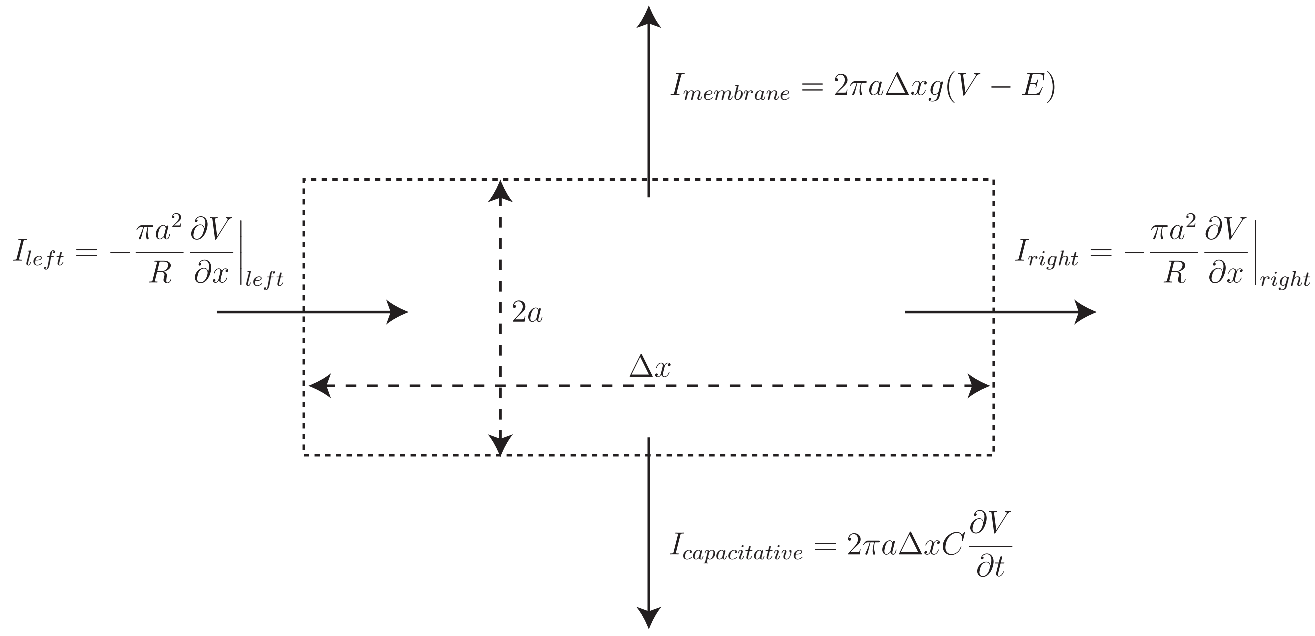
Numerical Computation for Branched Dendrites

Nan Xiao

November 15, 2006

1. **Single Cable** – The starting point for treating dendritic trees is to examine a single section of dendrite, represented by an unbranched cable. Consider a cylindrical section of the cable with radius a, width Δx, and axial resistivity R:


$I_{membrane} = 2\pi a \Delta x g(V - E)$
$I_{left} = -\frac{\pi a^2}{R}\frac{\partial V}{\partial x}\Big|_{left}$
$I_{right} = -\frac{\pi a^2}{R}\frac{\partial V}{\partial x}\Big|_{right}$
$2a$
Δx
$I_{capacitative} = 2\pi a \Delta x C\frac{\partial V}{\partial t}$

Current balance on the cylindrical section yields the equation:

$$I_{right} + I_{capacitance} + I_{membrane} = I_{left}$$
$$2\pi a \Delta x C\frac{\partial V}{\partial t} = -\frac{\pi}{R}\left[a^2\frac{\partial V}{\partial x}\Big|_{left} + a^2\frac{\partial V}{\partial x}\Big|_{right}\right] - 2\pi a \Delta x g(V - E)$$

Allow $\Delta x \to 0$:

$$\lim_{\Delta x\to 0} C\frac{\partial V}{\partial t} = \lim_{\Delta x\to 0}\frac{1}{2aR}\frac{1}{\Delta x}\left[a^2\frac{\partial V}{\partial x}\Big|_{right} - a^2\frac{\partial V}{\partial x}\Big|_{left}\right] - g(V - E)$$
$$C\frac{\partial V}{\partial t} = \frac{1}{2aR}\frac{\partial}{\partial x}\left(a^2\frac{\partial V}{\partial x}\right) - g(V - E)$$

We now have the cable equation:

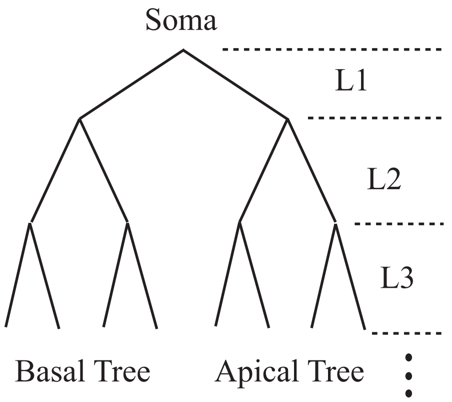
$$G(a^2V_x)_x = a[CV_t + g(V - E)] \tag{1}$$

where $G = 1/2R$

2. **Dendritic Tree** – The dendritic arbor can be represented as a binary tree – each junction is a branchpoint for three cables. We may split and enumerate the dendritic tree as N cables, represented

by one-dimensional domains $\Omega_1, \Omega_2, \ldots, \Omega_N$. $\Omega_k = [A_k, B_k]$. As such, entire tree is represented by the domain $\Omega = \Omega_1 \cup \Omega_2 \cup \ldots \cup \Omega_N$. The order in which we connect the branches may affect the complexity of the linear solve after spatial discretization. The ordering method proposed by Michael Hines allows us to take advantage of the structure of the matrices produced by finite element discretization.

- Arrange the branches by ‘depth’ of branching:

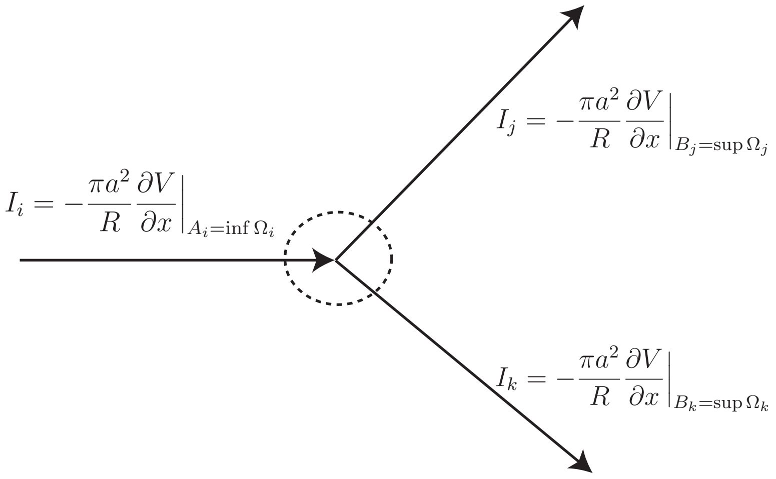


The Hines ordering is simple: we define the soma of the neuron to be represented by the point B_N. If L_Y is a ‘deeper level of branching’ than the L_X (say $L_Y > L_X$, L_Y represents branches farther away from soma than L_X), then for $\Omega_y \in L_Y$ and $\Omega_x \in L_X$, B_y is identical to A_x (a branchpoint). Finally, if some other branch j is connected to the soma, then B_j is identical to B_N.

- Thus, the set of all branchpoints is the intersection $W = \Omega_1 \cap \Omega_2 \cap \ldots \cap \Omega_N$.

- There are T levels in all.

- At each branchpoint, axial currents must be balanced. At a branchpoints, space derivatives of voltage exist in three directions. As such:


$I_j = -\frac{\pi a^2}{R}\frac{\partial V}{\partial x}\Big|_{B_j = \sup \Omega_j}$
$I_i = -\frac{\pi a^2}{R}\frac{\partial V}{\partial x}\Big|_{A_i = \inf \Omega_i}$
$I_k = -\frac{\pi a^2}{R}\frac{\partial V}{\partial x}\Big|_{B_k = \sup \Omega_k}$

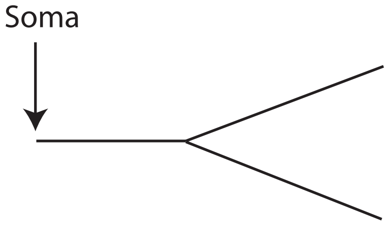
$$I_1 = I_2 + I_3$$

- The terminal branches don’t give rise to a further level of branching. The set of terminal points $\partial\Omega$ is the boundary of the dendritic tree. $\partial\Omega = \{B_k : B_k = \sup \Omega_k, \Omega_k \in L_T\}$.

