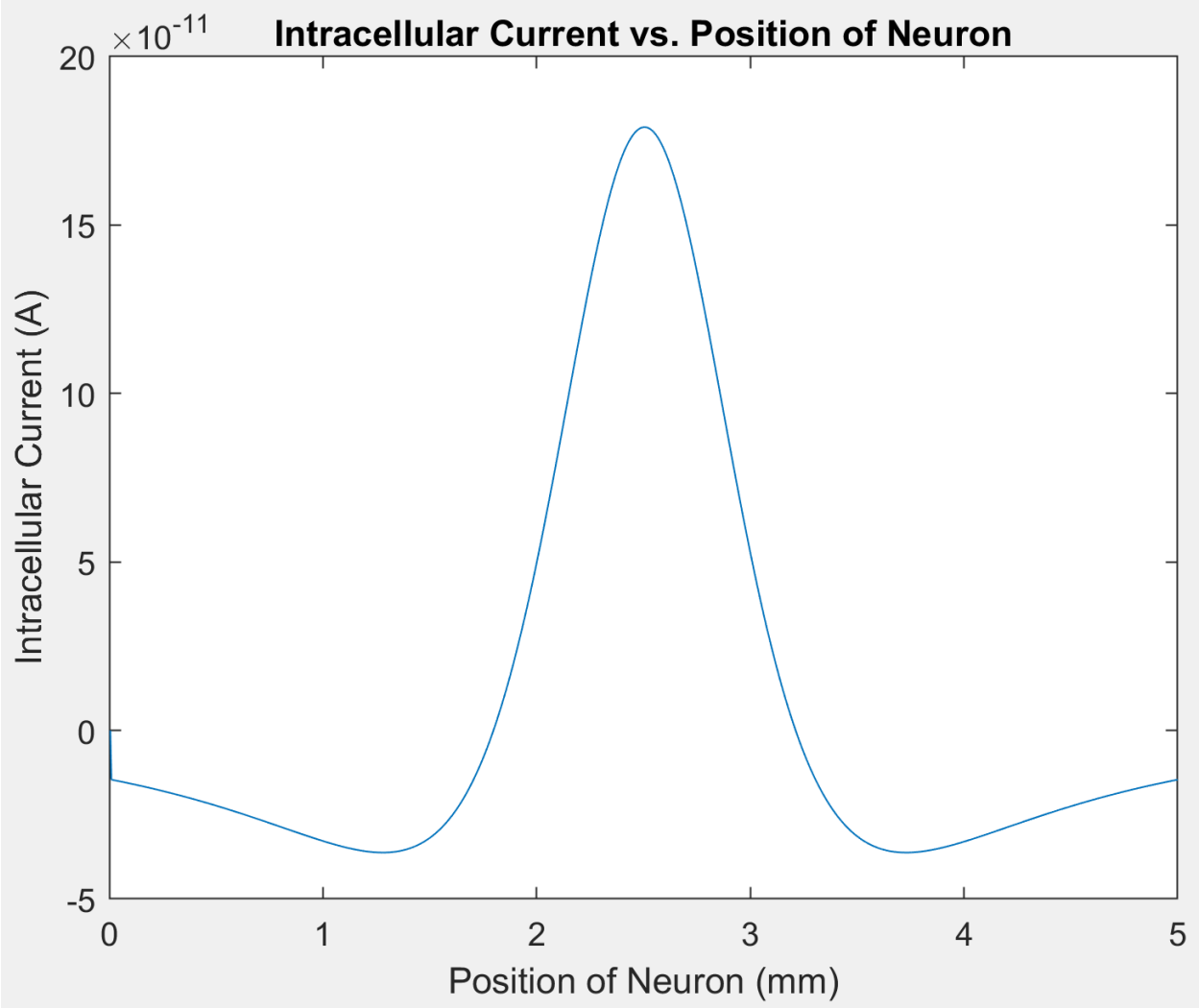
(Figure 9a and 9b) 9a shows the voltage recordings of 10 randomly distributed, randomly firing neurons with pink noise, and no dead zone. 9b shows the same thing but with a deadzone



(Figure 10) Extracellular voltage at neuron 50 um from the center using COMSOL data.



(Figure 11) Extracellular voltage of neuron 1 mm from point source in middle of axon.



(Figure 12) Intracellular current from point source 1 mm away from the middle of the axon.

