Numerical Solutions to the Passive Cable Equation: Using Finite Difference Approximations to Partial Derivatives

Marcello DiStasio
SUNY Downstate Medical Center & NYU-Poly

January 27, 2010

1 Second Order Difference Kernels

1.1 Finite Difference Approximation

Based on Taylor approximations, finite difference approximations to the derivatives of a function defined on a uniform grid $(x_i)\,,\, x_i = x_o + h\,,\, h > 0$ can be given using arbitrarily sized kernels. For *second order kernels*, we obtain the following 'forward difference' approximation to the first derivative:

$$f'(x) \simeq \frac{f(x_{i+1}) - f(x_i)}{\Delta x} + O(\Delta x)$$

and a 'center difference' approximation to the second derivative:

$$f''(x) \simeq \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{\Delta x^2} + O(\Delta x^2)$$

Note that these are not the only difference formulas that can be defined as computational molecules on the mesh; others such as backward or centered difference or the Crank-Nicolson molecule have various computational advantages related to ensuring stability while using differently spaced meshes.

1.2 Application to Cable Equation

The cable equation describes the passive spread of voltage in one dimension along a cable with an inner ('axial') conductor and outer ('membrane') resistive covering. The cable equation in a second order parabolic PDE, similar in form to the heat equation that describes the distribution of thermal energy along a uniform bar as a function of time and distance from the origin. The voltage in a cable as a function of time t and distance along the cable x can be given by

$$C_m \frac{\partial V}{\partial t} = \frac{1}{4R_i} \frac{\partial^2 V}{\partial x^2} - G_m V - R_m J$$

where C_m is the membrane capacitance; R_i is the longitudinal (internal) resistance; G_m is the membrane conductance ($G_m = \frac{1}{R_m}$, where R_m is the membrane resitance); and J is the injected current.

If we let $v_j^n = V(x_j, t_n)$, and partition time in steps Δt and space in steps Δx, we can use the second order difference approximations above to write the cable equation as

$$C_m \left[\frac{v_i^{n+1} - v_i^n}{\Delta t} \right] = \frac{d}{4R_i} \left[\frac{v_{i-1}^n - 2v_i^n + v_{i+1}^n}{\Delta x^2} \right] - G_m v_i^n - J_j^n$$

Rearranging terms so that the current terms at any grid step (v_j^n) can be expressed in terms of past values ($v_{j-j_m}^{n-n_m}$) we obtain

$$v_i^{n+1} = v_i^n + \frac{d\Delta t}{4R_i C_m \Delta x^2}(v_{i-1}^n - 2v_i^n + v_{i+1}^n) - \frac{\Delta t G_m}{C_m} v_i^n - J_i^n$$

Now we let $K = \frac{d\Delta t}{4R_i C_m \Delta x^2}$ and we let $B = \frac{-\Delta x^2 4R_i G_m}{d}$. Then we can write v_i^{n+1}as:

$$v_i^{n+1} = Kv_{i-1}^n + (1 - 2K - BK)v_i^n + Kv_{i+1}^n$$

We can write this in vector form if we take

$$\begin{array}{lllll} \mathsf{v}^n & = (v_i^n) & = & {}^t[\ldots, v_{i-1}^n, v_i^n, v_{i+1}^n, \ldots] & \text{and} \\ \mathsf{v}^{n+1} & = (v_i^{n+1}) & = & {}^t[\ldots, v_{i-1}^{n+1}, v_i^{n+1}, v_{i+1}^{n+1}, \ldots] & \text{and} \\ \mathsf{j}^n & = (J_i^n) & = & {}^t[\ldots, J_{i-1}^n, J_i^n, J_{i+1}^n, \ldots] & \end{array}$$

Then we have

$$\mathsf{v}^{n+1} = \mathsf{A}\mathsf{v}^n - \mathsf{j}^n$$

where A is defined as

$$\mathsf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ K & 1-K(2+B) & K & 0 & & & \\ 0 & K & 1-K(2+B) & K & & \vdots & \vdots \\ 0 & 0 & K & 1-K(2+B) & & K & 0 \\ 0 & 0 & 0 & K & \ddots & 1-K(2+B) & K \\ 0 & 0 & 0 & 0 & \ldots & K & 1 \end{bmatrix}$$

Thus, given a vector v^0 specifying the intial condition, you can simply iterate forward in time, repeatedly applying the matrix A to the output of each previous step. Note that the top and bottom rows are set to 1 (identity) to preserve the boundary condition. In the simulations below, Dirichlet (fixed) boundary conditions are used, with the voltage fixed are resting potential (analagous to a sealed, but not killed, axon ending). A simple MATLAB function implementing this entire method is given in the file CabSecondOrder.m.

2 Fourth Order Difference Kernels

For *fourth order kernels*, following the conventions above we obtain the following finite difference approximations:

$$f'(x) \simeq \frac{-f(x_{i-3}) + 6f(x_{i-2}) - 18f(x_{i-1}) + 10f(x_i) + 3f(x_{i+1})}{12\Delta x} + O(\Delta x^4)$$

$$f''(x) \simeq \frac{-f(x_{i-2}) + 16f(x_{i-1}) - 30f(x_i) + 16f(x_{i+1}) - f(x_{i+2})}{12\Delta x^2} + O(\Delta x^4)$$

Note that the fourth order kernels are not symmetrical about x_i, which would lead to a linear decrease in the error (but not change its order), but the asymmetrical kernel is used to simplify the programming. Here we let $K = \frac{d\Delta t}{C_m R_i \Delta x^2}$ and $B = (-30 - 30K - \frac{4\Delta t G_m}{C_m})$ to yield the following matrix:

$$\mathsf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & & & & 0 \\ 16K & B & -10K & -K & 0 & \cdots & & & & \\ -K & 16K & B & -10K & -K & & & & & \vdots \\ 0 & -K & 16K & B & -10K & & \ddots & & & \\ & & & 16K & & \ddots & & & & 0 \\ \vdots & & & & & & 16K & B & -10K & -K \\ & & & & & 0 & -K & 16K & B & -10K \\ 0 & & \cdots & & & \cdots & 0 & 0 & 0 & 1 \end{bmatrix}$$

Then we can calculate the voltage vectors at each time step (as defined above) by:

$$\mathsf{v}^{n+1} = \mathsf{A}\mathsf{v}^n + 54\mathsf{v}^{n-1} - 18\mathsf{v}^{n-2} + 3\mathsf{v}^{n-3} - \mathsf{j}^n$$

A MATLAB function implementing this method can be found in the file CabFourthOrder.m.

3 Computational Issues

The requisite density of the mesh spacing is highly sensitive to the parameters K and B, particularly for the fourth order difference method. In practice, the mesh size required to ensure a stable solution to the cable equation with realistic biological parameters using the fourth order method is too fine to allow for simulation on a standard desktop computer due to memory restrictions. Thus, to approximate higher order differences, more efficient computational algorithms such as Crank-Nicolson or Runge-Kutta should be employed.

4 Results

4.1 Comparison with results from NEURON

The following figure shows the voltage trace at the current injection site computed using the parameters:

- $R_m = 1 \times 10^3\ \Omega/cm$
- $R_i = 35.4\,\Omega \cdot cm$
- $C_m = 1 \times 10^{-6}\,F/cm^2$
- $d = 1 \times 10^{-13} cm$ (again, this is smaller than in reality)
- $T_{max} = .005\,s$
- $L = 0.01\,cm$
- $\Delta t = 1 \times 10^{-5} s$
- $\Delta x = 2 \times 10^{-4} cm$

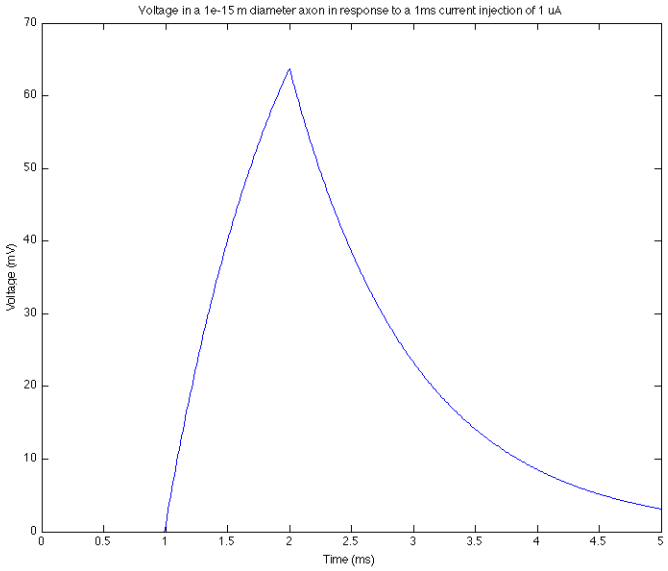

Figure 1: Voltage at the location of current input as a function of time as computed with the algorithm delineated in this report

Running the simulation in the NEURON environment (see http://www.neuron.yale.edu/) with the same parameters yields the following voltage trace:

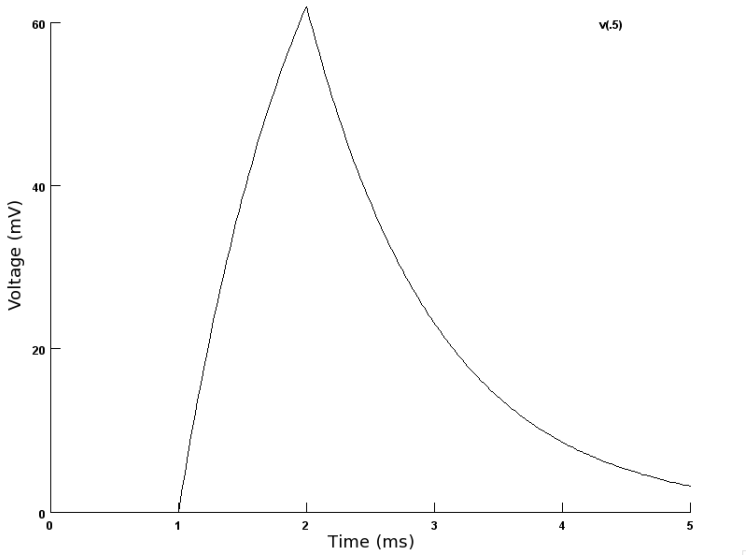

Figure 2: Voltage at the location of current input as a function of time as computed with the NEURON software

The similarity in shape (it is seen on further inspection that there are slight differences in amplitude) serves to validate the general approach outline above. However, the simulation of realistic biological parameters (similar to the default parameters given in NEURON) is computationally difficult for the reasons outlined above.

4.2 Visualizing current spread

The solution to the cable equation was computed using the second order difference method with the following parameters:

- $R_m = 2.5 \times 10^{11}\ \Omega/cm$
- $R_i = 29.7\,\Omega \cdot cm$
- $C_m = 1 \times 10^{-6}\,F/cm^2$
- $d = 1 \times 10^{-14} cm$ (this is smaller than in reality, but was used to allow a wider mesh space and thus allow the use of less memory)
- $T_{max} = 1.5\,s$
- $L = 0.001\,cm$
- $\Delta t = 6 \times 10^{-4} s$
- $\Delta x = 2 \times 10^{-6} cm$

A current injection of $-0.01 \mu A$ with a duration of 1 second was applied to the cable at a single location. The resulting voltage is shown below (Fig. 3) as a function of space and time.

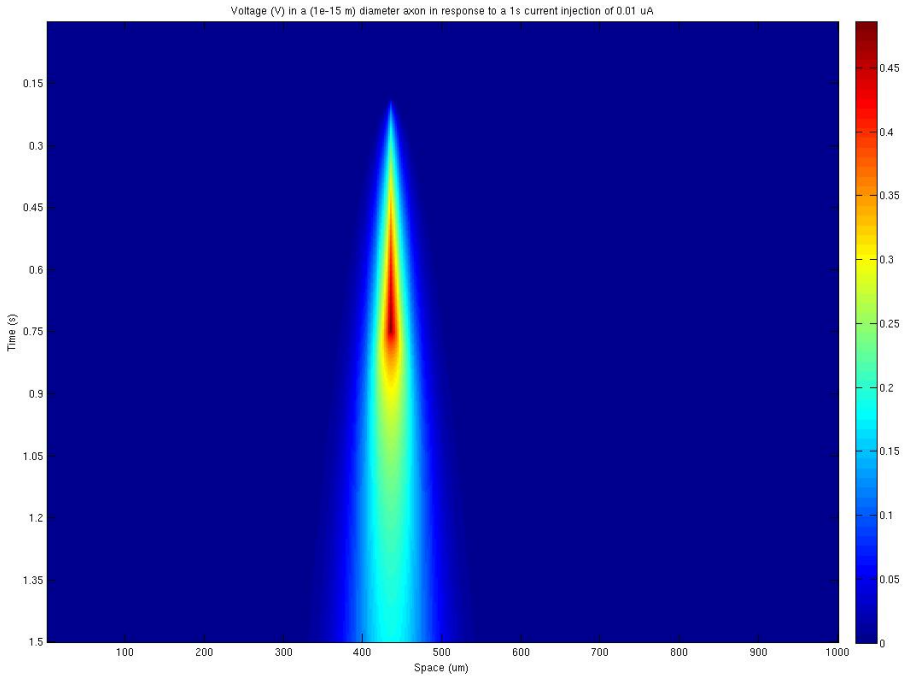

Figure 3: Spread of potential along the cable with a uniform current input at a single location

4.3 Response to s sinusoidal current input

Using the same parameters as section 4.2 the voltage response to the input of a sine wave current amplitude was calculated. The results are shown in Fig. 4, which uses the same conventions as the figures above.