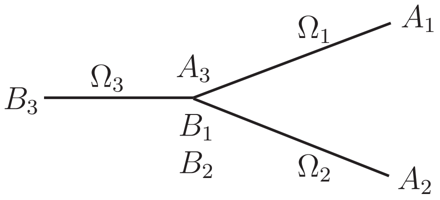
3. **Formulation of problem** – If we describe the dendritic tree as a binary tree, the problem is formulated as:

$$G(a^2V_x)_x = a[CV_t + g(V-E)]$$
$$V(x,t_o) = E \quad \forall x \in \Omega \quad \text{(Initial condition)}$$
$$V_x(x,t) = 0 \quad \forall x \in \partial\Omega \;\; t > t_o \quad \text{(sealed terminals boundary condition)}$$
$$V_x|_{B_N} + V_x|_{B_{N-1}} = \frac{RI(t)}{\pi a^2(B_N)} \quad t > t_o \quad \text{(stimulus current } I \text{ at soma)}$$

4. **Example** – This example problem will focus on the fundamental dendritic fork – the basic element in a dendritic arbor. Note that in this case, the soma is not a branchpoint.



It is easy to see that there are two levels of branching. According to the Hines ordering scheme, the dendrites are numbered as follows:



We proceed with solution by finite elements. The weak formulation:

$$\int_\Omega G(a^2V_x)_x U dx = \int_\Omega a[CV_t + g(V-E)]U dx$$

Examine each individual branch:

(a)

$$\int_{\Omega_1} G(a^2V_x)_x U dx = \int_{\Omega_1} a[CV_t + g(V-E)]U dx$$

(b)

$$\int_{\Omega_2} G(a^2V_x)_x U dx = \int_{\Omega_2} a[CV_t + g(V-E)]U dx$$

(c)

$$\int_{\Omega_3} G(a^2V_x)_x U dx = \int_{\Omega_3} a[CV_t + g(V-E)]U dx$$

Integrate by parts the left hand sides

(a)

$$Ga^2UV_x|_{B_1} - G\int_{\Omega_1} a^2V_xU_x dx = \int_{\Omega_1} a[CV_t + g(V-E)]U dx$$

(b)

$$Ga^2UV_x|_{B_2} - G\int_{\Omega_2} a^2V_xU_xdx = \int_{\Omega_2} a[CV_t + g(V-E)]Udx$$

(c)

$$\left.\frac{U\ I(t)}{2\pi}\right|_{B_3} - Ga^2UV_x|_{A_3} - G\int_{\Omega_3} a^2V_xU_xdx = \int_{\Omega_3} a[CV_t + g(V-E)]Udx$$

Current balance at a branch point dictates that

$$a^2V_x|_{A_3} = a^2V_x|_{B_1} + a^2V_x|_{B_2}$$

We can combine (a) (b) and (c) to obtain

$$\left.\frac{U\ I(t)}{2\pi}\right|_{B_3} - G\int_{\Omega_1} a^2V_xU_xdx - G\int_{\Omega_2} a^2V_xU_xdx - G\int_{\Omega_3} a^2V_xU_xdx =$$

$$\int_{\Omega_1} a[CV_t + g(V-E)]Udx + \int_{\Omega_2} a[CV_t + g(V-E)]Udx + \int_{\Omega_3} a[CV_t + g(V-E)]Udx$$

Rearrange to get

$$\begin{aligned}\left.\frac{U\ I(t)}{2\pi}\right|_{B_3} - G\int_{\Omega_1} a^2V_xU_xdx - G\int_{\Omega_2} a^2V_xU_xdx - G\int_{\Omega_3} a^2V_xU_xdx =& \\ C\int_{\Omega_1} aV_tUdx + C\int_{\Omega_2} aV_tUdx + C\int_{\Omega_3} aV_tUdx& \\ +\int_{\Omega_1} agVUdx + \int_{\Omega_2} agVUdx + \int_{\Omega_3} agVUdx& \\ -E\int_{\Omega_1} agUdx - E\int_{\Omega_2} agUdx - E\int_{\Omega_3} agUdx&\end{aligned}$$