Code
1/R relationship of voltage over distance.

```
voltages = load('Lab3Part2Voltages.txt');
x = voltages(:,1);
y = voltages(:,2);
z = voltages(:,3);
distance = 1:3799;
for i = 1:3799
    distance(i) = sqrt(x(i)^2+y(i)^2+z(i)^2);
end
plot(distance,voltages(:,4), 'o')
title("Distance From Stimulus vs. Voltage")
```

```
xlabel("Distance From Stimulus (m)")
ylabel("Voltage (V)")
```

Voltage through deep brain simulation model and activating function

```
voltages = load('Lab3Part3Voltages.txt');
size(voltages);
X = -1:.0001:1;
Y = zeros(1,20001);
Z = zeros(1,20001);
%size(X)
%size(Y)
%size(Z)
%interpolation = griddata(voltages(:,1),voltages(:,2), voltages(:,3), voltages(:,4),
neuron.XYZ(:,1), neuron.XYZ(:,2), neuron.XYZ(:,3));
interpolation = griddata(voltages(:,1),voltages(:,2), voltages(:,3), voltages(:,4),X, Y, Z);
size(interpolation);
figure(1);
plot(X, interpolation);
title("Voltage vs Distance")
xlabel("Distance along line (m)")
ylabel("Voltage")

secondDerivative = diff(diff(interpolation));
%size(interpolation)
%size(X)
length(secondDerivative)
length(X)
figure(2)
plot(X(1:19999), secondDerivative);
title("Second Derivative of Voltage vs Distance")
xlabel("Distance along line (m)")
ylabel("Second Derivative of Voltage")
```

Voltage recording over different distance for currents_big

```
currents_big = load('currents_big.mat');
```

```
%% Currents Big Electrode 50 away
positions50 = 1:531;

for index = 1:531
   positions50(index) = sqrt((currents_big.XYZ(index) - 50)^2 + 1^2);
end
voltages50 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_big.currents(k,index))/(4*pi*0.3333*10^-6*
positions50(k));
   end
   voltages50(index) = tempVoltage;
end
time = 1:758;

plot(time, voltages50*10^6);


hold on
%% Electrode 100 Away
positions100 = 1:531;

for index = 1:531
   positions100(index) = sqrt((currents_big.XYZ(index) - 100)^2 + 1^2);
end
voltages100 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_big.currents(k,index))/(4*pi*0.3333*10^-6*
positions100(k));
   end
   voltages100(index) = tempVoltage;
end
time = 1:758;
plot(time, voltages100*10^6)
hold on
%% Currents Big Electrode 200 Away
```

```
positions200 = 1:531;

for index = 1:531
   positions200(index) = sqrt((currents_big.XYZ(index) - 200)^2 + 1^2);
end
voltages200 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_big.currents(k,index))/(4*pi*0.3333*10^-6*
positions200(k));
   end
   voltages200(index) = tempVoltage;
end
time = 1:758;
plot(time, voltages200*10^6)
hold on
%% Currents Big Electrode 300 Away
positions300 = 1:531;

for index = 1:531
   positions300(index) = sqrt((currents_big.XYZ(index) - 300)^2 + 1^2);
end
voltages300 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_big.currents(k,index))/(4*pi*0.3333*10^-6*
positions300(k));
   end
   voltages300(index) = tempVoltage;
end
time = 1:758;
plot(time, voltages300*10^6)
legend("50um", "100um", "200um", "300um")
title("Currents Big Voltages vs Time")
xlabel("Time (ms)")
ylabel("Voltage (uV)")
hold off
```

