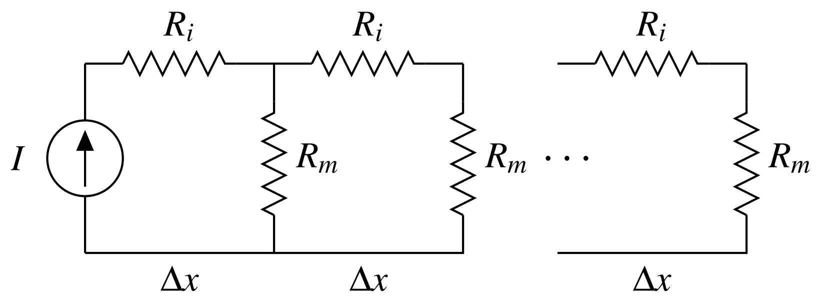
BIOM1010: Engineering in Medicine and Biology
Tutorial: Computational Modelling in Bioengineering

Socrates Dokos

30 September 2018

Voltage Distribution in a Passive Neural Membrane

Steady-state passive electrical behaviour of an unmyelinated nerve cell axon of length $L = 10$ mm can be approximated by a thin cylinder of radius $r = 25\ \mu$m, filled with axoplasmic medium of resistivity $\rho_i = 0.2\ \Omega\,$m, and surrounded by a membrane of resistance $r_m = 0.1\ \Omega\,\mathrm{m}^2$. If a current of $I = 1\ \mu$A is injected into one end of the axon, and assuming the extracellular potential is set everywhere to ground, this system can be represented as N discrete circuit elements, each of length Δx, shown below:



where the value of each R_i and R_m will depend on N according to

$$R_i = \frac{\rho_i \Delta x}{\pi r^2}, \quad R_m = \frac{r_m}{2\pi r \Delta x}, \quad \Delta x = \frac{L}{N-1}$$

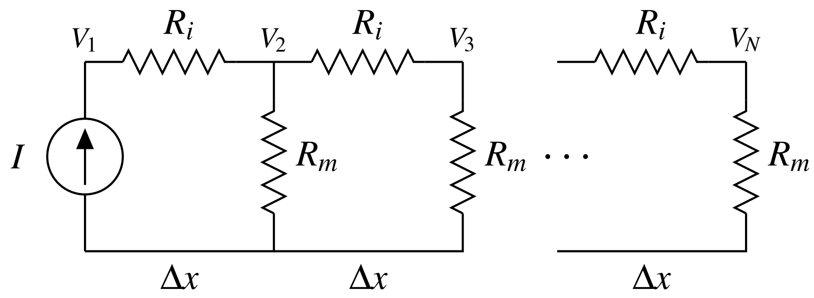
1. Use MATLAB to solve and plot the membrane voltage V_m across the R_m's as a function of length along the axon for $N = 200$. Note that V_m should monotonically decrease from the current source.

2. Plot the natural logarithm of V_m, i.e. $\ln(V_m)$, as a function of length along the first half of the axon (i.e. over the first 5 mm), and use MATLAB's curve fitting tools to determine the equation of the line of best fit.

3. Find the resulting equation representing V_m itself as a function of length along the first 5 mm of the axon.

Solution to 1

Label the membrane voltages at each node by V_1, V_2, V_3, V_N as shown in the diagram below:



The total current flowing into each node horizontally (i.e. from the adjacent R_i's) must equal the current flowing out of the node vertically through R_m. This is known as Kirchhoffs current law. For node k in general, Kirchhoffs current law can be written mathematically as:

$$\frac{V_{k-1}-V_k}{R_i}+\frac{V_{k+1}-V_k}{R_i}=\frac{V_k}{R_m}$$

which can be re-arranged to

$$\frac{V_{k-1}}{R_i}-\frac{2V_k}{R_i}-\frac{V_k}{R_m}+\frac{V_{k+1}}{R_i}=0$$

or

$$-\left(\frac{1}{R_i}\right)V_{k-1}+\left(\frac{2}{R_i}+\frac{1}{R_m}\right)V_k-\left(\frac{1}{R_i}\right)V_{k+1}=0$$

The only exceptions are for nodes 1 and N. For node 1, we have:

$$\frac{V_2-V_1}{R_i}=-I$$

or

$$\left(\frac{1}{R_i}\right)V_1-\left(\frac{1}{R_i}\right)V_2=I$$

For node N, we have:

$$\frac{V_{N-1}-V_N}{R_i}=\frac{V_N}{R_m}$$

or

$$\left(\frac{1}{R_i}+\frac{1}{R_m}\right)V_N-\left(\frac{1}{R_i}\right)V_{N-1}=I$$

Combining all the above equations, we can form the following matrix system of equations:

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_i} & 0 & \cdots & 0 \\ -\frac{1}{R_i} & \left(\frac{1}{R_m}+\frac{2}{R_i}\right) & -\frac{1}{R_i} & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & -\frac{1}{R_i} & \left(\frac{1}{R_m}+\frac{1}{R_i}\right) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The following MATLAB code generates this matrix system, $Av=b$ for $N=200$, where v is the vector of membrane voltages, and solves for these:

```
N = 200;
A = zeros(N);            % initializes A to a NxN matrix of 0's
b = zeros(N,1);          % initializes b to a Nx1 column vector of 0's
b(1) = 1e-6;             % 1 uA

Delta_x = 0.01/(N-1);
Ri = 0.2*Delta_x/(pi*26e-6^2);
Rm = 0.1/(2*pi*25e-6*Delta_x);
A(1,1) = 1/Rm+1/Ri;
A(2,1) = -1/Ri;
A(N,N) = 1/Rm+1/Ri;
A(N,N-1) = -1/Ri;
for k = 2:N-1
    A(k,k) = 1/Rm+2/Ri;
    A(k,k-1) = -1/Ri;
    A(k,k+1) = -1/Ri;
end
Vm = A\b;
x = (0:Delta_x:0.01)*1000;
plot(x,1000*Vm,'k'), xlabel('length along axon (mm)'), ...
    ylabel('membrane voltage (mV)');
```

This code produces the following plot: