Appendix A

Passive Membrane Parameters

A vexing and confusing question concerns the use of the all-important membrane parameters R_i, C_m, and R_m and their associated units.

A.1 Intracellular Resistivity R_i

In physics, the amount of *resistance* a piece of bulk material offers to electrical current flowing across the material is characterized by its *specific resistivity* ρ; its units are ohms-centimeter (Ω·cm). If the piece of material has a constant cross section A and length ℓ, its total resistance will be

$$R = \frac{\rho \cdot \ell}{A}. \tag{A.1}$$

In biophysics, the specific resistance of the intracellular medium, also known as the *intracellular resistivity*, is denoted by R_i, which can be thought of as the total resistance across a 1 cm cube of intracellular cytoplasm.

Given a cylindrical axon or dendrite of diameter d and length ℓ, its total resistance will be $4R_i\ell/\pi d^2$ and its resistance per unit length ℓ of cable,

$$r_a = \frac{4R_i}{\pi d^2}. \tag{A.2}$$

Conversely, given a neural process with axial resistance per unit length r_a, the total axial resistance will be ℓr_a. The units of r_a are ohms per centimeter (Ω/cm).

The resistivity of seawater, from where we all came half a billion years ago, is 20 Ω·cm, that of mammalian saline about 60, and that of physiological Ringer solution 80 Ω·cm. This is not surprising, since the resistivity decreases as salts are added to a solution. Computing the intracellular resistivity on the basis of the Nernst-Planck electrodiffusion equation yields 33.4 Ω·cm for K^+ ions and 267 Ω·cm for Na^+ ions (Eq. 11.33). Under the assumption that these are the dominant charge carriers, this leads to a final value of $R_i = 29.7$ Ω·cm (see Sec. 11.3.1).

Because the intracellular environment contains many structures in addition to electrolyte, such as endoplasmic reticulum, the cytoskeleton, the Golgi apparatus, and so on, that restrict charge redistribution within neurons, the above cited values set a lower bound on the true

value of R_i in neurons. The value of R_i conventionally used is in the 70–100 Ω·cm range, based on measurements in neurons and axons of marine invertebrates (Foster, Bidinger, and Carpenter, 1976; Gilbert, 1975) and on data taken from cat motoneurons (Barrett and Crill, 1974)

However, much higher values have been reported. Neher (1970) directly measured a value of 180 Ω·cm for the resistivity of the cell body of snail neurons, while Shelton (1985) and Segev et al., (1992) require values of 225–250 Ω·cm for cerebellar Purkinje cells in order to fit various experimentally observed records, such as pulse attenuation and input resistance, against detailed cable models.

In a careful study of this problem using *in vitro* cortical pyramidal cells recorded and stained with sharp intracellular electrodes and whole-cell patch pipettes, Major (1992; see also Major, Evans, and Jack, 1993a) concludes that a good match between his physiological records and a detailed cable model requires a R_i in the 200–300 Ω·cm range (see also Spruston and Johnston, 1992, and Spruston, Jaffe, and Johnston, 1994). Tissue shrinkage during the histological recovery of the cell's anatomy does, however, affect these numbers. After decreasing the diameters of all dendrites by 20% while simultaneously increasing all lengths by 20% to invert the effect of shrinkage, Major and his colleagues infer a reduced value of $R_i = 187$ Ω·cm.

A.2 Membrane Resistance R_m

The *passive membrane resistivity*, that is, the resistance associated with a unit area of membrane, of the electrical component of the neuronal membrane that does not depend on synaptic input or on the membrane potential is denoted by R_m, measured in units of ohms-square centimeter (Ω·cm^2). Its inverse is known as the passive conductance per unit area of dendritic membrane or, for short, as the *leak conductance* $G_m = 1/R_m$ and is measured in units of siemens per square centimeter (S/cm^2).

The molecular correlate of this leak conductance is not precisely known. A pure phospholipid in saline solution has an extremely high resistance of up to 10^{15} Ω · cm^2 (Hille, 1992). Since measured membrane resistances are considerably lower, some mechanism has to permit ions to pass across the membrane.

The evidence for voltage-independent "leak" channels is not strong. Patch-clamp studies of frog sympathetic neurons reveal a nearly ohmic region between −70 and −110 mV (Jones, 1989). The underlying conductance is only weakly voltage dependent and is insensitive to blockers that block other known conductances in the cell. After adjusting for a 10 mV contribution of the sodium-potassium pump, the remainder of the "leak" current is carried by potassium ions and reverses around −65 mV. Evidence from hippocampal pyramidal cells suggests a very weakly voltage-dependent potassium current that is active at rest and that can be blocked by cholinergic input (Madison, Lancaster, and Nicoll, 1987).

In a cylindrical fiber of length ℓ and cross section $A = \pi d \ell$, the membrane resistance per unit length of fiber is defined as

$$r_m = \frac{R_m}{\pi d} \tag{A.3}$$

in units of Ω·cm. The total membrane resistance in a fiber of length ℓ is identical to r_m/ℓ. R_m varies widely from preparation to preparation, with the quality of the intracellular recording, with the amount of synaptic input to the cell, and other parameters. As intracellular recording techniques become more mature and sophisticated, the estimates of R_m increase.

Note that the "effective" specific resistivity associated with the membrane, on the order of 10^5 $\Omega \cdot cm^2/40$ Å$= 2.5 \times 10^{11}$ $\Omega \cdot cm$, is approximately one billion times larger than that of the intracellular fluid, explaining why by far the largest fraction of the cytoplasmic current flows within the dendrite or axon rather than across the membrane.

A.3 Membrane Capacitance C_m

The capacitance of the neuronal membrane is characterized by the *specific capacitance* per unit area C_m, measured in units of farads per square centimeter (F/cm^2). The generally agreed upon value for C_m is 1 μF/cm^2. This amounts to 10^{-7} coulomb (C) charge being distributed on both sides of a 1 cm^2 piece of neuronal membrane in the presence of a 100 mV voltage difference across the membrane. Assuming a parallel plate capacitor configuration and a dielectric constant of 2.1 for the hydrocarbon chains making up the bilipid layers, this implies a separation of 23 Å (Hille, 1992). Hence, the reason for the slow time constants in the millisecond time range is the molecular dimension of the neuronal membrane.

It is intriguing to compare the biological value of C_m against the capacitance in the analog CMOS circuit technology used to design neuromorphic electronic circuits (Mead, 1989; Mahowald and Douglas, 1991; Douglas, Mahowald, and Mead, 1995). Here, a capacitance is created by separating two layers of polysilicon with a 40-nm-thick layer of silicon oxide. The specific capacitance is about 0.5 μF/cm^2, about 20 times lower than its biological counterpart (compatible with the larger separation between the two layers). The higher value of C_m in biology is partly compensated for by the fact that the voltages across membranes (on the order of 0.1 V) are much smaller than the typical gate voltages of around 5 V in electronic circuits. Ultimately, this is due to the multiple gating charges of the voltage-dependent channels (see Chap. 8) allowing them to work on a much steeper exponential than transistors.

The 1 μF/cm^2 value of C_m appears to be somewhat of an overestimate. The membrane capacitance of a pure bilayer lipid membrane without proteins is between 0.6 and 0.8 μF/cm^2 (Cole, 1972; Fettiplace, Andrews, and Haydon, 1971; Benz et al., 1975), placing a lower bound on C_m. Mammalian red blood cells have a measured C_m of between 0.8 and 0.9 μF/cm^2. The "traditional" value of unity for C_m is based on the membrane of the squid giant axon and includes the nonlinear capacitances associated with the gating currents (for a discussion see Adrian, 1975). In a detailed investigation on dissociated hippocampal pyramidal cells, Sah, Gibb, and Gage (1988) report $C_m = 1.0 \pm 0.2$ μF/cm^2, while a more recent investigation of neocortical pyramidal cells finds values between 0.65 and 0.8 μF/cm^2 (Major, 1992).

