NEURON: A Tool for Neuroscientists
M. L. Hines and N. T. Carnevale

The Neuroscientist 2001; 7; 123

The online version of this article can be found at:
http://nro.sagepub.com/cgi/content/abstract/7/2/123

Additional services and information for The Neuroscientist can be found at:

Email Alerts: http://nro.sagepub.com/cgi/alerts

Subscriptions: http://nro.sagepub.com/subscriptions

Reprints: http://www.sagepub.com/journalsReprints.nav

Permissions: http://www.sagepub.com/journalsPermissions.nav
NEURON: A Tool for Neuroscientists

M. L. HINES and N. T. CARNEVALE
Departments of Computer Science (MLH) and Psychology (NTC)
Yale University
New Haven, Connecticut

NEURON is a simulation environment for models of individual neurons and networks of neurons that are closely linked to experimental data. NEURON provides tools for conveniently constructing, exercising, and managing models, so that special expertise in numerical methods or programming is not required for its productive use. This article describes two tools that address the problem of how to achieve computational efficiency and accuracy. NEUROSCIENTIST 7(2):123–135, 2001

KEY WORDS Computer model, Computational neuroscience, Neural model, Spatial grid

Over the past 2 decades, the application of new experimental techniques has yielded a wealth of information about the anatomical and biophysical properties of neurons and neural circuits. This expansion of knowledge is essential for understanding the biological basis of brain function, yet it comes at a cost of its own, because most data can now only be interpreted in terms of the interaction of many complex mechanisms. Increasingly aware of the difficulty of establishing consistency between data and theory, growing numbers of neuroscientists have found empirically based modeling to be a useful tool for studying the functional implications of anatomy and biophysics.

Two important factors have facilitated the widening acceptance of modeling among experimentalists. The first is the availability of powerful yet inexpensive computing hardware, so that most small laboratories, and even students, can now afford machines whose performance rivals that of supercomputers of recent memory. The second factor is the development of domain-specific simulation tools such as NEURON (http://www.neuron.yale.edu), which is designed to provide a flexible and convenient environment in which neuroscientists can take advantage of this raw computing power.

These circumstances have driven a progressive shift in modeling away from speculation to models that are highly constrained by biological data. In this article, we provide a brief overview of why NEURON is particularly well suited to this kind of modeling and discuss in greater detail the most recent enhancements to this program that address existing and emerging needs of investigators who are concerned with reconciling theory and experiment.

Background

NEURON can simulate individual neurons and networks of neurons with properties that may include, but are not limited to, complex branching morphology, multiple channel types, inhomogeneous channel distribution, ionic diffusion, and the effects of second messengers. It provides tools for constructing, exercising, and managing models, so that special expertise in numerical methods or programming is not required for its productive use.

These attributes are responsible for the application of NEURON to a broad range of research questions, from the basic cellular mechanisms that underlie neuronal function, to information encoding and the operation of large-scale networks involved in consciousness, perception, learning, and memory, and for examining the roles of neuronal and network properties in diseases such as epilepsy, multiple sclerosis, and disorders of learning and memory. NEURON has been used in research reported in more than 220 scientific articles, 81 of which were published in the past 2 years. A survey of the recent literature finds that it has been used for modeling individual cells or subcellular components to address topics that include

- presynaptic and postsynaptic mechanisms involved in synaptic transmission (Ahmed and others 1998; Baccus 1998; Dzubay and Jahr 1999; Kits and others 1999; Neville and Lytton 1999; Thomson and Destexhe 1999)
- dendritic electrotonus and synaptic integration (Larkum and others 1998; Migliore and Culotta 1998; Raastad and others 1998; Thurbon and others 1998; Cameron and others 1999; Chitwood and others 1999; Destexhe and Pare 1999; Jaffe and Carnevale 1999; Kulagina 1999; London and others 1999; Winslow and others 1999)
- amplification and suppression of postsynaptic potentials by dendritic active currents (Destexhe, Neubig, and others 1998; Korogod and Kulagina 1998; Pare, Lang, and others 1998)
• spike initiation, including dendritic spikes (Lüscher and Larkum 1998; Pare, Lang, and others 1998; Migliore and others 1999; Shen and others 1999)
• intrinsic neuronal activity (Canavier 1999; Elaagoubi and Yuste 1999; Zhu, Lytton, and others 1999; Zhu, Uhlrich, and others 1999) and its modulation by neuropeptides (Sohal and others 1998)
• neural code (Mukherjee and Kaplan 1998; Brown and others 1999; Engel and others 1999; Neubig and Destexhe 1999; Shao and others 1999; Sheasby and Fohlmeister 1999; Tang and others 1999)
• neuronal changes during development (Ivanov and others 1999; Vabnick and others 1999)
• extracellular stimulation (Maccabee and others 1998; Greenberg and others 1999; McIntyre and Grill 1999) and recording (Sahin and Durand 1998)
• network modulation of cellular activity (Nadim and others 1998; Bernasconi and others 1999)
• mechanisms of motor control (Herrmann and Flanders 1998)
• cellular mechanisms involved in visual direction and orientation selectivity (Mel and others 1998; Anderson and others 1999) and stereocilia (Zackenhauser and others 1998, Simon and others 1999)

There are also many reports of network models implemented with NEURON. These models have been used to study phenomena such as

• thalamic and thalamocortical oscillations (Destexhe 1998; Sohal and Huguenard 1998; Destexhe and others 1999; Houweling and others 1999; Sohal and others 2000)
• synchronization of network oscillations by gap junctions (Moortgat and others 2000)
• encoding of temporal information (Lytton and Lipton 1999; Buonomano 2000)
• network mechanisms underlying orientation selectivity in vision (Bush and Priebe 1998)
• mechanisms of epilepsy (Lytton and others 1998; Bush and others 1999; Destexhe 1999)
• actions of anticonvulsant drugs (Thomas and Lytton 1998)

This diversity is testimony to the wide utility of NEURON in neuroscience research, particularly for experimentalists who are concerned with empirically based modeling. An important side effect of these applications of NEURON has been the emergence of a community of users who have made their own contributions to the utility of this program. Several authors have developed and published strategies for design of accurate and efficient models of individual cells and networks with NEURON (Destexhe and others 1995a, 1995b, 1996; Lytton 1996; Destexhe and others 1997; Jackson and Cauller 1997; Destexhe, Mainen, and others 1998; Mainen and Sejnowski 1998), while others have used it to implement new tools for the analysis of neuronal properties (Carnevale and others 1996; O’Boyle and others 1996; Carnevale and others 1997).

Overview of NEURON

NEURON was initially designed to facilitate dealing with neuronal models in which complex membrane properties and extended geometry play important roles (Hines 1989, 1993, 1995). Subsequently, its domain of applicability has been increased by adding facilities for describing longitudinal ionic diffusion and computationally efficient representation of connections in a network (Hines and Carnevale 2000).

The fundamental principles behind the design and implementation of NEURON are detailed elsewhere (Hines and Carnevale 1997, 2000), but it is useful to summarize them briefly here. NEURON is formulated around the notion of continuous cable “sections,” which can be connected together to form any kind of branched cable. A section can be assigned properties that vary continuously with position along its length. The aim is to completely separate the physical properties of the neuron from the numerical issue of size of spatial compartments and thus to help the investigator focus on the biology rather than computational details (Hines and Carnevale 1997).

User-defined biophysical properties of membrane (e.g., ion channels, pumps) and cytoplasm (e.g., buffers and second messengers) are described in terms of differential equations, kinetic schemes, and sets of simultaneous equations. These model descriptions are compiled so that membrane voltage and gating states can be computed efficiently using an implicit integration method optimized for branched structures (Hines and Carnevale 2000).

NEURON derives its flexibility and convenience from two features. The first is a graphical interface (GUI) that can be used to create models, run initial exploratory simulations, set parameters, control common voltage and current stimuli, and graph variables as functions of time and position. The second is an object-oriented interpreter that provides a complete programming language that is useful for customization of the GUI, advanced data analysis, and optimization.

Thus, NEURON puts a great deal of computational power at the disposal of the user, especially for the study of models that have a close relationship to experimental data. Yet, this facility as a vehicle for implementing empirically based models immediately raises a new set of problems that are related to managing anatomical and biophysical complexity so as to achieve computational efficiency and accuracy while minimizing the effort required of the user. In this article, we describe features of NEURON that can help users deal successfully with the problem of balancing computational efficiency with numeric accuracy in space and time.