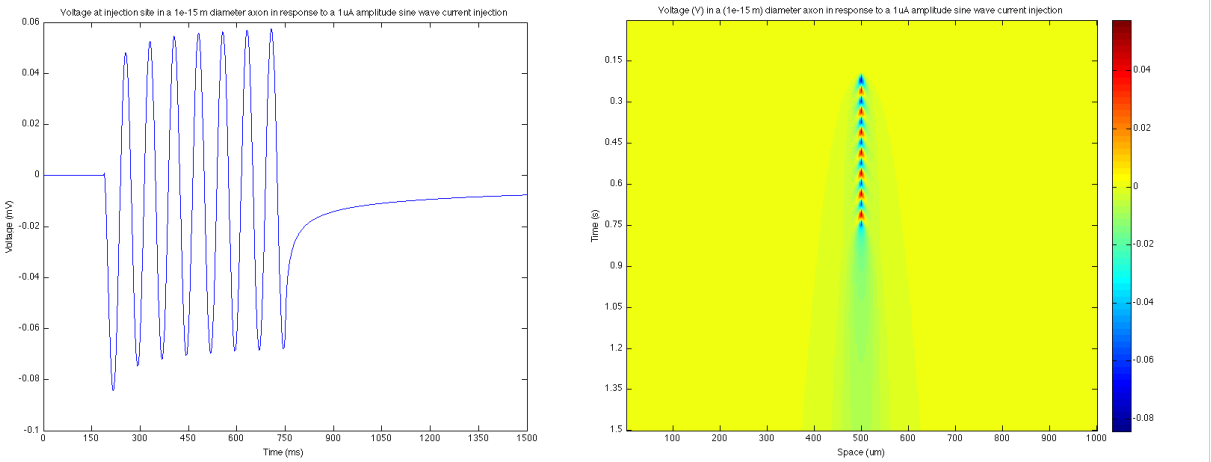

Figure 4: Voltage response in the $10\mu m$ length cable to a current input at a single location with a sinusoidally varying amplitude

5 Discussion

The comparison of the numerical solution presented above with a model run in NEURON reveals that our method yields very reasonable results when run with parameters that fulfill criteria for stability. The dynamics of the voltage change in repose to current inputs match quite well. The figures also show a pattern of voltage spread that matches expectation based on experiment. Nevertheless, because differential equations are inherently functions of both the independent variables and the discrete step size in those variables (e.g. Δx), the computation of the solution using this method has some key limitations:

- Computational efficiency: Simulation of completely biologically realistic parameters required a mesh size too small to achieve a guarantee (or even possibility) of a stable solution. This can be predicted by the value of K in the second order case, which should be less than $\frac{1}{2}$ to ensure stability (likewise for B in the fourth order case – a condition that could not be met in useful simulations using the memory available on a desktop computer).
- Stability (oscillations): The computation introduces error into the solution that is guaranteed to be less than the O terms in the formula, but these errors may wax and wane periodically. Such behavior can be observed in the color map plots in the spatial (x) direction, where ripples are seen in regions that should be smoothly decreasing.

There are a number of other options for computing the difference approximations. One simple change that would improve the stability of the solution without changing the computational load would be to replace the forward difference approximation to the first derivative with a centered difference approximation, as in $f'(x) \simeq \frac{f(x_{i+1})-f(x_{i-1})}{2\Delta x} + O(\Delta x^2)$. In this case the error would only be proportional to the square of the time step (and thus would be reduced, since $\Delta t < 1$). Finally, as mentioned above, in many situations the use of more efficient and more stable algorithms (Runge-Kutta, Crank-Nicolson), which are in packages already widely available, would remedy the problems described here.

6 References

1. Christof Koch. Biophysics of Computation. Information Processing in Single Neurons. 1999. Oxford University Press.
2. http://reference.wolfram.com/mathematica/tutorial/NDSolvePDE.html#c:4
3. Gerald W. Recktenwald. Finite-Difference Approximations to the Heat Equation. http://web.cecs.pdx.edu/~gerry/class/M
4. Numerical Analysis of Partial Difference Equations. Hall and Porsching. 1990. Prentice Hall.

5. Numerical Solution of Partial Difference Equations: Finite Difference Methods 3rd ed. G.D. Smith. 1985. Oxford University Press.

6. NEURON for empirically-based simulations of neurons and networks of neurons. http://www.neuron.yale.edu/neuron