A further complication are membrane invaginations and foldings which can multiply the effective membrane area—and therefore C_m—manyfold (as for instance in bullfrog sympathetic ganglion cells, where $C_m = 3$ μF/cm^2 has been inferred from the measured total cell capacitance and the area of these spherical cells (Yamada, Koch, and Adams, 1998; see also Segev et al., 1992).

Frequently, one defines

$$c_m = C_m \cdot \pi d \tag{A.4}$$

as capacitance per unit fiber of diameter d (in units of F/cm), in analogy to r_m. The total capacitance of all of the membrane in a process of length ℓ is given by $c_m \ell$.

Appendix B

A Miniprimer on Linear Systems Analysis

For those readers who forgot linear systems analysis, crucial to this book, we here provide the briefest of reviews.

A system is always linear or nonlinear with respect to some particular input and output variable. Indeed, the same physical system can be linear when using one sort of input-output pairing and nonlinear when considering a different one. If we restrict ourselves to the case when the input and output variables are single-valued functions of time, termed $x(t)$ and $y(t)$, respectively, then for a system $\mathbf{L}$, defined as

$$y(t) = \mathbf{L}[x(t)] \tag{B.1}$$

to be linear, it must obey two constraints. Firstly, it must be *homogeneous*, that is,

$$\mathbf{L}[\alpha x(t)] = \alpha \mathbf{L}[x(t)]\,. \tag{B.2}$$

For instance, doubling the input should double the output. Secondly, the system must also be *additive*,

$$\mathbf{L}[x_1(t) + x_2(t)] = \mathbf{L}[x_1(t)] + \mathbf{L}[x_2(t)]\,. \tag{B.3}$$

The response of the system to the sum of two inputs is given by the sum of the responses to the individual inputs. These two properties are sometimes also summarized in the *superposition* principle, expressed as

$$\mathbf{L}[\alpha_1 x_1(t) + \alpha_2 x_2(t)] = \alpha_1 \mathbf{L}[x_1(t)] + \alpha_2 \mathbf{L}[x_2(t)]\,. \tag{B.4}$$

A further property that some (not all) linear systems possess is *shift* or *time invariance*; in other words, if the input is delayed by some Δt, the output will be delayed by the same interval. A linear system is *shift invariant* if and only if

$$y(t) = \mathbf{L}[x(t)] \quad \text{implies} \quad y(t - t_1) = \mathbf{L}[x(t - t_1)]\,. \tag{B.5}$$

If a system possesses these three properties, then its entire behavior can be summarized by its response to an impulse or *delta* function $\delta(t)$.[1] The impulse response or Green's function of the system is exactly what its name imples, namely, the response of the system to an impulse,

1. The delta, unit, or *Dirac* "function" is defined by being zero for all values of t except at the origin, where it diverges. It has the immensely useful property that $\int_{-\infty}^{+\infty} f(t)\delta(t)dt = f(0)$.

$$h(t) = \mathbf{L}[\delta(t)]\,. \tag{B.6}$$

Any input signal can be treated as an infinite sum of appropriately shifted and scaled impulses, or

$$x(t) = \int_{-\infty}^{+\infty} x(t_1)\delta(t - t_1)dt_1\,. \tag{B.7}$$

The properties of homogeneity, additivity, and shift invariance ensure that the response to any arbitrary input $x(t)$ can be obtained by summing over appropriately shifted and scaled responses to an impulse function (or, equivalently, the output is a weighted sum of its inputs). In short,

$$y(t) = \int_{-\infty}^{+\infty} x(t_1)h(t - t_1)dt_1\,. \tag{B.8}$$

The shorthand form of this integral operation, known as a *convolution*, is $*$, as in

$$y(t) = (x * h)(t)\,. \tag{B.9}$$

We conclude that once we know $h(t)$, the response of a linear system to an impulse, the response to an arbitrary input waveform can be obtained by the linear convolution operation.

Before we end this short digression, we briefly want to remind the reader of another way to analyze time-invariant linear systems, namely, by using sinusoidal inputs. If the input to a linear system is a sinusoidal wave of a particular frequency f (in hertz), the output is another sinusoidal of the same frequency but shifted in time and scaled,

$$\mathbf{L}[\sin(2\pi ft)] = A(f)\sin(2\pi ft + \phi(f))\,. \tag{B.10}$$

The function $A(f)$ is known as the amplitude response and determines how much the output is scaled for an input at frequency f, while the phase $\phi(f)$ determines by how much the sinusoidal wave at the output is shifted in time with respect to the input.

Any input can always be represented as a sum of shifted and scaled sinusoidals. For a linear system, the impulse response function and the amplitude and phase functions are closely related by way of the Fourier transform. The Fourier transform of the impulse response function $h(t)$ is

$$\tilde{h}(f) = \mathbf{F}[h(t)] = \int_{-\infty}^{+\infty} e^{-i2\pi ft}h(t)dt\,. \tag{B.11}$$

The amplitude of this filter $\tilde{h}(f)$ corresponds to the amplitude of the Fourier transform of the impulse response function. Note that $\tilde{h}(f)$ (throughout the book, the $\tilde{h}$ symbol denotes the Fourier transform of some function h) is a complex function. We then have

$$\tilde{h}(f) = A(f)e^{-i\phi(f)}\,. \tag{B.12}$$

The reason we frequently talk in this book about the input being "filtered" by the filter function $\tilde{h}(f)$ is that formally the output can be obtained by convolving (another name for filtering) the input by the filter function. In the frequency space representation, convolution is turned into a straight multiplication, and Eq. B.8 can be rewritten as

$$\tilde{y}(f) = \tilde{h}(f) \cdot \tilde{x}(f)\,. \tag{B.13}$$

As emphasized before, when discussing linearity it is crucial to discuss with respect to what variable. There are a number of instances in which neurobiological systems can be treated,

to some degree of approximation, by linear systems analysis. Prominent examples are linear cable theory, when the input is current and the output voltage (but not when the input is a conductance change; see Chap. 1), or receptive field analysis of retinal or cortical neurons in the visual system (Palmer, Jones, and Stepnoski, 1991). Here, the input is usually the stimulus contrast and the output the mean firing rate.

Certain nonlinear systems can frequently be treated as a linear system with the addition of a simple type of static nonlinearities, such as a threshold (Palm, 1978; French and Korenberg, 1989).

A perhaps surprising linear system is the one that relates a continuous input, call it $\mu(t)$, to a continuous firing rate $f(t)$ via a spiking process. The input, suitably scaled, can be thought of as input into an integrate-and-fire unit (Chap. 14) with no leak and no refractory period. The threshold for generating spikes V_{th} is not fixed but is some probability distribution $p_{\text{th}}(V)$ (Gestri, 1971; Gabbiani and Koch, 1997): every time a spike has been generated the threshold is set to a new value drawn from $p_{\text{th}}(V)$.[2]

For any input $\mu(t)$ this unit generates a particular random spike train sequence, abbreviated here as $\sum_i \delta(t - t_i)$. Obviously, the relationship between the continuous input $\mu(t)$ and the discrete spike train is highly nonlinear. However, let us assume a population of independent but otherwise identically integrate-and-fire units with identical distributed voltage thresholds. If all receive the same input $\mu(t)$ one can average, as discussed in Sec. 14.1, over this ensemble and define an instantaneous output rate $f(t)$.

It can be proven (Gestri, 1971) that varying the input by $\alpha\mu(t)$ changes the instantaneous firing rate of this fictive population of cells by $\alpha f(t)$. In other words, the relationship between the input and the instantaneous output rate is a linear one.

2. If $p_{\text{th}}(V)$ is exponentially distributed, the spikes generated by this process have the convenient property that they are Poisson distributed (Chap. 15).

Appendix C

Sparse Matrix Methods for Modeling Single Neurons

Barak A. Pearlmutter and Anthony Zador

In this appendix we describe numerical methods used in the efficient solution of the linear and nonlinear cable equations that describe single neuron dynamics. Our exposition is limited to the scale of a whole neuron; we will ignore both the simulation of circuits of neurons (see, for example, the monographs by Bower and Beeman, 1998 and by Koch and Segev, 1998), as well as the simulation of the stochastic equations governing single ion channel kinetics (Skaugen and Walloe, 1979; Chow and White, 1996).

The appendix is divided into two parts. The first deals with the solution of the *linear* component of the cable equation. Since the cable as well as the diffusion equation are linear second-order parabolic partial differential equations (PDE), this part draws on techniques and principles developed in the many other fields that deal with similar equations, though naturally the discussion will focus on those problems of particular interest in neurobiology. The theme common to this part is that for the purposes of numerical solution, the cable equation is best discretized into a system of ordinary differential equations coupled by sparse matrices. The main difficulty is that the resulting equations are *stiff*, that is, they display time scales of very different magnitude; even so it is possible to apply widely available and efficient techniques for sparse matrices. The second part deals with the *nonlinear* components of neurodynamics, particularly equations of the Hodgkin–Huxley type and those arriving from calcium dynamics. These nonlinearities are surprisingly benign, and can readily be handled with a few simple techniques, provided that the linear component is treated properly.

Many of the techniques described are implemented in widely used and freely available neural simulators (in particular GENESIS and NEURON; see DeSchutter, 1992; Hines, 1998; Bower and Beeman, 1998) in a manner that is relatively transparent to the user. Nevertheless, there are at least three good reasons for understanding the foundations of these numerical methods. Firstly, when such simulators produce surprising—and possibly spurious—results, an understanding of their internal workings can help determine whether the numerical method is to blame. Secondly, when conducting original research it is inevitable that some problem will arise for which the software must be customized. Finally, and most important, understanding these techniques provides insight into the underlying neurodynamics itself, and hence into the behavior of neurons.

C.1 Linear Cable Equation

C.1.1 Unbranched Cables and Tridiagonal Matrices

Discretization in Space

The passive cable equation (Eq. 2.7) describes the spread of potential in one dimension along an unbranched homogeneous cable,

$$C_m \frac{\partial V(x,t)}{\partial t} = \frac{d}{4R_i} \frac{\partial^2 V(x,t)}{\partial x^2} - G_m V(x,t) - J(x,t), \tag{C.1}$$

where $G_m = 1/R_m$ and $J(x,t)$ is the injected current. (In this appendix, we use J rather than I to distinguish it from the identity matrix I below.) For the sake of convenience, we set the reversal potential in Eq. C.1 to zero; it can be thought of as being absorbed into the offset current $J(x,t)$. The membrane parameters C_m, R_i, G_m, and d are assumed to be independent of position along the cable. Note that the subscript on the intracellular resistivity R_i is not used here as a numerical index. We now discretize this partial differential equation in space by replacing the second spatial derivative with its simplest, second-order discrete approximation,

$$\frac{\partial^2 V(x,t)}{\partial x^2} \rightarrow \frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta x^2}, \tag{C.2}$$

resulting in the system of coupled, ordinary differential equations

$$C_m \frac{dV_i}{dt} = \frac{d}{4R_i} \frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta x^2} - G_m V_i - J_i. \tag{C.3}$$

Here $V_i = V(x_i,t)$ and $J_i = J(x_i,t)$ denote, respectively, the membrane potential and injected current, at some point x_i, where i is the discretization index, and the time variable t has been suppressed. Physically, these indices correspond to the locations along the cable—the *nodes* or *compartments*—at which the voltage is specified (see Fig. 2.2B). The resulting system of coupled ordinary differential equations can be written compactly in matrix form as

$$\frac{d\mathbf{V}}{dt} = \mathbf{C}^{-1}(\psi \mathbf{B}'\mathbf{V} - \mathbf{G}\mathbf{V}) - \mathbf{J}_{\text{inj}} = \mathbf{B}\mathbf{V} - \mathbf{J}_{\text{inj}} \tag{C.4}$$

where the matrix $\mathbf{B}$ is given by

$$\mathbf{B} = \mathbf{C}^{-1}(\psi \mathbf{B}' - \mathbf{G}), \tag{C.5}$$

with $\psi = d/(4R_i \Delta x^2)$. Here $\mathbf{B}'$ is a tridiagonal second difference matrix with elements -2 along the diagonal and $+1$ off the diagonal

$$\mathbf{B}' = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \tag{C.6}$$

and $\mathbf{C}$ and $\mathbf{G}$ are diagonal matrices with C_m and G_m, respectively, along their diagonals (that is, scalar multiples of the identity matrix). Since the matrices $\mathbf{C}$ and $\mathbf{G}$ are diagonal,

they do not contribute to the interaction between equations; coupling is exclusively through $\mathbf{B}'$. This is consistent with our physical expectations, since points along the cable interact only through the second spatial derivative (scaled by the effective axial R_i) captured in $\mathbf{B}'$. As discussed in detail below, the corner elements of this matrix determine the boundary conditions, which here correspond to $V(x) = 0$ at the boundaries—the *killed-end* or *Dirichlet* condition, which we have choosen here for the sake of convenience (Sec. C.1.3). For the *sealed-end* or *von Neumann* boundary condition, we have $dV/dx = 0$ at the boundary, and $\mathbf{B}'$ takes slightly different values in the upper and lower rows. Although the von Neumann condition arises more often, here we consider only the Dirichlet condition because it gives rise to a simpler form.

This set of coupled differential equations is (almost) the same set that arises from a network of discrete passive electrical components, for instance, the one shown in Figs. 2.3 and 3.4B. This is no coincidence. Another way to arrive at this set of equations is to first approximate the spatially continuous electrical structure specified by Eq. C.1 with a discrete electrical network that approximates its properties. As we shall see, however, the analogy with a discrete electrical circuit breaks down at the endpoints of the cable, so care must be taken to ensure the exact boundary conditions.

Eigenvalue Analysis

It is helpful to analyze the behavior of Eq. C.4 in terms of the eigenvalues of $\mathbf{B}$. For now we neglect the injected current term. We will first consider the eigenvalues of $\mathbf{B}'$—the second difference matrix from which $\mathbf{B}$ is derived—and then consider the eigenvalues of $\mathbf{B}$ itself. Recall that the eigenvalues m_z of any matrix $\mathbf{B}$ are those numbers that satisfy

$$\mathbf{B}\mathbf{V}^{(z)} = m_z\mathbf{V}^{(z)} \tag{C.7}$$

where $\mathbf{V}^{(z)}$ is the eigenvector corresponding to the z-th eigenvalue. For an $n \times n$ matrix $\mathbf{B}'$ there are n eigenvalues and n associated eigenvectors, and the index z ranges from 1 to n. We show below that the eigenvalues of the second difference matrix $\mathbf{B}'$ are given by

$$m'_z = -2 + 2\cos\frac{\pi z}{n+1} \tag{C.8}$$

and the corresponding eigenvectors are

$$V_i^{(z)} \propto \sin\frac{\pi z i}{n+1}. \tag{C.9}$$

The two sides are proportional rather than equal, since we are free to choose the magnitude of the right-hand side: any multiple of an eigenvector is itself and eigenvector. If the eigenvectors are scaled so that $||\mathbf{V}^{(z)}|| = 1$, then the solutions to Eq. C.4 can be expressed as a sum of exponentials,

$$\begin{pmatrix} V_1(t) \\ \vdots \\ V_i(t) \\ \vdots \\ V_n(t) \end{pmatrix} = c_1 \begin{pmatrix} V_1^{(1)} \\ \vdots \\ V_i^{(1)} \\ \vdots \\ V_n^{(1)} \end{pmatrix} e^{-t/\tau'_1} + \cdots + c_z \begin{pmatrix} V_1^{(z)} \\ \vdots \\ V_i^{(z)} \\ \vdots \\ V_n^{(z)} \end{pmatrix} e^{-t/\tau'_z} + \cdots + c_n \begin{pmatrix} V_1^{(n)} \\ \vdots \\ V_i^{(n)} \\ \vdots \\ V_n^{(n)} \end{pmatrix} e^{-t/\tau'_n}, \tag{C.10}$$

where $V_i^{(z)}$ is the i-th component of the z-th eigenvector, and τ'_z the reciprocal of the

corresponding eigenvalue.[1] The constants $c_1, \ldots, c_n$ are determined by the projection of the initial conditions $V_i(0)$ onto the eigenvectors,

$$c_z = \sum_i V_i(0) V_i^{(z)}, \tag{C.11}$$

or $\mathbf{c} = \mathbf{M}^T \mathbf{V}_0$. This last expression holds since the eigenvectors of the symmetric matrix $\mathbf{B}$ are orthogonal. If we denote by $\mathbf{M}$ the matrix that has the eigenvectors in Eq. C.9 as its columns, then the vector solution in Eq. C.10 can be rewritten compactly in matrix form as

$$\mathbf{V}(t) = \mathbf{M}^T \mathbf{V}_0 \mathbf{M} e^{\mathbf{m}'t} \tag{C.12}$$

where $\mathbf{V}_0$ is the vector of initial conditions and $\mathbf{m}'$ is the vector of eigenvalues.

These equations tell us several things. (1) The eigenvalues appear as the time constants in a sum of exponentials. This sum represents the decay of voltage with time. It is precisely these time constants that are "peeled" in the classical Rall analysis of electrotonic structure (see Sec. 2.3.2). Second, the eigenvalues m'_z range from $m'_1 = -2 + 2\cos[\pi/(n+1)] < 0$ to $m'_n = -2 + 2\cos[n\pi/(n+1)] > -4$. (2) The smallest eigenvalue m_1 of $\mathbf{B}$ is just less than 0, so the solution always decays to 0, even in the absence of a membrane leak term; this is due to the killed-end boundary condition we are considering. For the sealed-end case, the matrix is conservative for $G_m = 0$, charge only redistributes spatially without net loss or gain, so 0 is an eigenvalue (that is, the voltage decays to a constant). (3) Finally, note that the end elements V_1 and V_n do *not* equal zero, contrary to what we might expect from the boundary conditions. Only when we consider the "phantom" elements at $i = 0$ and $i = n + 1$ do we see that the boundary conditions, $V_0 = 0$ and $V_{n+1} = 0$, are satisfied.

How do the eigenvalues of $\mathbf{B}$—which is the operator that governs the physical behavior of the cable equation—relate to those of $\mathbf{B}'$? In general, if the eigenvalues of a matrix $\mathbf{B}$ are m_z, then the eigenvalues of $(\mathbf{B} - b\mathbf{I})/c$, where $\mathbf{I}$ is the identity matrix, are $(m_z - b)/c$; the eigenvectors remain unchanged. The dependence on b is called the *shifting property*.[2] The eigenvalues m_z can be obtained from those of the second difference matrix (Eq. C.8),

$$m_z = -\frac{1}{\tau_z} = \frac{\psi}{C_m}\left(2 - 2\cos\frac{\pi z}{n+1}\right) + \frac{G_m}{C_m}, \tag{C.13}$$

where as before z ranges from 1 to n. For $\mathbf{B}$, the smallest eigenvalue (associated with $z = 1$) is just larger than G_m/C_m (since $\cos[\pi z/(n+1)] \to 1$ for large n), which is just the inverse of the membrane time constant. The largest eigenvalue is just smaller than $4\psi/C_m + G_m/C_m$ (for large n). However, in a cable of fixed length as the number of compartments becomes larger and larger, the spatial discretization step Δx becomes smaller and smaller and this eigenvalue goes to infinity (and the corresponding time constant to 0), as expected from the corresponding eigenvalues of the continuous cable equation.

How do these expressions for the eigenvalues and eigenvectors arise? One way to understand their origin is by analogy to the continuous diffusion equation. The eigenfunctions of the continuous second derivative operator with a Dirichlet boundary condition (corresponding to a killed-end condition) on the interval [0, 1] are the sinusoidals

1. Following neuroscience convention, we express the solution in terms of $\tau'_z = -1/m'_z$ rather than m'_z.

2. This shifting property can be used in the change of variables $W = Ve^{t/\tau}$, with $\tau = C_m/G_m$, for reducing the cable equation (Eq. C.1) in V to the diffusion equation $C_m \partial W/\partial t = \frac{(dG_m)}{(4R_i)} \partial^2 W/\partial x^2$ in W. This shifting property is a special case of the property that for any polynomial (or more generally, any analytic function) $p(\cdot)$, the eigenvectors of $p(\mathbf{B})$ are $p(m_z)$, and the eigenvectors are unchanged.

$$\frac{\partial^2}{\partial x^2}\sin(\pi z x) = -(\pi z)^2 \sin(\pi z x), \tag{C.14}$$

where $-(\pi z)^2$ is the z-th eigenvalue. We need to find the corresponding expression for the discrete second difference operator $\mathbf{D}_2$:

$$\mathbf{D}_2 V_j = V_{j-1} - 2V_j + V_{j+1}. \tag{C.15}$$

We posit that the eigenvectors of $\mathbf{D}_2$ correspond to the eigenfunctions of the continuous operator, and we write the expression

$$\mathbf{D}_2 \mathbf{V}^{(z)} = \sin\frac{\pi(j-1)z}{n+1} - 2\sin\frac{\pi j z}{n+1} + \sin\frac{\pi(j+1)z}{n+1} = m'_z \sin\frac{\pi j z}{n+1} \tag{C.16}$$

and solve for m'_z. Application of the appropriate trigonometric identities confirms our expression Eq. C.8 for m'_z.

Another interpretation is in terms of the discrete Fourier transform (DFT) or rather its cousin, the discrete sine transform (DST). Although the DST is usually efficiently computed using the FFT algorithm, the DST is a linear transform, and can therefore be represented as a matrix $\mathbf{S}$. For an $n \times n$ operator $\mathbf{S}$—required to find the transform of a $1 \times n$ vector—the elements of $\mathbf{S}$ are given by

$$S_{jz} = \sin\frac{\pi j z}{n+1}. \tag{C.17}$$

The columns of this matrix are precisely the eigenvectors of $\mathbf{B}$. Therefore, $\mathbf{S}$ diagonalizes $\mathbf{B}$,

$$\mathbf{S}^{-1}\mathbf{B}\mathbf{S} = \mathrm{diag}(m'_1, m'_2, \ldots, m'_n). \tag{C.18}$$

This is entirely in analogy with the Fourier transform in the continuous case.

Explicit Discretization in Time

While the sum of exponentials suggests a possible solution to the equation, it does not generalize well to nonlinear problems and for many problems it is computationally inefficient (because n exponentials must be evaluated at each time point). In order to solve this system numerically, we transform the system of ordinary differential equations into a system of algebraic difference equations to be advanced by discrete time steps Δt. Just as before we replaced the spatial derivative by its finite difference approximation, we replace the temporal derivative by its first-order difference

$$\frac{d\mathbf{V}}{dt} \to \frac{\mathbf{V}^{t+1} - \mathbf{V}^t}{\Delta t}, \tag{C.19}$$

where the superscript refers to the index of discretized time. Now we have a choice for how to combine this with the right-hand side of Eq. C.4: do we use $\mathbf{V}^t$ or $\mathbf{V}^{t+1}$? The temptation is to use $\mathbf{V}^t$, since then $\mathbf{V}^{t+1}$ is an *explicit* function of $\mathbf{V}^t$,

$$C_m \frac{V_i^{t+1} - V_i^t}{\Delta t} = \frac{d}{4R_i}\frac{V_{i+1}^t - 2V_i^t + V_{i-1}^t}{\Delta x^2} - G_m V_i^t - J_i^t. \tag{C.20}$$

This choice for dV/dt is the basis of the *forward Euler* scheme. We can advance one time step at a very reasonable cost of three multiplications per node—one for the self-connection V_i and one each for the adjacent nodes $V_{i\pm1}$.

We can rewrite a single iteration of the forward Euler scheme as

$$\mathbf{V}^{t+1} = \mathbf{B}_f \mathbf{V}^t - \mathbf{J}_i^t, \tag{C.21}$$

where $\mathbf{B}_f$ is a tridiagonal matrix and $\mathbf{V}^t$ is a vector. The matrix $\mathbf{B}_f$ is obtained by solving for $\mathbf{V}^{t+1}$ in the previous equation and is equal to $\Delta t\ \mathbf{B}$ with an additional 1 along the diagonal,

$$\mathbf{B}_f = \mathbf{I} + \Delta t\ \mathbf{B} \tag{C.22}$$

where $\mathbf{I}$ is the identity matrix. If we set $\mathbf{J} = 0$, then after p iterations, the solution can be written as a power of $\mathbf{B}_f$,

$$\begin{aligned}\mathbf{V}^{t+p} &= (\mathbf{B}_f)^p \mathbf{V}^t \\ &= c_1 (f_1)^p \mathbf{V}^{(1)} + \cdots + c_z (f_z)^p \mathbf{V}^{(z)} + \cdots + c_n (f_n)^p \mathbf{V}^{(n)},\end{aligned}$$

where in the second line the solution is expressed in terms of the eigenvalues f_z of $\mathbf{B}_f$ and the c's are once again determined by the initial conditions. Note that while in the solution of the continuous differential equation (Eq. C.10) the eigenvalues appear in the exponent, here in the discrete difference equation they are raised to a power. This imposes a *stability* constraint on the eigenvalues. Stability refers to the behavior of the solution for $t \to \infty$, which for the discrete case translates into $p \to \infty$. For the discrete case the solution is stable if

$$|f_z| < 1 \tag{C.23}$$

for all z, since the eigenvalues are raised to a power. If any eigenvalue does not satisfy this condition, then for sufficiently large p it eventually diverges to infinity. By contrast, in the continuous case the eigenvalues appeared in the exponent, so the condition was that $m_z < 0$ for all z.

The forward Euler method is unstable if Δt is chosen too large. Consider the eigenvalues of the difference matrix $\mathbf{B}_f$, which we compute from Eq. C.13 using the shift property,

$$f_z = 1 + \Delta t\ m_z. \tag{C.24}$$

These eigenvalues are almost identical to those of the original differential system, but with the addition of the 1, which imposes a condition on the discretization parameters. Recall that we know only that $m_z < 0$, so there will exist discretization parameters such that $|f_z| > 1$. For $n \to \infty$ and $G_m = 0$, that is, no membrane leak, the stability condition simplifies to $|4\psi \Delta t / C_m - 1| < 1$, or $\Delta t \leq R_i C_m \Delta x^2 / d$. Thus if the spatial discretization Δx is made twice as small, the temporal discretization Δt must be made four times smaller to preserve stability. Since doubling the spatial discretization step also doubles the size of the matrix, computation time scales as the third power of n. This very rapidly becomes the limiting factor for large numbers of compartments.

Implicit Discretization in Time

One alternative is to use an *implicit* or *backward Euler* scheme, where the spatial derivative is evaluated at $t + 1$,

$$C_m \frac{V_i^{t+1} - V_i^t}{\Delta t} = \frac{d}{4R_i} \frac{V_{i+1}^{t+1} - 2V_i^{t+1} + V_{i-1}^{t+1}}{\Delta x^2} - G_m V_i^{t+1} - J_i^{t+1}. \tag{C.25}$$

Setting $J_i^{t+1} = 0$, we can rewrite (C.25) in matrix form as

$$\mathbf{V}^{t+1} = \mathbf{B}_b \mathbf{V}^t, \tag{C.26}$$

where

$$\mathbf{B}_b = [\mathbf{I} - \Delta t\, \mathbf{B}]^{-1}. \tag{C.27}$$

Using the result that the eigenvalues of the inverse of a matrix are just the inverse of the eigenvalues (and the eigenvectors unchanged), we write the eigenvalues b_z of $\mathbf{B}_b$,

$$b_z = (1 - \Delta t\, m_z)^{-1}. \tag{C.28}$$

The denominator is identical to Eq. C.24, except for the sign of the -1. This makes all the difference, since now $|b_z| < 1$ for all z, and the system is stable for all discretization steps. This does not guarantee, of course, that the solution to the discretized equation is close to the solution to the underlying continuous equations. This still requires small values of Δx and Δt.

The stability of the implicit scheme comes at a very high apparent cost: the inversion of an $n \times n$ matrix $\mathbf{B}_b$ at each time step. Now in general, the inversion of an $n \times n$ matrix requires $O(n^3)$ time; this is the same cost as the explicit scheme. Note that $\mathbf{B}_b$ is very sparse—in fact, it is tridiagonal. This sparseness is critical, since the solution to Eq. C.26 can be obtained by Gaussian elimination in $O(n)$ steps. (We defer the description of Gaussian elimination for the tridiagonal matrix until Sec. C.1.2, where it appears as a special case.) Thus, the implicit scheme is stable for all time steps and can be implemented in an efficient manner that scales linearly with the grain of spatial discretization.

Semi-Implicit Discretization in Time

The choice defining the implicit scheme, evaluating the spatial derivative at $\mathbf{V}^{t+1}$, is not the only one that leads to a stable scheme. We could also choose to evaluate the spatial derivative at the the midpoint between $\mathbf{V}^{t+1}$ and $\mathbf{V}^t$ (that is, by using the discretization scheme $\mathbf{V}^{t+1/2} = (\mathbf{V}^{t+1} + \mathbf{V}^t)/2$), to obtain

$$C\frac{V_i^{t+1} - V_i^t}{\Delta t} = \frac{d}{4R_i}\frac{\frac{1}{2}(V_{i+1}^t + V_{i+1}^{t+1}) - 2\frac{1}{2}(V_i^t + V_i^{t+1}) + \frac{1}{2}(V_{i-1}^t + V_{i-1}^{t+1})}{\Delta x^2}$$

$$- \frac{G_m}{2}(V_i^t + V_i^{t+1}) - \frac{J_i^t + J_i^{t+1}}{2}. \tag{C.29}$$

This choice is called the *semi-implicit* or *Crank-Nicolson* algorithm. Setting $J_i = 0$ as before, the resulting matrix equations can be expressed in terms of the implicit and explicit matrices,

$$\mathbf{B}_b\mathbf{V}^{t+1} = \mathbf{B}_f\mathbf{V}^t. \tag{C.30}$$

Stability is therefore determined by the eigenvalues e_z of $\mathbf{B}_b^{-1}\mathbf{B}_f$. Since $\mathbf{B}_b$ and $\mathbf{B}_f$ share eigenvectors, the eigenvalues of $\mathbf{B}_b^{-1}\mathbf{B}_f$ are simply the product of the eigenvalues of $\mathbf{B}_b^{-1}$ and $\mathbf{B}_f$,

$$e_z = \frac{1 + \Delta t\, m_z}{1 - \Delta t\, m_z}. \tag{C.31}$$

Observe that since $m_z < 0$ the numerator is always less than the denominator, so that the absolute value of this expression is less than unity. Hence the Crank-Nicolson method is stable for all choices of discretization parameters. This method is almost always preferable to the implicit method, because it is more accurate (as well as being stable).

The *accuracy* of any numerical method is measured by performing a Taylor expansion about $\Delta t = 0$. If there is a nonzero coefficient associated with Δt then the method is *first-order*, while if the first nonzero coefficient is associated with Δt^2 then the method is *second-order*. Both the explicit and the implicit discretizations are only first-order, while the semi-implicit discretization is correct up to second-order.

C.1.2 Branched Cables and Hines Matrices

So far we have developed techniques only for unbranched cables. Since many neurons have very extended and complex dendritic trees (see Fig. 3.1), these techniques must be extended if they are to be useful. The primary complication introduced by branching is that while the connectivity of an unbranched cable can be represented by a tridiagonal matrix, connectivity of a branched neuron can only be represented by a special matrix, a *Hines* matrix, of which the tridiagonal is a special case. This matrix has the same number of elements as a tridiagonal matrix, $3n - 2$ in the $n \times n$ case, but they can be scattered, rather than concentrated along the diagonals. Hines (1984) first introduced this transformation in the context of dendritic modeling and showed how to invert this matrix in $O(n)$ steps—the same as any tridiagonal matrix. This observation was key: until then implicit matrix techniques were believed inefficient for branched cables because the matrix inversion was thought to require $O(n^3)$ steps. We will see below that there are other cases where the sparseness of this matrix can be used to construct algorithms as efficient as those for the tridiagonal case.

The Hines matrix is defined by the structure of its sparseness. A matrix $\mathbf{B}$ is a Hines matrix if three conditions are satisfied: (1) The diagonal elements B_{jj} are nonzero; (2) B_{ij} is nonzero if and only if B_{ji} is nonzero; and (3) for any nonzero B_{ij} with $i < j$, there is no h such that $h > j$ and B_{ih} is nonzero. The second condition requires B to be structurally although not numerically symmetric, and the third requires that $\mathbf{B}$ have no more than one nonzero element to the right of the diagonal in any row, and by structural symmetry, no more than one nonzero element below the diagonal in any column. Notice that while there are only two tridiagonal matrices corresponding to an unbranched cable (since numbering can start from either end), there are many Hines matrices corresponding to a single tree structure.

In order to convert a graph into a matrix, the nodes must be numbered sequentially. There is a simple algorithm for appropriately labeling the nodes of a tree to construct a Hines matrix. The algorithm given here is a generalization of that proposed in Hines (1984).

> Sequentially number nodes that have at most one unnumbered neighbor until there are no unnumbered nodes left.

At each step in the algorithm there may be many choices for which node to number next; although all choices are equally suitable for the algorithms discussed, the particular choice of numbering adjacent nodes successively wherever possible leads to the matrix elements maximally concentrated near the diagonal.

The advantage of this numbering scheme is that Gaussian elimination can be applied directly. At the abstract level, it amounts to solving for $\mathbf{V}^{t+1}$ in the matrix equation $\mathbf{B}\mathbf{V}^{t+1} = \mathbf{V}^t$ by

Consider the matrix $[\mathbf{B}|\mathbf{V}^t]$.
Using row operations, put it in the form $[\mathbf{I}|h]$, where $\mathbf{I}$ is the identity matrix.
Now $h = \mathbf{V}^{t+1} = \mathbf{B}^{-1}\mathbf{V}^t$.

The standard scheme for solving tridiagonal systems is a special case of this algorithm.

C.1.3 Boundary Conditions

The partial differential equation for a single unbranched cable (Eq. C.1) has a unique solution only if *boundary conditions* (BC) are specified at the endpoints. The BCs determine the behavior of the membrane potential or its derivative at the boundary points. The membrane potential along a branched cable is governed by a system of coupled partial differential equations, one for each branch, and BCs must be specified at the branch points. In matrix notation the BCs at the origin of the cable correspond to the first row of the matrix $\mathbf{B}$, and at the end of the cable to the last row of the matrix $\mathbf{B}$. Similarly the BCs at a branch point correspond to the matrix elements at that point. We shall see that the implementation of BCs at a branch point is just a natural extension of the elements at any point along the cable and so poses few problems. The BCs at endpoints is rather more subtle and requires careful consideration.

In general there are many BCs corresponding to different physical situations. For example, in the *killed-end* or *Dirichlet* case (see Sec. 2.2.2) the voltage is clamped to zero and the axial current "leaks" out to ground. The matrix $\mathbf{B}'$ we have been considering above in the eigenvalue analysis implements this condition. To see this, assume that the fictive point at $x = 0$, that is, V_0 is assumed to be zero. Following Eq. C.3, the difference equation for the point at $x = \Delta x$ is proportional to $V_2 - 2V_1 + V_0 = V_2 - 2V_1$. Thus, the top two entries in $\mathbf{B}'$ are -2 and 1, as are the bottom two entries. The killed-end solution is a special case of the voltage-clamp condition, where the voltage at the endpoint is held at some arbitrary value. This condition is simply $V_0 = V_{\text{clamp}}$.

Here we limit our attention to the most important class of boundary conditions for nerve equations, the so-called *sealed-end* or *von Neumann* condition (Sec. 2.2.2). The sealed-end condition requires that no current flow out of the end. Mathematically, this is expressed as $\partial V/\partial x = 0$ at the boundary (Eq. 2.19).

There are several ways this boundary condition associated with the continuous equation can be implemented in a discretized system. Perhaps the most intuitive is the one that arises from consideration of an equivalent electrical circuit model, as in Fig. 3.4B. Following Ohm's law, the current flow in the axial dircetion is proportional to $V_2 - V_1$. Thus, the diagonal element is -1 or half the size of the elements along the rest of the diagonal. Because the endpoint is connected only to a single neighbor, while all other nodes are connected to two neighbors, it seems reasonable that the loss to neighbors should be half of that at all other nodes. With these choices the resulting matrix $\mathbf{B}'$ turns out to be symmetric, which is in line with our intuition. For a single unbranched cable with a sealed end at both ends, we have

$$\mathbf{B}' = \begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}. \tag{C.32}$$

This implementation has been widely used in neural simulators (but not by Mascagni, 1989, or Hines, 1989).

Our intuition can be misleading, that is, the associated electrical circuit (as in Fig. 3.4B) does not approximate well a finite cable with a sealed-end boundary. While such a matrix does implement a form of the von Neumann condition, it is correct only to first order, that is, the error is $O(\Delta x)$. This implementation introduces systematic errors or "phantom currents" (Niebur and Niebur, 1991).

A more accurate scheme can be derived by considering the Taylor expansion of the potential around $x = 0$ for a "ficticious" element just beyond the end of the cable at $x = -\Delta x$,

$$V_{-1} = V_0 - \Delta x \frac{\partial V}{\partial x}\bigg|_{x=0} + \frac{\Delta x^2}{2}\frac{\partial^2 V}{\partial x^2}\bigg|_{x=0} + O((-\Delta x)^3). \tag{C.33}$$

We can now solve for V_{-1} in terms of V_0 and V_1 by setting the first derivative to 0 (the BC corresponding to no axial current across the membrane at $x = 0$) and setting the second derivative to the second difference approximation, $\partial^2 V/\partial x^2|_{x=0} = (V_{-1}+V_1-2V_0)/\Delta x^2$, to obtain

$$V_{-1} = V_1. \tag{C.34}$$

This gives us the matrix with the second-order correct von Neumann boundary conditions:

$$\mathbf{B}' = \begin{pmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{pmatrix}. \tag{C.35}$$

C.1.4 Eigensystems and Model Fitting

We have seen how the morphology of a neuron—its branching geometry and connectivity—together with the passive membrane parameters—R_m, C_m, and R_i—collectively determine the sparse matrix $\mathbf{B}$ which governs the dynamic behavior of the neuron. Equation C.10 shows how the eigensystem acts as the link between dynamics and morphology: the time course of membrane potentials can be expressed as the sum of exponentials whose decay constants are the eigenvalues of $\mathbf{B}$. Thus, starting from the physical description of the neuron, we can use the eigensystem to compute its response to stimuli, although in practice it is usually more efficient to use an implicit matrix scheme.

Suppose we wish to determine the eigensystem of a model neuron. For very simple morphologies, for instance, a single terminated cylinder, and homogeneous membrane parameters, Rall (1977) has used the underlying partial differential equation to derive some relatively simple expressions for the eigenvalues as a function of the membrane parameters (as in Eq. 2.38). These expressions provide some insight into the behavior of passive cables. For somewhat more complex morphologies (for instance, several cylinders attached to a single lumped soma) there are correspondingly more complex expressions involving the roots of transcendental equations. But as the morphologies become more complex the expressions become more difficult to evaluate and provide less insight. Analytic solutions are not generally useful for arbitrary dendritic trees.

For arbitrary trees there are at least two options for determining the eigensystem. First, we could compute a solution $V_i(t)$ from initial conditions, using for example an implicit matrix scheme as outlined above, and then attempt to extract the time constants τ_z and associated coefficients $c'_{iz} = V_i^{(z)} c_z$ by fitting them to the solution. One method of fitting is the so-called "exponential peeling" or just "peeling" method (Rall, 1977). In fact, exponential fitting is intrinsically very difficult. The reason is that exponentials are a poor set of basis functions, because they are so far from orthogonal. The inner product of two exponentials over an infinite interval is given by

$$\int_0^\infty e^{-t/\tau_1} e^{-t/\tau_2} dt = \frac{\tau_1 \tau_2}{\tau_1 + \tau_2}. \tag{C.36}$$

Orthogonality would require that this expression be zero when $\tau_1 \neq \tau_2$, which is not the case. In fact, the general exponential fitting problem is equivalent (in the limit of a large number of time constants) to numerically inverting a Laplace transform, which is well known to be *ill-conditioned* (Bellman, Kalaba, and Locket, 1966). Since many combinations of time constants and coefficients fit almost equally well, no technique can reliably extract the time constants in the presence of noise. Typically only the first two or three eigenvalues can be extracted, as suggested by the characteristic decrease in the eigenvalues in Eq. C.8.

If we are working with a model of a neuron, a second and much more reasonable alternative is to extract the eigenvalues directly from the matrix **B**. (This option is not available to us if we are working with a real neuron, in which case we must fall back on some nonlinear fitting technique.) Typically we are interested only in the first few eigenvalues, so we exploit the sparseness of **B** by observing that multiplication of **B** with a vector is cheap—$O(n)$ for an $n \times n$ Hines matrix. We use the power method, which depends on the fact that the principal eigenpair dominates. Defining a new matrix $\mathbf{P} = \alpha \mathbf{I} - \mathbf{B}$ (where **I** is the identity matrix) with eigenvalues $p_z = \alpha - m_z$, the smallest eigenvalue of **B** (that is, the largest time constant) can be determined by the following algorithm, which finds the principal eigenpair of **P** for any initial **V**:

repeat

$$\mathbf{V} \leftarrow \frac{1}{\|\mathbf{V}\|} \mathbf{P}\mathbf{V} \tag{C.37}$$

until **V** is stable.

Standard deflation can be used to extend this procedure directly to the computation of the next several smallest eigenvalues of **B** (Stoer and Bulirsch, 1980).

C.1.5 Green's Functions and Matrix Inverses

Prior to the ascent of sparse matrix techniques, methods based on Green's functions (also called *transfer impedances*) and Laplace transforms were widely used (Butz and Cowan, 1984; Koch and Poggio, 1985a; Holmes, 1986; see Sec. 3.4). The Green's function gives the response at any point i to a delta pulse of current applied at some other point j; their properties were discussed in Sec. 2.3.1 (e.g., Eq. 2.31). These methods have been largely abondoned because they are typically less efficient and more difficult to generalize to synaptic and nonlinear conductances. Nevertheless, for theoretical work they are often very convenient, and can sometimes offer a different perspective (see also Abbott, 1992). One interesting application of the Green's function is the morphoelectrotonic transform

(Zador, Agmon-Snir, and Segev, 1995; see Sec. 3.5.4). Another is in analyzing the effects of Hebbian learning (Pearlmutter, 1995). Here we reconsider the Green's functions in terms of sparse matrix methods.

One way to compute the Green's function is in terms of the eigensystem of the matrix **B**. First we expand the solution $V_j(t)$ at a point x_j in terms of the initial conditions $V_i(0)$ at $t = 0$ and the Green's function $K'_{ij}(t)$,

$$V_j(t) = \sum_i V_i(0) K'_{ij}(t). \tag{C.38}$$

Notice that K'_{ij} is identical to K_{ij} of Eq. 3.16 and in the following, except for a constant of proportionality with the dimensions of an impedance. Because the eigenvectors $V_i^{(z)}$ form a complete basis, we can use them to represent the initial conditions,

$$V_i(0) = \sum_z c_z V_i^{(z)}, \tag{C.39}$$

with c_z constants. From Eq. C.10 we can write $V_j(t)$ as

$$V_j(t) = \sum_z c_z V_j^{(z)} e^{-t/\tau_z}. \tag{C.40}$$

By orthogonality and completeness we can compute the constants as the projection of the initial condition vector on the eigenvectors,

$$c_z = \sum_i V_i(0) V_i^{(z)}. \tag{C.41}$$

Hence

$$V_j(t) = \sum_i V_i(0) \underbrace{\sum_z V_i^{(z)} V_j^{(z)} e^{-t/\tau_z}}_{\text{Green's function}} \tag{C.42}$$

that is,

$$K'_{ij}(t) = \sum_z V_i^{(z)} V_j^{(z)} e^{-t/\tau_z}. \tag{C.43}$$

Understanding the eigenvalue expansion of the Green's function can be useful in theoretical work and for small test problems. In practice, computing the Green's function from the eigensystem is often not the most efficient way. Rather, it is often better to exploit the fact that the Green's function can be considered an inverse operator. This reduces the problem to computing the inverse of a matrix. Once again, we can exploit the sparseness of **B** to compute the Green's function.

Extension to Two and Three Spatial Dimensions

The techniques we have described for solving the cable and diffusion equations in one dimension are readily generalized to two or even three spatial dimensions. Although the solution of electrical potential in several dimensions is not common (see, however, Chap. 2), the diffusion of second messengers such as Ca^{2+} often requires a consideration of two- or three-dimensional effects.

The separability of the three-dimensional diffusion operator suggests an easy and efficient method for solving multidimensional diffusion. Consider the three-dimensional Laplacian operator in Cartesian coordinates,

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}. \tag{C.44}$$

Similarly we can write the discrete approximation using matrix operators,

$$\mathbf{L}_{3d} = \mathbf{L}_x + \mathbf{L}_y + \mathbf{L}_z. \tag{C.45}$$

Here $\mathbf{L}_x$, $\mathbf{L}_y$, and $\mathbf{L}_z$ are very sparse matrices that approximate the second derivative in the x, y, and z directions, respectively. These operators can be applied sequentially, using for example a stable implicit method, to advance the solution from t to $t + 1$.

The main difficulty associated with solving three-dimensional diffusion is bookkeeping, although the bookkeeping with this approach is simpler than if $\mathbf{L}_{3d}$ is used directly. The matrices that comprise $\mathbf{L}_{3d}$ are sparse, but not tridiagonal or Hines. The locations of the nonzero entries depend on the precise spatial discretization.

C.2 Nonlinear Cable Equations

We have focused on the numerical solution of the linear cable equation for two reasons. First, it is impossible to solve the nonlinear equations efficiently without first understanding the techniques for linear solution. Second, the methods of numerical solution provide insight into the equations themselves. Our approach was to compare the eigensystems of the discrete and continuous systems, and the correspondences were strong.

Here we consider two important classes of nonlinear cable equations that arise frequently in the study of single neurons. The first class is a generalization of the Hodgkin-Huxley equations, which describe the propagation of the action potential. The second class arises from the effect of nonlinear saturable ionic buffering in diffusion. Unfortunately, we know of no approach to understanding the numerical solutions of these nonlinear equations that provides the same kind of insight. This may be because most numerical techniques begin by linearization (eliminating the interesting properties of these equations), while the behavior of the nonlinear equations is best understood using phase space analysis (see Chap. 7) or by numerical simulation. In the first section we provide an overview of the most widely used and efficient method for solving generalizations of the Hodgkin-Huxley equations. It is a direct method that is so easy to implement that since its introduction a decade ago (Hines, 1984) it seems largely to have supplanted the predictor-corrector method of Cooley and Dodge (1966) that reigned before. It is now standard on many widely distributed neural simulators, including GENESIS and NEURON (see www.klab.caltech/MNM for more information on these). In the next section we describe a simple scheme for handling nonlinear saturable buffers.

C.2.1 Generalized Hodgkin–Huxley Equations

The method we consider applies to a class of nonlinear cable equations generalized from the Hodgkin–Huxley equations. They differ from the linear cable equation only by the addition of a term, $I_{\text{HH}}(V(x,t),t)$, which gives the current contributed by voltage-dependent membrane channels,

$$C_m \frac{\partial V(x,t)}{\partial t} = \frac{d}{4R_i} \frac{\partial^2 V(x,t)}{\partial x^2} - G_m V(x,t) - I_{\text{HH}}(V(x,t),t). \tag{C.46}$$

The most general form we need to consider for $I_{\text{HH}}(V,x,t)$ includes k different ionic currents,

$$I_{HH}(V(x,t),t) = \overline{g_1}\, g_1(V,t)\,(V(x,t) - E_1) + \cdots + \overline{g_k}\, g_k(V,t)\,(V(x,t) - E_k) \quad \text{(C.47)}$$

(see Eq. 6.20). The total current is the sum of the contributions of the k individual currents in the membrane patch considered. Here $\overline{g_1}$ is the density of type 1 membrane channels at any point along the neuron; $V(x,t) - E_1$ is the driving force for this conductance; and $0 \leq g_1(V,t) \leq 1$ is the fraction of total conductance of the current, that is, the fraction of total channels of that type that is open in one patch of membrane. The interesting term in this expression is the function $g(V,t)$, because it includes the nonlinearity. It has the general form

$$g(V,t) = y(V,t)^r, \quad \text{(C.48)}$$

where y is a gating particle, and r is an integer corresponding to the number of identical gating particles that need to be simultaneously present in order for current to flow (see Sec. 6.2). In many cases, there are two or even three gating particles, usually *activating* and *inactivating* particles. In this case, a slightly more general product of the form $g(V,t) = y_m(V,t)^{r_m} y_h(V,t)^{r_h}$ must be used. This introduces no conceptual difficulties but does clutter the notation.

We have suppressed the spatial dependence of the currents, since this occurs only through $V(x,t)$. This fact simplifies the solution, since it means that the system of equations governing g in the spatially discretized system is diagonal. The variable $y(V,t)$ (or variables y_m and y_h) in turn obeys a first-order, nonlinear differential equation of the form

$$\frac{dy}{dt} = \frac{y_\infty(V) - y}{\tau_y(V)}. \quad \text{(C.49)}$$

Here the function $y_\infty(V)$ governs the steady-state behavior of y. It is monotonic and bounded between 0 and 1—it must be, for g to be guaranteed to always fall in the same range. The $\tau_y(V)$ governs the rate at which equilibrium is reached. For numerical solution an equivalent but more convenient form is

$$\frac{dy}{dt} = (1 - y)\,\alpha_y(V) - y\,\beta_y(V). \quad \text{(C.50)}$$

The method we describe depends on the fact that the generalized Hodgkin–Huxley system is *conditionally linear* (Mascagni, 1989), meaning that the system is linear in V^{t+1} if g_h^t is known, and likewise linear in g_h^{t+1} if V^t is known. Thus we can alternate between solving for V^t at the times V^t, V^{t+1}, V^{t+2}, ..., and solving for g_h at intermediate points $g^{t+\frac{1}{2}}, g^{t+\frac{3}{2}}, \ldots$. We do this implicitly,

$$y^{t+\frac{1}{2}} = y^{t-\frac{1}{2}} + \Delta t[(1 - y^{t+\frac{1}{2}})\alpha_y(V^t) - y^{t+\frac{1}{2}}\beta_y(V^t)]. \quad \text{(C.51)}$$

We can solve this expression explicitly for $y^{t+\frac{1}{2}}$ in terms of $y^{t-\frac{1}{2}}$ and V^t, both of which are known. The algebra is straightforward and the details are well described in Hines (1984).

C.2.2 Calcium Buffering

Chapter 11 deals with how, under certain conditions, the equations describing calcium dynamics are formally equivalent to the cable equation. Furthermore, over a wide range of parameters, the linearized equations offer an adequate approximation to the nonlinear dynamics. Nevertheless, in some cases it is of interest to explore the behavior of the full nonlinear equations. (If linear two- or three-dimensional diffusional behavior is of interest,

the methods of Sec. C.1.5 can be used.) For the most part the semi-implicit method described above generalizes directly to calcium dynamics. A nonlinear saturable buffer raises special issues that must be considered separately.

Let us treat the common case of a second-order buffer (Sec. 11.4.1),

$$B + Ca^{2+} \underset{b}{\overset{f}{\rightleftharpoons}} M \tag{C.52}$$

where the rate constants f and b govern the equilibrium of free and bound calcium and $[Ca^{2+}]_i$ is the concentration of free, intracellular calcium. (To simplify our exposition, we consider only the case of a second-order buffer, but higher-order buffers introduce no qualitative differences.)

Together with the basic diffusion equation, with a simple calcium extrusion process $P([Ca^{2+}]_i)$, and a saturable nondiffusible buffer (see also Eqs. 11.37, 11.43 and 11.48) we have,

$$\frac{\partial [Ca^{2+}]_i}{\partial t} = D\frac{\partial^2 [Ca^{2+}]_i}{\partial x^2} - P([Ca^{2+}]_i) - f[Ca^{2+}]_i\,(T_B - [B \cdot Ca]) + b[B \cdot Ca] - \frac{4}{d} i(x,t) \tag{C.53}$$

$$\frac{\partial [B \cdot Ca]}{\partial t} = f[Ca^{2+}]_i\,(T_B - [B \cdot Ca]) - [B \cdot Ca] + D' \frac{\partial^2 [B \cdot Ca]}{\partial x^2}, \tag{C.54}$$

where the calcium concentration $[Ca^{2+}]_i(x,t)$ at time t and position x in response to the applied calcium current density $i(x,t)$ depends on the partial derivatives of $[Ca^{2+}]_i$, the diffusion constant D of Ca^{2+}, the diameter d of the cable, and the concentration of total buffer T_B (the sum of the free buffer and the bound buffer). The first equation specifies the rate of change in concentration of calcium as a function of diffusion, extrusion, buffer dynamics and influx. The second equation gives the rate of change in bound buffer concentration as a function of buffer diffusion and buffer binding and unbinding (see also Eq. 11.48).

The difficulty arises because $[Ca^{2+}]_i$ enters into Eq. C.53 in an essentially nonlinear way as $[Ca^{2+}]_i[B]$, and not in the conditionally linear way as in the case of the Hodgkin–Huxley equations. However, we can discretize Eq. C.53 in such a way that the system becomes conditionally linear. That is, we can rewrite the right-hand side of Eq. C.53 in terms of $[Ca^{2+}]_i^{t+1}$ and $[B \cdot Ca]^t$,

$$\frac{[Ca^{2+}]_i^{t+1} - [Ca^{2+}]_i^t}{\Delta t} = h([Ca^{2+}]_i^{t+1}, [B \cdot Ca]^t) \tag{C.55}$$

and then write Eq. C.54 in terms of $[Ca^{2+}]_i^t$ and $[B \cdot Ca]^{t+1}$,

$$\frac{[B \cdot Ca]^{t+1} - [B \cdot Ca]^t}{\Delta t} = g([B \cdot Ca]^{t+1}, [Ca^{2+}]_i^t). \tag{C.56}$$

These implicit equations can now be advanced each time step by an $O(n)$ step (a single sparse matrix inversion for the Hines matrix) for Eq.C.55, and another $O(n)$ step for the diagonal matrix specified in Eq. C.56.

C.2.3 Conclusion

Although this appendix has ostensibly been about efficient numerical methods for single neuron simulation, the real aim has been to provide a deeper understanding of the equations

that govern neurodynamics. In the case of the linear cable equation our approach has been to carefully analyze the properties of the discrete eigensystem corresponding to the continuous partial differential equations underlying the cable equation. This led to stable and efficient methods for simulating the cable equation, to an understanding of why fitting passive models to neurophysiological data is hard, and to a different way of looking at the Green's functions. For the nonlinear case our goals were more modest, namely, an overview of the best existing method for solving equations of the Hodgkin–Huxley class, and of some special problems that arise in the solution of calcium dynamics.