For this example, let's assume a is constant over space. Write V in terms of hat functions ϕ, where h is the step size between nodes assigned to each branch:

$$\phi(x, h_l, h_r) = \begin{cases} 1/h(x-(i-1)h_l) & \text{if } x_{i-1} < x < x_i, \\ -1/h(x-(i+1)h_r) & \text{if } x_i \leq x < x_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The potential function is approximated by linearly composing hats. The potential function is discretized into a vector of $N_1 + N_2 + N_3$ coefficients with v_b the membrane potential at the single branchpoint. N_k denotes the number of nodes assigned to branch k (including the branchpoint for nonterminal branches).

$$V \approx \sum_{i=0}^{N_3} v_i(t)\phi_i(x, h_i, h_{i+1}) + \sum_{i=N_3}^{N_3+N_2} v_i(t)\phi(y, h_i, h_{i+1}) + \sum_{i=N_3+N_2}^{N_3+N_2+N_1} v_i(t)\phi_i(z, h_i, h_{i+1})$$

where $x \in \Omega_3$, $y \in \Omega_2$, $z \in \Omega_1$.

Substituting V, we get the system of equations

$$\frac{\phi_j(B_3)I(t)}{2\pi a} - Gav_i\sum_i\left[\int_{\Omega_1}(\phi_i)_z(\phi_j)_zdz - \int_{\Omega_2}(\phi_i)_y(\phi_j)_ydy - \int_{\Omega_3}(\phi_i)_x(\phi_j)_xdx\right] =$$

$$\sum_i C(v_i)_t\left[\int_{\Omega_1}\phi_i\phi_j dz+\int_{\Omega_2}\phi_i\phi_j dy+\int_{\Omega_3}a\phi_i\phi_j dx\right]$$
$$+\sum_i\sum_k g_k v_i\left[\int_{\Omega_1}\phi_k\phi_i\phi_j dz+\int_{\Omega_2}\phi_k\phi_i\phi_j dy+\int_{\Omega_3}\phi_k\phi_i\phi_j dx\right]$$
$$-E\sum_k g_k\left[\int_{\Omega_1}\phi_k\phi_j dz+\int_{\Omega_2}\phi_k\phi_j dy+\int_{\Omega_3}\phi_k\phi_j dx\right]$$
$$j=0\ldots N_1+N_2+N_3$$

The system in matrix vector notation:

$$\vec{I}(t)+\vec{B}=\mathbf{M}\vec{v}_t+(\mathbf{L}+\mathbf{K})\vec{v}$$

Where

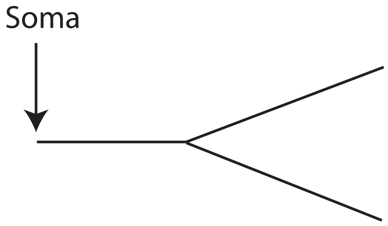
$$\mathbf{M}_{ij}=C\left[\int_{\Omega_1}\phi_i\phi_j dz+\int_{\Omega_2}\phi_i\phi_j dy+\int_{\Omega_3}a\phi_i\phi_j dx\right]$$
$$\mathbf{K}_{ij}=Ga\left[\int_{\Omega_1}(\phi_i)_z(\phi_j)_z dz-\int_{\Omega_2}(\phi_i)_y(\phi_j)_y dy-\int_{\Omega_3}(\phi_i)_x(\phi_j)_x dx\right]$$
$$\mathbf{L}_{ij}=\sum_k g_k\left[\int_{\Omega_1}\phi_k\phi_i\phi_j dz+\int_{\Omega_2}\phi_k\phi_i\phi_j dy+\int_{\Omega_3}\phi_k\phi_i\phi_j dx\right]$$
$$B_j=E\sum_k g_k\left[\int_{\Omega_1}\phi_k\phi_j dz+\int_{\Omega_2}\phi_k\phi_j dy+\int_{\Omega_3}\phi_k\phi_j dx\right]$$