Voltage recording over different distance for currents_small

```
currents_small = load('currents_small.mat');

%% Currents Small Electrode 50 Away
positions50 = 1:531;

for index = 1:531
   positions50(index) = sqrt((currents_small.XYZ(index) - 50)^2 + 1^2);
end
voltages50 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_small.currents(k,index))/(4*pi*0.3333*10^-6*
positions50(k));
   end
   voltages50(index) = tempVoltage;
end
time = 1:758;

plot(time, voltages50*10^6);


hold on
%% Currents Small Electrode 100 Away
positions100 = 1:531;

for index = 1:531
   positions100(index) = sqrt((currents_small.XYZ(index) - 100)^2 + 1^2);
end
voltages100 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_small.currents(k,index))/(4*pi*0.3333*10^-6*
positions100(k));
   end
   voltages100(index) = tempVoltage;
end
```

```
time = 1:758;
plot(time, voltages100*10^6)
hold on
%% Currents Small Electrode 200 Away
positions200 = 1:531;

for index = 1:531
   positions200(index) = sqrt((currents_small.XYZ(index) - 200)^2 + 1^2);
end
voltages200 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_small.currents(k,index))/(4*pi*0.3333*10^-6*
positions200(k));
   end
   voltages200(index) = tempVoltage;
end
time = 1:758;
plot(time, voltages200*10^6)
hold on
%% Currents Small Electrode 300 Away
positions300 = 1:531;

for index = 1:531
   positions300(index) = sqrt((currents_small.XYZ(index) - 300)^2 + 1^2);
end
voltages300 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      tempVoltage = tempVoltage + (currents_small.currents(k,index))/(4*pi*0.3333*10^-6*
positions300(k));
   end
   voltages300(index) = tempVoltage;
end
time = 1:758;
plot(time, voltages300*10^6)
legend("50um", "100um", "200um", "300um")
title("Currents small Voltages Over Time")
```

```
xlabel("Time (ms)")
ylabel("Voltage (uV)")
```