Spatiotemporal Accuracy versus Computational Speed

As mentioned above, NEURON is designed so that users can specify models without being concerned about
compartment size or time step duration. These are mere computational details that ought not to intrude on the process of prescribing what aspects of the biological system should be included in the model. The NEURON simulation environment allows such distractions to be put off until it is time to launch a simulation. Furthermore, it has features and tools that help users deal easily and effectively with these two previously vexing problems of modeling.

Time and space are continuous variables in biological neurons, and the spread of electrical and chemical signals is governed by the diffusion equation, a partial differential equation in which potential (voltage, concentration) and flux (current, movement of solute) are smooth functions of time and space (Rall 1977; Jack and others 1983). A standard strategy is to approximate the diffusion equation with a set of algebraic difference equations that can be solved numerically (Crank 1979; Carslaw and Jaeger 1980). This is analogous to approximating the original continuous system by another system that is discontinuous in time and space, and it is the approach used by NEURON (Hines and Carnevale 1997).

NEURON computes the values of spatiotemporally continuous variables over a set of discrete points in space (“nodes”) for a finite number of instants in time. When NEURON’s second-order correct integration method is used, these values are a piecewise linear approximation to the continuous system, so that linear interpolation will give the values of continuous variables at intermediate locations and times with second-order accuracy. The size of the time step Δt and the fineness of the spatial grid jointly determine the accuracy of the solution. How faithfully the computed solution emulates the behavior of the continuous system depends on the spatial intervals between adjacent nodes and the temporal intervals between solution times. These should be small enough that the piecewise linear approximation can follow the curvature of the solution for the continuous system in space and time.

Figure 1 shows how this works in a situation where the size of the time step is the only consideration. These charging curves were computed from a model of a small spherical cell with passive membrane that was subjected to a depolarizing current pulse. Because the cell was isopotential, the spatial grid consisted of a single node. Figure 1B shows the analytic solution for membrane potential $V_m$ (dashed orange line) along with numeric solutions that were computed using several different values of Δt (solid black lines). As time advanced, even the least accurate numeric solution became indistinguishable from the analytic solution. However, solutions computed with large Δt lack the high-frequency terms needed to follow the initial rapid change of $V_m$. Decreasing Δt produced a progressive improvement in how closely the piecewise linear approximation approached the smooth curve of the analytic solution, especially at early times (Fig. 1C). That is, using a smaller Δt allowed the numeric solution to better capture the curvature of $V_m(t)$.

But a short time step alone does not guarantee good temporal accuracy. If propagation of electrical or chemical signals through the cell involves significant delay, then the spatial grid is also important. To see how the spatial grid affects accuracy, we turn to a model of fast excitatory synaptic input onto a dendritic branch in mammalian brain. In this model, the synapse is attached...
to the middle of an unbranched cylinder (Fig. 2A) with passive membrane that is 5 DC length constants long to avoid possible confounding effects of complex geometry and active current kinetics. The biophysical properties are within the range reported for mammalian central neurons (Spruston and Johnston 1992). The time course of the synaptic conductance follows an alpha function (Fig. 2B) with time constant $\tau_s$ and reversal potential $E_s$, chosen to emulate an AMPA synapse (Kleppe and Robinson 1999) and $g_{\text{max}}$ selected to produce a peak depolarization of $\sim 10$ mV. We use this model to compare the analytic solution for $V_m$ as a function of space and time with the numeric solution computed with a very small time step ($\Delta t = 1 \mu s = 0.001$ ms) but a very coarse spatial grid ($\Delta x = 1 \lambda$).

The time course of $V_m$ at the site of synaptic input (Fig. 3) shows that the numeric solution (solid black line) rises and falls more slowly than the analytic solution (dashed orange line) and has a peak depolarization that is substantially smaller and delayed. These differences occurred even though $\Delta t$ was more than two orders of magnitude smaller than necessary to follow the EPSP waveform. They reflect the fact that solutions based on the coarse grid lack sufficient amplitude in the high frequency terms that are needed to reproduce rapidly changing signals. Such errors could lead to serious misinterpretations if the purpose of the model were to examine conditions under which synaptic input might activate depolarization-activated currents, especially those with fast kinetics like $I_{\text{CA}}$, spike sodium current, and transient $I_{\text{Na}}$.

The graphs in Figure 4 present the spatial profile of $V_m$ along the dendrite at two times, selected from the rising and falling phases of the EPSP. These curves, which are representative of the early and late response to synaptic input, show that the error of the numeric solution is most pronounced in the part of the cell where $V_m$ changes most rapidly, i.e., in the near neighborhood of the synapse. However, at greater distances the analytic solution itself changes much more slowly because of low-pass filtering produced by cytoplasmic resistivity and membrane capacitance. At these distances, the error of the numeric solution is surprisingly small, even though it was computed with a very crude spatial grid. Furthermore, error decreases progressively as time advances and high frequency terms become less important.

Figures 1 and 3 demonstrated how accuracy depends on both the size of the time step and the resolution of the spatial grid. Using an inappropriate value for either can result in excess computational burden or inaccurate solutions. Furthermore, solutions computed by NEURON’s second-order integration method may oscillate if the time step is too large for the spatial grid. This is illustrated in Figure 5, which shows the response of the model dendrite of Figure 2 to a brief current pulse injected at its midpoint. To prevent oscillations in the numeric solution, the normalized increments in time ($\Delta T = \Delta t / \tau_s$) and space ($\Delta X = \Delta x / \lambda$), where $\Delta x$ is the distance between adjacent nodes must satisfy the relationship $\Delta T / \Delta X \leq 1/2$ (see chapter 8 in Crank 1979). In this model with nodes spaced 20 $\mu$m apart, oscillations will occur if $\Delta t > 0.0128$ ms.

As these examples indicate, choosing an appropriate spatiotemporal grid is a recurring practical problem in neural modeling. The accuracy required of a discrete approximation to a continuous system depends on the anatomical and biophysical complexity of the original system and the question that is being asked. Thus, finding the steady-state (resting) $V_m$ of an isopotential model with passive membrane may require only a few large time steps at one point in space, but determining the time course of $V_m$ throughout a highly branched model with active membrane as it fires a burst of spikes may demand much finer spatiotemporal resolution. Particular care may be needed when selecting $\Delta x$ and $\Delta t$ for complex models, because the time required to compute a simulation run is directly proportional to the product of the number of nodes and the number of time steps.

Choosing the Spatial Grid

One time-honored way to check the adequacy of the spatial grid is to repeatedly increase the number of grid points and exercise the model until further increases cause no significant change in simulation results. A particularly convenient way to do this in NEURON is
through the command `forall nseg*=3`, which triples the number of nodes throughout the model. Because NEURON solutions are second-order accurate in space, this reduces spatial error by a factor of 9, allowing easy detection of inadequacies of the spatial grid. Use of an odd multiple (Fig. 6A) also has the distinct advantage of introducing new nodes into the gaps between existing nodes while leaving the positions of the latter unchanged. Existing nodes would be destroyed if an even multiple were used (Fig. 6B), making it impossible to tell whether an apparent difference between simulations should be attributed to different spatial errors or instead to the fact that the solutions were computed for different points in space.

The simple and convenient strategy of repeatedly tripling the number of nodes throughout an entire model is generally not computationally efficient, especially if geometry is complex and biophysical properties are nonuniform. We have found that models that incorporate quantitative morphometric data frequently contain at least a few branches that need nine or more nodes, yet many other branches need only one or three nodes.
In such models, by the time the spatial grid is just adequate in some regions, elsewhere it will be much finer than necessary, requiring more storage and prolonging run times.

Alternatively, one might try the common practice of keeping the distance between adjacent grid points less than a small fraction (e.g., 5%–10%) of the DC length constant \( \lambda_{\text{oc}} \) of an infinite cylinder with identical diameter, cytoplasmic resistivity, and specific membrane resistance (Mainen and Sejnowski 1998; Segev and Burke 1998). This plausible approach has two chief limitations. First, large changes in \( R_m \) and \( \lambda_{\text{oc}} \) can be produced by activation of voltage-dependent channels (e.g., \( I_\text{i} \); Magee 1998; Stuart and Spruston 1998), \( \text{Ca}^{2+} \)-gated channels (Wessel and others 1999), or synaptic inputs (Bernander and others 1991; Häusser and Clark 1997; Pure, Shink, and others 1998; Destexhe and Pare 1999). The second and more fundamental problem is that the spatial decay of transient signals is unrelated to \( \lambda_{\text{oc}} \). Cytoplasmic resistance \( R_c \) and membrane capacitance \( C_m \) constitute a spatially distributed low-pass filter, so transient signals are subject to greater distortion and attenuation with distance than DC or slowly changing signals are. In other words, by virtue of their high frequency components in time, transient signals also have high frequency components in space. Just as high temporal frequencies demand a short time step, high spatial frequencies demand a fine grid.

As a rational revision to the present practice, we propose a criterion based on the length constant \( \lambda \) computed at a frequency \( f \) that is high enough for transmembrane current to be primarily capacitive, yet still within the range of frequencies relevant to neuronal function. Ionic and capacitive transmembrane currents are equal at the frequency \( f_m = 1/2\pi \tau_m \), so \( R_m \) has little effect on the propagation of signals \( \geq 5f_m \). For instance, a membrane time constant of 30 ms corresponds to \( f_m \approx 5 \text{ Hz} \), so \( R_m \) would be irrelevant to signal spread at frequencies \( \geq 25 \text{ Hz} \). Most cells of current interest have \( \tau_m \approx 8 \text{ ms} (f_m \approx 20 \text{ Hz}) \), so we suggest that the distance between adjacent nodes should be no larger than a user-specified fraction of \( \lambda_{\text{oc}} \), the length constant at 100 Hz. This frequency is high enough for signal propagation to be insensitive to shunting by ionic conductances, but it is not unreasonably high because the rise time \( \tau_r \) of fast EPSPs and spikes is \( \approx 1 \text{ ms} \), which corresponds to a bandpass of \( 1/\tau_r \approx 400 \text{ Hz} \).

At frequencies where \( R_m \) can be ignored, the attenuation of signal amplitude is described by

\[
\log \left( \frac{V(x)}{V_0} \right) = 2 \sqrt{\pi f R_mC_m} \frac{d}{\lambda f},
\]

so the distance over which an e-fold attenuation occurs is

\[
\lambda_f = \frac{1}{2} \sqrt{\frac{d}{\pi f R_mC_m}}.
\]

where \( f \) is in Hz. As an example, the model dendrite of Figure 2 has \( \lambda_{\text{oc}} = 500 \mu\text{m} \), but \( \lambda_{\text{m}} \) is only \( \approx 225 \mu\text{m} \).

In NEURON, this rule is implemented in the CellBuilder, a GUI tool for constructing and managing models of cells. The CellBuilder allows the maximum anatomical distance between grid points to be specified as a fraction of \( \lambda_{\text{oc}} \) using an adjustable parameter called \( d_{\text{lambda}} \). The default value of \( d_{\text{lambda}} \) is 0.3, which is more than adequate for most purposes, but a smaller value can be used if \( \tau_m \) is shorter than 8 ms. For increased flexibility, the CellBuilder also provides two alternative strategies: specifying \( n_{\text{seg}} \), the actual number of grid points; specifying \( d_{\text{x}} \), the maximum anatomical distance between grid points in \( \mu\text{m} \). Each of these strategies deliberately sets \( n_{\text{seg}} \) to an odd number, which guarantees that every branch will have a node at its midpoint (e.g., Fig. 6). These strategies can be applied to any section or set of sections in a model, each section or set of sections having its own rule and parameter value. However, barring special circumstances, for example, localized high membrane conductance, it is usually sufficient to use the \( d_{\text{lambda}} \) strategy for the entire model. Regardless of which strategy is selected, it is always advisable to try a few exploratory runs with a finer grid to be sure that spatial error is acceptable.

To see how the \( d_{\text{lambda}} \) rule works in practice, consider the model in Figure 7A, which represents a granule cell from the dentate gyrus of the rat hippocampus. The complex architecture of this model is taken directly from quantitative morphometric data provided by Dennis Turner (http://www.neuro.soton.ac.uk/cells/cellArchive.html), and the biophysical parameters are the same as those reported by Spruston and Johnston (1992): \( R_m = 40k \Omega \text{ cm}^2 \), \( C_m = 1 \mu\text{F} / \text{cm}^2 \), and \( R_c = 200 \text{ k}\Omega \text{ cm}^2 \).
Ω cm. Attached to the soma is an excitatory synapse; this is identical to the AMPA synapse of Figure 2 except that $g_{\text{max}}$ has been reduced to $2 \cdot 10^{-9}$ siemens.

Figure 7B shows the time course of $V_{\text{soma}}$ computed using spatial grids with one or three nodes per branch (thick blue and thin black traces, respectively) or specified by the $d_{\text{lambda}} = 0.3$ criterion (dashed orange trace).

From this figure, we can see that even the crudest spatial grid ($n_{\text{seg}} = 1$) would suffice if the purpose of the model were to evaluate effects of synaptic input on $V_{\text{soma}}$ well after the peak of the EPSP ($t > 7$ ms). However, a finer grid is clearly necessary if the maximum somatic depolarization produced by the EPSP is of concern.

Additional refinements to the grid are necessary if we are interested in how the EPSP spreads into other parts of the cell, e.g., the path marked by orange in Figure 8A. To compute the maximum depolarization produced by a somatic EPSP along this path, the model can get along with a grid that has only three nodes per branch (Fig. 8B). If the timing of this peak is important, e.g., for coincidence detection or activation of voltage-gated currents, a finer grid must be used (Fig. 8C).
The computational cost of these simulations is approximately proportional to the number of nodes. Least burdensome, but also least accurate, were the simulations generated with 1 node per branch, which involved 28 nodes in the model. Increasing the number of nodes per branch to three (total nodes in model = 84) improved accuracy substantially, but noticeable errors remained (Fig. 8C) that disappeared only after an additional tripling of the number of nodes per branch (total nodes = 252; results not shown). The greatest accuracy without sacrificing efficiency was achieved with the grid specified by the d_lambda = 0.3 criterion, which contained only 110 nodes.

As these figures suggest, the advantages of the d_lambda strategy will be most apparent when signal propagation throughout the entire model must be simulated to a similar level of accuracy. If the focus is on a limited region, then a grid with fewer nodes and a simpler representation of electrically remote areas may be acceptable. Special features of the model may also allow a simpler grid to be used. For example, in principal neurons of mammalian cortex, proximal dendritic branches tend to have larger diameters (Rall 1959; Hillman 1979) and shorter lengths (Cannon and others 1999) than distal branches. In models based on quantitative morphometry, grids specified with either a d_lambda or d_X criterion will have fewer nodes in proximal branches than in more distal branches. Indeed, many proximal branches may have only one or three nodes, regardless of which criterion is applied; differences between gridding strategies will manifest only in the thinner and longer distal branches. Such differences will have little effect on simulation results if signals in the vicinity of the soma are the only concern, and the relative advantage of the d_lambda strategy will be smaller.

Choosing the Time Step

The choice of an appropriate time step Δt is the temporal corollary of assigning a spatial grid, and it raises similar concerns. We have seen how grid spacing affects the ability of the computational solution to follow spatial nonlinearities in state variables (e.g., curvature in the plot of Vm versus distance; Fig. 8); likewise, the size of Δt should be set according to the degree to which state variables change nonlinearly with time (Fig. 1).

There is a wide variety of problems for which an adaptive time step method would be expected to have much higher performance than a fixed step method, i.e., Δt could grow very large when all states are varying slowly, as in interspike intervals. On the other hand, in problems that involve propagating action potentials or networks of cells, it may happen that some state somewhere in the system is always varying quickly. In such cases, Δt must always be small in order to follow whichever state is varying fastest. It is often not obvious in advance whether the increased overhead of an adaptive time step method will be repaid with an occasional series of long time steps. Even so, the greatest benefit of an adaptive time step method may be that it offers the user a direct choice of local step accuracy rather than Δt.

To this end, NEURON has adopted CVODE (Cohen and Hindmarsh 1994), one of the standard variable timestep/variable order integration methods. With CVODE, the user specifies a maximum allowable absolute error rather than Δt. The integrator then dynamically adjusts Δt so that the estimated local error of each state variable is always less than the maximum absolute error. The default value chosen for the maximum absolute error was 0.01 so that the classical Hodgkin-Huxley action potential simulation at 6.3 °C had accuracy comparable to a second-order correct simulation with fixed Δt = 25 μs. The user can specify an error criterion that involves relative tolerance, but this is generally not advisable in neural modeling because there is rarely a reason to require increasing absolute accuracy around the 0 value of most states, especially voltage. However, the scale of states is often a crucial consideration, and the maximum absolute error must be consistent with the desired resolution of each state. An extreme example is a model of a calcium pump in which pump density is measured in moles/cm². Here, an appropriate value is 10⁻¹⁴ mole/cm², and an allowable error of 0.01 is clearly nonsense. For this reason, it is essential that each state that is badly scaled, e.g., [Ca²⁺], measured in mM, be given its own explicit maximum absolute error. NEURON accommodates this need by allowing the user to set specific error criteria for individual states that take precedence over any global criterion.

For an example of how CVODE can reduce the time necessary to produce accurate simulations, we turn to the neocortical layer V pyramidal cell model described by Mainen and Sejnowski (1996). We computed the response of this model over 1000 ms, during which a 900 ms depolarizing current applied to the soma evoked two bursts of spikes (see Fig. 9A). Global error of the simulation was assessed by observing the effect of reducing the integration time step or CVODE absolute tolerance on the variability of the time ti at which the last somatic action potential crossed above 0 mV. When the fixed step, second-order integration method was used, ti converged to 695.3 ms for Δt ≤ 0.01 ms, and a simulation performed with Δt = 0.01 ms took 807 seconds to complete. Solutions computed with CVODE converged to the same ti when absolute tolerance was 2.5 • 10⁻³ for all states except for [Ca²⁺], which had an absolute tolerance of 2.5 • 10⁻¹; the solution generated with these tolerances had a runtime of just 44 seconds. In other words, CVODE allowed us to achieve the same accuracy as the most accurate fixed time step solution but with a runtime that was more than 10 times faster.

Figure 9B reveals the control that CVODE exerted over the integration step size throughout the entire simulation, cutting Δt to values much smaller than 0.01 ms when states were changing most rapidly, and increasing it to a maximum of ~4.4 ms during the long interburst
interval. The smallest steps were restricted to narrow intervals that began just before the threshold and ended shortly after the depolarized peak of each spike, as illustrated by an expanded view of the transition from the interburst interval to the beginning of the second burst (Fig. 10). The remarkable acceleration of the simulation by CVODE reflects the fact that \( \Delta t \) was much larger than 0.01 ms for most of the run.

An important feature of the variable step method is the fact that it was incorporated in NEURON in such a way that users will find it as unobtrusive and easy to apply as possible. Care has been taken so that the same description of a model neuron or biophysical mechanism will work with each of NEURON’s integration methods. Furthermore, once a model specification has been created, switching between fixed and variable time step methods is as simple as a button press. This convenience is crucial because relative performance between high overhead variable step and low overhead fixed step methods ranges widely. For example, the demonstration simulations by Mainen and Sejnowski (1996) slowed down by a factor of 2 or sped up by a factor of 7, depending on the number of spikes in a simulation run and whether there are long intervals in which no state is rapidly changing.

NEURON provides a network connection class (NetCon) for network simulations in which cell to cell communication can be abstractly represented by the (possibly delayed) delivery of logical events, as opposed to graded interaction via gap junctions or electrical synapses. The notion of a cell driven by discrete input events naturally suggests a possible expansion of the simulation domain where variable time step methods provide substantial performance gains. Recall that in networks it may happen that only a few cells are active at any one time, but with a global time step these active cells govern the time step for all. The local variable time step method uses a separate CVODE solver instance for each cell, which integrates that cell’s states with time steps governed only by those state dynamics and the discrete input events, and can efficiently (without integrating equations) retreat from its current time to any time as far back as the beginning of its previous time step.
All cells are always on a list ordered by their current time and all outstanding events are on a list ordered by their delivery time. The network progresses forward in time by finding the least time cell or event and integrating that cell by one step or delivering the event to the proper cell. In the latter case, the cell retreats to the delivery time and becomes the least time cell. The event, of course, is removed from the list and discarded. In the former case, the cell is integrated according to its current time step and moves to a location on the cell list appropriate to its new time.

In periods of synchronous activity, the local time step method yields no benefit. If events are extremely numerous, neither the local nor the global variable time step method gives a performance boost. When multiple events per reasonable $\Delta t$ arrive regularly, the fixed time step nicely aggregates all events in a step without regard to their micro temporal structure, whereas the variable step method’s scrupulous handling of each event is out of all proportion to the conceptual approximation of the network. On the other hand, it is easy to devise networks in which the speed improvement of the local step approach is arbitrarily great, e.g., chains of neurons. The fact that the integration method is so dependent on both the problem and the intention of the user underscores the importance of allowing easy selection of any of them with no changes to the user-level specification of the model.

One limiting case of the variable step simulation style is the event-driven simulation, in which the cell jumps from event to event. Here, a single compartment is used merely as a stage in which the voltage never changes (the natural time step is infinite) and the cells are represented by POINTPROCESSes that receive events from, and provide events to, the NetCon instances. A wide range of abstract neuronal phenomena useful in artificial neural nets, such as integrate and fire, firing frequency dependent on input, and use dependent synaptic plasticity, have equations that can be solved analytically so that “cell” state needs only to be computed at the event.

Discussion

A computational model of a neurobiological system is actually a model of a model. The first level of modeling occurs when, motivated by some phenomenon of interest, one formulates a hypothesis that includes just those properties of the original system that are judged to be essential. This hypothesis itself is a conceptual model, derived from the real world by a process of abstraction and simplification that relies heavily on scientific insight about the biology. Some conceptual models are so simple that their implications are obvious. However, most interesting neural phenomena involve cells and circuits whose anatomical and biophysical complexities confound intuition. In such a case it can be helpful to create a computational model that emulates the operation of the conceptual model. This is the second level of modeling, and if it is to reliably illustrate the consequences of the hypothesis, then the translation from conceptual model to computational model must be as faithful as possible. The utility of a simulation environment in neuroscience research derives largely from how well it facilitates the creation of computational models that closely match their conceptual antecedents. This has always been a key consideration in the design and implementation of NEURON (Hines 1998).

The utility of a simulation environment also depends strongly on the ease with which computational models can be exercised. This degrades rapidly when users are forced to divert their attention from the biology of the problem to deal with arbitrary computer issues that have nothing to do with neuroscience. A major purpose of NEURON is to release the individual user from such concerns, or at least provide guidance regarding their safe management. The $\dlambda$ heuristic for dealing with the spatial grid and the CVODE method for automatically adjusting the order and time step used in numerical integration are effective and robust, and both have been incorporated in NEURON in such a way that they are quite easy to apply. These are vital attributes in a simulation environment that is designed to be particularly well suited for computational models that are closely linked to experimental data.

We must note that although the $\dlambda$ criterion for the spatial grid is based on an estimate of the spread of membrane potential, $V_m$ is not the only state that may vary nonlinearly in space. Other factors need to be weighed in models where spatially nonuniform chemical signals play an important role. The obvious example is $[Ca^{2+}]$, which may be subject to nonuniformities as a result of channel clustering, localized release from intracellular stores, and the effects of diffusion and buffering.

We should also point out that the default integration method in NEURON is a fixed-step-first-order implicit scheme that is numerically stable when extremely stiff ODEs and even algebraic equations are present in the system, as when voltage clamps are included in the model. All of the simulations presented in this article were generated with either NEURON’s Crank-Nicholson-like integration method or CVODE. The Crank-Nicholson-like method uses an algorithm with a performance that is almost identical to the simpler first-order implicit method (Hines, 1984), but it is second-order correct when channel membrane current is instantaneously linear in voltage (e.g., equations of Hodgkin-Huxley form). Consequently, it can use a larger $\Delta t$ to achieve the same accuracy as the first-order method, resulting in shorter runtimes.

An important issue in the use of the CVODE method is selection of appropriate values for local error control. Experience so far suggests that the absolute local error tolerance is much more important than the relative error. The default error setting (10 $\mu$V for membrane potential and 0.1 nanomolar for free $[Ca^{2+}]$) is approximately equivalent to the default fixed $\Delta t = 0.025$ ms for spike transients, but occasionally it gives inaccurate interspike
errors due to the finite spatial grid and essential that computational models be tested so that the "physiological accuracy," i.e., sufficiently accurate to give useful insight into the question under study. It is essential that computational models be tested so that the errors due to the finite spatial grid and error tolerance do not affect the interpretation of results.

References