5. Branch Ordering Revisited

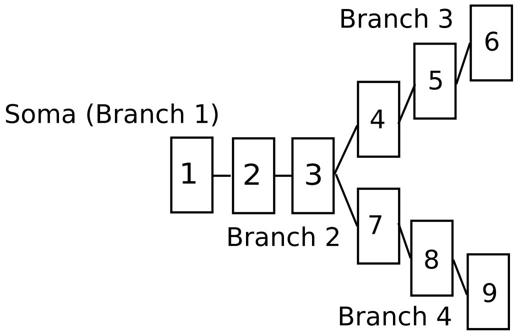
The spatial discretization essentially produce a matrix equation

$$\mathbf{A}\vec{x}=\vec{b}$$

to be solved for $\vec{x}$ at every time step. **A** not only contains numerical information; the neuronal morphology is encoded in the sparsity pattern. For the simulated single neurons, **A** is always symmetric and nearly tridiagonal. If the neuron consists of a soma and a straight fiber **A** will be tridiagonal. On the other hand, if each discretized compartment is a branch itself, **A** will hardly be tridiagonal. Though, if the Hines reordering method is used, **A** for the latter case will be essentially tridiagonal. Given the special nature of **A**, it is inefficient to use a general Guassian elimination algorithm to solve for $\vec{x}$, especially if the Hines ordering is used, in which case a barely modified tridiagonal solver, whose computation time scales proportional to the size of **A**, is sufficient. This example, the discretized triple-branched neuron, illustrates the importance of the Hines ordering method.



An intuitive method of numbering the nodes assigns increasing branch numbers moving away from the soma like so

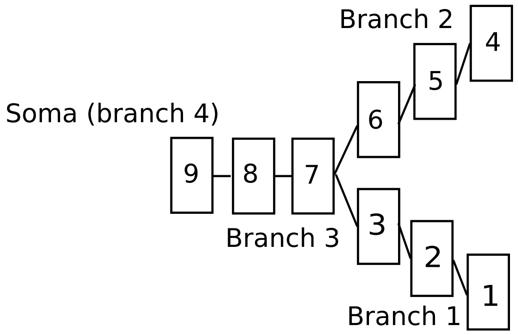


The left-hand-side matrix that corresponds to this numbering scheme is

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{3,2} & A_{3,3} & A_{3,4} & 0 & 0 & A_{3,7} & 0 & 0 \\ 0 & 0 & A_{4,3} & A_{4,4} & A_{4,5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{5,4} & A_{5,5} & A_{5,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{6,5} & A_{6,6} & 0 & 0 & 0 \\ 0 & 0 & A_{7,3} & 0 & 0 & 0 & A_{7,7} & A_{7,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{8,7} & A_{8,8} & A_{8,9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{9,8} & A_{9,9} \end{bmatrix}$$

A one-step-per-node diagonalizing algorithm used to eliminate the lower diagonal elements will run into problems at $A_{7,7}$ where unwanted fill-in of entry $A_{7,4}$ will occur – a more general elimination algorithm must be used. Consider a rather common pyramidal neuron with two large dendritic trees, ℓ branches in total (only bifurcating branches). Any numbering scheme that assigns higher branch numbers to branches that are "further away" (in terms of depth as defined earlier) will encounter at least $\ell/2 - 1$ entries in matrix $\mathbf{A}$ where a general elimination algorithm must be used. This dramatically lowers the overall efficiency of solving for $\vec{x}$.

The Hines method assigns larger branch indices to branches "closer" to the soma. Furthermore, the nodes within each branch end up numbered in increasing order moving closer to the branch/parent node.



This ensures that any branch in the neuron is only coupled to one other branch, which has a larger branch index. As such, $\mathbf{A}$ looks like

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{3,2} & A_{3,3} & 0 & 0 & 0 & A_{3,7} & 0 & 0 \\ 0 & 0 & 0 & A_{4,4} & A_{4,5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{5,4} & A_{5,5} & A_{5,6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{6,5} & A_{6,6} & A_{6,7} & 0 & 0 \\ 0 & 0 & A_{7,3} & 0 & 0 & A_{7,6} & A_{7,7} & A_{7,8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{8,7} & A_{8,8} & A_{8,9} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{9,8} & A_{9,9} \end{bmatrix}$$

A slightly modified tridiagonal solver is sufficient for the Hines matrix, which is essentially a tridiagonal matrix. Fill-in problems can now be avoided.