Voltage recording over time with 10 neurons and pink noise

```
neuron1 = load('currents_big.mat');
neuron1.XYZ(:,3) = 10;
voltages1 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron1.XYZ(k,1))^2 + (neuron1.XYZ(k,2))^2 + (neuron1.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron1.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages1(index) = tempVoltage;
end
%total1 = 1:40000;
total1 = zeros(40000,1);
for i = 1:2
   ind = floor((40000/3)*i);
   total1(ind + 1: ind + 758) = voltages1;
end

neuron2 = load('currents_big.mat');
neuron2.XYZ(:,3) = 10;
neuron2.XYZ(:,2) = 10;
voltages2 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron2.XYZ(k,1))^2 + (neuron2.XYZ(k,2))^2 + (neuron2.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron2.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages2(index) = tempVoltage;
end
total2 = zeros(40000,1);
for i = 1:6
   ind = floor((40000/7)*i);
   total2(ind + 1: ind + 758) = voltages2;
end
```

```
neuron3 = load('currents_big.mat');
neuron3.XYZ(:,3) = 10;
neuron3.XYZ(:,2) = -10;
voltages3 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron3.XYZ(k,1))^2 + (neuron3.XYZ(k,2))^2 + (neuron3.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron3.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages3(index) = tempVoltage;
end
total3 = zeros(40000,1);
for i = 1:8
   ind = floor((40000/9)*i);
   total3(ind + 1: ind + 758) = voltages3;
end

neuron4 = load('currents_big.mat');
neuron4.XYZ(:,3) = -78;
neuron4.XYZ(:,2) = 57;
voltages4 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron4.XYZ(k,1))^2 + (neuron4.XYZ(k,2))^2 + (neuron4.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron4.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages4(index) = tempVoltage;
end
total4 = zeros(40000,1);
for i = 1:11
   ind = floor((40000/12)*i);
   total4(ind + 1: ind + 758) = voltages4;
end


neuron5 = load('currents_big.mat');
neuron5.XYZ(:,3) = 88;
```

```
neuron5.XYZ(:,2) = -26;
voltages5 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron5.XYZ(k,1))^2 + (neuron5.XYZ(k,2))^2 + (neuron5.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron5.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages5(index) = tempVoltage;
end
total5 = zeros(40000,1);
for i = 1:18
   ind = floor((40000/19)*i);
   total5(ind + 1: ind + 758) = voltages5;
end


neuron6 = load('currents_big.mat');
neuron6.XYZ(:,3) = -15;
neuron6.XYZ(:,2) = 7;
voltages6 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron6.XYZ(k,1))^2 + (neuron6.XYZ(k,2))^2 + (neuron6.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron6.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages6(index) = tempVoltage;
end
total6 = zeros(40000,1);
for i = 1:21
   ind = floor((40000/22)*i);
   total6(ind + 1: ind + 758) = voltages6;
end


neuron7 = load('currents_big.mat');
neuron7.XYZ(:,3) = 25;
neuron7.XYZ(:,2) = -25;
voltages7 = 1:758;
```

```
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron7.XYZ(k,1))^2 + (neuron7.XYZ(k,2))^2 + (neuron7.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron7.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages7(index) = tempVoltage;
end
total7 = zeros(40000,1);
for i = 1:27
   ind = floor((40000/28)*i);
   total7(ind + 1: ind + 758) = voltages7;
end

neuron8 = load('currents_big.mat');
neuron8.XYZ(:,3) = -33;
neuron8.XYZ(:,2) = -54;
voltages8 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron8.XYZ(k,1))^2 + (neuron8.XYZ(k,2))^2 + (neuron8.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron8.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages8(index) = tempVoltage;
end
total8 = zeros(40000,1);
for i = 1:33
   ind = floor((40000/34)*i);
   total8(ind + 1: ind + 758) = voltages8;
end

neuron9 = load('currents_big.mat');
neuron9.XYZ(:,3) = -63;
neuron9.XYZ(:,2) = 82;
voltages9 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron9.XYZ(k,1))^2 + (neuron9.XYZ(k,2))^2 + (neuron9.XYZ(k,3))^2);
```

```
        tempVoltage = tempVoltage + (neuron9.currents(k,index))/(4*pi*0.3333*10^-6* distance);
    end
    voltages9(index) = tempVoltage;
end
total9 = zeros(40000,1);
for i = 1:43
    ind = floor((40000/44)*i);
    total9(ind + 1: ind + 758) = voltages9;
end

neuron10 = load('currents_big.mat');
neuron10.XYZ(:,3) = 44;
neuron10.XYZ(:,2) = -32;
voltages10 = 1:758;
for index = 1:758
    tempVoltage = 0;
    for k = 1:531
        distance = sqrt((neuron10.XYZ(k,1))^2 + (neuron10.XYZ(k,2))^2 +
(neuron10.XYZ(k,3))^2);
        tempVoltage = tempVoltage + (neuron10.currents(k,index))/(4*pi*0.3333*10^-6* distance);
    end
    voltages10(index) = tempVoltage;
end
total10 = zeros(40000,1);
for i = 1:48
    ind = floor((40000/49)*i);
    total10(ind + 1: ind + 758) = voltages10;
end

time = 1:40000;
total = total1 + total2 + total3 + total4 + total5 + total6 + total7 + total8 + total9 + total10;
%plot(time, total)
%title("Recording with 10 neurons")
cn = dsp.ColoredNoise('Color','pink','SamplesPerFrame',40000);
recordedNoise = cn();
subplot(2,1,1)
plot(time, total * 10^6,'b')
hold on
plot(time, recordedNoise,'b')
title("Multiple Neuron Voltage Trace Without Dead Zone")
```

```
xlabel("Time (ms)")
ylabel("Voltage (uV)")

%% Same With Dead Zone

neuron4 = load('currents_big.mat');
neuron4.XYZ(:,3) = -78;
neuron4.XYZ(:,2) = 57;
voltages4 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron4.XYZ(k,1))^2 + (neuron4.XYZ(k,2))^2 + (neuron4.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron4.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages4(index) = tempVoltage;
end
total4 = zeros(40000,1);
for i = 1:11
   ind = floor((40000/12)*i);
   total4(ind + 1: ind + 758) = voltages4;
end


neuron5 = load('currents_big.mat');
neuron5.XYZ(:,3) = 88;
neuron5.XYZ(:,2) = -26;
voltages5 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron5.XYZ(k,1))^2 + (neuron5.XYZ(k,2))^2 + (neuron5.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron5.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages5(index) = tempVoltage;
end
total5 = zeros(40000,1);
for i = 1:18
   ind = floor((40000/19)*i);
   total5(ind + 1: ind + 758) = voltages5;
```

```
end



neuron7 = load('currents_big.mat');
neuron7.XYZ(:,3) = 25;
neuron7.XYZ(:,2) = -25;
voltages7 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron7.XYZ(k,1))^2 + (neuron7.XYZ(k,2))^2 + (neuron7.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron7.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages7(index) = tempVoltage;
end
total7 = zeros(40000,1);
for i = 1:27
   ind = floor((40000/28)*i);
   total7(ind + 1: ind + 758) = voltages7;
end

neuron8 = load('currents_big.mat');
neuron8.XYZ(:,3) = -33;
neuron8.XYZ(:,2) = -54;
voltages8 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron8.XYZ(k,1))^2 + (neuron8.XYZ(k,2))^2 + (neuron8.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron8.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages8(index) = tempVoltage;
end
total8 = zeros(40000,1);
for i = 1:33
   ind = floor((40000/34)*i);
   total8(ind + 1: ind + 758) = voltages8;
end
```

```
neuron9 = load('currents_big.mat');
neuron9.XYZ(:,3) = -63;
neuron9.XYZ(:,2) = 82;
voltages9 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron9.XYZ(k,1))^2 + (neuron9.XYZ(k,2))^2 + (neuron9.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron9.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages9(index) = tempVoltage;
end
total9 = zeros(40000,1);
for i = 1:43
   ind = floor((40000/44)*i);
   total9(ind + 1: ind + 758) = voltages9;
end

neuron10 = load('currents_big.mat');
neuron10.XYZ(:,3) = 44;
neuron10.XYZ(:,2) = -32;
voltages10 = 1:758;
for index = 1:758
   tempVoltage = 0;
   for k = 1:531
      distance = sqrt((neuron10.XYZ(k,1))^2 + (neuron10.XYZ(k,2))^2 +
(neuron10.XYZ(k,3))^2);
      tempVoltage = tempVoltage + (neuron10.currents(k,index))/(4*pi*0.3333*10^-6* distance);
   end
   voltages10(index) = tempVoltage;
end
total10 = zeros(40000,1);
for i = 1:48
   ind = floor((40000/49)*i);
   total10(ind + 1: ind + 758) = voltages10;
end

time = 1:40000;
total = total4 + total5 + total7 + total8 + total9 + total10;
%plot(time, total)
```

```
%title("Recording with 10 neurons")
cn = dsp.ColoredNoise('Color','pink','SamplesPerFrame',40000);
recordedNoise = cn();
subplot(2,1,2)

plot(time, total * 10^6,'b')
hold on
plot(time, recordedNoise,'b')
title("Multiple Neuron Voltage Trace With Dead Zone")
xlabel("Time (ms)")
ylabel("Voltage (uV)")
```

Extracellular voltage over time using COMSOL voltages

```
voltages = load('Lab3Part2VoltagesAdaptive.txt');
neuron = load('currents_big.mat');
neuron.XYZ(:,2) = 50;
neuron.XYZ(:,1)
interpolation = griddata(voltages(:,1)*10^6,voltages(:,2)*10^6, voltages(:,3)*10^6, voltages(:,4),
neuron.XYZ(:,1), neuron.XYZ(:,2), neuron.XYZ(:,3));
%Vext = neuron.currents * interpolation
Vext = zeros(758, 1);
for i = 1:758
   Vtemp = 0;
   for k = 1:531
      Vtemp = Vtemp + neuron.currents(k,i) * interpolation(k);
   end
   Vext(i) = Vtemp;
end
time = 1:758;
plot(time, Vext*10^6);
title("COMSOL Voltages vs Time")
xlabel("Time (ms)")
ylabel("Voltage (uV)")
```

Finding minimum current amplitude for activation

```
neuron = 0:.005:5;
Vext = zeros(1,1001);
for i=1:1001
```

```
    Vext(i) = -.001/(4*pi*0.3333*sqrt((1*10^-3)^2 + ((neuron(i)*10^-3) - (2.5*10^-3))^2));
end

figure(1)
plot(neuron, Vext);
title("External Voltage vs. Position of Neuron")
xlabel("Position of Neuron (mm)")
ylabel("Extracellula Voltage (V)")

Iint = zeros(1,999);
for i = 2:1000
    Iint(i) = (3*10^-5)*(Vext(i-1)- 2*Vext(i) + Vext(i+1));
end

figure(2)
plot(neuron(2:1001), Iint)
title("Intracellular Current vs. Position of Neuron")
xlabel("Position of Neuron (mm)")
ylabel("Intracellular Current (A)")
%Peak negative current is -1.78983 * 10^-11 A
% 1 mA -> -1.78983 * 10^-11 A
% Minimum current inside is 76 nA
% minimum 1mm away is 4.26 Amps
max(Iint)
```

Finding minimum intracellular current to have a stimulus

```
axon2 = h.Section(name = 'axon2')
axon2.diam = 20 * um
axon2.L = 5000 * um                    # Hopefully these are in mm
axon2.nseg = 1000
axon2.insert("hh")
iclamp2 = h.IClamp(axon2(0))
iclamp2.delay = 1
iclamp2.dur = 0.1
iclamp2.amp = 76
v3 = h.Vector().record(axon2(.1)._ref_v)
t2 = h.Vector().record(h._ref_t)

h.finitialize(-65 * mV)
```

```
h.continuerun(40 * ms)
plt.plot(t, v2)
```