Hines, M. and Carnevale, N. T. (1994). Computer Simulation Methods for Neurons. In: *The Handbook of Brain Theory and Neural Networks* M. Arbib (ed.) MIT Press.

Computer Simulation Methods for Neurons

Michael Hines[1,2] **and Nicholas T. Carnevale**[1,3]
[1]Center for Theoretical and Applied Neuroscience,
[2]Dept. of Computer Science, and [3]Dept. of Psychology
Yale University, New Haven, CT 06520-8285

March 24, 1994

Introduction

Computer simulation allows us to compare the behavior of an idealized model (a computational model) with the observed behavior of a biological system (an experimental model). We see this relationship in terms of two phases of experimental inquiry. The first phase starts when the investigator decides to study a general phenomenon in a particular experimental system, formulates hypotheses or conceptual models, and judges which components of the system seem most important. This phase continues with the design of experiments, selection of experimental subjects, and interpretation of data to characterize these components. Each of these activities imposes inevitable choices and approximations: which details are essential, and which can be omitted? Simulation, the second phase of this experimental inquiry, faces a question that is superficially more straightforward: does recombining the components create a model that reproduces the behavior of the original system? However, answering this question forces another series of choices and approximations. Regardless of the degree of behavioral similarity that is achieved, every simulation is an analogy compounded from multiple levels of approximation, each of which must be carefully examined. The first level of approximation lies in our speculation about which components of the conceptual model are most important and how faithfully they must be represented in a mathematical or computational model to capture the essential features of the phenomena displayed by the biological system. In the case of electrical signals in neurons, the classic approximations are to describe the spread of current and voltage in the longitudinal direction with one-dimensional electrical cable theory, and to use phenomenological descriptions of membrane conductances to emulate transmembrane ionic currents.

The second level of approximation lies in the relation between the abstract computational model and its particular instantiation on the computer. Making something work on a computer compels us to move back from the abstract to the concrete. This often means adding details we'd rather ignore, or worse, being forced to invent them. Also, the computer program that embodies this instance necessarily contains a huge amount of scaffolding allowing one to interact with the computer model, solve the model equations, etc.. It may be very difficult to verify that all this scaffolding is functioning correctly for this particular instance.

Information processing in neurons is accomplished by the spread and interaction of electrical and chemical signals. These signals are characterized by spatial nonuniformity and complex temporal dynamics that are intrinsic to the operation of biological neural networks. Models of neurons or neural nets that are closely linked to experimentation must preserve the essential spatiotemporal features of these signals.

This article focuses on a single but critical part of the process of creating a computer program for neural simulation: the numerical methods for solving the cable equation with voltage dependent channel conductances. We concentrate on electrical signaling for two reasons. First, many models that include electrical signals while ignoring chemical signals have proven useful. Second, both electrical and chemical signaling are susceptible to the same computational strategies, since both are examples of the broad class of reactive diffusion problems (Oran and Boris 1987; Carnevale and Rosenthal 1992a,b) and are described by parabolic partial differential equations.

We attempt to provide an intuitive rationale for the methods. The only mathematical prerequisite is some algebra and two concepts from first year calculus that are used very intuitively: differentiation and Taylor's theorem.

Of the many previous articles that discuss numerical methods for solving the cable equation, Mascagni's chapter in Koch and Segev (1989) is notable for a reasonably complete explanation and bibliography. Douglas (1961) gives a fairly rigorous account of numerical methods for solving parabolic partial differential equations. Hines (1984) discusses special techniques for fast simulations of neurons. A textbook with good general coverage of numerical methods for differential equations has been written by Dahlquist and Bjö rck (1974)

The Basic Approach

The easiest and most commonly used method for dealing with quantities that vary continuously with position and time is to approximate the differential equations that describe the neuron's electrical behavior with difference equations. This involves replacing continuous derivatives by finite differences in a manner analogous to the definition of the derivative

$$\left.\frac{df}{dx}\right|_{x=x_o} \approx \frac{f(x_o + h) - f(x_o)}{h} \tag{1}$$

However, in a difference equation we don't take the limit as h goes to zero but instead treat it as a small constant. This converts differential equations into a set of algebraic equations (one for each discrete value of x and t) that can be solved by the computer. The practical issue is to choose a difference replacement that optimizes the accuracy of the simulation results, the time it takes for the computer to carry out the arithmetic, and the overhead involved in determining that a given method yields correct results for the specific discretization (Δx, Δt) of a particular simulation.

The Cable Equation

The physical principle of conservation of charge is combined with Ohm's law to derive the cable equation. We focus on these separately to provide insight into the process of spatial discretization and the meaning of boundary conditions.

Conservation of charge at any point along the cable requires that the sum of currents flowing into a region from all sources (e.g., adjacent interior regions, transmembrane ionic fluxes, and microelectrodes) must equal zero.

$$\sum i_a - \int i_m dA = 0 \tag{2}$$

where the sum is over all the axial currents i_a in units of charge/time (e.g. milliAmperes) flowing into the region through cross-section boundaries, i_m is the transmembrane current density (mA/cm^2), and the integral is taken over the membrane area A of the region. This is illustrated in Figure 1. The usual convention is that outward transmembrane current flow is positive and axial current flow into a region is positive.

The standard approach in computer simulation is to divide the neuron into regions or compartments small enough that the spatially varying i_m

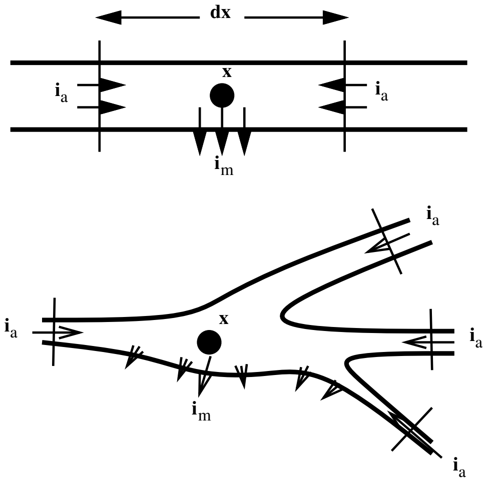

Figure 1: The sum of axial currents flowing into a region equals the current flowing out of the membrane area. Note that the axial current i_a varies with position.

in any compartment j is well approximated by its value at the center of compartment j. Therefore equation 2 becomes

$$i_{mj} A_j = \sum i_a \tag{3}$$

where A_j is the surface area of compartment j. Each axial current at the cross-section between compartment j and an adjacent compartment k is approximated by the voltage drop between the centers of the compartments divided by the resistance of the path between them. This transforms equation 3 into

$$i_{mj} A_j = \sum_k \frac{V_k - V_j}{r_{jk}} \tag{4}$$

This automatically takes care of the direction of axial current flow, since $V_j < V_k$ means that current flows into the compartment. The total membrane current $i_{mj} A_j$ is the sum of capacitive and ionic components, $c_j \frac{dV_j}{dt} + I(V_j, t)$, where c_j is the membrane capacitance of the compartment and $I(V_j, t)$ includes the effects of varying ionic channel conductances. In summary, the spatial discretization of branched cables yields a set of ordinary differential equations of the form

$$c_j \frac{dV_j}{dt} + I(V_j, t) = \sum_k \frac{V_k - V_j}{r_{jk}} \tag{5}$$

Injected currents would be added to the right hand side of this equation.

In deriving this equation we made two approximations: representing spatially varying i_m by its value at the center of each compartment, and specifying axial current i_a in terms of the voltage drop between the centers of adjacent compartments. If the compartments are of equal size, it is easy to use Taylor's series to show that both of these approximations have errors proportional to the square of compartment length. Thus doubling the number of compartments reduces the error by a factor of four.

Forward Euler: simple, inaccurate, unstable

Spatial discretization reduced the cable equation to a set of equations with first order derivatives in time. This section covers the simplest possible method for solving these kinds of equations, Euler's method. While it can suffer from low accuracy and can be numerically unstable; it has the advantage of being the easiest to understand, provides concrete examples of the concepts of accuracy and stability, and is a platform from which we can branch out to more complicated methods that have fewer pitfalls.

We illustrate Euler's method with models of a passive neuron (constant membrane resistance) that have only one or two compartments. In the one compartment model the right hand side of Eq. 5 disappears, so

$$\frac{dV}{dt} + kV = 0 \tag{6}$$

where the membrane capacitance and conductance have been subsumed in the constant k — the inverse of the membrane time constant. In this easy linear case we can readily compare the results of our computer methods to the analytic solution of this equation, which is

$$V(t) = V(0)e^{-kt} \tag{7}$$

The numerical methods that we use to understand and control the error are immediately generalizable to the nonlinear case.

Euler's method says that, since we know the initial value of our dependent variable ($V(0)$, given by the initial conditions) and its initial slope

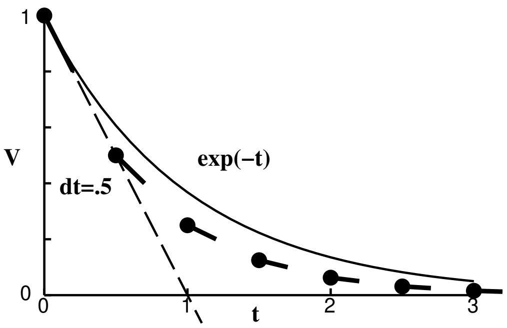

Figure 2:
Simulate the behavior of equation 6 by successively marching forward by a fixed interval assuming the current is constant within each interval. The current value that is used for a given interval is found from the value of the voltage at the beginning of the interval. The voltage values are shown by filled circles. The slope of the line segment emanating from the voltage value depends on the current *at that time step*. The dashed line shows the value of the voltage after the first time step as a function of Δt.

$(-kV(0)$, given by the equation), let's assume the slope is constant for a short period of time and extrapolate to a new value a brief interval into the future. In Fig 2, we start with the initial condition $V(0) = 1$ and use a rate parameter $k = 1$. The time interval over which we extrapolate is $\Delta t = 0.5$.

Consider the error in Fig. 2. For the first few time steps the absolute error increases but then decreases as the analytic solution and the simulation solution approach the same steady state ($V = 0$). The total or global error is a combination of errors from two sources. First, there is the local error, which is due to the extrapolation process within a time step. This is easily analyzed with Taylor's theorem truncated at the term proportional to Δt^2

$$V(t + \Delta t) = V(t) + \Delta t V'(t) + \frac{\Delta t^2}{2} V''(t*), t \leq t* \leq t + \Delta t \tag{8}$$

Euler's method ignores the second order term, so the error at each step is proportional to Δt^2. Integrating over a fixed time interval T requires $T/\Delta t$ steps, so the error that accumulates in this interval is on the order of $\Delta t^2 \cdot T/\Delta t$, i.e. proportional to Δt . We can always decrease the error as much as we like by reducing Δt.

The second contribution to the total error has to do with the cumulative effect of past errors, which have moved the computed solution away from the trajectory of the analytic solution. Thus, if our computer solution has a nonzero total error at time t_1, then even if we were to thereafter solve the equations exactly using the state values at t_1 as our initial condition, the future solution will be inaccurate because we are on a different trajectory.

The total error of the simulation is therefore not easy to analyze. In the example of Fig. 2, all trajectories end up at the same steady state so total error tends to decrease, but not all systems behave in this manner. Particularly treacherous are systems that behave chaotically so that, once the computed solution diverges even slightly from the proper trajectory, it subsequently moves rapidly away from the original and the time evolution becomes totally different.

The question is not so much how large the error of a simulation is relative to the analytic solution but whether the simulation error puts us on trajectories that are different from the set of trajectories defined by the error in our parameters. There may be some benefit in treating the model equations as sacred runes which must be solved to an arbitrarily high precision — removal of any source of error has value. But judgment is required in order to determine the meaning of a simulation run. For example, consider the Hodgkin-Huxley membrane action potential elicited by a short but strong current stimulus and one elicited by a much weaker stimulus. The top panel of Fig. 3 compares these action potentials with those calculated by Euler's method using a time step of 25 μs. While the voltage hovers near threshold, a little bit of error due to our time step is amplified into a considerable error in the actual time of occurrence of the spike. However the behavior around the threshold is highly sensitive to almost any parameter as is seen by changing the sodium channel density by only 1%. Clearly it is crucial to know the sensitivity of our results to every parameter of the model, and the time step is just one more parameter which is added as a condition of being able to simulate a model on the computer.

It might seem that using extremely small Δt would be the best way to reduce error. However, computers represent real numbers as floating point numbers with only a fixed number of digits, so if you keep adding 10^{-20} to 1 you may always get a value of 1 — even if you iterate the process 10^{20} times. Operations that involve the difference of similar numbers, as when differences are substituted for derivatives, are especially prone to roundoff error. Consequently there is a limit to the accuracy improvement that can be achieved by decreasing Δt.

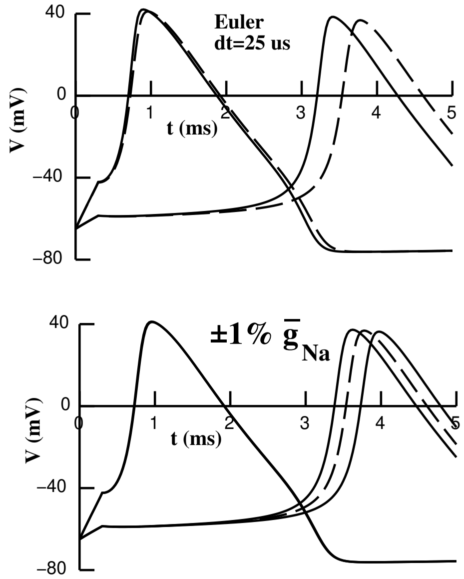

Figure 3: Hodgkin Huxley membrane action potentials elicited by a current stimulus of duration 0.3 ms and amplitudes 0.08 mA/cm^2 and 0.022 mA/cm^2. In the top panel the solid lines are for Euler's method with $\Delta t = 0.025$ ms, and the dashed line is computed using a very much smaller Δt. All action potentials are calculated with $\overline{g}_{Na} = 0.12 mA/cm^2$. The bottom panel shows very accurate simulations with $\overline{g}_{Na} \pm 1\%$. In this panel, the three simulations that involve the large stimulus superimpose.

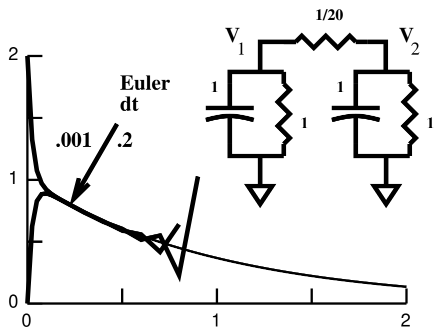

Figure 4: Euler's method is numerically unstable whenever Δt is greater than twice the smallest time constant. The analytic solution is the sum of two exponentials with time constants 1 and 1/41. The solution step size is 0.001 ms for the first 0.2 ms, after which it is increased to 0.2 ms.

Generally speaking it would be nice to be able to use what might be called "physiological" values of Δt. That is, time steps that give a good representation of the state trajectories without having a numerical accuracy that is many orders of magnitude better than the accuracy of our physiological measurements.

Numerical stability

What would happen if the time step was very large in Figure 2, e.g. $\Delta t = 3$? Now, the first step would extrapolate down to $V = -2$, the second step would end up at $V = -2 + 6 = 4$, and each successive step would oscillate with geometrically increasing magnitude – the simulation is numerically unstable. An important aspect of instability is most easily illustrated with a two compartment simulation in which the compartments are connected by a small axial resistance so that the membrane potentials are normally in quasi-equilibrium and at the same time are decaying fairly slowly. Figure 4 shows the time course of the two compartments when the initial condition is $V = 0$ in one and $V = 2$ in the other. Now, if we use the Euler method with $\Delta t = 0.5$ we realize that there will be a great deal of trouble during the time where the voltages are changing rapidly, so we might think that all we need to do is choose a Δt which will carefully follow the time course of the voltage changes, i.e. a Δt which is small when they are changing rapidly and larger when they are changing slowly. Figure 4 illustrates the results of this strategy as well. After 0.2 units of time with $\Delta t = 0.001$, the two voltages have nearly come into equilibrium. Then we changed to $\Delta t = 0.2$ — still small enough to follow the slow decay closely. Unfortunately what happens is that, no matter how small the difference between the voltages (even if it consists only of roundoff error), the difference grows geometrically at each time step. For Euler's method, the time step must never be more than twice the smallest time constant in the system.

The notion of "time constant" and its relationship to stability are clarified by appealing to linear algebra. For a linear system with N compartments, there are exactly N spatial patterns of voltage over all compartments such that only the amplitude of the pattern changes with time, and the shape of the pattern is preserved. These patterns, called eigenvectors, have the property that the time course of change of the ith pattern is given by $e^{\lambda_i t}$, where $\lambda_i$is called the eigenvalue of the ith eigenvector. If the real part of λ_i is negative, then the ith pattern decays exponentially to 0; if the real part is positive, the amplitude grows catastrophically. If λ_i has an imaginary component, then the pattern oscillates with frequency, $\omega_i =$Im(λ_i).

Our two compartment model has two such patterns. In one pattern, the voltage is the same in both compartments. This pattern decays with the time course e^{-t}. The other pattern, in which the voltages in the two compartments are equal but opposite in sign, decays with the time course e^{-41t} . The key idea is that a problem involving N coupled differential equations can always be transformed into a set of N independent equations, each of which is solved separately as in the single compartment of Equation 6. It is essential to use a small enough Δt that the solution of each equation is stable. This is the reason why stability criteria that involve Δt depend on the smallest time constant.

Systems that have a very large ratio between their slowest and fastest time constants are said to be stiff. Stiffness is a serious problem because we may need to use a small Δt to follow changes due to the fast time constant, while running the simulation for a very long time in order to observe changes governed by the slow time constant.

Whether an imposed driving force changes the stability properties depends on whether it alters the time constants that describe the system. A current source (perfect current clamp) will not change the time constants and therefore will not affect stability. Any other signal source introduces a load into the compartment to which it is attached, changing the time constants and their corresponding eigenvectors. The more closely it approx-

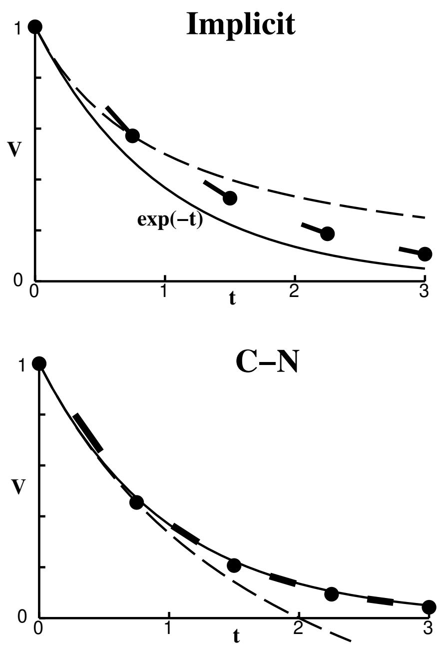

Figure 5: First order fully implicit (backward Euler) and second order Crank-Nicholson methods. At the end of each implicit step the slope at the new value points back to the beginning of the step. In the Crank-Nicholson method the slope at the midpoint of the each step is used to determine the new value. The dashed lines show the voltage after the first time step as a function of Δt.

imates a voltage source (perfect voltage clamp), the greater this effect will be.

Implicit (Backward Euler) method: inaccurate but stable

It turns out that we can avoid the numerical stability problems of Euler's method by evaluating the equations at time $t + \Delta t$, i.e.

$$V(t + \Delta t) = V(t) + \Delta t f(V(t + \Delta t), t + \Delta t) \tag{9}$$

This is called the implicit method or sometimes "backwards Euler" since it is derived from Taylor's series truncated at the Δt term but with $t + \Delta t$ in

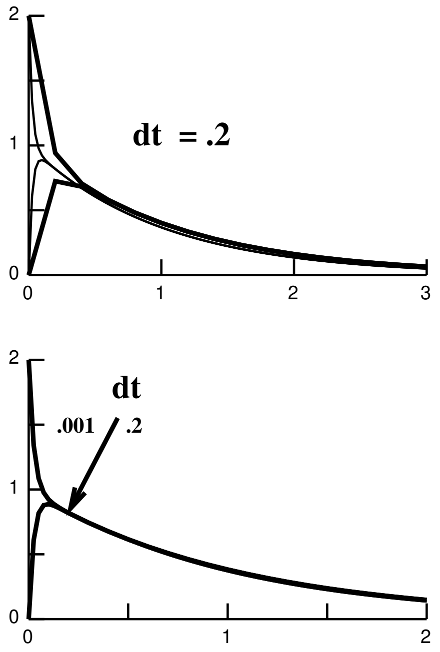

Figure 6: Two compartments as in Figure 4 simulated with fully implicit method. Top: Δt is much larger than the fast time constant. Bottom: for the first 0.2 time units, Δt is small enough to accurately follow the fast time constant. Thereafter, Δt is increased to 10 times the fast time constant, yet the simulation remains numerically stable.

place of t. For our simple example we have

$$V(t + \Delta t) = \frac{V(t)}{1 + k\Delta t} \tag{10}$$

Several iterations are shown in the top half of Fig. 5. At each step we move to a new value such that the slope there points back to the beginning of the step. If Δt is very large, then instead of geometrically increasing error oscillations, we get an exponential approach to the steady state.

The attractive stability properties of the implicit method are illustrated in Figure 6 where we use it to simulate the two compartment model. Notice that although a large Δt does not allow us to follow the fast changing concentrations at the early times, it does give a reasonable qualitative understanding of the behavior. Also the artifice of changing the step size depending on how quickly the states are changing does work now without blowing up on us. Of course, as in Euler's method, the global error is proportional to Δt for small Δt. Unfortunately we now have to solve a set of nonlinear simultaneous equations at each step. Obviously this takes a lot of extra work and we'll want to use a step size a large as possible while still getting good quantitative accuracy. It is safest to use the first order implicit method for initial exploratory simulations because its robust stability properties give fast simulations that are almost always qualitatively correct given reasonable values of Δt, and one does not have to worry about large error oscillations between very tightly coupled compartments that really should quickly come into equilibrium.

Central Difference (Crank-Nicholson) Method: stable and more accurate

This motivates us to look into a variation of the implicit method which has global error that is proportional to the square of the step size and is basically just an average of the two methods (or more precisely, advancing by one half step using the implicit method and then advancing one half step using Euler's method). The bottom half of Figure 5 illustrates the idea. The value at the end of a step is along the line determined by the estimated slope at the midpoint of the step.

Generally, for a given Δt we can expect a large accuracy increase with the central difference or Crank-Nicholson method (Crank and Nicholson, 1947). It's proper to wonder, though, what effect the Euler half step has on numerical stability. Figure 7 shows the two compartment model using this central difference method with a much larger Δt than the fast time constant. We say the method is stable because the error oscillations eventually decay away. Clearly, this method will also work with a variable time step approach. What is going on here is that the combination of implicit half step followed by an explicit half time step approximates an exponential decay by

$$V(t + \Delta t) = V(t)\frac{1 - \Delta t/2}{1 + \Delta t/2} \tag{11}$$

As Δtgets very large the step multiplier approaches -1 from above so the solution oscillates with decreasing amplitude.

The most interesting feature of the central difference method is that the amount of computational work for the extra accuracy beyond the backward

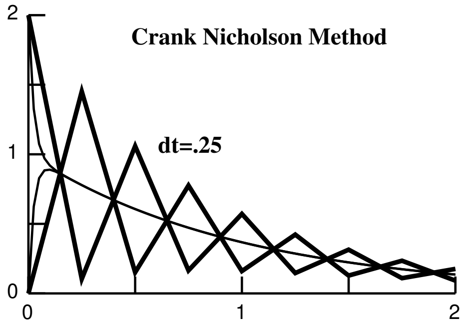

Figure 7: The Crank-Nicholson method can have significant error oscillations when there is a large amplitude component in the simulation that has a time constant much smaller than Δt. However, the oscillation amplitude decreases at each step, so the simulation is numerically stable.

difference method is trivial since after computing $V(t + \Delta t/2)$ we just have

$$V(t + \Delta t) = 2V(t + \frac{\Delta t}{2}) - V(t) \tag{12}$$

so the extra accuracy does not cost extra computations of the model functions.

This completes the basic ideas involved in simulation of branched cables. But several details about the structure of these equations can be exploited to greatly reduce the computation time over that required to solve general sets of nonlinear ordinary differential equations.

Efficiency

Nonlinear equations generally need to be solved iteratively to maintain second order correctness. However, voltage dependent membrane properties, which are typically formulated in analogy to Hodgkin-Huxley (HH) type channels, allow the cable equation to be cast in a linear form, still second order correct, that can be solved without iterations. A direct solution of the voltage equations at each time step $t \rightarrow t + \Delta t$ using the linearized membrane current $I(V,t) = G \cdot (V - E)$ is sufficient as long as G and E are known to second order at time $t + 0.5\Delta t$. (Note: G is called the slope conductance (dI/dV) and E is the effective reversal potential). HH type channels are easy to solve at $t + 0.5\Delta t$ since the conductance is a function of state variables which can be computed using a separate time step that is offset by $0.5\Delta t$ with respect to the voltage equation time step. That is, to integrate a state from $t - 0.5\Delta t$ to $t + 0.5\Delta t$ we only require a second order correct value for the voltage dependent rates at the midpoint time t. Figure 8 contrasts this approach with the common technique of replacing nonlinear coefficients by their values at the beginning of a time step. For HH equations in a single compartment, the staggered time grid approach converts four simultaneous nonlinear equations at each time step to four independent linear equations that have the same order of accuracy at each time step.

Neuronal architecture can also be exploited to increase computational efficiency. Since neurons generally have a branched tree structure with no loops, the number of arithmetic operations required to solve the cable equation by Gaussian elimination is exactly the same as for an unbranched cable with the same number of compartments. That is, we need only O(N) arithmetic operations for the equations that describe N compartments connected in the form of a tree, even though standard Gaussian elimination generally takes O(N^3) operations to solve N equations in N unknowns. The tremendous efficiency increase results from the fact that, in a tree, one can always find a leaf compartment i which is connected to only one other compartment j, so that: 1) the equation for compartment i (Equation 13a) involves only the voltages in compartments i and j, and 2) the voltage in leaf compartment i is involved only in the equations for compartments i and j (Equations 13 and 14).

$$a_{ii}V_i + a_{ij}V_j \qquad = b_i \tag{13}$$

$$a_{ji}V_i + a_{jj}V_j \quad + \text{terms from other compartments} \quad = b_j \tag{14}$$

Using Equation 13 to eliminate the V_i term from Equation 14, which requires O(1) (instead of N) operations, gives Equation 15 and leaves N-1 equations in N-1 unknowns.

$$a'_{jj}V_j + \text{terms from other compartments} = b'_j \tag{15}$$

where $a'_{jj} = a_{jj} - (a_{ij}a_{ji}/a_{ii})$ and $b'_{jj} = b_{jj} - (b_{ij}a_{ji}/a_{ii})$. This strategy can be applied until there is only one equation in one unknown.

Assume that we know the solution to these N-1 equations, and in particular that we know V_j. Then we can find V_i from Equation 13 with O(1) step. Therefore the effort to solve these N equations is O(1) plus the effort needed

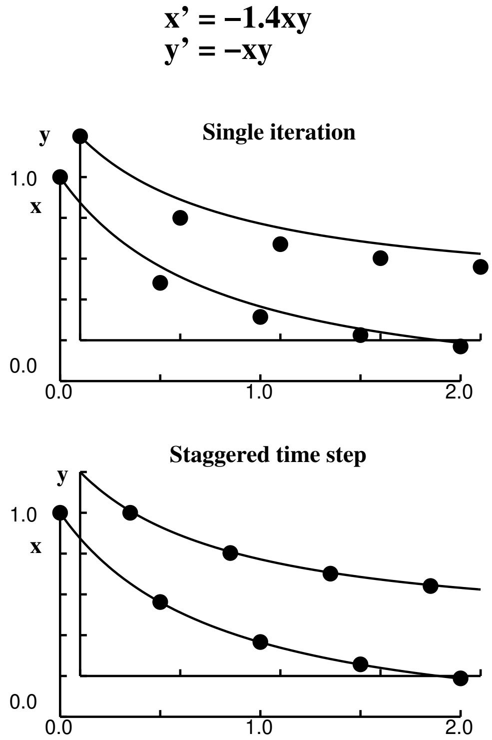

Figure 8: The equations shown at the top of the figure are computed using the Crank-Nicholson method. Top: $x(t+\Delta t)$ and $y(t+\Delta t)$ are determined using their values at time t. Bottom: staggered time steps yield decoupled linear equations. $y(t+\Delta t/2)$ is determined using $x(t)$, after which $x(t+\Delta t)$ is determined using $y(t+\Delta t/2)$.

to solve N-1 equations. The number of operations required is independent of the branching structure and thus a tree of N compartments uses exactly the same number of arithmetic operations as a one-dimensional cable of N compartments.

Efficient Gaussian elimination requires an ordering that can be found by a simple algorithm that merely chooses the equation with the current minimum number of terms as the equation to use in the elimination step. The minimum degree ordering algorithm is commonly used in standard sparse matrix solver packages. One example written in C is Sparse1.3, the Sparse Linear Equation Solver by Kundert and Sangiovanni-Vincentelli. This and many other sparse matrix packages are freely available on the Internet via anonymous ftp from netlib.att.com.

REFERENCES

Carnevale, N.T. and Rosenthal, S. Kinetics of diffusion in a spherical cell. I. No solute buffering. *J. Neurosci. Meth. 41*:205–216, 1992.

Carnevale, N.T. and Rosenthal, S. Kinetics of diffusion in a spherical cell. II. Solute buffering included. *J. Neurosci. Meth. 41*:217–229, 1992.

Crank, J., and Nicholson, P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Proc. Cam. Phil. Soc. 43*:50–67, 1947.

*Dahlquist, G. Bjö rck,and Anderson, N . Numerical Methods Prentice-Hall, Englewood Cliffs, N.J., 1974.

Douglas, J. A survey of numerical methods for parabolic differential equations. In: Advances in Computers vol. 2, ed. F. Alt, Academic Press, New York, ch.1, 1961.

Hines, M. Efficient computation of branched nerve equations. *Int. J. Bio-Med. Comp. 15*:69–76, 1984.

Mascagni, M. Numerical methods for neuronal modeling. In: Methods in Neuronal Modeling eds. Koch, C. and Segev, I., MIT Press, Cambridge, Mass., ch.13, pp. 439–484, 1989.

*Oran, E.S. and Boris, J.P. Numerical Simulation of Reactive Flow Elsevier, N.Y., 1987.